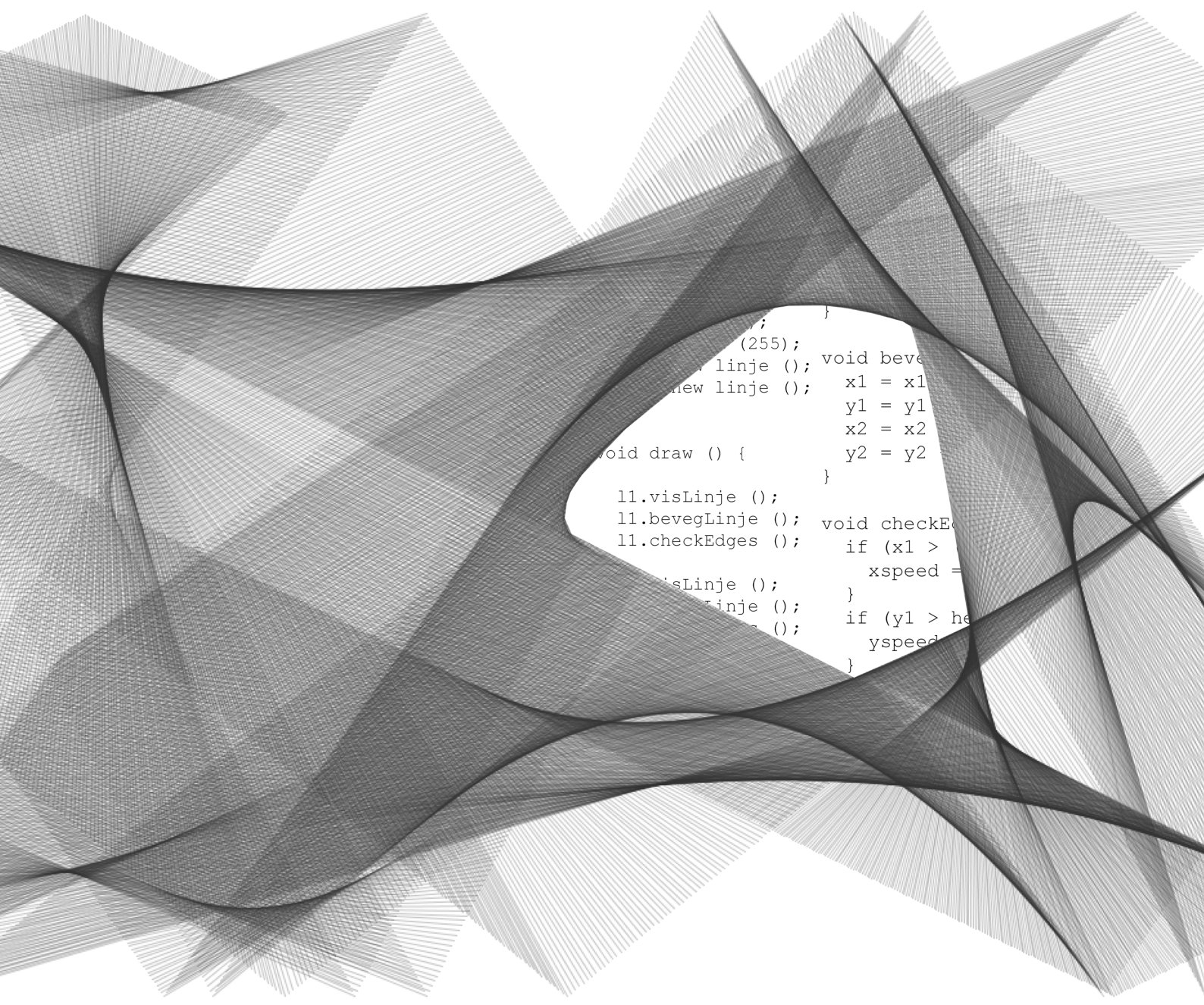


Programmering i Kunst og håndverk

Hvorfor programmere i Kunst og håndverk på grunnskolen?



Programmering i Kunst og håndverk
Hvorfor programmere i Kunst og håndverk på grunnskolen?

Master i estetiske fag: Fagdidaktikk - Kunst og design
Kandidatnummer 128

OsloMet – storbyuniversitetet
Fakultetet for teknologi, kunst og design
Institutt for estetiske fag
Emnekode: MEST5900

Stephanie Hoebeke, 2019

Forsidebilde:
Stephanie Hoebeke, 2018

En stor takk til...

Mine veiledere Peter Haakonsen og Ingri Strand for gode refleksjoner, konstruktive tilbakemeldinger og støtte.

Mine informanter Roger Antonsen, Liv Klakegg Dahlin, Anette Seip og Jon Øivind Hoem for deling av deres kunnskap og innsikt.

Min familie og venner for støtte og oppmuntring, og en spesiell takk til min mor og Marthe for gjennomlesning og korrektur og Robert for hjelp med oppsett og at du alltid er der for meg.

Mine medstudenter for faglige og interessant diskusjoner.

Sammendrag

Denne masteroppgaven ble basert på en pågående diskusjon om programmeringen i den norske grunnskolen. I skolepolitiske debatter angående fagfornyelsen som forventes å iverksettes i 2020, har programmering ofte vært knyttet til realfagene, men det har også vært knyttet til fagene Kunst og håndverk, Musikk og Samfunnsfag. Det har blitt mer satsing på algoritmisk tankegang og programmering i grunnskolen fra et internasjonalt perspektiv. Hovedargumentene for undervisning i programmering er at elevene må dannes og utdannes til et samfunn der digital teknologi stadig blir viktigere. I tillegg må elevene lære å skape teknologi, ikke kun skape med teknologi. Konseptet med *kreativ koding* har vært knyttet til en kontekst hvor uttrykk er viktigere enn funksjon, og hvor programmering har blitt referert til som et nytt materiale. Nye programmeringsplattformer har blitt utviklet og utvikles fortsatt, noe som gjør det enklere for barn og kunst studenter å lære seg å programmere. Med dette som utgangspunkt har jeg valgt å undersøke følgende problemstilling:

Hvilke mulige årsaksforklaringer kan belyse hvorfor programmering skal implementeres i kunst og håndverk på grunnskolen?

I formulering av problemstillingen, forskningsstrategien og strukturen for drøftingen har jeg benyttet kritisk realisme som vitenskapsteoretisk ståsted og rammeverk. For å besvare problemstillingen har empirien blitt innhentet ved hjelp av en blandet metode bestående av fire semistrukturerte forskningsintervju og en praktisk-estetisk undersøkelse gjennom *research by design*. Undersøkelsene har foregått parallelt og har komplementert hverandre. Basert på undersøkelsene ble det observert en uoverensstemmelse mellom fokuset til fagfeltet i Kunst og håndverk og det skolepolitiske. I tillegg kan dagens rammefaktorer innen kunst- og håndverksutdanningen gjøre det vanskelig for at programmering skal bli en del av faget. Likevel viste undersøkelsen at programmering har mye å tilby faget. Programmering gir elevene mulighet til å skape med teknologi innen mange forskjellige domener innen kunst og design. Det kan brukes som et verktøy for å forbedre fagets egenart der elevene får muligheten til dybdelæring innen sentrale begreper i Kunst og håndverk. I tillegg kan programmering belyse og styrke det algoritmiske i faget. Elevene kan få kjennskap til ulike materialer og et komplekst designbegrep. Programmeringen kan også bidra til et kunst- og håndverksfag som relaterer seg til elevenes hverdag, det kan hjelpe elevene til å utvikle en etterspurt ferdighet i fremtiden, samtidig som det gir mulighet for å styrke kritiske demokratiske skapeferdigheter.

Summary

This master's thesis was based on an ongoing discussion about programming in the Norwegian primary and secondary schools. In political debate on the renewal of the Norwegian curriculum, which is expected to take effect in 2020, programming has often been linked to the science subjects, but it has also been linked to the subjects of music, arts and crafts, and social studies. There has also been a great focus on computational thinking and programming in primary and secondary school from an international perspective. The main argument for teaching programming is that pupils must be prepared for and are educated in a society in which digital technology is becoming increasingly important. In addition, students must learn to create technology, not just to create with technology. The concept of *creative coding* has been linked to context where expression is more important than function. Thus, programming has been referred to as a new creative media. New programming platforms have been and are still being developed, which makes it easy for young children and art students to learn how to program. With this as a starting point, I have chosen to investigate the following issue:

What possible explanations of reasons can illustrate why programming should be implemented in arts and crafts at primary and lower secondary school levels?

In formulating the issue, the research strategy and the structure of the discussion, I have used critical realism as a scientific theoretical point of view and framework. In order to address the problem, the empirical data has been obtained using a mixed method consisting of four semi-structured research interviews and a practical aesthetic study involving *research by design*. The methods were conducted in parallel and complemented each other. Based on the studies, a discrepancy between the focus of the fields of arts and crafts and politics was observed. In addition, current frameworks in arts and crafts education can make it difficult for programming to become a part of the subject. Nevertheless, the study showed that programming has much to offer to the subject. Programming provides students with the opportunity to create with technology within many different domains in art and design. It can be used as a tool to enhance the subject's distinctive character, whereby students may receive in-depth learning in key arts and crafts concepts. In addition, programming can illuminate and strengthen computational thinking within the subject, allowing students to gain knowledge of different materials and complex design concepts. Programming is also a means whereby arts and crafts lessons can relate to students' everyday lives, and it may help students to develop a skill sought-after in the workplace while offering the opportunity for them to engage in critical democratic creation in the future.

Innholdsfortegnelse

Introduksjon til undersøkelsesfeltet	1
Problemområde	4
Oppgavens avgrensning	4
Problemstilling	4
Begrepsavklaring	5
Oppgavens struktur	6
Teoretisk forankring	11
Programmering i et historisk perspektiv	11
Programmering i grunnskolen	12
Programmering som en allmenn ferdighet	14
Digital kompetanse	15
Digital kompetanse i Kunst og håndverk	16
Programmering i Kunst og håndverk	17
Tredje utkast til kjerneelementene i Kunst og håndverk i fagfornyelsen	19
Kjerneelementene og skisser til den nye læreplanen i Kunst og håndverk	20
Forskningsstrategi	23
Kritisk realisme	23
Årsaksforklaringer og mekanismer	24
Aktør og struktur	24
Kvalitative metoder	25
Research by design	26
Semistrukturerte forskningsintervju	27
Utvalg	28
Transkribering og fremlegg av intervju	29
Analyse	29
Validitet og reliabilitet	29
Praktisk estetisk undersøkelse	33
Programmeringsspråk	33
Valg av programmeringsspråk og -plattformer	34
De ulike programmeringsspråkene og fremgangsmetode	34
Processing	35
BBC micro:bit	37
Arduino	38
Hva kan man programmere innen Kunst og håndverk	39
Presentasjon av intervjuene	43
Fremlegg av informantenes posisjonering	43
Roger Antonsen	43
Anette Seip	44
Jon Øyvind Hoem	44
Liv Klakegg Dahlin	44
Mulige positive årsaker	45
Aktualisering av Kunst og håndverk	45
Bevisstgjøring	47

Designprosess med integrert teknologi	48
Forståelse for verktøyenes særegenhet.	49
Tverrfaglighet	50
Til fordel for programmering	51
Mulige begrensende årsaker	52
Et fagfelt med manglende fokus og kompetanse	52
Miljø	53
Økonomi	53
Noe annet må ut	54
Oppsummering	54
Drøfting	59
Strukturelle betingelser	59
Fagfeltets holdninger	59
Programmering blir for teoretisk	60
Fokus på tradisjon	60
Politiske styringer	61
Et udefinert område	61
En utvidet definisjon	62
Rammefaktorer	63
Tid	63
Tid til kompetanseheving og planlegging av undervisning	64
Tid til undervisning i programmering	65
Økonomi	65
Økonomi til utstyr	65
Økonomi til kurs	66
Økonomi til mindre klasser	66
Strukturell utvikling	67
Indre kriterier	67
Dybdelæring ved bruk av programmering	67
Det algoritmiske i Kunst og håndverk	68
Kjennskap til ulike materialer	69
Et komplekst designbegrep med interaksjon	69
Ytre kriterier	70
Et kunst- og håndverksfag som relaterer seg til elevenes hverdag	70
En ettertraktet ferdighet	71
Kritisk demokratiske skapeferdigheter	72
Oppsummering	75
Veien videre	76
Litteraturliste	79
Figurliste	90
Vedlegg	92
Vedlegg 1, meldeskjema til NSD	92
Vedlegg 2, godkjenning fra NSD	97
Vedlegg 3, Informasjonsskriv	98
Vedlegg 4, intervjuguide	100

Introduksjon til undersøkelsesfeltet

Den stadig hurtige utvikling innen teknologi er noe som påvirker de fleste disipliner på en eller annen måte. Dette har også vært tilfelle innen kunst og design, og det finnes nye felt som er blitt skapt av ny teknologi, slik som spillutvikling, interaktiv media, multimedia og webutvikling. Teknologi som emne har lenge vært diskutert også i grunnskolen og er fortsatt like aktuell når det kommer til den nye fagfornyelsen som skal implementeres i 2020. Med læreplanen, Kunnskapsløftet som tredde i kraft i 2006 (LK06) ble det opprettet et tverrfaglig emne *Teknologi og design*, innen Matematikk, Naturfag og Kunst og håndverk. Både private- og statlige organisasjoner og interesseorganisasjoner har arbeidet sammen om å få inn teknologi i grunnskolen i nettverket *Teknologiinspiratørene* (Sanne et al., 2016). Teknologibegrepet blir ofte knyttet opp mot design og i rapporten *Teknologi og programmering for alle* av Sanne et al. (2016) skriver de «Et teknologisk produkt eller system kjennetegnes ved at det er et redskap, skal fylle en oppgave eller ha en funksjon. Produktene og systemene som utvikles må få en form, og teknologi og design er tett forbundet» (Sanne et al., 2016, s. 12). Videre sier de også at teknologien bygger på håndverkstradisjoner og «det kreative og utøvende i utformingen, utviklingen og design av produkter er en sentral del av teknologien» (Sanne et al., 2016, s. 11). Videre skiller de mellom naturvitenskapens mål og teknologiens mål, der vitenskapen prøver å forstå verden, men teknologien forsøker å løse problemer og skape noe nytt.

Selv om teknologi fortsatt er et hett område i den skolepolitiske diskusjonen, har det med den nye fagfornyelsen blitt et stadig større fokus på diskusjoner rundt programmering, algoritmisk tankegang og koding, som går innunder teknologibegrepet. Sanne et al. (2016) sier at den digitale teknologien er en så sentral del av både teknologien og dagens samfunn at dagens skole må beskrive og gi en tydeligere plass til digital teknologi og programmering. Deres anbefaling er å opprette et nytt grunnleggende fag innen teknologi og programmering i grunnskolen. Likevel har det i den nye fagfornyelsen ikke blitt opprettet et nytt fag. Begrepene programmering, koding og algoritmisk tankegang har blitt brukt og har blitt plassert i ulike fag med ulik tyngde, også i Kunst og håndverk (Kunnskapsdepartementet, 2018a; Utdanningsdirektorater (Udir), 2017c, 2017e, 2017f, 2018a, 2018b, 2018c). I media har også programmeringens plass i grunnskolen fått mye dekning, med ulike synspunkter og meninger. Noen har argumentert for at programmering er det nye håndverket og at det bør ha en sentral plass i flere fag i skolen eller på lik linje med Kunst og håndverk (Bergersen, 2017; Stærk, 2016). Andre, slik som Nina Andresen-Borud som er dramapedagog, teaterlærer og skuespiller stiller seg heller negative til at programmering skal bli en del av skolen, og påpeker at skolen heller må tilrettelegge for kreativitet. Hun legger til at kreativitet er den viktigste ferdigheten for den teknologiske utviklingen (Andresen-Borud, 2018).

Som nevnt over blir teknologi ofte knyttet til design i den norske grunnskolen. Det kan likevel virke som om programmering, algoritmisk tankegang og koding har fått et større fokus innen realfagene, enn i praktisk-estetiske fag. Som en del av og forberedelse til masteroppgaven har jeg vært med på ulike workshops på *Teknoteket Makerspace* ved Tekniske museum og *Makeriet* ved OsloMet. Jeg har også deltatt på fagfrokoster

ved Universitetet i Oslo, på *Seminar om programmering i skolen*, den 14. november 2018 og *Programmering i skolen – muligheter og utfordringer*, den 12. oktober 2018. Jeg var også deltaker på *Oslo Skaperfestival 2018* ved Deichman Hovedbibliotek. En opplevelse som jeg sitter igjen med er at programmering særlig blir nevnt i sammenheng med Matematikk og Naturfag, og at Kunst og håndverk kun nevnes i en bikommentar eller ikke blir nevnt. Det samme gjelder for brosjyrer slik som *Programmering i skolen* av Karin Nygård (2018), der fagene Kroppsøving, KRLE, Samfunnsfag og Naturfag blir nevnt, men ikke Kunst og håndverk. Masteroppgaver innen programmering og grunnskolen i Norge viser at de fleste er innen realfagsdidaktikk eller IKT (Bjørnevoll, 2016; Haaland & Rosvold, 2006; Iversen, 2015; Jåtten, 2006; Lang-Ree, 2016; Verstad, 2017). På udirbloggen.no blir det skrevet om programmering i matematikkfaget den 21. september 2017 av Kjerneelementgruppen for Matematikk. Her skriver de

«Styrende myndigheter har imidlertid kommet fram til at det ikke er ønskelig å etablere et nytt skolefag, men at denne opplæringen skal legges inn i matematikkfaget (...). Matematikk er det av dagens obligatoriske skolefag som best kan ta utfordringen ved å lære elevene algoritmisk tenkning og programmering» (Kjerneelementgruppen for Matematikk, 2017).

Likevel skriver Sevik (2016) fra *Senter for IKT i utdanningen* at:

«Det er mange fag i skolen der programmeringskompetanse kan være nyttig og bidra til å gjøre faget mer relevant og motiverende; realfag er åpenbare kandidater for programmering, men vi mener mulighetene er minst like store i de estetiske fagene. Programmering er et språk som muliggjør visuelle, musikalske og kunstneriske uttrykk, i tillegg til regneoperasjoner, algoritmer og måle- instrumenter» (Sevik, 2016, s. 26).

Diskusjoner om programmerings plass strekker seg også lengre enn det skolepolitiske. Som nevnte innledningsvis har teknologien ført til nye områder innen design og media, dette finner vi også igjen i kunsten. Det finnes grener innen dagens kunstinstusjon der programmering har blitt en del av utviklingen av kunstverk. Kreativ koding har også blitt et kjent begrep og brukes mye innen kunstretninger som har fokus på programmering som et middel for skapelse av uttrykk, for eksempler generativkunst, newmedia art og installasjoner. Innen den generative kunsten er målet å skape noe organisk ved hjelp av prinsipper innen det mekaniske (Pearson, 2011). Matt Pearson (2011) er en kreativ koder og beskriver generative kunst slik og refererer spesielt til programmering:

«Programming is an interface between man and machine: It's clean, logical discipline, with clearly defined aims. Art is an emotional subject, highly subjective and defying definition. Generative art is the meeting place between the two; it's the discipline of taking strict, cold, logical processes and subverting them into creating illogical, unpredictable, and expressive results» (Pearson, 2011, s.18)

I Persons beskrivelse, tolket av meg, kan det virke som om programmering og kunst er to helt forskjellige retninger. Et slikt skille kan virke kunstig og det eksisterer pågående diskusjoner om programmering er en kunst eller en kald og logisk vitenskap slik jeg oppfatter sitatet av Pearson. John Maeda er leder for designteamet ved *Automatic, inc* som blant annet har laget WordPress.com. Han har også arbeidet ved *MIT Media Lab* og har blant annet arbeidet med å få begrepet STEM (Science, Technology, Engineering and Mathematics) omgjort til STEAM (Science, Technology, Engineering, Art and Mathematics). Han har skrevet bøker som *Creativ Code* (2004) og *Media @Maeda* (2000). Maeda (2004) skriver følgende om teknologi og kunst:

« (...) the two complement each other in a necessary union of relevant vision united with relevant construction. To assume that the only model for art technology is through collaboration between the

artist and technologist, is to gamble the entire future of our culture upon the custom of figuratively grafting artist's eyes and senses directly into technologists' hands and minds » (Maeda, 2004, s. iv).

Donald Knuth (1974) som er professor i informatikk ved Stanford University var en av de første som startet denne diskusjonen i artikkelen *Computer Programming as an Art*. I artikkelen skriver Knuth om programmering:

«it applies accumulated knowledge to the world, because it requires skill and ingenuity, and especially because it produces objects of beauty. A programmer who subconsciously view himself as an artist will enjoy what he does and will do it better» (Knuth, 1974, s. 673).

Oppfatningen om at programmering har flere likheter med design og håndverk og blir definert som kunst finner vi hos flere (Amiri, 2011; Graham, 2004; Hunt & Thomas, 2001; Price, 2007). Colin B. Price (2007) som er professor i informatikk ved Universitetet i Worcester mener at man må begynne å se på programutvikling som en kreativ prosess og sier «As artists, we paint, as programmers, we code. While these are two different media of expression, they share one goal, to produce new artefacts» (Price, 2007, s. 44). Tanken om at programmering og koding er et håndverk slik som Bergersen (2017) og Stærk (2016) påpeker finner vi også igjen hos professor i sosiologi Richard Sennett. Sennett (2008) tillegger begrepet *Craftmanship* en bredere definisjon enn hvordan det norske begrepet definerer håndverk. Sennett mener at en håndverker er opptatt av to ting: « The first is the craftsman's desire to do good work: the second lies in the abilities required to do good work» (Sennett, 2008, s. 241). I Sennetts forklaring av begrepet nevner han også programmering i Linux til å være *craftmanship*. Truls Bergersen som er 1. vara til utvalg for barn og unge i Bærum Frp omtaler i artikkelen *Programmering i Bærumsskolen* den 18. Oktober 2017 i Budstikka, at programmering er det nye håndverket. Han argumenterer for at digitalisering er en stor del av samfunnet vårt, dermed må man bygge opp en grunnleggende forståelse av programmering. Han mener ikke at alle skal lære seg programmering flytende, men at alle bør kunne kode i noe grad. Han setter IT-prosjekter opp mot byggeprosjekter for å vise behovet for mange ulike kompetanser innen IT-verden. Slik som i byggebransjen har man mange ulike kompetanser innen IT-bransjen, Bergersen forklarer det slik

«Omsatt til IT-verdenen trenger man også prosjektledere og arkitekter, man trenger dem som har peiling på lagring, andre som har peiling på nettverk. Noen som tar seg av servere og andre som sørger for databasen. Og så har man dem som bygger applikasjonene» (Bergersen, 2017).

Jeg har fulgt diskusjoner angående problematikken rundt programmering i studietiden min på masteren i *Fagdidaktikk kunst og design* ved OsloMet. Interessen for programmering har sitt utspring i valg av mine fag som er Matematikk, Naturfag og Kunst og håndverk, men også i arbeidet som forskningsassistent. Våren 2018 startet jeg som forskningsassistent ved OsloMet, i et prosjekt som sammenkoblet kunstig intelligens og kunst. Denne erfaringen ga meg mulighet til å utforske området som ligger mellom kunst og teknologi og har gitt meg en større forståelse for betydningen av programmering. Som innledningen viser, menes det mye og mangt om programmering som fag i skolen. Jeg mener at arbeidet med denne masteroppgaven har gitt meg en spennende mulighet for å undersøke problematikken i større grad, og jeg vil med dette redegjøre for problemområde og problemstilling nedenfor.

Problemområde

I denne oppgaven har jeg som mål å undersøke hvorfor programmering bør bli en del av Kunst og håndverk. Som innledningen viser er det kanskje ikke naturlig at programmering har en plass i Kunst og håndverk. Likevel er det flere som kommenterer at programmering er et nytt håndverk, et materiale og at det er en kreativ prosess. Mine refleksjoner og diskusjoner i løpet av det første året på masterutdanningen har omhandlet hvilke fag programmering bør bli plassert. I denne masteroppgaven ønsker jeg å sette et søkelys på og fortsette diskusjonen om hvorfor programmering bør bli en del av Kunst og håndverk i grunnskolen. Et underliggende ønske er at oppgaven kan være et bidrag i diskusjonen som omhandler problematikken, og oppgaven vil derfor ikke gi et endelig svar.

Oppgavens avgrensning

For å avgrense oppgavens omfang har jeg valgt å fokusere på *hvorfor* og ikke på *hva* og *hvordan*. *Hva* og *Hvordan* programmering skal bli implementert i Kunst og håndverk er fortsatt et veldig relevant spørsmål, jeg håper at denne oppgaven kan være en start på å besvare disse problemstillingene på et senere tidspunkt. I oppgaven har fokuset vært på å intervjuere personer som har kjennskap til og kunnskap innen programmering, men også innen Kunst og håndverk på grunnskolen. Grunnen for valget er at det har vært vanskelig å finne lærere som kan noe om programmering og som samtidig underviser i Kunst og håndverk, da programmering i grunnskolen fortsatt er et relativt nytt emne. I oppgaven har det også vært benyttet *Research by design* for å kartlegge og få oversikt over programmering som en utøvende praksis, og for å få svar på hvilke skapemuligheter som finnes i programmering knyttet mot Kunst og håndverk. Refleksjonen rundt den praktisk-estetiske undersøkelsen har også vært et bidrag og grunnlag for drøftingen, selv om den i større grad besvarer hvilke skapemuligheter som ligger i programmering knyttet mot Kunst og håndverk. For å vite *hvorfor*, er det også grunnleggende å ha en innsikt i hva programmering omhandler når det knyttes til kunst og håndverksfaget. Ønsket er ikke å argumentere for at programmering skal bli implementert fremfor andre emner i Kunst og håndverk, ei heller å argumentere for at programmering skal inn i Kunst og håndverk istedenfor andre fagdisipliner. Formålet har vært å kartlegge mulige årsaker for hvorfor programmering kanskje bør inngå i faget. Gjennom intervjuene har det også blitt belyst begrunnelser som kan gjøre det vanskelig for at programmering skal bli en del av grunnskolen. For å svare på problemstillingen vil begge aspekter bli tatt med.

Jeg har også valgt å skrive fag i grunnskolen med store bokstaver, dette for å presisere at det er skolefag jeg omtaler. I oppgaven vil det også bli representert ulike bilder som skiller hvert kapittel, dette er for å belyse hva programmering kan være i Kunst og håndverk, for slik å gi en større innsikt i hvorfor. Illustrasjonene er laget av meg ved hjelp av programmering i programmeringsspråkene Processing og Scratch. Kodene som er laget for utvikling av bildene vil bli vist på den neste siden. Illustrasjonene vil bli nevnt i kapittel om den praktisk-estetiske undersøkelsen.

Problemstilling

Jeg har valgt å posisjonere meg i et kritisk realistisk perspektiv. Ifølge byplanforsker Petter Næss (2012) kan man innen vitenskapsposisjonen komme med predikasjoner og dermed kunne si noe om en mulig fremtid, selv om jeg er innenfor et samfunnsvitenskapelig domene. I denne oppgaven er dette en viktig faktor der

jeg både ønsker å se på eksisterende forhold rundt programmering og Kunst og håndverk som kan ha innvirkning på om programmering bør bli en del av faget, såvel som på mulige fremtidsperspektiver. Utfra valg av problemområde og kritisk realisme som vitenskapeligstøttet har jeg valgt å ta for meg følgende problemstilling:

Hvilke mulige årsaksforklaringer kan belyse hvorfor programmering skal implementeres i Kunst og håndverk på grunnskolen?

Begrepsavklaring

Programmering

Definisjonen av programmering har vært under stadig forandring i og med at programvare, maskinvare og bruken av datamaskiner har forandret seg, man kan også tenke seg at definisjonen vill forandre seg med tiden, siden den teknologiske industrien er i konstant forandring (Duncan, Bell & Tanimoto, 2014). Et av de vanligste kjennetegnene til programmering er at brukeren ikke deltar direkte i manipulasjon. Dette betyr at sekvensen av handlingene er bestemt på forhånd, den som programmerer må deretter vente og se om sekvensene av det en har gjort gir et ønsket resultat. Som et eksempel er det å skrive i et tekstbehandlingsdokument ikke programmering (Duncan, Bell & Tanimoto, 2014). På grunnlag av dette ønsker jeg i denne oppgaven å benytte meg av en bred definisjon av programmering. Programmering vil bli forstått som å kunne bryte et gitt problem i mindre deler som kan bli omgjort til kommandoer som datamaskinen kan gjennomføre (Sanne et al., 2016, s. 18).

Koding

Koding blir ofte brukt som et synonym til programmering. I denne oppgaven ønsker jeg likevel å skille begrepene fra hverandre, der koding vil omhandle ferdigheten «å skrive kode», og dermed blir koding en gjennomføringsdel av programmeringen (Hiltunen, 2016, s. 8). I undervisningssammenheng er det viktig å skille begrepene fra hverandre og ha tydelige forklaringer på hva som menes med hvert begrep. Hvis koding kun betyr å taste inn riktig syntaks, kan dette føre til at elevene kun lærer å bruke eksisterende koder. Men hvis koding er en del av programmering, vil det å lære koding bety at en elev også må kunne skape nye koder.

Algoritmisk tankegang

For å kunne si noe om programmering og koding i utdannelsen kommer man ikke unna begrepet *Computational thinking* (CT) som på norsk er blitt oversatt til *algoritmisk tankegang*. Det er likevel viktig å ikke blandet det engelske begrepet *algorithmic thinking* med det norske begrepet, da det engelske begrepet inneholder noe annet enn det norske. Det engelske begrepet *algorithmic thinking* blir ofte lagt under paraplybegrepet CT (Bocconi, Chiocciariello & Earp, 2018). Jeannette Wing (2006) var en av de første som definerte og beskrev CT, der «Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science. Computational thinking includes a range of mental tools that reflect the breadth of the field of computer science» (Wing, 2006, s. 33). Ferdighetene som kommer inn under CT trenger ikke å bli utviklet ved å drive med programmering, men programmering kan være et verktøy til å utvikle ferdigheten (Grover and Pea, 2013). Med dette som grunnlag, velger jeg i oppgaven å støtte meg på Sanne et al. (2016) sin beskrivelse om hva som er målet med algoritmisk tankegang. De skriver at algoritmisk tankegang er «å løse et problem gjennom å spesifisere en presis sekvens av kommandoer (...) en slik presis sekvens av kommandoer kalles en algoritme» (Sanne et al. 2016, s.18).

Maskinvare

Maskinvare vil bli forstått som de fysiske delene av en datamaskin som det er mulig å programmere. Dette er for eksempel ledninger, skjerm, lys eller servomotorer som man kan se eller røre (Vorderman, 2017, s. 219). Dermed blir maskinvare det fysiske som det er det mulig å programmere. Når jeg i denne oppgaven skriver om programmering av maskinvare, er det innforstått at man skaper en programvare til de fysiske aspektene ved en datamaskin, og uten programvare vil ikke maskinvarene hatt en funksjon.

Programvare

Programvare vil bli brukt om de programmene som kjøres på en datamaskin og er med på å styre hvordan datamaskinen fungerer (Vorderman, 2017, s. 219). Dermed er programvare det som gjør maskinvare funksjonelt. For at man skal kunne interagere eller styre en maskin er man dermed nødt til å ha en programvare. Slik som at maskinvare ikke ville vært funksjonelt uten programvare, vil heller ikke programvareutvikling være mulig uten maskinvare.

Informatikk

I oppgaven har jeg valgt å bruke informatikk istedenfor *datavitenskap* ved oversettelse av det engelske begrepet *computer science*. Jeg velger å støtte meg på Sanne et al. (2016) sin beskrivelse, der de beskriver informatikk slik «(...) Informatikk som kan defineres som læren om hvordan datasystemer designes, konstrueres og brukes, og faget handler om programmering, design, datamaskinarkitektur, nettverk og systemer» (Sanne et al, 2016, s. 19).

Programmeringsplattform

Sammen med de ulike programmeringsspråkene kommer det også andre konsepter som utviklingsmiljø, maskinvare, utviklingskort med mikrokontroller osv. Jeg velger å kalle dette for en programmeringsplattform i denne oppgaven. Et eksempel er BBC micro:bit. BBC micro:bit er ikke et programmeringsspråk, men er en ettkortsdatamaskin som kan programmeres i et utviklingsmiljø. Et utviklingsmiljø er en programvare der man skriver koden. BBC micro:bit kan bli programmert med mange ulike programmeringsspråk. Når det kommer til for eksempel Scratch og Processing heter både programmeringsspråket og utviklingsmiljøet det samme. Jeg kommer til å bruke begrepene programmeringsplattformer og programmeringsspråk om hverandre, selv om dette blir noe upresist. Grunnen til dette er for å bruke begreper som leseren kanskje har kjennskap til, men også for å poengtere at man trenger maskinvare og programvare for at programmeringen skal være mulig.

Oppgavens struktur

Oppgaven innledes med kapittel 1, *Introduksjon til undersøkelsesfeltet*. Her presenteres oppgavens bakgrunn, avgrensning, problemstilling og begrepsavklaring.

I kapittel 2, *Teoretisk forankring* fremlegges relevant teori rundt oppgavens problemstilling. Kapitlet starter med en kort presentasjon av programmeringsposisjon i skolen. Så vil relevante teori om programmering knyttet til grunnskolen bli belyst sammen med meninger om hvorfor programmering og algoritmisk tankegang bør bli en del av grunnskolen. Deretter kommer en presentasjon rundt diskusjoner om digital kompetanse og digital kompetanse i Kunst og håndverk. Kapitlet avsluttes med teori rundt programmering, Kunst og håndverk og programmeringsposisjon i fagfornyelsen med tilknytning til Kunst og håndverk.

I kapittel 3, *Forskningsstrategi* vil de metodiske valgene og det vitenskapelige ståstedet bli representert.

Valgene som er tatt underveis i oppgaven vil bli fremlagt og kapittelet avsluttes med refleksjoner rundt validitet og reliabilitet.

I Kapittel 4, *Praktisk-estetisk undersøkelse* belyses den praktiske undersøkelsen som har foregått parallelt med det teoretiske arbeidet, dette for å få en praktisk og teoretisk forståelse for programmering, men også for å for å svare på hvilke skapemuligheter programmering har i Kunst og håndverk. Her belyses valg av programmeringsspråk, den praktiske prosessen, samt skapermuligheter.

I kapittel 5, *Presentasjon av intervjuene*, presenteres empirien som er hentet fra intervjuene. Kategoriene som er satt og som blir belyst er hentet gjennom en tilnærming basert på Grounded Theory.

I kapittel 6, *Drøfting*, drøftes empirien fra intervjuene og refleksjonene fra den praktisk-estetiske undersøkelsen opp mot relevant teori. Strukturelle betingelser og strukturell utvikling som er begreper innen kritisk realisme vil bli brukt for å finne mulige årsaksforklaringer på hvorfor programmering bør blir en del av Kunst og håndverk. Dette både for å belyse eksisterende strukturer, men også for å kunne se på mulige fremtidige strukturer.

Opgaven avsluttes med kapittel 7, *Oppsummering og veien videre*. Her oppsummeres funnene fra oppgaven og refleksjoner rundt veien videre vil bli lagt frem.



```
void setup () {  
  background (255);  
  size (650,700);  
}  
  
void draw () {  
  strokeWeight (0.5);  
  stroke (100,70);  
  if (mousePressed) {  
line(mouseX,350,mouseX,mouseY);  
  }  
}
```

Teoretisk forankring

I dette kapitlet vil det komme et fremlegg av relevant litteratur knyttet til problemområdet. Jeg vil først starte med et historisk perspektiv om programmering i skolen, som følges opp av hvordan programmering blir fremstilt i dagens skole. Deretter vil jeg legge frem argumenter som blir brukt for hvorfor programmering og algoritmisk tankegang bør være en del av grunnskolen. Jeg vil så gjøre rede for rammeverket til digital kompetanse og vil spesielt se på Kunst og håndverk. Kapitlet avsluttes med relevant teori rundt programmering, kjerneelementene og Kunst og håndverk. Litteraturen består av både vitenskapelige publikasjoner, politiske dokumenter, artikler og meninger vedrørende problematikken.

Programmering i et historisk perspektiv

Et av de tidligste forsøkene med å få programmering inn i skolen er gjort av Seymour Papert på 1960-tallet med *Mindstorms*, der han var med på å utvikle det som kalles LOGO-språket og tok det med inn i klasserommet. LOGO-programmeringsspråket var laget slik at nesten alle elever kunne lære seg å programmere hvis riktige og gode ramme faktorene lå til rette for det (Duncan et al., 2014). LOGO-språket har senere utviklet seg til LEGO Mindstorms og videre til programmeringsplattformen Scratch (Nygård, 2018). Papert var også med å utvikle læringsteorien *Constructionism*. Constructionism bygger på og deler samme syn som konstruktivismen. Constructionism fokuserer i større grad på kunsten å lære, altså lære det å lære og betydningen av å lage og skape ting når man lærer, og henter inspirasjon fra kunstundervisningen. I motsetning til Piaget, som hadde mer fokus på den indre mentale stabiliteten av erfaringer og kunnskap, fokuserte Papert heller på «How knowledge is formed and transformed within specific context, shaped and expressed through different media, and processed in different people`s minds» (Ackermann, 2001, s. 8). Papert så på algoritmer som en måte å beskrive våre handlinger i verden på, og dermed ble algoritmen også en måte å kunne reflektere over sine egne handlinger. Ved å få barn til å skrive ned algoritmene deres, mente Papert at man bedre kunne hjelpe barna med å reflektere over sine egne ideer og dermed også påvirke deres konstruksjoner av kunnskap. Dermed ble datamaskiner et godt medium som skapte kontekst for å konstruere kunnskap (Ko, 2017).

På 80-tallet ble det tilbudt et valgfag i den norske ungdomskolen som het *elektronisk databehandling* (EDB) der det ble gitt en grunnleggende opplæring i programmering. Dette faget ble også et tilbud på noen få barneskoler, og faget kunne velges som linjefag på enkelte videregående skoler (Braathe & Bakketun (1981) og Fylgje (1983) omtalt i NOU 2013:2). Men, per dags dato er ikke programmering lenger et valgfag på barneskoler, og kun noen utvalgte ungdomskoler tilbyr valgfaget Programmering som et forsøk siden 2016-2017 (Udir, 2017a).

Programmering i grunnskolen

I 2013 kom Digitutvalget som ble oppnevnt i statsråd 24. juni 2011, med rapporten *Hindre for digital verdiskaping*. Her skriver Digitutvalget blant annet at det i den norske skolen er en vektlegging på teknologi som verktøy for å lære annen fagkunnskap (NOU 2013:2). I rapporten skriver Digitutvalget at «Bruken av digitale verktøy for kommunikasjon i digitale medier har vært hovedmålet for norsk grunnutdanning. Det har blitt lagt for lite vekt på å skape teknologi og det er for dårlig allmenn forståelse av teknologisamfunnet» (NOU 2013:2, s. 10). Digitutvalget mener at den norske skolen vektlegger kompetanse innen bruk av digitale medier på å formidle og kommunisere, og dermed vil elevene bli konsumenter og videreformidlere, men ikke skapere av digital teknologi. I rapporten anbefaler Digitutvalget å innføre programmering som et valgfag og utvide definisjonen av digitale ferdigheter slik at det «ikke bare omfatter kommunikasjon og presentasjon, men også beregning, analyse og en generell teknologiforståelse som igjen bidrar til generelle skaperferdigheter» (NOU 2013:2 s. 106). Rundt 2013 oppsto det også ulike kurs- og aktivitetstilbud utenfor skolen. Lær Kidsa koding og nettstedet code.org ble opprettet i 2013, siden da har en hel programmering- og skaperbevegelse oppstått internasjonalt så vel som i Norge, som også har røtter i skolesystemet (KidsaKoder, u.å.b).

I 2016 kom rapporten *Teknologi og programmering for alle*, der en ekstern arbeidsgruppe på oppdrag fra Utdanningsdirektoratet fremlegger hvilke kompetanser elevene innen den norske grunnskolen skal ha i fremtiden når det kommer til teknologi, digitale ferdigheter og programmering (Sanne et al., 2016). De påpeker, slik som Digitutvalget også gjorde, at det er store avvik mellom den teknologiske virkeligheten elevene møter og det elevene lærer på skolen i dag. En av konklusjonene er at det bør bli opprettet et nytt fag i grunnskolen innen teknologi og programmering. Senter for IKT i utdanningen mener at programmering på lengre sikt bør komme inn under et eget fag, slik som blir foreslått i rapporten *Teknologi og programmering for alle*. På kort sikt mener de at det er mer realistisk at programmering kommer under eksisterende fag og læreplaner (Sevik, 2016). Før denne tid, sommeren 2015 kom Ludvigsen-utvalget med utredningen *Fremtidens skole – Fornyelse av fag og kompetanser*. I utredningen kommer utvalget med en vurdering av hvilke kompetanser som vil være viktige for elevene i fremtiden. Her vektlegges det å kunne bruke ulike digitale verktøy og datamaskinen på en sikker måte som de mest sentrale ferdighetene innen digital kompetanse. Utvalget skriver ikke noe om programmering, koding eller algoritmisk tankegang. Ludvigsen-utvalget viser til kompetansene å kunne tenke kritisk, kommunisere og samhandle og kommer inn på at digital kompetanse er avgjørende for innovasjon og teknologisk utvikling i næringslivet og i offentlig virksomheter (NOU 2015:8). Det kan stilles spørsmål ved om utvalgets beskrivelse av digital kompetanse stemmer overens med den digitale kompetansen som etterspørres av næringslivet og offentlig virksomhet.

I forbindelse med fagfornyelsen kom regjeringen den 25. august 2017 ut med pressemeldingen *Mer koding og teknologi inn i skolen*. I pressemeldingen omtaler statsminister Erna Solberg, kunnskapsminister Torbjørn Røe Isaksen og barne- og likestillingsminister Solveig Horne *Fremtid, Fornyelse og digitalisering - Digitaliseringsstrategi for grunnopplæringen 2017-2021* (Kunnskapsdepartementet, 2017). I strategidokumentet for digitalisering kommer det frem at regjeringen vil

«Vurdere hvordan teknologiforståelse, algoritmisk tenkemåte og programmering skal inngå i bestemte fag, innføre valgfag i programmering som permanent ordning fra 2019 og innføre nasjonalt forsøk med programmering og modellering som programfag i videregående opplæring fra høsten 2018.» (Kunnskapsdepartementet, 2017, s. 20).

De ønsker også i forbindelse med fagfornyelsen å vurdere «hvordan teknologi, programmering og

algoritmisk tenkemåte kan inngå i bestemte læreplaner for fag, særlig matematikk og naturfag» (Kunnskapsdepartementet, 2017, s. 18). Her ser vi at det ikke er en overenstemmelse med rapporten *Teknologi og programmering for alle* utredet av Sanne et al. (2016), som anfører at teknologi og programmering bør bli et eget fag. De politiske føringene henviser dermed til at programmering bør inngå i de eksisterende fagene i grunnskolen med vektlegging på Matematikk og Naturfag og at valgfaget programmering skal bli en permanent ordning (Kunnskapsdepartementet, 2017, s. 18). I *Jeløya-plattformen* den 14. januar 2018 som er en politisk plattform utgått av Høyre, Fremskrittspartiet og Venstre, er et av punktene «Sikre at elever får kunnskap om og forståelse for teknologi, algoritmisk tenkning og koding i flere fag allerede tidlig i skoleløpet» (Statsministerens kontor, 2018, s. 59). I *Fremtid, fornyelse og digitalisering - Digitaliseringsstrategi for grunnsopplæringen 2017-2021* (Kunnskapsdepartementet, 2017) blir Matematikk og Naturfag nevnt som fag som spesielt skal ha ansvar for programmering, men i den politiske plattformen bruker de benevnelsen flere fag uten å henviser til spesifikke fagfelt. I *Granavolden-plattformen* som kom ut den 17 januar 2019, vektlegges digital kompetanse og innovasjon. Også her går regjeringen videre med å si at de ønsker et digitalt løft på skolen med vekt på utvikling av teknologiforståelse, digital dømmekraft og skaperkraft, og vil sikre at elevene allerede tidlig i skoleløpet får kunnskap om og forståelse for teknologi, algoritmisk tankegang og koding i flere fag (Statsministerens kontor, 2019)

Fra 2017 til 2018 har en kjerneelementgruppe i hvert fag arbeidet med å utvikle kjerneelementer for alle fagene i skolen. I løpet av denne perioden har det vært tre innspillsrunder på kjerneelementene i hvert fag, og den 26. juni 2018 fastsatte Kunnskapsdepartementet kjerneelementene i de ulike fagene (Kunnskapsdepartementet, 2018a). Programmeringen har blitt plassert under ulike fag med ulik tyngde gjennom hele prosessen. I det første og andre utkastet av kjerneelementene ble *algoritmisk tenkemåte* plassert under matematikkfaget og under algoritmisk tenkemåte nevnes programmering (Udir, 2017c, 2017e). I det andre utkastet har algoritmisk tankegang og programmering også kommet inn i Naturfag under kjerneelementet, *teknologi i naturvitenskap* (Udir, 2017f). I det tredje utkastet av kjerneelementene har programmering fått en sentral plass i Matematikk og en mindre plass i Naturfag, men her blir også *kode* nevnt under kjerneelementet *visuell kommunikasjon* i Kunst og håndverk med punktet «kode og modellere visuelle uttrykk» (Udir, 2018a, 2018b, 2018c). I det endelige utkastet av kjerneelementet har programmering blitt plassert under Matematikk, Naturfag, Musikk og Kunst og håndverk, der Samfunnsfag får et spesielt ansvar for digitale ferdigheter (Kunnskapsdepartementet, 2018a).

Sett fra et internasjonalt perspektiv ligger Norge etter mange land i Europa når det kommer til å implementere programmering og algoritmisk tankegang i utdanningsløpet. I 2017 var det mer enn 20 europeiske land som hadde inkorporert programmering eller algoritmisk tankegang som en del av læreplanen (Balanskat, Engelhardt & Ferrari, 2017). De ulike land har ofte ulike argumenter og mål for hvorfor algoritmisk tankegang og programmering har blitt implementert, som for eksempel å utvikle elevenes ferdigheter i logiske tenkning, problemløsning og programmering, få flere elever til å studere informatikk senere, og for å øke kompetansen i matematikk (Bocconi, Chiocciariello, Dettori, Ferrari & Engelhardt, 2016).

I de nordiske landene, ble programmering og algoritmisk tankegang inkorporert i læreplanene i 2017 i Sverige og i 2016 i Finland. I begge landene har programmering blitt inkludert som en del av den digitale kompetansen og er en del av målene i spesifikke fag. I Finland er programmering en sentral del i matematikkundervisningen fra første til tredje klasse, og brukes i håndverk fra tredje til niende klasse. I håndverk er målet at elevene skal praktisere programmering i funksjoner slik som robotikk. Deretter er målet at elevene skal designe og produsere produkter som har implementerte systemer og programmering (Kwon & Schroderus, 2017). Dette blir også referert til som teknisk sløyd, som har programmering og ulike dataassisterende produksjonsteknikker sammen med håndverk (Porko-Hudd, 2017). I Sverige er det

Matematikk, Teknologi og Samfunnsvitenskap som har programmering som en del av den faglige læreplanen (Bocconi et al., 2018). Den europeiske kommisjon (EU-kommisjonen) kom med en handlingsplan for digital utdannelse, der to av målene er å få kodeklasser i alle skoler i Europa samt å øke alle skolers deltakelse i EU kode uka (EU-Kommisjonen, 2018).

Programmering som en allmenn ferdighet

Ulike land har ulike mål når det kommer til å implementere programmering og algoritmisk tankegang inn i skolen. Likevel er det ofte noen overordnede mål for hvorfor programmering og algoritmisk tankegang har blitt og skal bli en del av læreverket. De tre hyppigste årsakene for hvorfor, er for det første at programmering er knyttet til nødvendige ferdigheter for det 21. århundre, for det andre at næringslivet vil ha behov for personer med programmeringsferdigheter, og for det tredje at elevene skal ha en forståelse for et stadig mer digitalisert samfunn.

Det første argumentet går ut på at elevene skal utvikle viktige ferdigheter for det 21. århundre. Her går det første perspektivet ut på at for å skape teknologi og produsere digitalt, kreves det forståelse og ferdigheter innen programmering, men også andre sentrale ferdigheter, argumentet knytter seg opp mot behovet for denne kompetansen i næringslivet (Sevik, 2018). Det andre perspektivet handler om at kunnskap og ferdigheter i algoritmisk tankegang vil støtter andre tverrfaglige og viktige ferdigheter innen i det 21. århundre slik som problemløsning, analytisk- og logisktenkning (Balanskat et al., 2017; EU-kommisjonen, 2018; Sevik, 2018).

Det andre argumentet går ut på at næringslivet i fremtiden vil ha behov for personer som kan programmering og at det er en økende etterspørsel i jobbmarkedet, der ferdigheter innen informatikk i større grad blir etterspurt (Balanskat et al., 2017; Sevik, 2018). Noen argumenterer for at denne forskjellen er spesielt rettet mot kultur, der yrker innen programmering og informatikk er høyt ettertraktet i blant annet Kina og India (Duncan et al., 2014). Det er ikke kun etterspørsel i arbeidsmarkedet som er et av problemene, i digitaliseringsstrategien fra Kunnskapsdepartementet (2017) nevner departementet at det norske samfunnet står ovenfor store utfordringer og nevner blant annet eldrebølgen, det grønne skifte og endringer i arbeidsformer. Videre skriver departementet:

«For å utnytte mulighetene digitalisering gir er det behov for stadig både mer spesialisert og bedre generell IKT-kompetanse i samfunnet. På den ene siden trenger samfunnet flere IKT-spesialister og flere personer med tverrfaglig IKT-kompetanse. På den andre siden trenger alle en generell digital kompetanse som gjør oss i stand til å utnytte tjenester som utvikles, utføre arbeidsoppgaver ved bruk av IKT, gjøre sikre valg i vår digitale hverdag og sikre vårt personvern» (Kunnskapsdepartementet, 2017, s. 6).

Det siste argumentet handler om at elevene skal forstå et samfunn som er stadig mer digitalisert (Sevik, 2018). Ved å ha en forståelse for algoritmisk tenkning og programmering vil elevene kunne ha en fullverdig deltagelse i den digitale verden (Bocconi et al., 2016). Dette betyr at alle fagfelt trenger en forståelse av hvordan et digitalisert samfunn fungerer. Som tidligere nevnt mener Digitutvalget at en slik kompetanse «forutsetter at representanter for alle profesjoner og utdanninger ikke bare må se på digitale tjenester som verktøy, men også forstå hvordan de fungerer» (NOU 2013:2 s. 106). Sanne et al. (2016) skriver

«Det norske samfunnet bygger på grunnleggende verdier som demokrati, likhet, likeverd, likestilling og respekt for mangfold. Det forutsetter at innbyggerne forstår de sentrale, grunnleggende ideene og de prinsippene som preger samfunnet. Dermed er bred kunnskap om teknologi et viktig element i det

å sørge for at disse verdiene kan forbli grunnleggende for samfunnet vårt» (Sanne et al., 2015, s. 22).

Videre pekes det på at en bredde i teknologifaget må til, og at dette gir mulighet for å nå ulike elevgrupper (Sanne et al., 2016).

En annen grunn for at det argumenteres for at programmering og algoritmisk tankegang skal bli en del av grunnskolen for alle elever og ikke bare være et valgfag, er for å redusere gapet som eksisterer mellom elever med høy og lav sosioøkonomisk bakgrunn og redusere kjønnsulikheter når det kommer til valg innenfor studier knyttet til informatikk. I hele Europa ser man et fåtall av kvinner som velger studier innen realfag, teknologi, ingeniørfag og matematikk (EU-Kommisjonen, 2018). I året 2017 til 2018 var det 5176 elever (3%) som valgte valgfaget *Programmering* på ungdomskolen og 83,3 % av disse var gutter (Udir, 2018d). Dermed er programmering det valgfaget som har flest gutter i forhold til jenter, i motsetning til *Design og redesign* som har flest jenter i forhold til gutter (Udir, 2018d). I Norge har denne problemstillingen også vært undersøkt. Hilde G. Corneliusen og Fay Tveranger (2018) mener at pilotprosjektet ved å introdusere programmering ikke la til rette for å inkludere ulike kjønn. For det første mener Corneliusen og Tveranger at man må skape en bevissthet rundt problematikken. For det andre mener de at man må skape et nytt bilde av programmering som er mer mangfoldig, variert og inkluderende for begge kjønn. En kan likevel stille spørsmål om implementering av programmering og algoritmisk tankegang hindrer kjønnsgapet, da man i land slik som Malaysia, ser at omtrent like mange kvinner som menn velger karrierer innenfor informatikk (Duncan et al., 2014). Det kan være andre faktorer slik som kultur, som påvirker om kvinner eller menn kommer til å fortsette med programmering, eller ikke. Det kan også vises til Finland, der jenter gjør det bedre innen teknologi og realfag, men at det er guttene som velger en karriere innen informatikk (Kwon & Schroderus, 2017). Videre kan det bli interessant å se om en innføring av teknisksløyd vil påvirke jentenes karrierevalg i Finland.

Digital kompetanse

Algoritmisk tankegang og programmering blir ofte nevnt under digital kompetanse i utdanningsammenheng. Men hva er digitalkompetanse og hva vil digitalkompetanse innebære i Kunst og håndverk? Digital kompetanse er en flytende størrelse, da teknologi stadig er i forandring. Begrepet er også avhengig av konteksten, da tid, sted, alder og bruksområde vil være med på å definere hva begrepet innebærer (Bjarnø, Giæver, Johannesen & Øgrim, 2017). Dette forårsaker ofte at elevenes kompetansemål blir overordnede og generelle da det faglige innholdet enda er i endring (Hatlevik & Thronde, 2015). Det finnes ulike rammeverk som definerer overordnede mål for hva digital kompetanse vil innebære. I det reviderte rammeverket for digitale ferdigheter under grunnleggende ferdigheter i LK06 blir det nevnt fem ulike ferdighetsområder innen digitale ferdigheter (Udir, 2016). De fem områdene er *bruke og forstå, finne og behandle, produsere og bearbeide, kommunisere og samhandle og utøve digital dømmekraft*.

Senter for IKT i utdanning ga i 2016 ut *Monitor skole 2016 – Skolens digitale tilstand* med formålet å kartlegge skolens digitale tilstand ved hjelp av undersøkelser. I rapporten sammenligner de det eldre norske rammeverket for grunnleggende ferdigheter innen det digitale med *The European Digital Competence Framework for Citizens* (DigComp) sitt rammeverk. I den eldre definisjonen av digitale ferdigheter blir det nevnt fire områder, *tilegne og behandle, produsere og bearbeide, kommunisere og digital dømmekraft* (Egeberg, Hultin & Berge, 2016). DigComp har skapt en standard for å forbedre samfunnets digitale kompetanse, planen består av fem generelle områder, med 21 kompetansemål. De fem overordnede områdene er *Information and data literacy, Communication and collaboration, Digital content creation, Safety og Problem*

solving (Vuorikari, Punie, Carretero & Van Den Brande, 2016). I *Monitor skole 2016 – Skolens digitale tilstand* blir det kommentert at to av områdene som skiller rammeverkene er *programmering* og *problemløsning*, der programmering blir nevnt under *Digital content creation*. En av konklusjonene fra sammenligningen er at «Det norske rammeverket er mer orientert mot å skape med teknologi enn å skape teknologi (blant annet gjennom programmering)» (Egeberg et al., 2016, s.22). Denne vurderingen kan også sammenlignes med Digitutvalget som sier at man må utvide definisjonene av digital kompetanse til også å omhandle det å skape teknologi, og ikke bare skape med teknologi. I den reviderte definisjonen av digitale ferdigheter er heller ikke programmering eller problemløsning nevnt som kunnskapsområder under digital kompetanse. Likevel kan man tolke punktet; *produsere og bearbeide* som et område der programmering kan bli plassert, da det forklares som å «lage digitale produkter ved hjelp av digitale ressurser» (Udir, 2016b).

Digital kompetanse i Kunst og håndverk

Det digitale ble en del av kunst og håndverksfaget med læreplanverket fra 1997 (L97), der skannere og digital billedbehandling ble nevnt som innhold i lærerplanen. Dette var en speiling av daværende samfunn, da det på 1990-tallet ble et økt fokus og vekst innen blant annet PC, dataspill og TV (Nielsen, 2009). Ved utarbeidelse av den gjeldende læreplanen LK06, fortsatte veksten innen IKT. Dette medførte at visuell kommunikasjon ble viktigere og det ble derfor et økende behov for visuell kompetanse (Nielsen 2016). Under grunnleggende ferdigheter i Kunst og håndverk i LK06 ble det opprettet fem grunnleggende ferdigheter, der en av de er å kunne bruke digitale verktøy i faget. Her går fokuset på å søke informasjon, produsere informasjon i tekst og bilde, produsere digitale bilder, kildekritikk, personvern, opphavsrett og multimedier (Udir, 2006). I artikkelen *Å være digital i kunst og håndverk* (2009) deles ferdighetene elevene må kunne innen det digitale i to områder. På den ene siden skal elevene kunne søke informasjon og stille seg kritisk til informasjon og yringer. På den andre siden skal elevene skape informasjon i form av tekst og bilder (Dahlin & Gjerde, 2009). Ser man på tendenser innen LK06 er det der også en vektlegging av at visuell kommunikasjon er et område der digitale verktøy spesielt kan trekkes inn, her nevnes foto, film og grafisk design (Nielsen, 2009). Det digitale ble innført i L97 og ble vektlagt i større grad i LK06. Selv om disse føringene kom i faget, ser man at den utøvende bruken av digitale verktøy i Kunst og håndverk er mindre enn i andre fag. I *International Computer and Information Literacy Study* (ICILC) fra 2013 kommer det frem at halvparten av elevene i 9 klasse sier at de aldri bruker datamaskin i de estetiske fagene, Kunst og håndverk og Musikk (Ottestad, Throndsen, Hatlevik & Rohatgi, 2014). Dette er likevel en eldre kilde, som ikke nødvendigvis gjenspeiler dagens situasjon. ICILS gjennomførte en ny datainnsamling i 2018 der de også undersøker algoritmisk tankegang, rapporten er forventet en gang i 2019 (IEA, u.å). I den nye fagfornyelsen står det beskrevet at man ønsker at digitale ferdigheter i større grad knyttes mot fagspesifikk kompetanse i fagene (NOU 2015: 8).

I mastergradsavhandlingen til Ingri Strand (2017) undersøker Strand hvordan digitale verktøy blir brukt i Kunst og håndverk og lærerens forklaringer av bruken. I undersøkelsen kommer Strand frem til at lærere i større grad vektlegger tradisjonelle teknikker enn digitale verktøy. Dette understøttes også i mastergradavhandlingen til Ida Linett Olsen (2014), der hun undersøker hva digital kompetanse kan innebære i Kunst og håndverk. I avhandlingene kan det virke som om det er en tredeling av rådene praksiser. Den første er at læreren ser på digitale ferdigheter som teoretisk, og vektlegger det praktiske og tradisjonelle der elevene arbeider taktilt med tradisjonelle materialer. Den andre praksisen går på at man arbeider tradisjonelt og praktisk, men søker inspirasjon og informasjon samt dokumenterer og presenterer det praktiske ved hjelp av digitale verktøy. Den tredje praksisen går ut på å ha et fokus på fagspesifikke digitale verktøy i undervisningen og det blir en vektlegging av både tradisjonelle og digitale verktøy. I mastergradsavhandlingen til Strand kommer det fram at bruken av digitale verktøy ofte omhandler å søke etter informasjon eller inspirasjon og til presentasjoner

og dokumentasjon av fagstoff og eget arbeid. I andre rekke kommer det å arbeide med fagspesifikke digitale verktøy som digital tegning, fotografering, bildebehandling, 3D-modellering og animasjonsfilm (Strand, 2017). I mastergradsavhandlingen til Janne H. Hjelmberg (2015) undersøker hun nettbrettet sitt potensial for å skape. Hun undersøker også om ulike faktorer har innvirkning på om nettbrettet blir tatt i bruk på en skapende måte eller ikke. Hun finner ulike faktorer for hvorfor dette ikke skjer, der hun nevner manglende kunnskap og engasjement rundt oppdatering av egen kunnskap hos lærerne. Hun finner, slik som Olsen (2014) og Strand (2017) at lærerne har større fokus på tradisjonelle teknikker og materialer enn å skape gjennom nettbrett.

I sine mastergradsavhandlinger støttet både Olsen og Strand seg på en deling og definering av det digitale utfra den eldre forklaringen av digital kompetanse gjort av Monitor 2013 og Monitor 2011. Her blir det beskrevet fem mestringsområder for digitalkompetanse som er, *operativ bruk av IKT, å tilegne seg og behandle digital informasjon, å produsere og bearbeide digital informasjon, digital dømmekraft og å kommunisere digitalt* (Egeberg et al., 2012; Hatlevik, Egeberg, Guðmundsdóttir, Loftsgarden & Loi, 2014). Denne inndeling av digital kompetanse tar ikke for seg en revidert definering av digitalkompetanse som også kan ha konsekvenser for kunst og håndverksfaget. Avhandlingene tar heller ikke for seg programmering eller algoritmisk tankegang i Kunst og håndverk, og dette blir ikke nevnt som mulige digitale verktøy. Derfor vil jeg i neste delkapittel se på programmering knyttet opp mot Kunst og håndverk. De fleste av kildene er likevel knyttet til et internasjonalt perspektiv.

Programmering i Kunst og håndverk

Kreativ koding er et begrep som ofte blir brukt i sammenheng med kunst. Det finnes også flere artikler som beskriver hvordan kreativ koding har blitt brukt i kunstrelatert undervisning, eller hvorfor det bør være en del av kunstutdanningen (Greenberg, Kumar & Xu, 2012; Knochel & Patton, 2015; Peppler & Kafai, 2009). Når en snakker om kreativ koding blir ofte programmering betegnet som et nytt materiale, der formålet er å skape noe uttrykksfullt istedenfor funksjonelt (Artut, 2017; Knochel & Patton, 2015; Maeda, 2000; Reas & Fry, 2006;). Jon Hoem som er førsteamanuensis, forsker og underviser på nye digitale medier ved Institutt for kunstfag, Høgskolen på Vestlandet – Bergen, kommer inn på tema og snakker om uttrykk og *meningsvare* koblet til programmering. Hoem har skrevet to innlegg på HVL bloggen med navnene *Barn og koding – ikke bare realfag* den 22. oktober 2017 og *Generativ kunst og tankefullt håndverk* den 13. november 2017. I innleggene argumenterer han for at programmering omhandler mer enn kun realfag, der han mener at programmering også er et håndverk og uttrykksmiddel. Dette synspunktet har også blitt lagt frem i innledningen, der Sennett (2008), Bergersen (2017) og Stærk (2016) omtaler programmering som et håndverk. Hoem henviser til noe som heter *meningsvare*, og forklarer at programvare og maskinvare er vanskelig å plassere andre steder enn i realfag, men med en gang dette skal brukes til noe befinner man seg i *meningsvare*.

Torgeir Waterhouse (2017) som er direktør for internett og nye medier i IKT-Norge hadde et innlegg på *Kunst og håndverkskonferansen 2017*, den 27. januar. I sin presentasjon *Teknologi i Kunst og håndverk* sier Waterhouse at Kunst og håndverk kanskje er et av de viktigste faget når det kommer til teknologi. Waterhouse mener at det er viktig med Kunst og håndverk når det kommer til møte mellom teknologi og mennesket. Han mener at teknologisk utvikling også forutsetter at man har forståelse for blant annet formgivning, kommunikasjon, farge og visuell kommunikasjon. I fremlegget henviser han til programmering, og at dette også bør bli brukt av kunst og håndverksfaget. Waterhouse mener at teknologi trenger kunst og håndverk, men at kunst og håndverk også trenger teknologi, og sier videre at mye av verktøyene som brukes i Kunst og håndverk er basert på teknologisk utvikling. Waterhouse gjengitt i *Når robotene tar over* av Hans Skjong

(2017), sier om programmering i skolen «Det er mange som tror realfag blir det viktigste (...) men kunst og håndverk vil også bli sentralt, fordi det handler om møte mellom mennesket og produktet». Sevik (2016) mener også at programmering har en plass i Kunst og håndverk på grunnskolen. Hun sier «Utøvelsen av kunst- og håndverksfaget og musikkfaget baserer seg også i økende grad på digitale teknologier og programmeringsferdigheter. Vi har lenge hatt strikke- og veve-maskiner der mønsteret programmeres på ulike måter» (Sevik, 2016, s. 12). Kylie Peppler er universitetslektor i informatikk og utdanning og Yasmin B. Kafai forsker blant annet på utdanning som bruker programmering til design relaterte emner og har vært med på å utvikle Scratch programmeringsspråket. Kafai og Peppler (2009) skriver at det kan være vanskelig å definere verk som er skapt i sjiktet mellom kunst og teknologi, siden dette er under konstant forandringen. De velger heller å bruke betegnelsen *media art*, og i artikkelen nevner de eksempler som dataspill, datahåndverk (engelsk: computational craft), robotikk, animasjon og datagrafikk.

Fra et internasjonalt perspektiv er det flere som argumenterer for at programmering bør bli en del av undervisningen innen kunstrelaterte fag, fra barnehagen og i hele det obligatoriske skoleløpet. Aaron D. Knochel og Ryan M. Patton er begge forskere på nye medier i kunstutdanning og underviser ved ulike universiteter i USA. Knochel og Patton (2015) kommer med ulike synspunkter for hvorfor programmering bør bli en del av kunstundervisningen. De legger også frem argumenter på hvordan kunstutdannelsen kan tilføye ulike aspekter til algoritmisk tankegang på tvers av ulike disipliner. De mener at elever bør få arbeide med kreativ koding og elektronikk, slik at elevene får arbeidet med nye materialer innen kunsten, men også for å utvikle kritisk tenkning ved skapning av teknologi. Et av argumentene Knochel og Patton kommer med er at «Coding is not only a technical practice or a series of steps to implement mathematical algorithms, but rather a process of design informed through social, political, and cultural frameworks» (Knochel & Patton, 2015, s. 28). De samme tankene kan man også se i argumentene for videreutviklingen av STEM til STEAM. En av grunnleggeren og forskeren bak STEAM, Georgette Yakman (2008) definerer praksisen til STEAM og sier: «Science and Technology, interpreted through Engineering and the Arts, all based in a language of Mathematics» (Yakman, 2008, s. 18). Altså ligger det en tanke om at teknologi og programmering må bli forstått i en kontekst. Dette reflekterer igjen det Hoem (2017a; 2017b) og Waterhouse (2017) sier, at med en gang man skal bruke teknologien til noe, må man tenke på møtet mellom produkt og mennesket. Selcuk Artut (2017) er kunstner innen nye medier og legger frem argumentet med at kunstens utvikling etter den moderne perioden er sterkt tilknyttet til utvikling innen den seneste teknologien. Han stiller dermed spørsmål om i hvilken grad nåværende kunstutdanning bør ha teknologi som en del av utdannelsen. Artut konkluderer med at programmering bør bli implementert som et nytt materiale til å skape kunst i undervisningen med. Peppler og Kafai (2009) mener også at kunstundervisningen på grunnskolen ikke har fulgt utviklingen til samtidens kunstpraksis. De skriver:

«(...) multimedia design, particularly that which uses Adobe Photoshope and Hyperstudio software, have been taken as the main artistic expressions of digital media in primary and secondary educational settings, whereas professional artists are using advanced programming to manipulate and create digital expressions. If there is any movement in education, it has not paralleled the developments in professional art» (Peppler& Kafai, 2009, s. 1).

Selv om det over refereres til forskning i andre land, kan mastergradsavhandlingen til Strand (2017) og Olsen (2014) vise til samme tendenser når det kommer til hva elevene lærer i Kunst og håndverk på grunnskolen. I Strand (2017) sin avhandling er fagspesifikke digitale verktøy blitt beskrevet som digital tegning, fotografering, bildebehandling, 3D-modellering og animasjonsfilm.

Et annet argument som blir brukt for at programmering skal bli en del av kunstundervisningen er at

arbeidsmarkedet ser etter studenter som har kunnskap innen kunst og design, men også innen programmering spesielt når det kommer til interaktiv media (Amiri, 2011). Faramarz Amiri (2011) underviser i digitale medier på University of Wolverhampton, og fokuserer på spilldesign og interaksjons design gjennom å kode. Amiri (2011) velger å dele nivået for hva man må kunne, der han trekker linjer mellom studenter innen informatikk med lingvister, og Kunst og design studenter med turister. Turistene trenger språket for å kunne kommunisere med mennesker, utforske miljøet og komme seg rundt. Lingvistene er mer interessert i syntaksen og semantikken selv om de aldri trenger å bruke språket til kommunikasjon.

Det finnes prosjekter og forskning som har brukt kreative aktiviteter for å lære programmering. Målet har både vært å lage fysiske gjenstander med integrert teknologi og programmering, eller animasjoner og videospill basert på programmering. Forskningen viser at elevene ofte har en positiv opplevelse av undervisningen og at et skapende rammeverk er en god måte for å promotere og undervise programmering (Adams, 2007; Burke & Kafai, 2010; Giannakos, Jaccheri & Proto 2013; Knobelsdorf & Romeike, 2008; Lau, Nagai, Chan & Cheung, 2009;) Slike undervisningstilnæringer blir ofte knyttet til skaperbevegelsen (Maker-bevegelsen) og skapeverksteder (Makerspace). Veena Vasudevan og Kafai (2016) sier at materialene som blir brukt i et skaperverksted kan falle innenfor to kategoriene, håndverk (Engelsk: craft) og elektronikk. I Norge ser man en økning når det kommer til skapeverksteder, der det var to skaperverksteder i 2010 og 69 skaperverksteder i 2018, dette er i ulike institusjoner som blant annet vitensentre, skoler, universiteter og høyskoler (Norwaymakers, u.å). *Teknoteket Makerspace* ved Teknisk Museum støtter seg både på tradisjonelle teknologier slik som tresløyd, men også på moderne teknologi slik som programmering med BBC micro:bit. De mener ideen bak et skaperverksted er viktig for fremtiden, altså skaperferdigheten som barna utvikler ved å få muligheten til å arbeide kreativt. De mener det er viktig at elevene blir skapere og ikke kun konsumenter av ny teknologi (TekniskMuseum, u.å.a).

Tredje utkast til kjerneelementene i Kunst og håndverk i fagfornyelsen

For å få en bedre forståelse av programmering i Kunst og håndverk i den norske grunnskolen har jeg undersøkt hva fagfornyelsen sier om temaet. Kjerneelementene i Kunst og håndverk ble satt til å være håndverksferdigheter, kunst- og designprosesser, visuell kommunikasjon og kulturforståelse. Programmering ble plassert under kjerneelementet visuell kommunikasjon i det tredje utkastet av kjerneelementene og var et punkt under 8. til 10. trinn som lød «kode og modellere visuelle uttrykk» (Udir 2018a). Den 4. juni 2018 kom utdanningsdirektoratet med en oppsummering av tilbakemeldingene på siste innspillsrunden på kjerneelementene i Kunst og håndverk. I tilbakemeldingen kommer det frem at en stor andel (69%) mener at kjerneelementene dekker det viktigste innholdet i faget, 22% svarer at de ikke synes kjerneelementene dekker det viktigste, og de resterende har enten ikke svart eller svart «vet ikke» (Udir, 2018e). Det nevnes også at det har kommet tilbakemeldinger på at fagfeltet stiller seg negativt til at programmering kommer under Kunst og håndverk og er en del av kjernen i faget. Videre nevnes det på bakgrunn av dette at vurderingen av programmering i faget har blitt tatt opp igjen og at programmering skal knyttes mot visuell modellering på ungdomstrinnet (Udir, 2018e). Daglig leder for Kunst og design i skolen Lennart Johansson (2018) skriver

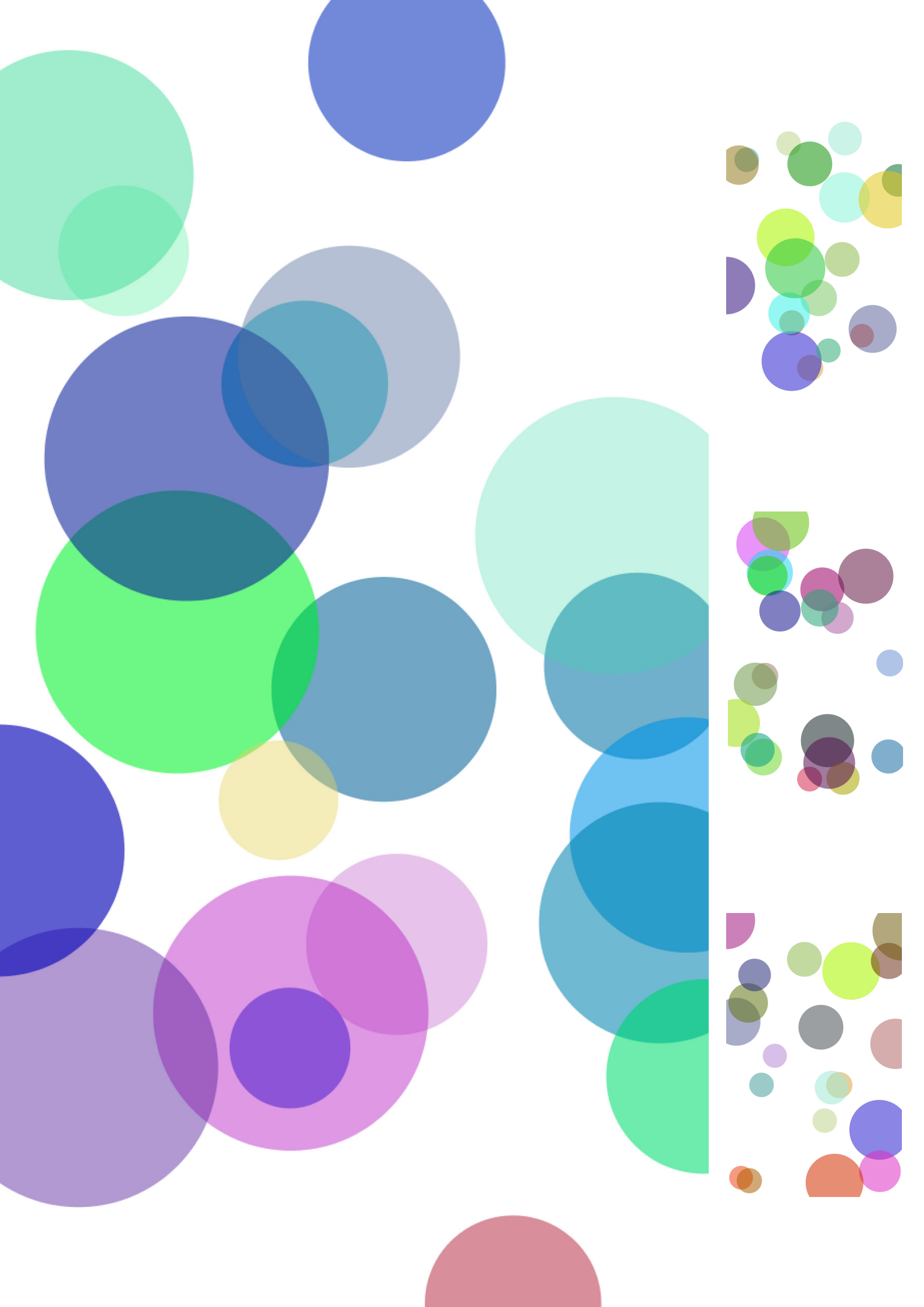
«Selvsagt skal også vårt fag henge med i utviklingen innen IKT, men å innføre det å «kode» og «programmere» som sentrale deler i kunst og håndverk er et feilgrep. Opplæringen i dette mener jeg kan foregå i matte og i realfag – så kan vi selvsagt bruke det, gjerne i tverrfaglige opplegg» (Johansson, 2018, s. 9).

Kjerneelementene og skisser til den nye læreplanen i Kunst og håndverk

I det endelige utkastet av kjerneelementene i Kunst og håndverk blir det digitale nevnt både under håndverksferdigheter og visuell kommunikasjon. Under håndverkferdigheter står det «De skal bruke harde, plastiske, myke materialer og digitale verktøy på en etisk, miljøbevisst og trygg måte gjennom hele skoleløpet» (Kunnskapsdepartementet, 2018a, s. 2) og under visuell kommunikasjon er et av punktene «Elevene skal bruke visuelle virkemidler bevisst og eksperimenterende i to-, tredimensjonale, og digitale uttrykk» (Kunnskapsdepartementet, 2018a, s. 2). Her blir det digitale koblet sammen med visuelle virkemidler og visuell kommunikasjon som er knyttet mot uttrykk. Dette kan se ut som en videreføring av tankene i LK06, der det digitale spesielt er knyttet opp mot visuell kommunikasjon. Det nye her er at det digitale også blir koblet opp mot håndverksferdigheter og det blir til dels sammenlignet med tradisjonelle materialer innen Kunst og håndverk (Kunnskapsdepartementet, 2018a). Det kan likevel være noe vanskelig å tolke hvordan det digitale skal bli brukt når det plasseres under håndverksferdigheter. Dette kan den første og andre skissen til læreplanen i Kunst og håndverk og duodji kanskje gi eksempler på.

Den 18. oktober 2018 kom den første skissene til den nye læreplanen, med mulighet til å gi innspill frem til 14. november 2018 (Udir, 2018f). Slik som ved kjerneelementene blir det digitale nevnt flere steder. Målene åpner også opp for å bruke programmering og teknologi. Ett av kompetansemålene under Kunst og håndverk, nevner programmering eksplisitt, da elevene etter 7. trinn skal «analysere ulike mønster og programmere en mønsterrapport som lages i myke materialer» (Udir, 2018f). Både digitale verktøy og digital representasjon blir nevnt, men planen legger også frem nye begrep som teknologi, digitale materialer og programmering (Udir, 2018f). Dette er begreper som ikke er brukt i den nåværende læreplanen LK06. Under digitale ferdigheter som vil fortsette å være en generell kompetanse i den nye lærerplanen blir digitale ferdigheter beskrevet som «Digitale ferdigheter i kunst og håndverk er å kunne bruke digitale verktøy og materialer for å gi form til opplevelser, meninger og ideer i skapende prosesser og produkter» (Udir, 2018f). Som det kommer fram kan dette igjen vise til en bredere definisjon av hva det digitale er i Kunst og håndverk enn i gjeldene læreplan LK06. I oppsummeringen av tilbakemeldingen på læreplanen var et av punktene at språket var for vagt og at vanskelige ord må lukes bort (Udir, 2018h). Det kan tenkes at digitale materialer, programmering og teknologi er eksempler på slike ord.

Den 18. mars 2019 kom den andre skissen til den nye læreplanen i Kunst og håndverk. Her har ord som digitale materialer falt bort, men ordene teknologi og programmere er fortsatt en del av planen. Digitale ferdigheter blir i planen blant annet beskrevet som «Utvikling av digitale ferdigheter i kunst og håndverk går fra å benytte enkle digitale ressurser til å beherske og forme egne digitale produkter for å skape opplevelser og uttrykke følelser, ideer og meninger» (Udir, 2019). Programmering og mønster er fortsatt et kompetansemål etter 7.trinn men har blitt omgjort til «gjenkjenne og programmere mønstersekvenser til visuelle uttrykk». Etter 10. trinn blir det vist til teknologi og dette blir knyttet opp mot «nye kommunikasjonsformer og opplevelser i skapende prosesser og produkter» (Udir, 2019).



```

ball b1;
ball b2;
ball b3;
ball b4;
ball b5;
ball b6;
ball b7;
ball b8;
ball b9;
ball b10;

ball b11;
ball b12;
ball b13;
ball b14;
ball b15;
ball b16;
ball b17;
ball b18;
ball b19;
ball b20;

void setup () {
  fullscreen ();
  //background (255);
  b1 = new ball ();
  b2 = new ball ();
  b3 = new ball ();
  b4 = new ball ();
  b5 = new ball ();
  b6= new ball ();
  b7 = new ball ();
  b8= new ball ();
  b9 = new ball ();
  b10= new ball ();

  b11 = new ball ();
  b12 = new ball ();
  b13 = new ball ();
  b14 = new ball ();
  b15 = new ball ();
  b16= new ball ();
  b17 = new ball ();
  b18= new ball ();
  b19 = new ball ();
  b20= new ball ();
}

void draw () {
  background (255);
  b1.visBall ();
  b1.bevegBall ();
  b1.checkEdgesBall ();

  b2.visBall ();
  b2.bevegBall ();
  b2.checkEdgesBall ();

  b3.visBall ();
  b3.bevegBall ();

  b3.checkEdgesBall ();

  b4.visBall ();
  b4.bevegBall ();
  b4.checkEdgesBall ();

  b5.visBall ();
  b5.bevegBall ();
  b5.checkEdgesBall ();

  b6.visBall ();
  b6.bevegBall ();
  b6.checkEdgesBall ();

  b7.visBall ();
  b7.bevegBall ();
  b7.checkEdgesBall ();

  b8.visBall ();
  b8.bevegBall ();
  b8.checkEdgesBall ();

  b9.visBall ();
  b9.bevegBall ();
  b9.checkEdgesBall ();

  b10.visBall ();
  b10.bevegBall ();
  b10.checkEdgesBall ();

  b11.visBall2 ();
  b11.bevegBall ();
  b11.checkEdgesBall ();

  b12.visBall2 ();
  b12.bevegBall ();
  b12.checkEdgesBall ();

  b13.visBall2 ();
  b13.bevegBall ();
  b13.checkEdgesBall ();

  b14.visBall2 ();
  b14.bevegBall ();
  b14.checkEdgesBall ();

  b15.visBall2 ();
  b15.bevegBall ();
  b15.checkEdgesBall ();

  b16.visBall2 ();
  b16.bevegBall ();
  b16.checkEdgesBall ();

  b17.visBall2 ();
  b17.bevegBall ();
  b17.checkEdgesBall ();

  b18.visBall2 ();
  b18.bevegBall ();
  b18.checkEdgesBall ();

  b19.visBall2 ();
  b19.bevegBall ();
  b19.checkEdgesBall ();

  b20.visBall2 ();
  b20.bevegBall ();
  b20.checkEdgesBall ();
}

class ball {
  float x = random (width);
  float y = random (height);
  float xspeed = (random
(0.5, 4));
  float yspeed = (random
(0.5, 4));
  float diameter = (random
(100, 280));
  float fargeR = ( random
(0,244));
  float fargeG = (random
(0,244));
  float fargeB = (random
(0,244));

  void visBall () {
    noStroke ();
    //stroke (40);
    fill (fargeR, fargeG,
fargeB, 140);
    ellipse (x,y, diameter,
diameter);
  }

  void visBall2 () {
    noStroke ();

    fill (random(0,244),
r a n d o m ( 0 , 2 4 4 ) ,
random(0,244), 140);
    ellipse (x,y, diameter,
diameter);
  }

  void bevegBall () {
    x = x + xspeed;
    y = y + yspeed;
  }

  void checkEdgesBall () {
    if (x > (width - 15) ||
x < 0) {
      xspeed = xspeed * -1;
    }
    if (y > (height - 15) ||
y < 0) {
      yspeed = yspeed * -1;
    }
  }
}

```

Bilde 2

Animasjon av sirkler i ulike farger (2018) Eget bilde

Forskningsstrategi

Kritisk realisme

Men utgangspunkt i problemstillingens ordlyd og eget ståsted er denne oppgaven skrevet innenfor et kritisk realistisk paradigme. Kritisk realisme ble utviklet av den engelske filosofen Roy Bhaskar på 1970-tallet. Retningen oppsto som en kritikk av positivismens empiristiske ontologi (Næss, 2012), og anklager positivismen og sosialkonstruktivismen for å være overfladiske og som utfører forskningen på en lite teoretiske måte (Alvesson & Sköldberg, 2009, s. 39). Kritisk realisme støtter seg på en *ontologisk realisme*, *epistemologisk relativisme* og en *vurderingsmessig rasjonalisme*.

En *ontologisk realisme* innen kritisk realisme forstår jeg som at det eksisterer en virkelig verden der ute, uavhengig om vi har kunnskap om den eller ikke. I denne oppgaven blir dette forstått som det kritisk realisme kaller det *reelle domenet*. Det reelle domenet består av strukturer og mekanismer som forårsaker begivenheter og fenomener som vi har mulighet til å observere (Alvesson & Sköldberg, 2009; Buch-Hansen & Nielsen, 2014; Næss, 2012). I min undersøkelse blir det dermed viktig å bruke det jeg samler inn av empiri for best mulig å kunne si noe om den virkelige verden og det reelle domenet. Innsamlingen av empiri og analysen vil dermed fokusere på å finne mulig mekanismer som støtter eller bestrider problemstillingen som blir undersøkt. Under vil jeg videre forklare hva som menes med mekanismer.

Selv om kritisk realisme mener at man kan si noen om den virkelige verden har den en *epistemologisk relativisme*. Dette betyr at all kunnskap er sosialt og historisk betinget. I min undersøkelse vil jeg kun støtte meg på en viss mengde med empiri, da det finnes begrensninger ved alle forskningsprosjekter. Empirien er innhentet ved hjelp av semistrukturerte forskningsintervjuer med fire ulike informanter og en praktisk-estetisk undersøkelse gjennom *research by design*. Den mengden med empiri som blir utgangspunkt for min analyse er det kritisk realisme kaller for det *empiriske domenet*. Det er derfor viktig å huske at det vil være meninger og årsaker om hvorfor programmering bør eller ikke bør bli en del av Kunst og håndverk som ikke er en del av oppgaven. Kritisk realisme kaller dette for det *faktiske domenet*, som er alle de fenomener og begivenheter som eksisterer og finner sted selv om de blir erfart eller ikke av meg som forsker (Alvesson & Sköldberg, 2009; Buch-Hansen & Nielsen, 2014; Næss, 2012). Slik jeg har forstått kritisk realisme kan man ikke umiddelbart og ufeilbarlig lese verden akkurat som den er, all viten blir dermed konstruert og feilbarlig. Man kan ikke få direkte adgang til den eksterne virkeligheten, alle oppfatninger filtreres gjennom sansene til meg som forsker. Likevel er ønsket at selv om jeg kun kan holde meg innen det empiriske domenet, så kan jeg uttale meg om noen årsaksforklaringer innen det reelle domenet (Alvesson & Sköldberg, 2009; Buch-Hansen & Nielsen, 2014; Næss, 2012).

Til slutt støtter kritisk realisme seg på en *vurderingsmessig rasjonalisme*, dette betyr at visse sannheter er mer troverdig enn andre. I masteroppgavens undersøkelser blir det dermed viktig at mine begrunnelser og utsagn er teoretisk og empirisk velbegrunnet (Næss, 2012), slik at oppgaven i større grad vil kunne si noe om årsaksforklaringer i det reelle domenet. Hvordan jeg velger å strukturere kunnskapen på feltet, metoden

som blir brukt, hvordan jeg analyserer og drøfter vil alle være med på å påvirke hva slags kunnskap som blir konstruert. Nedenfor vil jeg argumentere og beskrive hvordan innsamlingen av data har blitt gjennomført.

Årsaksforklaringer og mekanismer

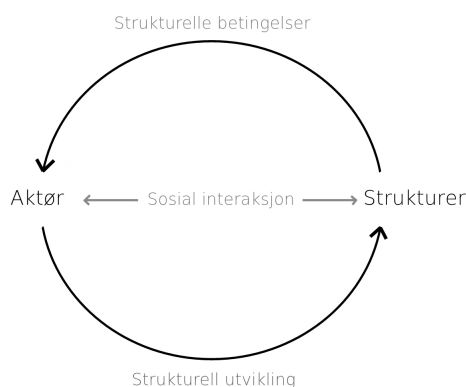
Årsaksforklaringer har en sentral rolle innen kritisk realisme og «Målet med vitenskapelig forskning er å forklare observerbare fenomener ved å avdekke slike underliggende årsaksmekanismer» (Næss, 2012, s. 5). Å finne årsaksforklaringer som støtter eller bestrider påstanden om hvorfor programmering bør bli en del av Kunst og håndverk, blir en sentral del av denne oppgaven. Årsaksforklaringer forstås i denne oppgaven som årsaksmekanismer og jeg støtter meg på Bendik Bygstad og Bjørn Erik Munkvold (2011) sin definisjon av mekanismer: «causality is expressed in the term mechanism, simply defined as a causal structure that explains a phenomenon» (Bygstad & Munkvold, 2011, s. 1). Dette betyr at årsaksforklaringer vil bli det samme som en mekanisme. Videre støtter jeg meg på Petter Næss sin utdypning av årsaksforklaringer som kan være «(...) individers holdninger og kunnskapsressurser så vel som intersubjektive meningsdannelser» (Næss, 2012, s. 5). Da jeg i denne oppgaven ønsker å benytte meg av semistrukturerte intervjuer og *research by design*, blir informantenes svar og mine refleksjoner over den praktisk-estetiske undersøkelsen, samt teori, grunnlaget for årsaksforklaringene. Alvesson & Sköldbberg (2009) mener at begrepet mekanisme ofte kan være vanskelig å definere, og begrepet kan blande seg med andre underliggende lag. Videre mener de at sosiale fenomener kan bli over-simplifisert, og det kan være vanskelig å bruke begrepet i komplekse sosiale kontekster, men som jeg kommer inn på i neste delkapittel mener jeg at det finnes noen regelmessigheter selv i en sosial kontekst slik som skolen. Målet i denne oppgaven er ikke å avdekke generelle lover da dette kan være vanskelig å fastslå i sosiale kontekster, ønske er å komme med mulige bør-perspektiver, og kunne si noe om mulig eksisterende styringer og holdninger. Det er likevel viktig å være klar over at disse aspektene kan være mer komplekse enn hvordan de blir fremstilt i denne oppgaven.

Aktør og struktur

En av grunnene til valget av kritisk realisme som vitenskapelig ståsted, er anvendelsen av begrepet *strukturell utvikling* og *strukturelle betingelse*, som kommer inn under forklaringen av struktur og aktør. Innen kritisk realisme deler man den sosiale verden i to grunnleggende nivåer, aktører og strukturer, og teorien opererer innenfor en *aktør/struktur dualisme*. Når jeg i denne oppgaven skal undersøke hvorfor et nytt emne bør eller ikke bør bli implementert i Kunst og håndverk, kan dette bli forklart ved hjelp av nåværende strukturer og aktører og eventuelt kommende strukturer. Når noe nytt skal implementeres i en veletablert organisasjon som skolen, er det viktig å se på hvordan skolen fungerer, og hvem som er med på å påvirke en slik institusjon. Kritisk realismen åpner opp for at det kan finnes noen regelmessigheter selv når konteksten er svært kompleks, dette fordi noen relasjoner og reguleringer er mer sentrale enn andre (Næss, 2012, s. 5).

Margaret Archer (1995) er den som i størst grad har vært med på å videreutvikle og presisere aktør og struktur relasjonen i det hun kaller *sosial morfogenese* (Buch-Hansen & Nielsen, 2005, Bråten, 2015). I oppgaven støtter jeg meg på Archer (1995) sin teori som beskrevet av Buch-Hansen & Nielsen (2005) i boken *Kritisk realisme*. Buch-Hansen & Nielsen, med utgangspunkt i Archer, mener at det er mulig å analysere samspillet mellom aktør og struktur over tid, som består av en syklus med *strukturelle betingelser*, *sosial interaksjon* og *strukturell utvikling* (se figur 1). *Strukturelle betingelser* vil i denne sammenhengen si eksisterende strukturer, som er et produkt av menneskelige aktiviteter, som aktøren blir påvirket av uavhengig om aktøren er bevist på det eller ikke (Buch-Hansen & Nielsen, 2005). Videre vil aktøren kunne påvirke strukturene, da aktøren «besidder

unikke egenskaber, som bevirker, at deres adfærd ikke lader sig reducere til den strukturelle kontekst, som de indgår i» (Buch-Hansen & Nielsen, 2005, s. 52). Aktøren vil heller aldri kun møte en struktur, da man innen kritisk realisme mener at det alltid er et nettverk med strukturer som er forbundet til hverandre, dette blir benevnt som *sosial interaksjon*. Den tredje delen av syklusen er *strukturell utvikling* der aktøren vil gjenskape eller omdanne eksisterende strukturer (Buch-Hansen & Nielsen, 2005).



Figur 1: Samspeillet mellom aktør og struktur, basert på Archer (1995) sin modell gjengitt av Buche-Hansen, H & Nielsen, P. (2005) s. 52. (2018) Egen modell

En kritikk av kritisk realisme er synet på at det er strukturene som deler populasjonen inn i ulike kategorier. Alvesson & Sköldberg (2009) skriver «of course groups can be more or less arbitrary defined, but still a group is never internally homogeneous [...] is in fact just another guise for the researcher, who, albeit often unconsciously, has a certain interest in dividing the population in this or that way» (Alvesson & Sköldberg, 2009, s. 45). I min oppgave ønsker jeg i større grad å benytte meg av begrepene strukturelle betingelser og strukturell utvikling (figur 1). Grunnen for dette er at jeg ønsker å undersøke eksisterende strukturer, som i denne oppgaven vil være politiskstyringer, fagfeltet sine holdninger og rammeverket i Kunst og håndverk, men jeg ønsker også å se på muligheten for strukturell utvikling. Jeg ønsker å se om Kunst og håndverk i grunnskolen bør utvikle seg til også å inkludere programmering. Jeg er bevist på at inndelingen som er valgt er basert på mine konstruksjoner av gjeldene og fremtidige strukturer. Det finnes for eksempel aktører som både er en del av fagfeltet, men også er en del av de politiske styringene.

Kvalitative metoder

For å kunne besvare problemstillingen og ha velbegrunnede antagelser har jeg i denne oppgaven valgt å støtte meg til kvalitative metoder, da kritisk realisme støtter både kvalitative og kvantitative tilnærminger (Næss, 2012). Denne metodefriheten er noe kritisk realisme har fått kritikk for, da man mener at dette er et naivt utgangspunkt for forskningen. Alvesson & Sköldberg (2009) skriver «it is almost as if the object of study discloses itself and then tells the researcher how it is most appropriately studied» (Alvesson & Sköldberg, 2009, s. 45). Uansett forskning tror jeg at antagelsen om at ulike perspektiver og forskningsmetoder vil lede til hva slags svar man får. Det er derfor viktig å være transparent i forskningen og fremlegge hvordan man samler inn empiri og hvordan man kommer frem til de ulike refleksjonene i drøftingen. Jeg er ikke helt enig i kritikken, da jeg som forsker mener at man må reflektere over hvilke metoder man bruker, og finne en metode som kan gi svar på problemstillingen. Jeg vil argumentere for de valgte metodene nedenfor.

Problemstillingen som er valgt er i større grad en beskrivende problemstilling, der målet med undersøkelsen er å komme til en deskriptiv og eksplorerende forklaring (Jacobsen, 2015). Ved bruk av kvalitative metoder kan empirien som samles inn, ha en større dybde som kan gi svar som er mer nyansert, helhetlig og dyptgående, enn om jeg skulle ha samlet inn empiri ved hjelp av kvantitative metoder. Valget av en kvalitativ tilnærming er også mer passende her, fordi kvalitativ data er mer undersøkende, der resultatet ender i foreløpige hypoteser, i motsetning til en kvantitativ tilnærming som søker å teste hypoteser (Alvesson & Sköldberg, 2009). På grunn av liten kunnskap på fagfeltet i den norske grunnskolen, er det dermed vanskelig å teste hypoteser, det blir derfor viktig å kartlegge mulig hypoteser, for så å se om hypotesene kan gjelde i senere forskning.

Formålet med det kvalitative forskningsintervjuet er å forstå og få innsikt i fagpersoners meninger og holdninger om programmering i Kunst og håndverk, sett i et perspektiv fra hans eller hennes ståsted (Kvale & Brinkmann, 2015). Men, jeg ønsker også å samle empiri gjennom egne refleksjoner fra den praktisk-estetiske undersøkelsen. Grunnen for valg av disse metoden er fordi det finnes lite forskning og teori på feltet og fordi det er et fåtall av personer som har kunnskap innen programmering og Kunst og håndverk i grunnskolen. Det er også et felt som er lite praktisert, og dermed vil andre metoder slik som observasjon eller spørreundersøkelse bli vanskelig.

Research by design

Det skapende arbeidet har som mål å utforske programmering i sammenheng med Kunst og håndverk, og dermed produsere nye kunnskap ved å bruke design som et mål gjennom å programmere. Siden det eksisterer lite kunnskap og forskning angående problemområdet, har jeg et ønske om at en praktisk tilnærming også vil kunne gi svar på hvorfor eller hvorfor ikke programmering skal inn i Kunst og håndverk på grunnskolen. Jeg ønsker at den praktiske tilnærmingen bringer meg nærmere til å finne svar på hvilke skapemuligheter som finnes i programmering, som kan bli med i å danne grunnlag for drøftingen.

Når jeg i denne oppgaven skal forske på min egen praksis gjennom å programmere, har jeg valgt å benytte meg av begrepet, *research by design*. *Research by design* blir beskrevet av Birger Sevaldson, forsker på designprosesser, som «A special research mode where the explorative, generative and innovative aspects of design are engaged and aligned in a systematic research inquiry» (Sevaldson, 2010, s. 11). *Research by design* kommer inn under begrepet *Research through art and design* (Sevaldson, 2010). Christopher Frayling (1993) anerkjennes ofte til å være en av de som gjorde begrepet *research through art and design* kjent, og skapte en distinksjon mellom tre typer forskning innen kunst og design, *research into art and design*, *research through art and design* og *research for art and design*. *Research through art and design* kan enten baserer seg på forskning på material, utviklingsarbeid eller aksjonsforskning. Videre kan man beskrive metoden som en skapende prosess, der forskeren også blir designeren som skaper nye produkter, eksperimenterer med materialer og prosesser (Godin & Zaheid, 2014). Som jeg har beskrevet tidligere kan programmering ansees som et nytt materiale og det er også slik jeg ønsker å bruke programmering i denne oppgaven. Sevaldson mener likevel at det som skiller *by* og *through*, er at *through* inkorporerer muligheten for å genere kunnskap for eksterne agendaer, men *research by design* «emphasises the special inherent nature of design and design as explorative and generative actions» (Sevaldson, 2010, s. 13). Jeg har også valgt denne retningen for å presisere at jeg ønsker å bruke forskningen gjennom det praktiske for å få mer kunnskap til programmering knyttet til Kunst og håndverk. Det er likevel mulig å diskutere om det praktiske arbeide kan komme under begrepet design, da jeg både undersøker digitale og fysiske gjenstander. Noen av gjenstandene kan i større grad nærme seg kunstbegrepet enn designbegrepet. Som problemstillingen og innledningen påpeker, kan det være vanskelig å plassere programmering under kunst eller design. Likevel sier Sevaldson (2010) at design ofte er et rikt felt

som er interdisiplinær og som gjerne inkorporerer ulike teknologier, og mener dermed at *research by design* også bør reflektere spennet som designpraksisen har.

Ved å bruke *research by design* arbeider man innen en vekselvirkning mellom handling og refleksjon. Donald Schön (1983) sine begreper, *reflection-in-action* og *reflection-on-action* blir ofte nevnt som viktige redskaper innenfor forskning i design og kunst (Godin & Zaheid, 2014; Morrison & Sevaldson, 2010; Selvaldson, 2010). Refleksjonene jeg har hatt gjennom prosessen og refleksjonene jeg sitter igjen med etter prosessen har påvirket utviklingen av intervjuguiden, gitt forståelse for hva som blir sagt i intervjuene og er en del av drøftingen. Refleksjonene i den praktisk-estetiske undersøkelsen har også bidratt til utvikling av tabell 2 for hvilke skapemuligheter programmering kan ha tilknyttet Kunst og håndverk på grunnskolen.

Semistrukturerte forskningsintervju

I oppgaven har jeg valgt to forskjellige fremgangsmåter når det kommer til utføring av intervjuet, jeg har både utført datastøttede intervju og klassiske intervjuer, der jeg fysisk møtte informantene (Kvale & Brinkmann, 2015). Det datastøttede intervju ble utført via e-post og google docs, der jeg sende spørsmålene på mail og fikk svar på google docs, og jeg kunne videre legge inn oppfølgings spørsmål. Grunnen for valg av to forskjellige former for intervju, er begrensningen av utvalg informanter og hvor de befinner seg i Norge. En annen grunn er fordi det innen forskningsspørsmålet i denne oppgaven ikke er like relevant med ikke-språklig informasjon som gester og ansiktsuttrykk (Kvale & Brinkmann, 2015). Spørsmålene jeg er interessert i å få svar på er av faglig karakter og begrunnelser som går på profesjonelle meninger. Mulige konsekvenser av å bruke datastøttet intervjuet var at jeg ikke fikk like detaljert og nyansert svar, likeså kan det argumenteres for at svarene var mer presise og at informanten fikk tid til å tenke og reflektere rundt svarene.

Før intervjuene, søkte jeg godkjenning for behandling av personopplysninger (vedlegg 1) på *Norsk senter for forskningsdata* (NSD). Da søknaden ble godkjent av NSD (vedlegg 2) sende jeg mail til utvalgte personer, om de ønsket å stille på intervju angående programmering og Kunst og håndverk. Når de meldte sin interesse fant vi et tidspunkt og sted for intervjuet og det ble også tilsendt et informasjonsskriv (vedlegg 3), slik at informantene fikk tid til å lese dette før intervjuene. Før jeg begynte å intervju personene gikk vi igjennom hva som sto på informasjonsskrivet og deres rettigheter, og dette ble signert før intervjuet startet. Det datastøttede intervjuet ble lagt ut på nett, og det var dermed ikke nødvendig med signatur fra informanten.

Grunnen for valg av et semistrukturert oppsett er for å oppnå en balanse mellom en åpen og lukket samtale. Ønsket var at informantene kunne dele deres syn, og at de ikke ble ledet inn i retningen av et bestemt svar. En annen grunn er fordi jeg ønsket at informantene ikke skulle føle at det ble et forhør angående deres kunnskap, men heller en deling av deres kunnskap. Alle intervjuene var semistrukturerte og spørsmålene ble tilpasset intervjuobjektet. I forkant av intervjuene baserte jeg meg på en intervjuguide (vedlegg 4). I Intervjuguiden har ønsket vært å ha hovedspørsmål, med underordnede tilleggsspørsmål. Hovedspørsmålene ønsket jeg å være mer åpne, slik at informantene selv kunne bestemme retning. «Hvorfor» spørsmål var vanskelig å begrense, da dette ofte er veldig åpne spørsmål. Kvale & Brinkmann (2015) skriver at «Spørsmålet om hvorfor (...) er det primært forskerens oppgave og vurdere» (Kvale & Brinkmann, 2015, s. 165), dermed prøvde jeg å ha spørsmål som gikk på, *hvilke, hva* og *om*. Utfra tidligere intervju og ulike informanter ble det en variasjon mellom hvilke spørsmål som ble stilt, dette spesielt knyttet til oppfølgingsspørsmålene.

Utvalg

For å kunne svare på problemstillingen har jeg valgt å finne informanter som har kunnskap om eller en tilknytning til programmering og Kunst og håndverk. Ikke alle informantene som er valgt har en tilknytning til Kunst og håndverk på grunnskolen, men heller en generell erfaring med programmering knyttet opp mot Kunst og håndverk. Grunnen for valget er at det finnes et begrenset antall personer som har en praksis som utøver kunstrelaterte emner basert på programmering. Det er også et lite utvalg når det kommer til personer som har kunnskap og forståelse for programmering og har en undervisningspraksis i Kunst og håndverk i grunnskolen. Jeg mener også at en større bredde i utvalg av informanter vil gi meg et bredere datamateriale, som i sin tur vill gi en mer nyansert drøfting. I utvalget ble det også viktig å finne informanter som kunne se problematikken fra ulike synsvinkler. I oppgaven har jeg valgt å ikke anonymisere informantene, og det finnes to grunner for dette. Den ene er fordi informantens kunnskap og posisjon vil ha noe å si på oppgavens og svarenes troverdighet. Den andre er fordi det finnes et fåtall av personer som kan noe om programmering, grunnskolen og Kunst og håndverk og det kan dermed bli vanskelig å anonymisere. Med samtykke fra hver enkel informant, vil jeg nedenfor redegjøre for erfaringer, navn, yrke og utdanning.

Anette Seip

Er lektor i Kunst og håndverk med tilleggsfagene, valgfag programmering og arbeidslivsfag. Seip har faglærer utdanning og mastergrad ved Høyskolen i Oslo og Akershus (OsloMet) innen formgivning, kunst og håndverk. Hun har jobbet som lærer siden 2013. Seip er også initiativtaker i å etablere et skaperverksted på sin skole og har blant annet holdt foredrag på *Kunst- og håndverkskonferansen 2019* om skaperverkstedet.

Roger Antonsen

Er førsteamanuensis ved institutt for informatikk ved Universitetet i Oslo og gjesteforsker ved UC Berkeley i California. Han har også en grad som Candidatus Philologiae (Cand. Philol.) fra universitetet i Oslo. Antonsen har opprettet og bidratt til flere prosjekter som omhandler matematikk, informatikk, filosofi og kunst, slik som YouTube serien *Magiske Mønstre*. Han er forfatter og skribent og har blant annet skrevet boken *Logiske Metoder* (2014) og skriver en fast matematikkspalte om mønstre i Aftenposten. Antonsen er også kunstner, og skaper kunst basert på matematiske strukturer gjennom å programmere. Han har blant annet blitt stillet ut ved *Catharine Clarck Gallery* i San Francisco med Greg Niemeyer i prosjektet *The Network Paradox* fra januar til februar i 2019. Antonsen er en kjent foredragsholder og har bidratt med mange foredrag som omhandler en rekke forskjellige temaer, alt fra matematikk og informatikk til filosofi og kunst.

Liv Klakegg Dahlin

Er førstelektor og studieleder ved institutt for estetiske fag, OsloMet – storbyuniversitet. På oppdrag fra Utdanningsdirektoratet har Dahlin ledet arbeidet med utvikling av kjerneelementene i kunst og håndverksfaget, og leder nå arbeide med fornyelsen av læreplanen i Kunst og håndverk. Dahlin har i en årrekke ledet større og mindre utviklingsprosjekter med relevans for kunst og håndverksfaget, samt etter- og videreutdanninger. Hun er forfatter av flere bøker om tema innen teknologi og design og innen digitale medier i skolen.

Jon Øivind Hoem

Er førsteamanuensis ved institutt for kunsthøgskulen på Vestlandet – Bergen. Hoem har en doktorgrad (ph.d) innen IKT og læring. Han forsker særlig på robotassistert undervisning, sfæriske medier, utforskning av interaktiv installasjonskunst og bruk av mobilteknologi. Hoem leder forskernettverket IKT og medier i UH-nett Vest. Hoem har skrevet om programmering og Kunst og håndverk på HVL bloggen.

Transkribering og fremlegg av intervju

Ved transkriberingen har jeg valgt å skrive det informantene sier så ordrett som mulig, men i fremlegget av intervjuene har jeg valgt en mer leservennlig fremstilling, for å få en bedre flyt i teksten. Jeg har valgt å ta bort fyllord. Dette kan være utsagn som, «men, men, det er liksom, det er...» «da» eller andre ting som har lite å si for hva som blir sagt. I noen deler har jeg valgt å ta bort lengre partier av hva informantene sier, som ikke er like relevant for hva som blir sagt og dette har blitt markert med (...) i fremlegget av empirien. I fremlegget av analysen, er ikke relevante endinger som for eksempel «ikke sant» på slutten av setningen blitt tatt bort. Det er også viktig å huske at bruken av komma og punktum også kan ha blitt brukt feil og som videre kan ha en konsekvens for betydningen. Som blir fremlagt av kritisk realisme, filtreres all kunnskap og informasjon gjennom sansene til meg som forsker. Min bakgrunn, tanker og holdninger kan dermed ha hatt en innvirkning på hvordan empirien fremlegges og drøftes. Når det kommer til det datastøttede intervjuet var det allerede transkribert, likevel var det noen ord som var skrevet litt feil og disse har jeg valgt å rett opp i. Jeg har også valgt å ta bort det jeg selv sier, slik som, «mhm,» «ja» og «absolutt». Jeg har også valgt å putte ord i [] der jeg ønsker å presisere det som bli sagt. Disse ordene har ikke vært brukt av informantene, men har vært valgt av meg for å vise til hva informantene sikter til. I denne oppgaven har det vært viktigere hva som har blitt sagt enn hvordan noe har blitt sagt.

Analyse

Innenfor kritisk realisme finnes det ikke en bestemt måte å analysere empirien på. Dermed startet analysen i denne oppgaven ved å markere setninger fra det transkriberte intervju der det ble sagt noe interessant og relevant. Etter dette besto analysen i å finne ulike kategorier som informantenes svar kunne plasseres i. Bygstad og Munkvold (2011) forklarer at det eksisterer to ulike måter å identifisere viktige komponenter fra empirien, dette er enten ved en grounded tilnærming, eller at man sorterer empirien i et eksisterende rammeverk. I denne undersøkelsen har jeg valgt å kategorisere empirien med en grounded tilnærming. Likevel kan kategoriene ha blitt påvirket av intervjuguiden sin struktur og det tenkte rammeverket for intervjuene som er basert på mine tanker og refleksjoner gjort med utgangspunkt i teorien. Kategoriene vil bli fremlagt og gjennomgått i kapittel, *presentasjon av intervjuene*. Deretter vil kategoriene bli drøftet sammen med den praktisk-estetiske undersøkelsen i kapittel, *Drøfting* sammen med relevant teori, altså vil det få en mer abduktiv tilnærming.

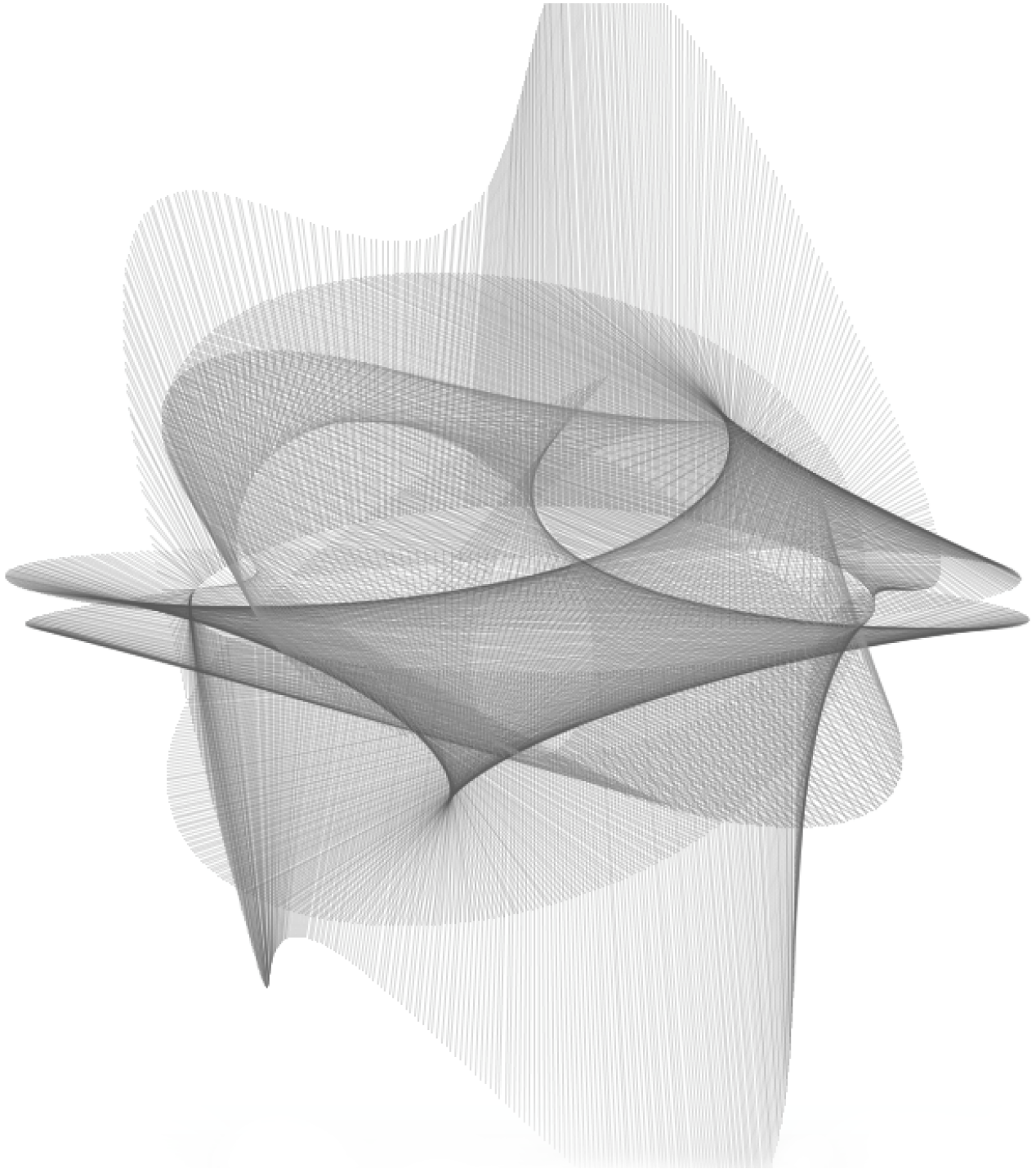
Validitet og reliabilitet

Innenfor kritisk realisme mener man at kunnskap er et sosialt produkt, denne kunnskapen er feilbarlig, men ikke alle utsagn er like feilbarlige, siden man har en vurderingsmessig rasjonalisme. Dermed blir det vesentlig at mine begrunnelser og utsagn er teoretisk og empirisk velbegrunnet (Næss, 2012). Ved velbegrunnete antagelser kan man si at utsagnet har høy troverdighet, som går på validitet. Siden sannhet blir et relativt fenomen er det viktig at jeg som forsker er i stand til både *personlig refleksivitet*, hvordan jeg påvirker forskningen og *epistemologisk refleksivitet*, hvordan valg av problemstilling og metode former innholdet i forskningen (Jacobsen, 2015).

Indre validitet omhandler hvor korrekt forskningsfunnene belyser problemstillingen (Østerud, 1998 s. 121) og ytre validitet omhandler i hvilken grad funnene kan generaliseres til andre omgivelser (Østerud, 1998). Da

jeg har valgt to metoder for innsamling av empiri kan dette øke validiteten. Metode triangulering går ut på at man har ulike metoder som undersøker problemstillingen, om empiriene som kommer av de ulike metodene viser at det er en konvergens vil dette bety at den indre validiteten øker (Krumsvik, 2014). Likevel kan valg av metodene svekke validiteten. Når det kommer til de semistrukturerte forskningsintervjuene har jeg et ganske lite utvalg, dette kan påvirke nyansen og bredden av det jeg forsøker å finne svar på i denne oppgaven. Da jeg ikke har målt alle mulige aspekter rundt forholdet til hvorfor, vil oppgaven kun belyse en liten del av problemstillingen, noe som er poengtert i innledningen, men også problemstillingen. De spørsmålene som er valgt til intervjuene kan også påvirke om spørsmålene belyser det som er mest sentralt og det er mulig å se problemområdet fra ulike perspektiver. Likevel har jeg valgt personer som har mye kunnskap innen programmering og grunnskolen. Utvalget består også av forskjellige mennesker, med ulike inngangsvinkler som kan være med på å styrke validiteten. Valg av programmeringsspråk i den praktisk-estetiske undersøkelsen kan også ha påvirket mulige refleksjoner som har kommet, men også den valgte prosessen. I prosessen har jeg ikke fått mulighet til å undersøke alle mulige aspekter som omhandler programmering. Som igjen kan ha konsekvenser på hvordan jeg har tolket intervjuene, hvilke spørsmål som har blitt stilt og hvilke refleksjoner som har blitt gjort.

Reliabilitet omhandler i hvilken grad de funnene jeg finner i denne oppgaven kan bli gjentatt eller eventuelt reproduisert av en annen (Østerud, 1998). Ved bruk av forskningsintervju og praktisk-estetisk utforskning vil jeg si at det kan være vanskelig for en annen å få samme svar som meg. Den praktisk-estetiske utforskningen handlet mye om refleksjoner i handlingen som videre kan ha fått konsekvenser for hvilken retning og hva som har blitt undersøkt. Likevel kan hvem som helst ha en utforskning i samme programmeringsspråk og kanskje noen av refleksjonene vil ha blitt de samme. Da læring av BCC micro:bit og Scratch omhandlet å prøve og feile, kan det være noe vanskelig for en annen å gjenta samme handling. Ved læring av Processing og Arduino, brukte jeg veiledningsvideoer på Youtube, som andre også kan følge. Dette kan gjøre det enklere for andre å ha lignende prosess som meg. Teknologisk utvikling kan også bidra til at det blir gjort andre valg og refleksjoner. Det kan også tenkes at siden jeg både er matematikk- og naturfagslærer, så kan jeg ha et annet inntrykk av programmering, enn en som kun er kunst og håndverkslærer. Dette kan påvirke refleksjonene både i handling og i refleksjonene av handlingene. Når det kommer til semistrukturert forskningsintervju kan andre spørsmål bli stilt og valg av andre informanter velges, dette kan påvirke hva som blir samlet inn av empiri. Det kan også tenkes at kommunikasjonsforståelse slik som hvordan informantene har oppfattet spørsmålene og hvordan jeg har tolket svarene kan ha en innvirkning på hva som har blitt spurt, og hva som har blitt svart. Det er likevel mulig å bruke intervjuguiden og samme informanter. Som kanskje kan gi noe likt svar. Videre kan det også tenkes at informantene også utvikler sin kunnskap tilknyttet problemområdet, som igjen kan gi andre svar. Kommunikasjonsforståelse- og analyseforståelse kan også ha påvirkning på transkriberingen og tolkningen av intervjuene. Likevel kan valg av to forskningsmetoder styrke reliabiliteten, da problemstillingen blir undersøkt fra to forskjellige vinkler.



```

static final int NUM_LINES = 700;
static final int NUMM_LINES = 700;

float t;
float s;
float m;

void setup() {
  size(800, 700);
}

void draw() {
  strokeWeight(1);
  background(255);

  translate(width/2, height/2);
  stroke(90, 60);
  for (int i=0; i < NUM_LINES; i += 1) {
    line(x1(t + i), y1(t + i), x2(t + i), y2(t + i));
    t += 2;

    strokeWeight(1);
    stroke(147, 80);

    for (int i=0; i < NUMM_LINES; i += 1) {
      line(x3(s + i), y3(s + i));
      s += 0.6;

      stroke(90, 60);
      strokeWeight(1);

      for (int i=0; i < NUMM_LINES; i += 1) {
        line(x5(m + i), y5(m + i));
        m += 0.6;
      }

      float x1(float t) {
        return sin(t/60) * 300 + sin(t/35) * 13;
      }

      float y1(float t) {
        return cos(t/50) * 24;
      }

      float x2(float t) {
        return sin(t/70) * 210 + sin(t/42) * 12;
      }

      float y2(float t) {
        return cos(t/90) * 190 + sin(t/55) * 10;
      }

      float x3(float s) {
        return sin(s/10) * 100 + sin(s/20) * 100;
      }

      float y3(float s) {
        return cos(s/10) * 100;
      }

      float x4(float s) {
        return sin(s/10) * 120 + sin(s/10) * 100;
      }

      float y4(float s) {
        return cos(s/10) * 100 + sin(s/10) * 100;
      }

      float x5(float m) {
        return sin(m/40) * 200 + sin(m/20) * 20;
      }

      float y5(float m) {
        return cos(m/40) * 50;
      }

      float x6(float m) {
        return sin(m/40) * 220 + sin(m/10) * 35;
      }

      float y6(float m) {
        return cos(m/40) * 320 + sin(m/10) * 82;
      }
    }
  }
}

```

Bilde 3

Animasjon cosinus og sinus bølger (2018) Eget bilde

Praktisk estetisk undersøkelse

Det praktisk-estetiske har vært en del av undersøkelsen i oppgaven. I den praktisk-estetiske undersøkelsen har jeg forsøkt å finne svar på hvilke skapemuligheter som ligger i programmering knyttet til Kunst og håndverk som videre kan belyse hvorfor det bør inngå i Kunst og håndverk på grunnskolen. Et ønske har også vært å gi et innblikk i programmering. Programmering kan være et nok så nytt emne for mange, og det er ønskelig å åpne opp for en forståelse av programmering både for meg, men også for leseren. Undersøkelsen startet med en utforskning av hva programmering innebærer og hva slags programmeringsspråk som kan være relevant for kunst- og håndverksfaget på grunnskolen. Undersøkelsen inneholdt blant annet søking på nett, deltakelse ved workshops på *Teknoteket Makerspace* ved Tekniske museum og *Makeriet* ved OsloMet. Jeg har også deltatt på fagfrokoster på Universitetet i Oslo, på *Seminar om programmering i skolen* den 14. november 2018 og *Programmering i skolen – muligheter og utfordringer* den 12. oktober 2018. Jeg var også deltaker på *Oslo Skaperfestival 2018* ved Deichman Hovedbibliotek. Den praktisk-estetiske undersøkelsen har gått parallelt med det teoretiske arbeidet, dette både for å få en praktisk forståelse over hva som blir sagt i intervjuene, men også for å få en teoretisk forståelse av hva programmering innebærer. Undersøkelsen har gitt meg en større innsikt over hva som bør blir spurt om i intervjuene, men intervjuene har også gitt inspirasjon på hva den praktisk-estetiske undersøkelsen skal inneholde.

Programmeringsspråk

Til dags dato finnes det opp mot tusen ulike programmeringsspråk, der alle har forskjellige formål og passer til ulike brukergrupper fra nybegynner til mer erfarne utviklere (Nygård, 2018). Det er også slik at ulike programmeringsspråk har blitt utviklet av og bygger på hverandre. Man kan dele programmeringsspråk i ulike sjangre, blant annet *lavnivå programmeringsspråk* og *høynivå programmeringsspråk*. Høynivå programmeringsspråk er nærmere den menneskelige taleformen og lavnivå programmeringsspråk er nærmere original språket til datamaskinen, som består av binære sekvenser. Høynivå programmeringsspråk blir ofte omgjort av en *kompilator*, som er en programvare som gjør om et programmeringsspråk til et annet, og språket blir ofte omgjort til et lavnivåprogrammeringsspråk slik som maskinkode (Nygård, 2018). Det finnes også *blokkbasert programmeringsspråk* og *tekstbasert programmeringsspråk*, der blokkbasert programmering går ut på å bruke visuelle blokker som sammenkobles, slik at man får større blokker og programmer. Eksempler er Scratch, Snap, BBC micro:bit, Alice, Greenfoot etc. Blokkbasert programmering kan også komme inn under betegnelsen *grafisk programmeringsspråk* (Nygård, 2018). Ved tekstbasert programmering bruker man tekst i form av bokstaver, tall og symboler. Koding går her ut på å skrive kommandoer, der det er viktig å bruke riktig syntaks. Eksempler på språk er Python, Processing, Java, C++, etc (Nygård, 2018). De ulike programmeringsspråkene brukes også til ulike funksjoner og plattformer, slik som C++ brukes til Windows og Google Chrome, Python til Instagram, Spotify, Pinterest og Youtube, og Java brukes til Google, EBay, LinkedIn og Amazon (Nygård, 2018).

Valg av programmeringsspråk og -plattformer

I utforskningen av programmeringsplattformer ble blant annet Scratch, BBC micro:bit, Processing, Arduino, p5.js, CSS og JavaScript testet, men også formateringsspråket HTML. Grunnleggerne av Processing utviklingsmiljøet Casey Reas og Ben Fry (2006) sammenligner programmeringsspråk med ulike materialer, og mener at ulike materialer er passende for ulike oppgaver. Dermed blir noen programmeringsspråk passende for noen formål og andre programmeringsspråk passende for andre formål. Dette er også noe min utforskning viste, der det blant annet var et klart skille mellom programmeringsplattformer som passet bedre til programvareutvikling av digitale plattformer og andre til maskinvare. I utvalget av programmeringsplattformer med tilhørende programmeringsspråk har jeg valgt to programmeringsspråk som er tekstbaserte, dette er Processing og Arduino (APL), og to som er blokkbaserte, BBC micro:bit (MakeCode Editor) og Scratch. Processing og Scratch brukes i større grad for å programmere visuelle- og audiovisuelle uttrykk som er digitale, men Arduino og BBC micro:bit brukes i større grad for å programmere fysiske gjenstander. Grunnen for valget av de forskjellige programmeringsplattformene er for å kunne få en kjennskap til både blokkbasert og tekstbasert programmering, men også kunne lage ulike gjenstander innen det digitale og det fysiske.

	Språkform	Formål
Scratch	Blokkbasert	Programvare til digitale produkter
BBC micro:bit	Blokkbasert (Tekstbasert)	Programvare til digitale og fysiske produkter
Processing	Tekstbasert	Programvare til digitale produkter
Arduino	Tekstbasert	Programvare til digitale og fysiske produkter

Tabell 1: Valg av programmeringsplattformer og mulige bruksområder (2018) Egen tabell

De ulike programmeringsspråkene og fremgangsmetode

Under vil det komme en forklaring av de ulike programmeringsspråkene jeg har valgt å undersøke videre i denne oppgaven. Det vil også komme en kort beskrivelse av hvordan jeg har arbeidet med de ulike programmeringsplattformene. Deretter vil det komme et delkapittel der jeg har utviklet en tabell ut fra refleksjonene jeg har hatt i og av den praktisk-estetiske undersøkelsen. I det praktiske arbeidet under den teoretiske delen av masteroppgaven har jeg i hovedsak fokusert på Scratch og Processing. Jeg startet arbeidet mitt i Processing og forsettet med Scratch, deretter arbeidet jeg parallelt med de to nevnte programmeringsplattformene. Jeg startet også med Arduino og BBC micro:bit og undersøkte ulike måter å programmere maskinvare på. I utforskningen med BBC micro:bit og Arduino fikk jeg ikke startet med rammeverket som skal være rundt maskinvaren jeg programmerer, som er et ønsket i masterutstillingen.

Processing

Processing er et programmeringsspråk og -utviklingsmiljø som er utviklet for kunst relaterte emner, som design, arkitektur og visuell kunst (Reas & Fry, 2006). Grunnleggerne av Processing, Casey Reas og Ben Fry (2006) forklarer programvaren slik «processing is created to teach fundamentals of computer programming within a visual context, to serve as a software sketchbook, and to be used as production tool for specific contexts. It is used by students, artist, design professionals, and researchers for learning, prototyping, and execution» (Reas & Fry, 2006, s. 527). En av grunnene til at Processing ble utviklet, var for å bedre kunne undervise i konsepter innen programvare og interaksjon i kunst- og designskoler. Reas og Fry mener at endring i teknologien bidrar til ulike former for kommunikasjon og uttrykk. De mener at programvare generelt er «unique among artistic mediums in its ability to produce dynamic form, process gestures, to produce behavior, simulate natural systems, and integrate various media including sound, image and text» (Reas & Fry, 2006, s. 528). Programmeringsspråket har to fokuspunkter når det kommer til utdanning, den ene er at Processing kan gi læring om konsepter innen programvare i en kunstrelatert konteksten, det andre er å åpne opp for kunstkonsepter for mer tekniskanlagte personer (Reas & Fry, 2006). Programmet er basert på Java, som igjen har noen likhetstrekk med programmeringsspråket C. Dermed kan personer som allerede kan litt om programvareutvikling i C lett lære seg Processing.



Bilde 4: Skjermbilde av frihåndstegning gjort med kode som vist i bilde 5 (2018) Eget bilde

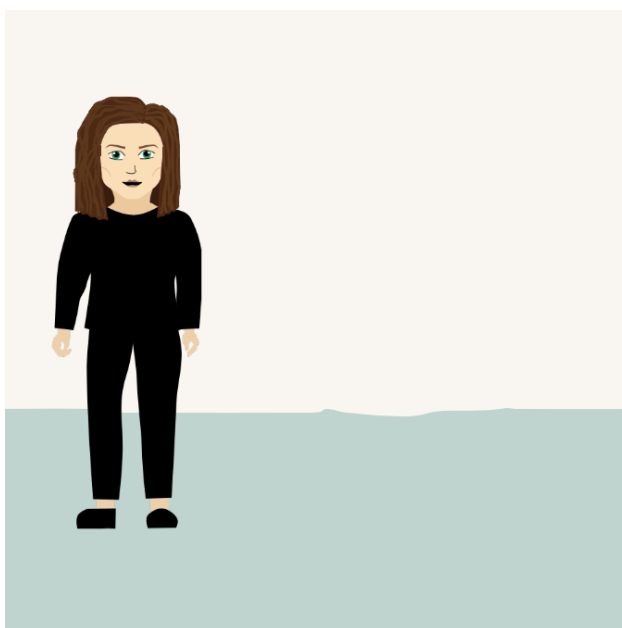
```
1
2 void setup () {
3   background (240);
4   size (650,700);
5 }
6
7 void draw () {
8   strokeWeight (0.5);
9   stroke (100,70);
10
11  if (mousePressed) {
12    line(mouseX,mouseY,mouseX,400);
13  }
14 }
```

Bilde 5: Skjermbilde av kode i Processing utviklingsmiljøet. Kode gir et tegneredskap som er brukt i bilde 4 og 1 (2018) eget bilde

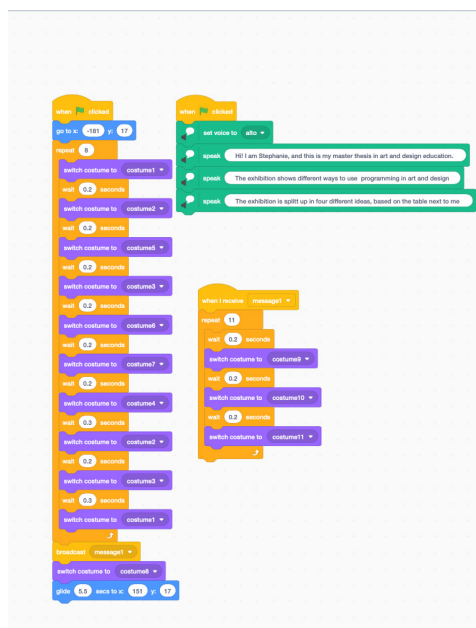
Arbeidet med Processing startet med å følge en Youtube leksjon fra kanalen The Coding Train av Daniel Shiffmans (2015). Leksjonene heter *Learning Processing: A Beginner's Guide to Programming Images, Animation and Interaction*. De første leksjonene var en introduksjon til Processing, de nesten omhandlet piksler, Processing miljøet, interaksjon, variabler, boolsk uttrykk, løkker, funksjoner, objekt-orientert programmering, tabell (Engelsk: arrays) og til slutt bilder og piksler. Etter hver leksjon arbeidet jeg med å lage egne uttrykk ved hjelp av det jeg hadde lært. Andre ganger forsøkte jeg å lage uttrykk på egenhånd og bruke kunnskapen fra de ulike leksjonene. De første visualiseringene besto av enkle former og utforskning av farger. Deretter gikk fokuset mer mot å skape uttrykk ved hjelp av interaksjon. Altså programmerte jeg ulike former som det var mulig å bevege og tegne med (bilde 1, 4 og 5). Jeg laget også former som det var mulig å interagere med (bilde 13). Senere i prosessen arbeidet jeg med visualiseringer med bevegelser og tilfeldighet (forsidebilde og bilde 2) og laget systemer som kunne settes i gang slik at de tegnet egne uttrykk (bilde 3 og forsidebilde).

Scratch

Scratch ble lansert i 2007, og bygger på prinsipper fra ideene bak programmeringsspråket LOGO og Etoys (Maloney, Resnick, Rusk, Silverman & Eastmond, 2010). Scratch er et prosjekt som tilhører *Lifelong Kindergarten Groupe* ved MIT media Lab (Scratch, u.å). Formålet med Scratch er å lære barn fra 8 til 16 år prinsipper bak programmering. Utviklerene har fokusert på at programmeringsplattformen skal bli lært på egenhånd - uten en lærer - gjennom utforskning, samarbeid med andre og deling. På programmeringsplattformen har man ulike områder for å arbeide med ulike ting. I et område kan man utvikle egne figurer, som kalles *spirit*, og lage ulike kostymer til spiritene. Det er også mulig å laste ned bilder og foto fra andre steder. I et annet område skriver man koden og i et tredje område kan man spille inn lyder. Det er mulig å skape innenfor mange ulike former for media, slik som visualiseringer, bilder, musikk og lyd.



Bilde 6: Skjermbilde av en tegning som viser et kostyme i Scratch utviklingsmiljøet, brukt til kode som vist i bilde 7 (2019) Eget bilde

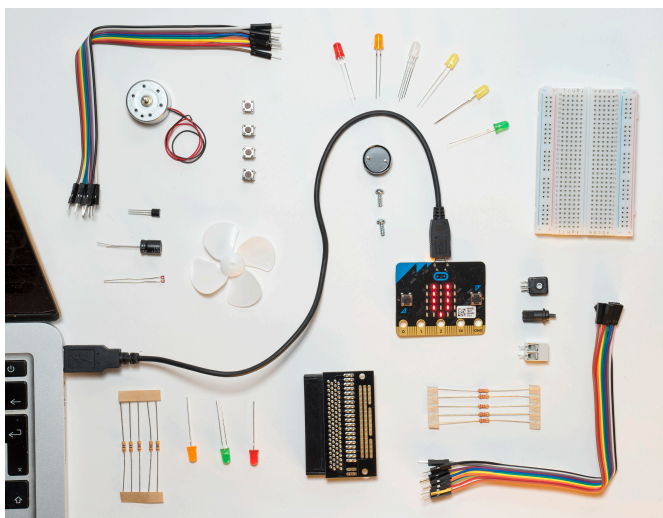


Bilde 7: Skjermbilde av utviklingsmiljøet i Scratch med kode til bilde 6 (2019) Eget bilde

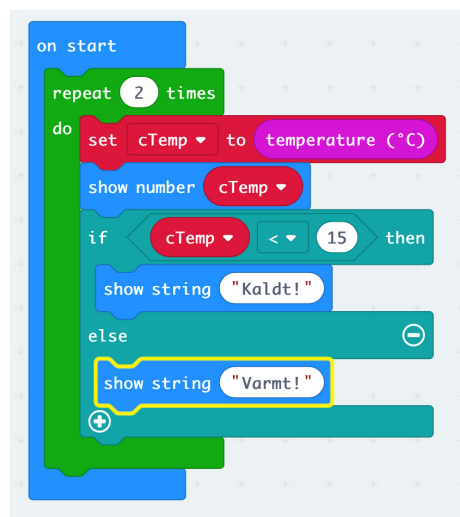
Jeg begynte å lære Scratch ved å være en frivillig på en kodeklubb. Undervisningen gikk ut på å følge en oppgavetekst fra KidsaKoder (u.å.a) med tittelen *Straffespark*. Jeg forsto fort at programmeringsverktøyet var nok så intuitivt, dermed besto min læringsprosess i å prøve og feile. Det er heller ikke nødvendig å lære seg syntakser, da kodene allerede ligger som blokker. Hvis jeg sto fast eller om det var noe jeg lurte på søkte jeg etter svar på nettet. Jeg brukte Scratch noe annerledes enn Processing, og jeg fokuserte heller på å utvikle animasjoner og spill. Jeg arbeidet mye med en animasjon som mest sannsynlig skal brukes som en informasjonsfilm til utstillingen (bilde 6 og 7). Her var fokuset mest på å utvikle spirits med tilhørende kostymer. Jeg arbeidet også med å prøve å lage ulike mønstre som er et punkt under den første og andre skissen til den kommende læreplanen (bilde 12).

BBC micro:bit

I hovedsak er BBC micro:bit en ettkortsdatamaskin (et lite programmerbart kort). BBC micro:bit startet som et Initiativ i Storbritannia der alle elever i syvende klasse fikk en BBC micro:bit hver. BBC micro:biten ble utviklet av et samarbeid mellom utdanningsinstitusjonen i Storbritannia, BBC og programvare- og maskinvareselskaper. BBC micro:biten er laget slik at det skal være enkelt å komme i gang, men at det også skal være mulig å skape mer kompliserte applikasjoner. Den lille datamaskinen som BBC micro:bit er, inneholder en prosessor, ulike input og output komponenter og muligheten for trådløs forbindelse. Den har 25 LED lys, to knapper, akselerometer, magnetometer (kompass), temperatursensor, lys-sensor og alt innenfor et 4 x 5 cm Brett (Ball, Protzenko, Bishop, Moskal, Halleux & Braun, 2016; Schmidt, 2016;). Designet på BBC micro:bit er utformet slik at den skal være enkelt og morsom å bruke, og bidra til engasjement, fantasi og kreativitet (Ball et al., 2016). BBC micro:biten kan bli programmert ved hjelp av ulike utviklingsmiljø slik som Code Kingdoms, Microsoft Block Editor, Microsoft Touch Develop og Python (Schmidt, 2016). Det er også mulig å kjøpe ulike kitt som ekspanderer hva man kan programmere av maskinvare (bilde 8). I den norske grunnskolen er det nå lovet at 100 000 grunnskolebarn skal få en BBC micro:bit (Teknisk Museum, u.å.b)



Bilde 8: Foto av maskinvare i BBC micro:bit inventorskit og BBC micro:bit programmert som et kompas (2019) Eget foto

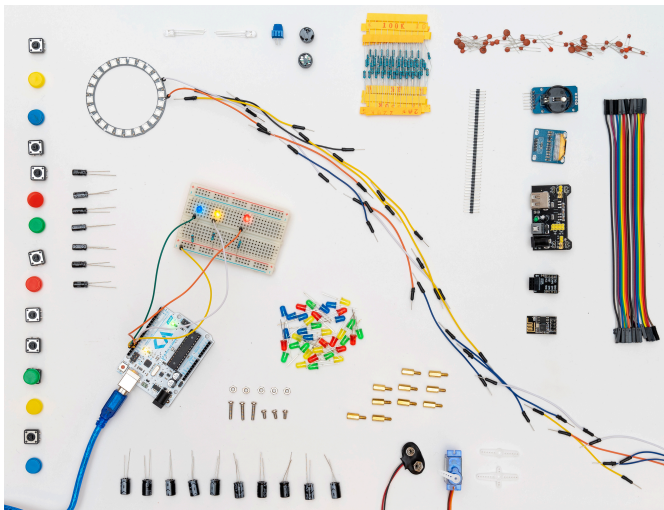


Bilde 9: Skjerm bilde av en kode i BBC micro:bit utviklingsmiljøet. Kode av temperatursensor og ledlys på ettkoertdatamaskinen til BBC micro:bit (2018) Eget bilde.

I arbeid med BBC micro:bit brukte jeg MakeCode Editor som baserer seg på JavaScript. Her kan man bygge opp koden på en blokkbasert måte, for så å se denne koden i JavaScript. Der er også mulig å arbeide med JavaScript inne på utviklingsmiljøet. For å teste ut BBC micro:bit hadde jeg samme strategi som ved Scratch, altså å prøve meg litt frem, men også se etter oppgaver på nett. Jeg startet først med å teste ut temperatursensor, magnetometer (kompass), akselerometer, og lys-sensoren for å finne ut om de fungerte og hvordan de fungerte. Jeg laget blant annet en temperaturmåler som lyste på ledlysene «kaldt» om det var under 15 C, og «varmt» om det var over 15 C (bilde 9). Her fulgte jeg en leksjon av That Cat (2016) på YouTube med navn *How to Read the Temperature Sensor on the BBC MicroBit*. Jeg forsøkte også å lage et kompass ved å bruke magnetometeret (bilde 8). Når man beveget på BBC micro:biten dukket det opp bokstavene N, S, V eller Ø på ledlysene alt utfra i hvilken retning ettkortsdatamaskin ble pekt. Jeg testet så ut akselerometret og kodet den til å lage en graf som viste hvilke bevegelser jeg gjorde med den. Inspirasjonen hentet jeg fra makecode.microbit (u.å).

Arduino

Arduino er skapt for at både nybegynner og erfarne programmerere kan skape og bygge egne elektronikkprosjekter. Man kan bruke Arduino-plattformen til å skape noe utenfor mobilens-, nettbrettets- eller datamaskinens skjerm ved hjelp av programmering. Arduino kan ligne på BBC micro:bit, men utviklingskortet til BBC micro:bit har en liten skjerm på 25 pixler. Arduino består av mange forskjellige ting, blant annet et utviklingskort, et utviklingsmiljø som kalles Arduino IDE og et programmeringsspråk som kalles Arduino Programming Language (APL). I motsetning til BBC micro:bit er originalspråket man programmerer i tekstbasert og ikke blokkbasert, dette kan gjøre at plattformen passer til litt eldre elever. Utviklingskortet inneholder en liten mikrokontroller, som er en liten datamaskin med prosessor, minne og inn- og utganger (Söderström & Nikka, 2017). Resten av utviklingskortet inneholder ulike inn- og utganger, inngang til strømforsyning, USB tilkobling, omstartsknapp og mikrokontroller. Mikrokontrolleren kan bli programmert slik at man kan styre de tilkoblede delene på utviklingskortet (Söderström & Nikka, 2017). Det finnes forskjellige modeller av utviklingskort som har ulike bruksområder, og i min oppgave bruker jeg Arduino Uno (bilde 10).



Bilde 10: Foto av maskinvare til Arduino og Arduino Uno kortet som har blitt programmert til ledlys (2019) Eget foto.

```
LED3 | Ardui
int LED1 = 13;
int LED2 = 12;
int LED3 = 11;
int LED4 = 10;
int LED5 = 9;

void setup() {
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
  pinMode(LED5, OUTPUT);
}

void loop() {
  digitalWrite(LED1, HIGH);
  delay(70);
  digitalWrite(LED2, HIGH);
  delay(70);
  digitalWrite(LED3, HIGH);
  delay(70);
  digitalWrite(LED4, HIGH);
  delay(70);
  digitalWrite(LED5, HIGH);
}
```

Bilde 11: Skjerm bilde av Arduino utviklingsmiljøet, med kode av dioder (2018) Eget bilde

I arbeid med Arduino fant jeg veiledningsvideoer på Youtube, den første jeg startet med var *Arduino Tutorial: LED Sequential Control- Beginner Project* (2015) av Drone How. Leksjonen gikk på å lage en elektronisk krets, der man kunne kontrollere lysene ved hjelp av programmering (bilde 10 og 11). Jeg gikk så videre med leksjoner i hvordan å styre servomotoren, og jeg eksperimenterte med ulike settinger om hvor fort den skulle gå rundt, eller om den bare skulle bevege seg 180-, 360- eller 90 grader. Til slutt fulgte jeg en annen YouTube leksjon av Hardik Rathod (2017) med navn *Control Arduino Using GUI (Arduino + Processing)*. Leksjonen gikk på å lage en kontroll i Processing som kan kobles til Arduino slik at man kan styre ledlysene. Jeg eksperimenterte videre med å legge til flere lys og flere knapper på kontrollen.

Hva kan man programmere innen Kunst og håndverk

I den utforskende fasen har jeg kommet frem til fem hovedområder for skapemulighetene til programmering innenfor Kunst og håndverk sitt rammeverk. Nedenfor følger en tabell med de ulike områdene, hva jeg legger i det og noen eksempler på praktiske prosjekter. Tabell 2 har blitt utviklet gjennom refleksjoner jeg har hatt under arbeidet, i etterkant og informasjon fra informantene.

Område	Forklaring	Eksempler
Kreativ koding av digitale produkter	Digitale produkter som enten er to- eller tredimensjonale. Det kan være audiovisuelle uttrykk, animasjoner, visuelle uttrykk. Her skapes det ingen form for funksjon, kun den estetiske dimensjonen. Området blir knyttet mer til kunstbegrepet.	Visualiseringer (bilde 2 og 3), grafiske fremstillinger, animasjoner (bilde 6), film, mønsterdannelse (bilde 12), form og installasjoner.
Programvare av digitale produkter	Digitale produkter som enten er to eller tredimensjonale. Det kan være audiovisuelle uttrykk, animasjoner, visuelle uttrykk. Her skapes det både funksjon og uttrykk. Området blir knyttet mer til designbegrepet.	Digitale kontroller, apper, hjemmesider, visualiseringer med parametere (bilde 13), dataspill og programvare med tegneredskaper (bilde 1 og 4).
Kreativ koding av fysiske produkter	Fysiske produkter med integrert maskinvare. Her skapes det ingen form for funksjon, bortsett fra den estetiske dimensjonen. Området blir knyttet mer til kunstbegrepet.	Installasjoner eller skulpturer med integrert maskinvare.
Programvare av fysiske produkter	Fysiske produkter med integrert maskinvare. Her skapes både form og funksjon. Området blir knyttet mer til designbegrepet.	Design med integrert maskinvare, slik som smarte tekstiler og lamper.
Verktøy til å utforske begreper i kunst og håndverk	Å bruke programmeringen til å utforske og lære om sentrale begreper i Kunst og håndverk.	Utforske mønster (bilde 12), former (bilde 13) og lysfarger (bilde 2).

Tabell 2: Mulige skapemuligheter med programmering i Kunst og håndverk (2018) Egen tabell

Tabellen har to hoveddimensjoner, dette er enten *kreativ koding* og *programvare*, eller *digitale produkter* og *fysiske produkter*. Kjetil Sømoe (2010) beskriver Kunst og håndverk utfra tre dimensjoner. Der den ene akse har design (bruksform) på den ene siden og kunst (bilde og skulptur) på den andre siden. I tabellen har jeg valgt denne formen for inndeling, der kreativ koding av digitale produkter og fysiske produkter er rettet mot kunstpraksisen og programvare av digitale produkter eller fysiske produktet er rettet mot designpraksisen. Som jeg har utredet før, kan koding bli sett på som et materiale, der formålet ikke er å skape noe funksjonelt, men å skape noe uttrykksfullt. Dette begrepet har jeg knyttet mot kunstbegrepet i Kunst og håndverk. Programvare har jeg knyttet til designbegrepet i Kunst og håndverk, for å vise at dette også omhandler funksjon. Jeg har også valgt denne inndelingen fordi produktene som blir skapt under kreativ koding er mer rettet mot et publikum som er betraktere eller deltagere, heller enn forbrukere. Under kreativ koding og

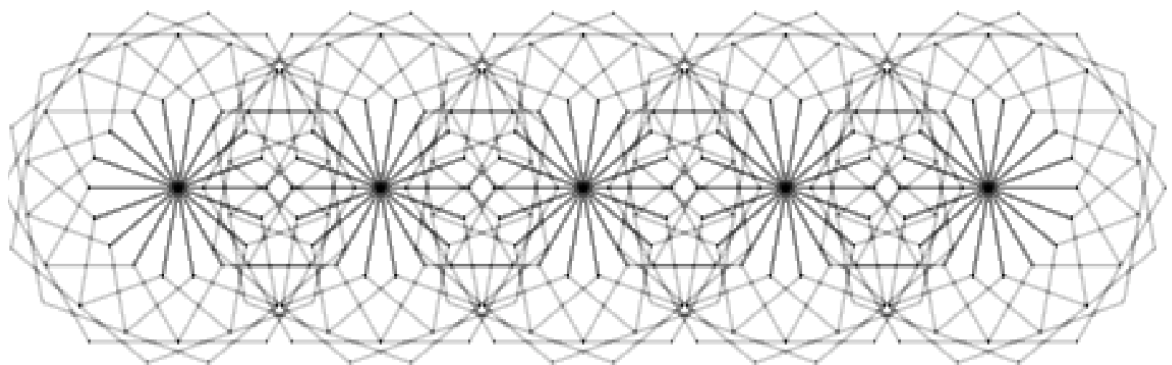
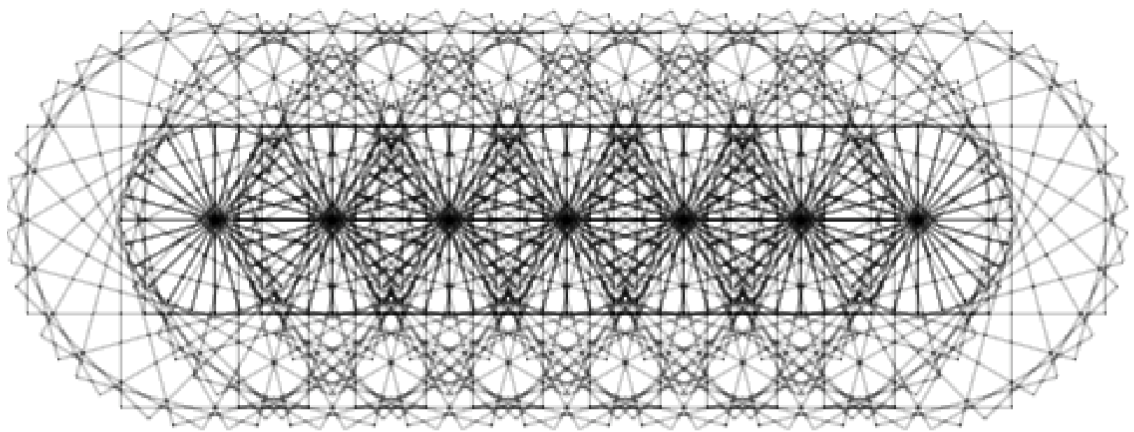
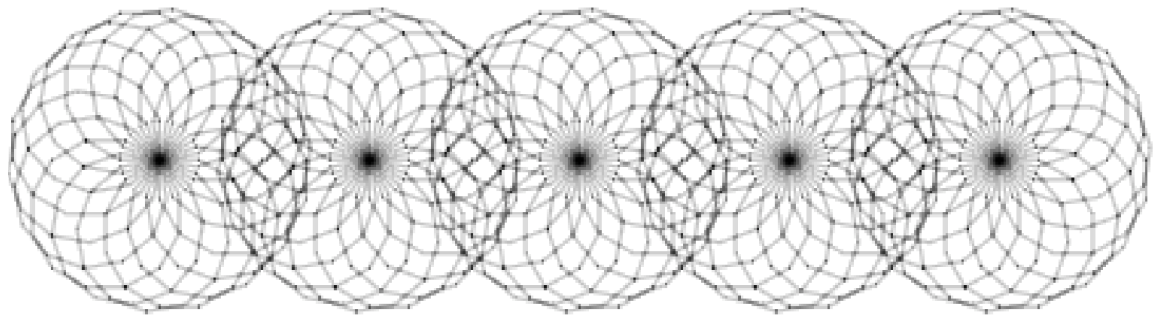
programvare av digitale produkter kommer også uttrykkene som er programmerte, men som kan gå utenfor skjermens rammer, eksempler er former og uttrykk som det er mulig å 3D-printet eller laser kutte. Under programvare har jeg plassert gjenstander som i større grad fokuserer på samhandlingen mellom produkt og mennesket. Og der mennesket som interagerer med gjenstanden i større grad blir forbruker og bruker, enn deltager og betrakter.

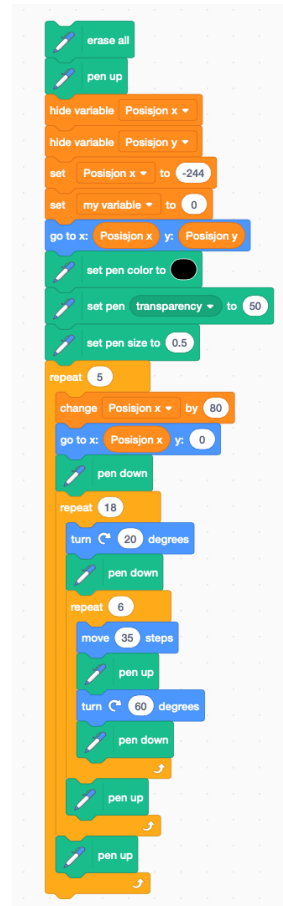
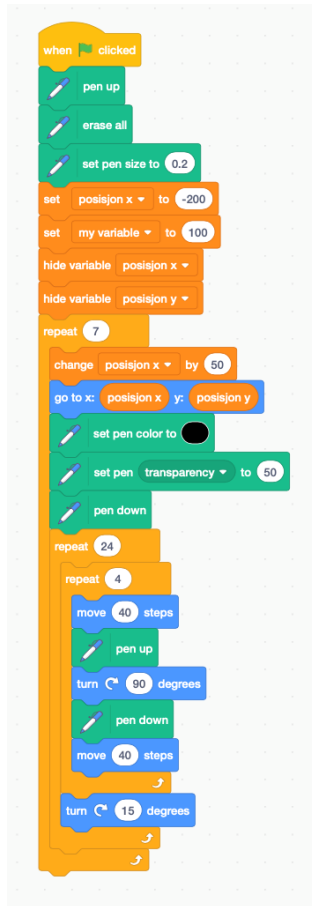
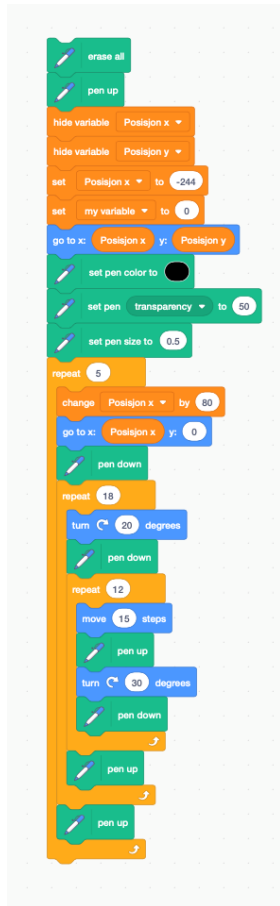
Jeg har også valgt en inndeling av digitale produkter og fysiske produkter, dette for å skille mellom maskinvare og programvare, men også fordi programvare ofte blir skapt til enten digitale uttrykk på en skjerm eller maskinvare slik som lys og bevegelse. Dette for å trekke linjer mot digitale uttrykk og digital media, men også fysiske gjenstander slik som smarte tekstiler og skulpturer. Digitale produkter omhandler både kunstneriske uttrykk men også designprodukter. I fysiske produkter går større deler av det praktiske arbeidet ut på å lage rammeverk eller koble sammen maskinvare som kan bli programmert. Her er det ikke kun fokus på å programmere deler som allerede eksisterer i en datamaskin, men å gå utenfor skjermens rammer. En slik praksis forandrer skapehandlingen til å i større grad omhandle taktile og håndverksmessige handlinger. Selv om denne delingen har blitt brukt, er det også mulig å skape på tvers av de ulike kategoriene for å skape kunstverk eller designprodukter.

Den siste kategorien i tabell 2 *Verktøy til å utforske begreper i Kunst og håndverk*, er også et viktig aspekt ved programmering knyttet til Kunst og håndverk. Denne kategorien har i hovedsak blitt til utfra hvordan programmering omtales i den nye fagfornyelsen, men også utfra informasjon fra informantene. Kategorien er tenkt som et verktøy til å utforske sentrale begreper i Kunst og håndverk. Begrepet lysfarger var noe jeg oppdaget at jeg kunne lite om da jeg startet å programmere i Processing. Ved å benytte denne plattformen har jeg fått en mye større innsikt og forståelse for lysfarger, enn det jeg hadde fra før. Sett ut fra et slikt rammeverk, kan tekstbasert programmering bidra med noe annet enn det blokkbasert programmering kan.

Et viktig aspekt som jeg også har innhentet gjennom refleksjoner i det praktisk arbeidet, er interaksjon. Programmering kan bidra til at man i større grad kan interagere med ulike objekter eller uttrykk. Posisjon, orientering, bevegelse, berøring, lys og lyd kan påvirke funksjonen eller uttrykket. Som eksempel kan man blant annet lage en kontroll som påvirker hvordan lyset i en lampe skal lyse. Det er mulig å lage visualiseringer som blir påvirket av hvordan betrakteren eller deltakeren interagerer med uttrykket. Det kan være slik som å lage et tegneredskap, som reagerer på posisjonen til musepilen på skjermen (bilde 1 og 4), eller sensorer som plukker opp en lyd, som påvirker hvordan uttrykket reagerer. En vurdering var å lage tabellen basert på dette aspektet siden det er en så viktig del av programmeringen. Ved en slik deling mistet jeg likevel de to andre viktige faktorene ved programmering som er kreativ koding, programvare.

Et ønske under masterutstillingen er å skape produkter som ligger innenfor alle de fire øverste domenene av tabell 2. Dette for å gjenspeile funnene i denne oppgaven, men også for videre å reflektere over det fysiske aspektet ved programmeringen. Begrepene som er brukt i tabell 2, og tabell 2 vil bli omtalt og henviset til i drøftingen.





Bilde 12:
Utforskning av mønstre (2018) Eget bilde

Presentasjon av intervjuene

I denne delen vil jeg legge frem empirien som er innhentet gjennom intervjuene. Analysen er gjennomført slik det beskrives i delkapittel *Analyse*. Først kommer et generelt fremlegg av hver informants posisjonering angående tematikken som er tolket av meg. Informantenes bakgrunn er beskrevet i delkapittel *Utvalg*. Deretter vil punktene som nevnt under, bli utdypet og fremlagt med utsagn fra de ulike informantene. Først vil punkter som støtter problemstillingen *Hvilke mulige årsaksforklaringer kan belyse hvorfor programmering skal implementeres i Kunst og håndverk på grunnskolen* bli kartlagt. Deretter fremlegges mulige fallgruver og begrensninger for å implementere programmering i Kunst og håndverk.

Fremlegg av informantenes posisjonering

Roger Antonsen

Antonsen har et positivt syn når det kommer til at programmering skal bli innført i grunnskolen. Han mener likevel at selv om programmering blir innført så trenger det ikke å føre til at fagene eller skolen blir noe bedre. Han mener at programmering har en mulighet til å tilføre fagene noe, om det brukes på riktig måte. Antonsen ser også at programmering har potensiale i alle fag, der han fremlegger: «(...) jeg mener at alt kan programmeres. Alt fra gym til biologi, til kunst, til matematikk og fysikk». Han mener likevel at det er viktig å vite hvordan man skal bruke programmering i de ulike fagene og at det ikke må bli en pedantisk undervisning med mekaniske øvelser. Senere i intervjuet sier Antonsen også at man kan argumentere for at programmering må være en del av noen fag, siden det er en så essensiell del av fagene. Her bringer Antonsen frem Matematikk og Kunst og håndverk, og legger til at programmering kan være et essensielt verktøy i begge fagene. Antonsen mener også at det er unaturlig å skille på koding, programmering og algoritmisk tankegang, da han mener at begrepene hører sammen. Antonsen bruker programmering og koding om hverandre, men er kjent med tanken om å dele begrepene, der koding omhandler det håndverksmessige og programmering omhandler å se strukturer og systemer. Antonsen er likevel opptatt av algoritmisk tankegang i kunst, og fremhever dette med:

«Jeg tror man har sunt av å tenke algoritmisk i kunst og håndverksfaget. Det er bare å gå tilbake i litteraturen og se på The book of Kells, det keltiske knutetekstverket og Lindisfarne Gospels. Hvis du ikke ser den algoritmiske tankegangen i knutemønstrene der, så har du gått glipp av noe. Hvis du reiser til Alhambra og ser på mønstrene der, og hvis du ikke ser den algoritmiske tankegang og alle mønstrene og strukturene som ligger bak, så har du gått glipp av noe. Når du strikker et skjerf, det er jo algoritmisk tankegang, fordi du har fem rette og fem vrangle, og så snur du. Det er en algoritme!»

Antonsen mener at algoritmisk tankegang og programmering kan åpne opp for en dypere forståelse i Kunst og håndverk og vitenskap. Han mener også at algoritmer er en viktig del av kulturen vår nå og før, der han

både henviser til historien og samtiden. Antonsen belyser at programmering er like mye en sosial aktivitet og at det benyttes som et språk til å kommunisere ideer.

Anette Seip

Anette Seip mener at programmering har en plass i grunnskolen og nevner spesielt Matematikk og Kunst og håndverk som relevante fag. Hun ser også på et tverrfaglig opplegg som gunstig, og sier at innføring av programmering kan føre til et bedre samarbeid mellom de ulike disiplinene på skolen, her vektlegger hun spesielt Matematikk og Kunst og håndverk. Seip mener at Matematikk er et tungt fag for mange elever, dermed vil det å plassere programmering i faget kunne gjøre det enda vanskeligere for noen elever om de skal lære tekstbasert programmering. Seip legger til at det er mange fag som kunne ha nytte av programmering, likevel mener hun også at programmering er et veldig praktisk fag, og dette kommer også til syne i hennes svar på intervjuet. Seip ser på programmering som et nytt språk hun må lære seg, men også at man kan bruke det til praktiske oppgaver og at elevene kan lære igjennom å gjøre morsomme ting som ikke involverer en bok. Seip anser også programmering som vanskeligere enn koding, da programmering inneholder en dypere forståelse enn koding. Algoritmisk tankegang synes hun hører i større grad til i Matematikk enn til Kunst og håndverk. Seip legger også til at hun ikke kan nok om begrepene til å kunne bedømme hvilket hun synes passer best i Kunst og håndverk.

Jon Øyvind Hoem

Hoem ser også på programmering i grunnskolen som noe positivt, og nevner at programmering har en spesiell rolle innen Kunst og håndverk og språkfagene. Hoem vektlegger at det han holder på med, ville han kalt for koding, og han synes koding er et bedre begrep i Kunst og håndverk enn programmering og algoritmisk tankegang. Hoem kategoriserer programmeringsspråk slik som Scratch, som ferdige rammeverk som er programmert av andre, og anser ikke koding innen dette språket som programmering. Hoem ser på programmering som en mer systematisk tilnærming der en lager mer komplette løsninger innen et programmeringsspråk. Han mener at begrepet algoritmisk tankegang ofte blir forstått feil, eller at begrepet får forskjellige innhold utfra hvem som snakker om begrepet. Han mener også at begrepet ofte blir knyttet mot det realfaglige. Likevel legger han til at algoritmisk tankegang har mange aspekter som passer Kunst og håndverk. Han nevner at algoritmisk tankegang er en del av den skapende prosessen, men at det er viktig å ikke glemme andre aspekter som er mer innovative.

«Kreativitet og algoritmisk tankegang er ikke motsetninger, men snarere ulike faser i en prosess. Får en for mye fokus på algoritmene (oppskriftene) kan det imidlertid gå på bekostning av andre tilnærminger som ligger nærmere de innovative egenskapene som vi ønsker at elevene skal utvikle gjennom undervisningen».

Hoem sier også at i kunst- og håndverksfaget må elevene lage algoritmer - ikke kun implementere dem - slik at prosessen blir mer utforskende.

Liv Klakegg Dahlin

Dahlin mener at programmering bør bli undersøkt og at det er viktig å komme med definisjoner på hva

algoritmisk tankegang, programmering og koding er i relasjon til Kunst og håndverk. Hun er usikker på hvilke begreper som eventuelt bør bli brukt i kunst- og håndverksfaget. Likevel kan det virke som om hun foretrekker algoritmisk tankegang med benevnningen:

«(...) særlig det utforskende, altså den algoritmiske tenkningen tror jeg er viktigere for [Kunst og håndverk] å koble oss på, enn selve programmeringen. Det er ikke bare tasting, men du kan gjøre det på mange måter. Tankesettet ligner på kunst- og designprosesser, som vi har kalt det som et kjerneelement. Det er jo litt den forståelsen vi jobber ut ifra nå, at det egentlig er det samme, men at vi bruker våre begreper på det».

I utarbeidelsen av kjerneelementene og læreplanmålene har fokuset heller ligget på de underordna begrepene til algoritmisk tankegang, da Dahlin mener at algoritmisk tankegang ikke nødvendigvis er et begrep som hører hjemme i faget. Likevel er hun mer opptatt av teknologi som helhet og ikke nødvendigvis programmering i Kunst og håndverk, og sier «(...) vi må snakke mye større om det, og at det er andre ting rundt teknologiutviklingen som kunst og håndverksfaget kan gripe». Hun mener likevel at teknologi er noe kunst og håndverksfagfeltet i større grad bør utforske og inkorporere i læreplanen. I læreplanarbeidet har det også vært mer fokus på hva man skal bruke det til, enn å bruke begreper som algoritmisk tenkning, programmering og koding. Grunnen er å vektlegge at det praktiske arbeidet er det mest sentrale i faget. Hun mener også at kanskje teknologi burde ha vært et eget fag, siden det blir viktigere og viktigere. Dahlin viser også til det tverrfaglige emnet teknologi og design som ikke hadde like god gjennomslagskraft i praksis, og er redd for at dette kan skje igjen om det ikke blir tatt gode nok grep.

Mulige positive årsaker

I denne delen ønsker jeg å redegjøre for de seks kategoriene som er beskrevet nedenfor, som jeg har hentet fra intervjuene. Utfra hva informantene legger i de ulike begrepene, programmering, koding og algoritmisk tankegang ser man at de blir brukt ulikt og vektlegges på forskjellige måter. Deres valg av begreper kan også påvirke hva de legger i de ulike svarene, og dette er noe som er viktig å tenke over når jeg fremlegger informantenes svar. Deres oppfatning av begrepene trenger ikke å være de som jeg har benyttet i denne oppgaven.

De mulig positive årsaker er kategorisert som følger:

- Aktualisering av Kunst og håndverk
- Bevisstgjøring
- Designprosess med teknologi
- Forståelse for det særegne ved verktøyet
- Tverrfaglighet
- Til fordel for programmering.

Aktualisering av Kunst og håndverk

Både Hoem, Seip og Dahlin kom inn på tema at programmering har en mulighet til å aktualisere faget kunst og håndverk i grunnskolen. I deres forklaringer og utsagn er det ulike ting som blir vektlagt når det kommer

til hvorfor det kan gjøre faget mer relevant både nå og i fremtiden. Hoem sier:

«Jeg tror et godt grep om koding vil kunne øke den opplevende relevansen av faget, ut over de som i dag er innsidere (de som har en forståelse for fagets kvaliteter). Samtidig tror jeg koding kan være en måte å utvikle faget i en retning som faget uansett må ta inn over seg, knyttet til at verktøy og grensesnitt mot fysiske objekter i økende grad blir styrt av programvare».

Slik som det kommer frem her er en av forklaringene på hvorfor, er at faget må følge den trenden som finnes i samfunnet der fysiske objekter i økende grad blir styrt av programvare. En annen ting Hoem poengterer er at utenforstående kanskje i større grad vil kunne se relevansen av faget. Hoem utdyper det siste poenget og sier:

«Jeg tror imidlertid at dersom koding og programmering og mer omfattende bruk av programvare ikke kommer inn i kunst og håndverk, så vil andre fag gripe dette, eller det kommer helt nye fag. Kunst og håndverk risikerer da å få en mer bevarende funksjon, som absolutt er viktig, men som jeg tror det vil bli stadig vanskeligere å vinne gehør for».

En av de tingene Dahlin er opptatt av er ikke nødvendigvis programmering eller koding i Kunst og håndverk, men teknologi. Om programmering og koding er en del av denne teknologien er ikke noe som kommer helt klart frem. Teknologi eksemplifiseres av Dahlin i forbindelse med klær. «Jeg tror – om ikke for lang tid vil klærne vår kunne ha elementer av teknologi i seg». Hun poengterer videre at den teknologiske utviklingen vil gjelde for flere typer produktet og at faget Kunst og håndverk bør ha en aktiv rolle i prosessen fremover med å bidra til kunnskapsgrunnlaget for den teknologiske utviklingen. Dahlin sier «Fryktelig mange produkter kommer til å ha det. Om ikke kunst og håndverksfaget er med i vurderingen, eller utformingen eller i kunnskapsgrunnlaget som leder opp mot produktene, så har vi egentlig ikke gjort jobben vår.». Dahlin sier at slik hun ser det står Kunst og håndverk ved et veiskille, der man på den ene siden kan ta innover seg digital teknologi i Kunst og håndverk eller faget kan jobbe for å være i fred. I følge Dahlin har kanskje faget jobbet i fred i for lang tid.

Både Hoem og Dahlin er inne på noe av det samme, og det er at faget må følge de tendensene som eksisterer i samfunnet. Her trekker begge inn at elevene må møte teknologi på skolen som de senere vil møte i arbeidslivet. De er også inne på at Kunst og håndverk kan ha et for stort fokus på tradisjon, der Hoem legger til at det dermed kan være vanskelig å vinne gehør for faget senere. Hoem og Seip nevner en annen dimensjon når det kommer til relevans og dette er å fange oppmerksomheten til flere elever og relatere undervisningen til elevenes hverdag. Seip sier:

«Du kan fange en hel elevmasse som kanskje enten kjeder seg i faget, eller føler at de ikke får nok utfordringer, eller ikke liker teknikkene som er tradisjonelle. [Elevene vil] føle seg mer møtt på sitt nivå, fordi vi jobber med noe som er knyttet til det de holder på med. Lager vi et dataspill, så vil de automatisk tenke 'Å! dette gjør jeg hjemme, læreren skjønner meg, hun er ikke bare en sånn gammel person'».

Hoem sier han ikke er så opptatt av hvilke materialer elevene skaper med, så lenge elevene får lage noe med utgangspunkt i egne ideer og behov. Det kan virke som om Hoem synes det er viktige å lære elevene prosesser og teknikker i hvordan å skape og er ikke like opptatt i hvilke materialer eller verktøy dette skjer i. Hoem mener at det kanskje er ønskelig å relatere verktøyene og materialene mer til elevenes hverdag. Her trekker han også frem at mye er i endring rundt elevene, og sier «barn og unge skaper med digitale verktøy og Kunst og håndverk må gripe dette». Et annet perspektiv som også kommer frem er at kunst og håndverksfaget må være med å møte utfordringene som kommer i fremtiden. Dahlin seier:

«Hvis vi skal nå det med innovasjon og nye ting - som vi skal kunne leve av i Norge etter oljen - bærekraftig utvikling også videre, så trenger vi at kunst og håndverksfaget kobles på teknologi».

Hun viser også til at det innenfor arbeidslivet allerede finnes etterspørsel om designere som kan programmere. Dahlin mener også at det kommer til å bli mer attraktivt i fremtiden.

I følge informantene sine svar kan man dermed si at programmering har mulighet til at flere får en forståelse av Kunst og håndverk, at en innføring av programmering kan fange oppmerksomheten til flere elever og i større grad relatere undervisningen til elevene sin hverdag. Siste punkt går på at fremtidens utfordringer også må bli møtt av kunst og håndverksfaget og at kunnskap innen teknologi vil bli mer attraktivt for fremtiden.

Bevisstgjøring

Denne kategorien er det i hovedsak Antonsen og Seip som snakker om, der de nevner ulike dimensjoner av kategoriene. Seip nevner at det å bevisstgjøre elevene på arbeide som ligger bak teknologi er viktig, men også at elevene blir bevisste på hvilke gjenstander som faktisk har programmerte deler i seg. Seip sier:

«(...) men jeg håper jo at det blir mer fokus på hva bruker du programmering til. Hva er det du har rundt deg i dag som er basert på programmering? For alt er jo, fra kjøleskapet ditt til nettsiden du bruker hver dag, til apper, til alt. [Elevene] tenker jo ikke på at kjøleskapet er programmert. Altså det er som et sjokk når jeg sier det. Hvordan er det dette termostaten fungerer da dere? 'Ah, oi' ikke sant? Og det er jo det de trenger å bli bevisstgjort på, alt som er rundt seg».

Seip er også opptatt av at elevene skal få en forståelse og bevissthet av at tradisjonelle teknikker fortsatt er viktig i dagens teknologisamfunn, og sier:

«Jeg vet det er veldig mange som sier at vi må fokusere på det tradisjonelle håndverket, men vi må også ta inn det som er nytt. Vi må kunne ha digitalsløyd og tradisjonell sløyd, og kunne sammenligne de to. Det er kjempe viktig. Kanskje bruke de sammen, og bevisstgjøre. Fortsette å jobbe fremover, skape nye ting. Det er ikke nok å bare tenke på det vi har gjort tidligere, og hvordan det fungerte. Det må inn i en større sammenheng. Og da tror jeg kanskje at, når man jobber på nye måter også, så vil man se at det tradisjonelle også har noe for seg. Vi begynte jo der, og at vi fortsatt holder på med en god del tradisjonelle teknikker, og kanskje trenger en god del tradisjonelle kunnskaper for å kunne jobbe nytt»

Antonsen legger mer vekt på underholdningssjangeren når han snakker om bevisstgjøring, der han nevner at de fleste filmer og det meste av musikk er laget av dataprogrammer. Der dataprogrammene har blitt programmert til visse formål. Antonsen viser også til at det kanskje er mulig å kunne se forskjellen mellom det som er laget av en maskin og det som er laget for hånd og sier:

«Hvis vi skal kalle det digital årvåkenhet, når du ser noe, så kan du si at 'okei! Det er kanskje noe en datamaskin har vært med på å lage'. Det synes jeg er veldig interessant, at når jeg ser rundt meg så kan jeg se ting som har blitt laget for hånd, som er laget av en maskin og vice versa».

Dette punktet handler dermed om at elevene igjennom å arbeide med programmering innen Kunst og håndverk vil kunne lære om arbeidet som ligger bak teknologien de bruker hver dag. For det andre vil de kunne bli mer bevisste på gjenstander innen både design, kunst og visuelle media som er laget ved hjelp av

programmering. For det tredje vil undervisningen kunne åpne opp for at elevene har en forståelse for hva som er laget av en datamaskin og hva som er laget for hånd, og dermed prosessen bak.

Designprosess med integrert teknologi

Programmering kan åpne opp for å skape gjenstander med integrert teknologi og at elevene får mulighet til å arbeide med oppgaver der de må tenke på form og funksjon. Når informantene snakker om funksjonen nevner de blant annet interaksjon, bevegelse, sensorer eller andre teknologiske funksjoner. Det er i hovedsak her fokuset til Dahlin, Hoem og Seip ligger når de snakker om hva elevene skal bruke programmering til i Kunst og håndverk. Selv om de nevner ting som kan lages innen visuelle representasjoner, snakker de i hovedsak om fysiske produkter med programmerte deler. Her vektlegges spesielt det med fysiske artefakter som det er mulig å interagere med eller fysiske gjenstander som har bevegelse. Når det er snakk om interaksjon så nevnes sensorer som et eksempel og både Dahlin og Seip knytter det mot smarte tekstiler. Seip viser også til at dette er noe elevene synes er veldig fasinende, og sier:

«Så har jeg veldig sansen for smarte tekstiler. Jeg har veldig lyst til å prøve for eksempel tekstiler som lyser når du går ut, (...) som er temperaturavhengig for eksempel, eller lysavhengig. Sann type sensor oppgaver er kjempe spennende. Ofte synes elevene det er helt fantastisk fasinende også».

Det informantene snakker mest om under denne kategorien er likevel det med bevegelse og dette utdyper Seip og Hoem. Hoem mener at koding i større grad kan lede elevene i retningen av en planmessig tilnærming til design, der han sier:

«Koding kan gi en inngang til utforming av fysiske objekter som i større grad/på nye måter leder elevene i retning av en planmessig tilnærming til design (forstått som en planmessig tilnærming til et kommende arbeid med fysiske materialer) og senere fysisk utforming».

Både Seip og Hoem er opptatt av at man skal lage de fysiske objektene selv. Seip sier blant annet:

«Det som er populært blant en del elever her, er å lage ting som beveger seg, ting som har en funksjon. Og da kan du jo med programmering få inn en liten dings, en robot, et eller annet bevegelig som kan settes inn i den praktiske tingen, som for eksempel sløyden, så har du jo plutselig digital sløyde. De praktiske oppgavene kan gjøres tekniske, rett og slett, og det er jo kjempespennende».

Her snakker Seip om at de ulike tingene elevene lager ved hjelp av programmering må ha en funksjon. Seip henviser også til at design feltet er viktig for forståelsen av produktutvikling og for at de praktiske produktene både skal ha en estetisk og funksjonell side ved seg.

I denne kategorien er informantene opptatte av at elevene skal lære en prosess, og ikke nødvendigvis håndverksteknikker og materialforståelse. Dahlin er opptatt av at elevene skal gå igjennom en prosess der de bruker teknologi. Hun er ikke så opptatt av hvilke produkter elevene skal lage, så lenge de utvikler ferdigheter og forståelse for produkt- og designutvikling. Som jeg beskrev i punktet med aktualisering, forklarer Hoem at han er mer opptatt av prosessene elevene skal igjennom, enn av hvilke materialer de bruker. Han påpeker likevel at man kan bruke materialer og verktøy fra elevenes hverdag for å henvise til elevens verden. Seip drar linjer mellom problemløsning, design og programmering. Hun mener at elevene får arbeidet med problemløsning når de driver med programmering. Problemløsning er også noe hun anser som svært viktig for Kunst og håndverk, og som hun arbeider mye med i undervisningen. Seip sier om programmering: «(...)

hele oppgaven er å finne ut av ting, tenke annerledes, prøve forskjellige løsninger, variere, prøve igjen når det går feil, spare på skisser og så starte helt på nytt og lage noe eget, også starte på nytt». Seip mener også at det å få forståelse for gjenstander kan øke om elevene får muligheten til å arbeide med praktiske oppgaver, der de må lage noe for hånd. Antonsen ser også relevansen når det kommer til fysiske objekter, men innen dette område sier han selv at han ikke er så kjent, så dette blir det ikke snakket så mye om.

I denne kategorien går fokuset på at elevene får arbeidet med designutvikling der teknologi i større grad kan lede dem i en planmessig tilnærming til design. Ved at elevene får arbeide med sensorer og bevegelse må de både tenke på funksjon og estetikk. Informantene er ikke så opptatte av hvilke materialer dette skjer i, men er mer opptatte av at elevene lærer ulike prosesser for å skape. De mener at programmering eller teknologi kan føre til et slikt fokus.

Forståelse for verktøyenes særegenhet.

Dette punktet ble i utgangspunktet presentert og presisert av Antonsen, der han vektlegger programmeringen sin særegenhet i Kunst og håndverk. Antonsen knytter det både opp mot at programmering kan være et særegent verktøy til å utforske mønstre og former, men også at det er et særegent verktøy til å uttrykke seg. Det siste punktet vil bli belyst i større grad nedenfor. Når Antonsen snakker om at programmering kan være et særegent verktøy til å utforske, snakker han om hastighet og antall operasjoner, samt det å parametrisere og finne konstanter. Han er veldig opptatt at hvert verktøy har sin særegne natur, og at dette er noe man bør lære elevene. Antonsen sier «Det å se på datamaskinen som noe helt nytt, eller helt spesielt, det er jeg ikke enig i. Det har sine særegne trekk, hastighet og antall operasjoner». Videre forklarer han hvordan man kan bruke de særegne trekkene til datamaskinen og knytter det spesielt opp mot mønstre.

«(...) Å få en forståelse av algoritmisk kunst som er omkring oss. Islamsk kunst og keltiske knuter har i seg algoritmiske trekk. Det er veldig gøy å lete etter regler. Strikkemønstre, heklemønstre og veving har også det tilsvarende ved seg, eller sør-amerikanske ponsjoer, har på en måte systemer i seg. Som er systemer du kunne hermet etter med et dataprogram(...). Vi har alle brukt mønstre opp igjennom historien og datamaskinen er et verktøy for å utforske mønstre. Så kommer det magiske, når du har gjenskapt mønstrene, så kan du parametrisere det og endre på det. Kanskje du vil at det skal være smalere, bredere, kanskje du vil ha mer av noe og mindre av noe annet».

Han legger også vekt på at det er når datamaskinen kan hjelpe oss til å gjøre noe mer, at det blir interessant å bruke en datamaskin og eventuelt programmering. Antonsen belyser også at om man kan gjøre det på en annen måte så er det ikke et poeng å bruke programmering. Det kan også virke som at parametrisering er et viktig begrep for han og at programmeringen er en god måte å gjøre dette på. Antonsen knytter også parametrisering og programmering opp mot former, der han snakker om hjerteformen og sier:

«Fordi kunsten er å finne ut, hva er det som sitter fast, hva er det som er den konstante verdien, og hvordan kan vi gjøre den om til en parameter som kan bevege seg (...). Fordi da finner jeg ut hva som er essensen med den, hvis du gjør den, hvis du gjør den (beveger hendene sine). Det er når man driver og parametriserer ting at man lærer om ting, tror jeg».

Antonsen påpeker også at man kan bruke utforskningen til å lage egne uttrykk og skape noe fra hva man finner ut. Antonsen sier:

«Så vi kan utforske verden og kunsten på en helt annen måte. Hvis du har laget noe, og du vet at her er

det noen parametere som styrer det, så kan du bruke de parameterne til å finne det riktige uttrykket»

Dette kan også bli forstått som at programmeringen kan åpne opp for en dypere læring av former og mønstre. Antonsen bruker matematiske begreper slik som parametrisere og konstanter. Dette er ikke vanlige ord som blir brukt i Kunst og håndverk, men likevel viser det til en dypere forståelse av hva former og mønstre er. Han mener også at programmering kan være et godt redskap til denne forståelse, og at elevene også kan få denne type forståelse. Slik som at programmering er et særegent verktøy til å utforske, snakker Antonsen også om at det er et særegent verktøy til å kunne uttrykke seg.

«(...) hva er det som er spesielt med strikking? Hva er det som er spesielt med tegning og maling? Hva er det som er spesielt med silketrykk eller andre ting? Og hva er det som er spesielt med datamaskinen i det å uttrykke seg? Det er som å høre på lyden av en saksofon, for da finner du ut at, 'å ja! Det høres kanskje bra ut med en saksofon, men kanskje ikke på gitar'. Da får man en følelse for verktøyet, for mediumet som man bruker. Og det er jo enklere i dag å skrive et lite dataprogram, enn å lære seg å strikke eller å spille saksofon».

Her blir litt av det samme som ovenfor diskutert, altså må man finne det særegne ved programmering når det kommer til å uttrykke seg. Antonsen belyser også at programmering er enklere å lære seg enn for eksempel det å strikke.

Antonsen legger også til en annen dimensjon ved det særegne for programmering og dette er generative mekanismer og prosedurale prosesser. Personen som programmerer kan skrive noen kommandoer som gjør at datamaskinen tegner mange punkter som tilslutt utgjør uttrykket. Hoem har mindre fokus på det særegne ved programmering, men vektlegger heller at det er et godt redskap til å lage audiovisuelle uttrykk. Audiovisuelle uttrykk slik jeg forstår det kan være ting som spill og animasjon. Både Hoem og Antonsen legger frem at om noe kan løses uten programmering, så finnes det ikke noe poeng i å bruke programmering. Antonsen sier «Hvis du gjør noe som du like gjerne kunne gjort med penn og papir, så hva var poenget?»

Fokuset i denne kategorien har vært at programmering kan være et særegent verktøy til å utforske og skape uttrykk i Kunst og håndverk. Her vektlegges at programmering spesielt er nyttig når det kommer til antall operasjoner og hastighet. Programmering kan også brukes til å skape generativ kunst, der elevene kan utforske prosedurale prosesser. En annen måte å bruke programmering på er å sammenkoble det visuelle og lyd, slik at elevene får arbeidet med audiovisuelle uttrykk.

Tverrfaglighet

I denne kategorien vil argumentet ligge i at programmering har muligheten for å åpne opp for en dialog og et tverrfaglig samarbeid med andre fagfelt. Antonsen mener at programmering kan bygge broer mellom ulike disipliner, og anser programmering til å være veldig tverrfaglig. Seip ser på tverrfaglig opplegg som gunstig og mener programmering kan tilrettelegge for slik opplegg. Seip legger spesielt vekt på et bedre samarbeid mellom Matematikk og Kunst og håndverk, men mener også at det er mulig å få til tverrfaglig undervisning med flere fagfelt. Dahlin er også opptatt av å styrke samarbeidet med Naturfag og Matematikk, og få til en kobling mellom fagene. Dahlin mener at teknologi kan være et slik felt og kanskje også programmering. Dahlin sier: «En positiv side av programmering i skolen i utgangspunktet er jo at det kanskje er veldig forskjellige fagfelt som plutselig er nødt til å jobbe sammen».

Dahlin, Hoem og Seip mener også at kunst og håndverksfaget ofte blir glemt, blir plassert utenfor diskusjonen

eller blir ansett som et «kosefag». Dahlin nevner at det er sjeldent at Kunst og håndverk blir tatt med inn i varmen når det kommer til diskusjoner av tverrfaglige emner slik som teknologi. Likevel legger hun til at Matematikk og Naturfag samtidig ser at det er viktig at Kunst og håndverk blir tatt med inn i en diskusjon angående teknologi. Det kan virke som om Dalin har en oppfattelse av at faget ofte blir glemt. Dahlin sier at de har arbeidet mye med å få inn begreper som problemløsning og utforskning i arbeidet med kjerneelementene. Likevel har hun lest kommentarer fra politisk hold om at fagene ikke ivaretar slike begreper og opplever at de glemmer Kunst og håndverk.

Hoem er også inne på noe av det samme som blir belyst i kategorien om aktualisering. Her nevnes blant annet at mange utenforstående ikke vet hva kunst og håndverksfaget innebærer. Informantene mener at hvis programmering blir en del av faget, vil kanskje flere fagfelt se en verdi i å arbeide med Kunst og håndverk. Dahlin sier: «(...) så er det de som ser hvordan det kan gi Kunst og håndverk en bedre plassering i skolens hverdag, som et fag som samarbeider med flere fag. Altså å få Kunst og håndverk ut av en skygge». Seip er også inne på litt av det samme der hun sier at Kunst og håndverk ofte blir sett på som et «kosefag», og at kanskje programmering kan bidra til at andre anser at faget har litt mer tyngde. Seip sier også at innenfor læreryrket finnes det mange tidstyver og at lærerne ofte har mye arbeide, og at det dermed kan være vanskelig å gjennomføre samarbeid. Hun mener at hvis læreplanen er bygget opp på en måte som krever samarbeid så kan det bli lagt mer vekt på tverrfaglighet, og dette kan programmering bidra med.

Under dette punktet snakker alle informantene om at programmering er veldig tverrfaglig. Hoem, Seip og Dahlin mener også at programmeringens tverrfaglighet kan bidra til et bedre samarbeid med flere fagfelt eller få flere fagfelt i skolen til å arbeide sammen.

Til fordel for programmering

I denne kategorien presenteres meninger som tilsier at programmering i Kunst og håndverk kanskje kan være en fordel for fremtiden til programmering. Dette var ikke noe som ble presisert i stor grad. Antonsen sier:

«At mennesker som kanskje ikke hadde gjort det, gjør det mer og bruker det på en litt annerledes måte. Jeg tror det er sunt og bra. Jeg tror generelt sett, at et mangfold av perspektiver er sunt og bra. Hvis noe kan bidra til at man ser ting på en annen måte så tror jeg det er veldig sunt. Jeg tror kanskje også at det tvinger en bedre brukervennlighet på plass, fordi når flere mennesker bruker et verktøy, så finner man ut at det kanskje er bedre måter å organisere det på (...). Jeg tror jo flere som bruker det, og flere som bruker det kreativt, jo bedre blir det tilrettelagt for det. Det kan man se på hvordan digitale verktøy har utviklet seg de siste ti, femten årene»

Det Antonsen snakker om her, er for det første at om programmering blir lært under flere fagfelt, så kan verktøyet utvikle seg til å passe ulike disipliner. For det andre kan også programmeringsverktøyene få en bedre brukervennlighet for ulike personer om det er mange ulike fagdisipliner som bruker programmering. Et tredje perspektiv Antonsen nevner er at det kan inspirere til å bruke datamaskinen til andre ting, ikke kun til å regne eller gjøre mekanisk anvendte ting. Her legger han til at kanskje flere vil finne ut at man kan bruke datamaskinen til å uttrykke seg.

Seip snakker i hovedsak om at ved å ha programmering i Kunst og håndverk så kan det bidra til at flere jenter velger realfag senere, der hun sier «Det at man får inn et sånt realfagsfokus, så kanskje en klarer å fange flere jenter, for eksempel. Jeg tror det er veien å gå»

Under denne kategorien ligger fokuset på at programmering kan utvikle seg til å passe ulike disipliner, men også flere forskjellige mennesker med ulike bakgrunn. Programmeringen kan også bli mer brukervennlig hvis det eksisterer flere ulike brukere. Her blir det også nevnt at kanskje flere jenter vil kunne velge realfag senere, og kanskje flere som vil velge å bruke datamaskinen på nye måter om det blir plassert i Kunst og håndverk.

Mulige begrensende årsaker

Mulige begrensende årsaker har blitt kategorisert som følger:

- Et fagfelt med et annet fokus og manglende kompetanse
- Miljø
- Økonomi
- Noe annet må ut

Et fagfelt med manglende fokus og kompetanse

I intervjuet med Dahlin kommer det klart fram at hun anser det problematisk at fagfeltet har lite fokus på tematikken. Dahlin sier at det er et politisk styrt ønske om at programmering skal bli en del av flere fag i grunnskolen. I arbeidet med kjerneelementene opplevde hun at hovedfokuset til fagfeltet på Kunst og håndverk er at faget skal ha et større fokus og ivareta det praktiske arbeidet. Dette ønsket kommer i kollisjon med at programmering ofte blir ansett som teoretisk der man sitter mye foran en skjerm. Antonsen er redd for at noe slikt kan skje. Han er bekymret for at programmering kun blir tekniske øvelser der man skal følge regler og ikke blir en praktisk prosess med skapelse som formål.

I intervjuet med Dahlin kommer det også frem at begrepet algoritmisk tenkning og programmering kanskje kan være et fremmedord for mange kunst- og håndverkslærere. Følgelig har faggruppen for kjerneelementene lagt mer fokus på hva man skal bruke det til, enn å bruke begrepene. Dahlin sier også at å implementere nye og fremmede begreper i læreplanen ikke nødvendigvis løser problematikken og at å bruke begreper som folk ikke er vant til innen fagfeltet kan være problematisk. Dahlin nevner de to kompetansemålene som læreplangruppen plasserte i forslaget på kompetansemål i Kunst og håndverk, der den ene gikk på digitalt materiale og den andre på å programmere en mønsterrapport. Hun opplevde at tilbakemeldingene i hovedsak var negative, der de enten ikke ønsket kompetansemålene, eller at de ikke forsto hva begrepene inneholdt. Dahlin mener at dette kommer av for liten kompetanse når det kommer til programmering og algoritmisk tankegang knyttet opp mot Kunst og håndverk. Personer innen faget ser heller ikke koblingen mellom det digitale og det praktiske. Dahlin nevner fagkunnskap koblet sammen med programmering som noe vanskelig og sier:

«(...) Og det er det som jeg tenker mangler her. Det er diskusjonen mellom de som har den spisskompetansen og de som jobber i feltet og som ikke er negative. Som ikke skal overbevises eller noe sånt, men som sier 'ja okei, kanskje jeg ser at dette blir viktig, men jeg vet ikke hvordan'».

Seip er også inne på problematikken om at det er for liten kunnskap. Hun trekker dette opp mot både tilpasset opplæring og å kunne legge opp undervisningsopplegg som fungerer. Seip viser til at det eksisterer en holdning der mange kun ser for seg utfordringer ved å implementere programmering. Hun mener dermed

at det kan bli vanskelig at programmering blir en del av Kunst og håndverk om feltet kun ser problemer og ikke mulige løsninger. Seip mener at det finnes utfordringer med alle undervisningsopplegg og som lærer mener hun at det er viktig å sette seg inn i hvordan man løser det. En annen faktor hun påpeker er at hun tror mange lærere er redde for å undervise i noe de ikke kan selv og dermed se dum ut foran elevene. Hun mener det finnes mye frykt i at lærerne ikke har nok kunnskap.

Dahlin er også redd for at om programmering blir en del av Kunst og håndverk så kan man miste viktige fagpersoner. De som har drevet med faget i lang tid har ofte en faglig tyngde som ikke nødvendigvis henger sammen med digital teknologi. Likeså er det et fåtall av de som er utdannet innen Kunst og håndverk som har kompetanse eller kjennskap til programmering. Lærere kan dermed miste motivasjonen for sitt eget fag, om de ikke føler de mestrer det. Dahlin mener at i politiske sammenhenger blir Kunst og håndverk gjerne glemt med tanke på programmering. Hun er redd for at hvis man kjører i gang en lærerplan der programmering er en del av faget og man ikke har en forståelse for at kunst- og håndverkslærere også skal være der, kan det bli et stort problem. Da vil man få fagpersoner som ikke har kunnskap innen programmering som skal undervise i det.

Denne kategorien handler i hovedsak om at det er manglende fokus på programmering når det kommer til fagfeltet innen Kunst og håndverk. Det handler også om at fagfeltet har liten kunnskap om problematikken og at dette kan ha negative konsekvenser for de som arbeidet med faget, men også for selve faget.

Miljø

En annen negativ konsekvens av å implementere programmering inn i Kunst og håndverk som blir belyst av Dahlin er at det kan føre til mye elektrisk avfall. Innen Kunst og håndverk er det en vanlig praksis at elevene får ta med det de lager hjem. Hvis dette blir tilfelle kan det ende med mye elektriskavfall. Dahlin utdyper dette med:

«Eller at man tror at ved at man kjører inn teknologi at elevene får lage produkter som de skal ha med seg hjem, som er en tenkning som ligger i Kunst og håndverk. Da kommer vi til å spy ut billige sensorer, som de programmerer og som blir liggende (...). Det blir elektrisk avfall»

Dahlin sier at dette strider med prinsippene i bærekraftig utvikling og den miljøprofilen som faget ønsker å ha. Det kan her tenkes at Dahlin snakker om fysiske gjenstander som er programmert og ikke om visuelle og audiovisuelle uttrykk som blir programmert på dataen. Dahlin kommer selv med en løsning på problemet og det er å ha klassesett med elektronikk, og at elevene lager integrerte prosjektet der man kan prøve det elektroniske for så at læreren samler det inn igjen. Seip anser BBC micro:biten til å være gjenbrukbar og derfor også kanskje miljøvennlig. Da dette er en liten skjerm man kan bruke flere ganger, ved at man laster opp programmene, for så å viske det ut igjen.

Økonomi

En annen problemstilling er økonomi. For å få til en programmeringspraksis kreves det utstyr og kompetanse, og dette koster. Både Seip og Dahlin nevner dette som mulig utfordring i å implementere programmering i Kunst og håndverk. Seip sier:

«Så har du eventuelt elever som synes det er vanskelig å tenke logisk, det kan også bli tungt (...). Men jeg tenker det er litt opp til læreren å legge til rette for. Og det er ikke umulig å få det til. Man må bare vite hva slags ressurser man skal bruke, så alle må jo på kurs, hvis det skal inn i skolen. Og det er jo kanskje en større utfordring, sånn budsjettmessig».

Her blir det økonomiske knyttet opp mot kostnadene forbundet med kompetansebygging for at lærere skal få nok kunnskap til å skape gode undervisningsopplegg. Seip nevner også at noen oppgaver innen programmering kan bli vanskelige å gjennomføre i store grupper da elevene trenger mer individuell veiledning. Seip underviser programmering i forskjellige grupper der det er forskjell på antall elever. Hun sier at grafiske oppgaver knyttet til 3D-modellering ofte krever mer veiledning og følgelig kan det være vanskelig å gjøre slike oppgaver i store grupper. Elevene kan også slite med den logiske tenkningen noe som igjen også vil føre til mere veiledning. Dette kan ha en indirekte konsekvens på det økonomiske, der mer veiledning vil kreve mindre klasser.

Noe annet må ut

Under denne kategorien fremlegges en annen begrensende faktor for at programmering skal bli en del av Kunst og håndverk, og dette er at det skaper mindre plass for noe annet. Dahlin belyser dette ved å problematisere dagens elever sine praktiske ferdigheter. Hun sier at barn og unge sine praktiske ferdigheter er nesten på et kritisk nivå, altså at de mangler finmotorikken. Hun er redd for at hvis programmeringen kun blir en teoretisk øvelse der elevene må sitte og taste på et tastatur, så kan de gå glipp av nødvendige praktiske ferdigheter. Hun mener også at ferdigheter innen finmotorikk ikke er en ferdighet elevene lærer i så stor grad hjemme, og dermed er det viktig at grunnskolen legger til rette for denne utviklingen. Seip er også inne på det samme, at kunst og håndverksfaget har mye pensum som elevene skal igjennom. Seip sier: «Det er ikke ideelt at det kommer inn, da må noe annet ut (...). Jeg har ikke så mye tid å gi egentlig». Hun sier videre at det er mål i kunnskapsløftet som kan være utfordrende å gjennomføre i undervisningen slik som filmmaging. Dette er på grunn av manglende utstyr og utfordringer rundt å bruke elevens telefoner til filming. Her legger hun til at elevene kanskje kan få mer bruk for programmering i fremtiden og at programmering kan bli brukt i en mer utstrakt grad. Hoem er også enig at det kan være et problem at noe må ut for at programmering skal få plass. Han sier at diskusjonen rundt dette kan bli vanskelig og splittende. Antonsen mener at kvalitet og kvantitet ofte blir blandet for mye sammen når det kommer til tema om programmering skal bli en del av grunnskolen. Antonsen mener at problemet ligger helt andre steder enn tidsperspektivet.

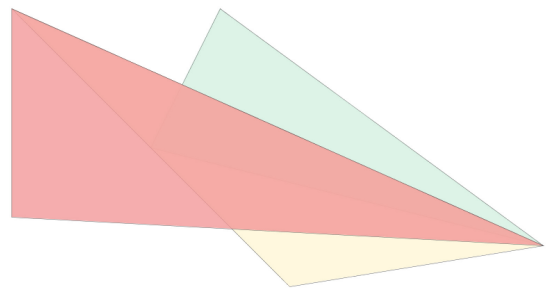
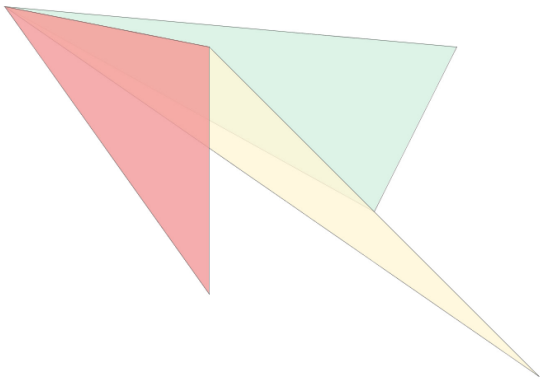
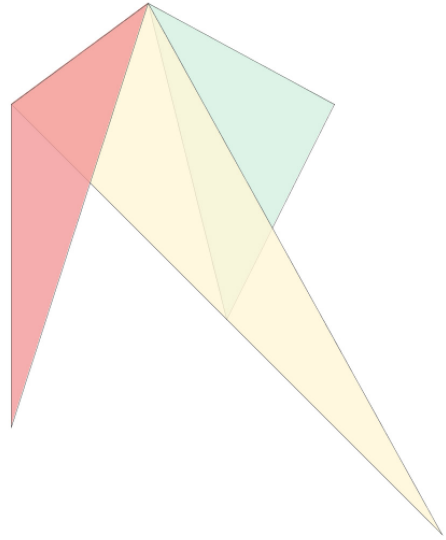
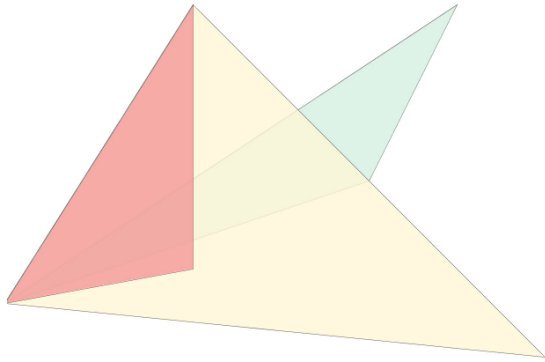
Oppsummering

I fremlegget av intervjuene har jeg delt det informantene har sagt i to hovedkategorier, der den ene handler om mulige positive årsaker og den andre om mulige begrensende årsaker til hvorfor programmering skal bli en del av Kunst og håndverk. Under de positive årsakene har jeg valgt å plassere kategoriene, *aktualisering av Kunst og håndverk*, *bevisstgjøring*, *designprosess med integrert teknologi*, *forståelse for verktøyenes særegenhet*, *tverrfaglighet* og *til fordel for programmering*.

Under *aktualisering av Kunst og håndverk* går fokuset på at man kan fange oppmerksomheten til flere elever om programmering blir en del av faget. Faget kan også i større grad relatere seg til elevenes hverdag. Flere som ikke kjenner til fagfeltet kan også få en forståelse av verdien til Kunst og håndverk. Det vektlegges at fremtidens utfordringer må bli møtt av kunst- og håndverksfaget, og at teknologi innen design vil bli mer

attraktivt i fremtiden. Under kategorien *bevisstgjøring* blir det fremlagt at elevene kan bli mer bevisste på gjenstander som er laget ved hjelp av programmering og arbeidet som ligger bak. Elevene kan også bli mer bevisste på hva som er laget med programmering og hva som er laget for hånd. I kategorien *designprosess med integrert teknologi*, er det viktig for informantene at elevene går igjennom ulike prosesser der formålet er å skape og at programmering kan føre til et større fokus på prosesser som innovasjon, problemløsning og design- og produktutvikling. I kategorien *forståelse for verktøyenes særegenhet* diskuteres det særegne med programmering som er både antall operasjoner og hastighet, i både det å uttrykke seg, men også for å utforske former og mønstre. Her blir det også lagt frem at programmering egner seg svært godt til å lage audiovisuelle uttrykk. I kategorien *tverrfaglighet* ligger argumentet for at programmering har et potensial til å være tverrfaglig, og som kan føre til et bedre samarbeid mellom ulike fag i grunnskolen. I kategorien *til fordel for programmering* omhandler at programmering i Kunst og håndverk kan gjøre at programmering kan tilpasses flere ulike disipliner og at kanskje flere jenter vil fortsette med det. Det kan også føre til at det blir en bedre brukervennlighet og at programmering kan brukes på nye måter.

Under mulige begrensende årsaker har jeg valgt å plassere kategoriene, *et fagfelt med manglende fokus og kompetanse, miljø, økonomi og noe annet må ut*. I kategorien *et fagfelt med manglende fokus og kompetanse* bli det lagt frem argumenter om at fagfeltet heller har hatt fokus på tradisjon og håndverksferdigheter, enn teknologi og programmering. Under manglende kompetanse diskuteres at det eksisterer lite kunnskap om programmering knyttet til Kunst og håndverk, og at mange lærere kan miste motivasjon og mestringsfølelse om programmering blir innført. I kategorien *miljø* fremlegges argumenter som går på at programmering kan føre til mye elektronisk avfall, om det fortsatt skal eksistere en praksis der elevene får ta med produktene hjem. I kategorien *økonomi* blir det fremlagt at alle lærere i Kunst og håndverk må få kurs, og at undervisning i programmering kanskje krever mindre klasser. Til slutt legges det frem at *noe annet må ut* som problematisk, da faget inneholder mange forskjellige ferdigheter og kompetanser og at en slik diskusjon kan bli splittende.



```
void setup () {
  fullScreen ();
}

void draw () {
  background (255);
  strokeWeight (0.5);
  stroke (90);

  fill (210,238,224,200);
  beginShape ();
  vertex (600,300);
  vertex (500,500);
  vertex (mouseX,mouseY);
  vertex (600,300);
  endShape ();

  fill (254,245,214,200);
  beginShape ();
  vertex (300,300);
  vertex (700,700);
  vertex (mouseX,mouseY);
  vertex (300,300);
  endShape ();

  fill (240,150,152,200);
  beginShape ();
  vertex (300,300);
  vertex (300, 600);
  vertex (mouseX, mouseY);
  vertex (300,300);
  endShape();
}
```

Bilde 13:
Utforskning av former (2018) Eget bilde

Drøfting

Når jeg i problemstillingen ønsker å undersøke hvorfor programmering bør bli en del av Kunst og håndverk, opererer jeg i et fremtidsperspektiv, altså blir det viktig å se på hvordan fremtidige strukturer kan og bør bli, som jeg har knyttet til det kritisk realisme kalles for strukturell utvikling. For å kunne si noe om fremtidige strukturer, er det også viktig å se på eksisterende strukturer, altså det som innfor kritisk realisme blir kalt strukturelle betingelser (Buche-Hansen & Nielsen, 2005). Ut fra de kvalitative metodene som er brukt i oppgaven kan det være vanskelig å generalisere og si noe om fremtiden, likevel er ønsket å kunne komme med mulige hypoteser eller bør-perspektiver, slik at det kan føre til videre diskusjoner. Slik jeg har tolket kritisk realisme og Næss (2012) sin begrunnelse, kan bruken av kritisk realisme også kunne forutsi noe om fremtiden, fordi det eksisterer regelmessigheter selv i en svært kompleks kontekst, slik som skolen. Jeg ønsker først å diskutere eksisterende perspektiver som kan ha konsekvenser for fremtidige perspektiver, for så å legge frem fremtidsperspektiver. Ønsket er ikke å argumentere for at programmering skal bli implementert fremfor andre emner i Kunst og håndverk, og det er heller ikke ønskelig å argumentere for at programmering skal inn i Kunst og håndverk istedenfor andre fag. Jeg vil heller ikke forsøke å argumentere for at programmering skal tilhøre et spesifikt fag, om det skal være tverrfaglig, eller om det skal være et eget fag. Med utgangspunkt i problemstillingen er ønsket å besvare om det finnes noen årsaksforklaringer for at programmering bør eller ikke bør inngå i Kunst og håndverk.

Strukturelle betingelser

Strukturelle betingelser blir beskrevet som eksisterende strukturer som påvirker aktøren. De eksisterende strukturene er skapt av aktørene i samfunnet. Aktørene blir også påvirket av de gjeldende strukturene, enten man er bevisst på det eller ikke. I denne oppgaven har mulige strukturer som fagfeltets holdninger, politiske styringer og rammefaktorer blitt nevnt, og jeg vil nedenfor drøfte funnene av disse strukturelle betingelsene i lys av relevant teori.

Fagfeltets holdninger

Noe som kommer tydelig frem i min undersøkelse er at det eksisterer et manglende fokus hos fagfeltet når det kommer til programmering i Kunst og håndverk. Dette kommer både frem hos informantene, men også fra relevant litteratur. Det er spesielt to faktorer som har kommet frem som kan være årsaker til hvorfor ikke fagfeltet er opptatt av programmering. Den ene er at de anser programmering som noe teoretisk og at dette strider mot grunnleggende verdier om at faget skal utvikle ferdigheter og kompetanse i håndlag. Det andre er at programmering går imot holdninger om at elevene skal lære tradisjonelle teknikker og arbeide med tradisjonelle materialer. Et annet funn er at Hoem, Seip og Dahlin er opptatt av at elevene skal lære prosesser, og ikke nødvendigvis er så opptatt av hvilke materialer dette skjer i.

Programmering blir for teoretisk

I arbeidet med den nye nasjonale læreplanen i Kunst og håndverk har den rådene innstillingen vært at læreplanen må legge opp til at elevene må arbeide praktisk og at elevene skal tilbake til verkstedene (Kjerneelementgruppa for kunst og haandtverk, 2017a). Dette påpekes også av Dahlin i intervjuet. I undersøkelser som har gått på digitale verktøy i Kunst og håndverk, har et funn vært at lærerne heller fokuserer på praktiske oppgaver, der elevene får arbeide med finmotorikk og håndverksteknikker (Olsen, 2014 Strand, 2017;). Selv om programmering ikke er nevnt under digitale verktøy i mastergradsavhandlingene til Strand (2017) og Olsen (2014), er det mulig at lærerne prosjekterer lignende holdninger når det kommer til programmering.

Slik det kommer frem i mastergradsavhandlingen til Strand (2017) er bruken av fagspesifikk digitale verktøy ikke så utbredt i Kunst og håndverk. Dette kommer også frem i ICILS undersøkelsen fra 2013, der halvparten av elevene i 9. klasse sier at de aldri bruker datamaskinen i estetiske fagene (Ottestad, et al., 2014). Det kan derfor argumenteres mot at det skal innføres enda flere nye digitale verktøy, når de eksisterende verktøyene ikke blir tatt i bruk. Det kan også argumenteres for at fokuset heller bør ligge på å gi lærere kompetanse og utarbeide et repertoar innenfor de gjeldene fagspesifikke digitale verktøy i faget og at fokuset bør gjenspeile LK06 der det digitale spesielt blir lagt under visuell kommunikasjon. Likevel kan en implementering av programmering fungere som en katalysator for et større fokus på digitale ferdigheter. En av styrkene til programmeringen kan være som Seip sier, at praktiske oppgaver kan få en teknologisk vri, da vil tankene om at faget skal være taktilt fortsatt bli ivaretatt. Det kan tenkes at dette er forsøkt med kjerneelementene i Kunst og håndverk. Her har det digitale også fått et fokus under håndverksferdigheter og ikke kun under visuell kommunikasjon (Kunnskapsdepartementet, 2018a). Gjennom utforskningen i den praktisk-estetiske undersøkelsen, har jeg funnet to ulike hovedtilnærminger til arbeidet med programmering, nemlig utforming av digitale eller fysiske produkter. I arbeid med digitale produkter på en skjerm, i ulike digitale media, har mye av prosessen vært å sitte foran en dataskjerm, der tasting har vært det taktile. Derimot har jeg sett at arbeid med maskinvare kan føre til et større fokus på taktilt arbeid. Dette kan kanskje være en av grunnene til at Dahlin, Hoem og Seip fokuserer på fysiske gjenstander med integrert teknologi, som kan være basert på programmering. En mulig diskusjon er om fokuset i Kunst og håndverk bør ligge på å programmere fysiske gjenstander slik at det ivaretar det håndverksmessige i faget. Å bruke programmering på denne måten vil kanskje kunne sikre en mer taktil tilnærming der man fokuserer mer på håndverksferdigheter, enn om man programmerer innenfor skjermens rammer.

Fokus på tradisjon

Det som også kommer frem i min undersøkelse er at programmering kan aktualisere Kunst og håndverk. Dahlin og Hoem har begge en oppfattelse om at Kunst og håndverk ofte kan ha en bevarende funksjon, der det tradisjonelle blir mer ivaretatt. I Strand (2017) sin avhandling virker dette som en av de viktigste årsaksforklaringene til hvorfor lærerne ikke gjennomfører undervisning i fagspesifikke digitale verktøy. Strand finner ut i sin avhandling at lærerne ikke er negative til IKT, men at de verdsetter teknikker og materialer som er tradisjonelle. Mange av lærerne mener at motoriske egenskaper og taktile erfaringer er noe av de viktigste egenskapene elevene skal lære i Kunst og håndverk. Om tradisjon skal være en gjeldende verdi for faget, kan det bli vanskelig å innføre nye emner i Kunst og håndverk. Som informantene er inne på, må også fremtidens utfordringer bli møtt av Kunst og håndverk og faget må være med i kunnskapsgrunnlaget for teknologisk utvikling.

Seip, Hoem og Dahlin virker også mer opptatt av at elevene skal lære seg prosesser og ikke nødvendigvis tradisjonelt håndverk i tradisjonelle materialer. De samme posisjoneringene kan vi også finne igjen i avhandlingen til Olsen (2014), der Olsen velger å dele lærerens posisjoneringer innen digitale verktøy i to ulike retninger, *det tradisjonelle* og *det nye*. Posisjoneringen av *det nye* illustreres med nøkkelordene som problemløsning, oppgaveløsning, kreativitet, produksjon og tankesett. Området Olsen har kalt for *det nye* kan etter mitt syn også bli kalt for *prosessorientert posisjonering*. Med utgangspunkt i mine informanter og Olsens mastergradsavhandling kan det virke som om de som har fokus på digitale materialer og verktøy, også har et større fokus på at elevene skal lære seg prosesser innen kunst og design, og ikke nødvendigvis tradisjonelle teknikker og materialer. Kunst- og designprosesser så vel som håndverksferdigheter har blitt satt til å være to av fire kjerneelementer i den nye fagfornyelsen (Kunnskapsdepartementet, 2018a). Det praktiske arbeidet må dermed gjenspeile begge områdene i kjerneelementene, og Seip er inne på at det er en verdi i å sammenligne digital sløyd og tradisjonell sløyd. Hun sier videre at for å arbeide nytt, trenger man også kunnskap om det tradisjonelle. Dette er noe jeg gjenkjenner fra den praktisk-estetiske undersøkelsen. I arbeid med Scratch ønsket jeg å lage en spirit, som lignet på meg selv. For å få til dette, bygget jeg på kunnskaper innen portrettmaling og -tegning. I arbeidet med Processing bygget jeg videre på prinsipper innen det formalestetiske, for å oppnå et uttrykk jeg var fornøyd med. Et slikt fokus finner man også igjen i undersøkelsen til Greenberg et al. (2012), der de bruker formalestetiske prinsipper for målet til programmeringen i undervisningen. En mulig negativ konsekvens som Dahlin poengterer er at lærere kan miste motivasjonen og interessen for sitt eget fag dersom programmering bli innført. Om lærerne ikke føler mestring for sitt eget arbeid kan jobben bli vanskelig å gjennomføre. Dette kan spesielt tenkes om det ikke blir tilrettelagt for etterutdanning, som vil bli diskutert nedenfor.

Politiske styringer

Dahlin sier i sitt intervju at innføringen av programmering i flere fag er et politisk styrt ønske, og at dette ikke nødvendigvis henger sammen med fagfeltet sine ønsker. Jeg har skrevet en del om de politiske rammene om programmering i teorikapittelet. Under ønsker jeg likevel å drøfte dette, da jeg mener at de politiske styringene både viser til et udefinert område som kan gjenspeiles i det informantene sier, og at det er et ønske om en utvidet definisjon av hva digital kompetanse innebærer fra et politisk perspektiv.

Et udefinert område

Det som spesielt har kommet frem fra relevant teori er at programmering er et udefinert område, dette kommer også frem fra politiske dokumenter angående tematikken. Det kan virke som om programmering i første omgang hadde et realfaglig fokus. Programmering tilknyttet Matematikk og Naturfag ble blant annet uthevet i fagvurderingen *Teknologi og programmering for alle* (2016) Sanne et al. og *Fremtid, fornyelse og digitalisering - Digitaliseringstrategi for grunnsopplæringen 2017-2021* (2017) Kunnskapsdepartementet. Programmering, koding og algoritmisk tankegang ble også plassert under Matematikk i de to første utkastene til kjerneelementene og ikke i Kunst og håndverk (Udir, 2017b, 2017c, 2017d, 2017e). I det tredje og siste utkastet av kjerneelementene kom også programmering inn under fagene Kunst og håndverk og Musikk, der Samfunnsfag skal ha hovedansvaret for digital kompetanse (Kunnskapsdepartementet, 2018; Udir, 2018a). Videre kan forskning som har vært gjort på koding, programmering og algoritmisk tankegang i den norske grunnskolen vise til et manglende fokus på Kunst og håndverk, da mastergradavhandlingene om temaet har vært skrevet under realfag og IKT (Haaland & Rosvold, 2006; Iversen, 2015; Jåtten, 2006; Lang-Ree, 2016; Verstad, 2017).

I pressemeldingen *Mer koding og teknologi inn i skolen* (Statsministerenskontor, 2017) blir det nevnt at man må vurdere hvordan programmering skal inngå i bestemte fag. En slik vurdering har jeg enda ikke klart å finne. I ulike land har det blitt lagt ulik vekt på hvilke fag som skal ha algoritmisk tankegang og programmering. I Finland har opplæringen av programmering blitt plassert i Matematikk for så at det brukes i teknisksløyd (Porko-Hudd, 2017; Kwon & Schroderus, 2017). Er en mulig ordning kan være å plassere opplæringen i programmering i andre fag slik som Matematikk, for så at programmering brukes i Kunst og håndverk i de praktiske prosjektene til elevene. Utfra informantenes posisjonering kan vi også se at algoritmisk tankegang, koding og programmering blir vektlagt forskjellig. Hoem påpeker at det algoritmiske ofte blir knyttet til realfag, og at det dermed kan være vanskelig å bruke begrepet i Kunst og håndverk. Informantene har også en ulik vektlegging på om programmering skal være tverrfaglig, inngå i ulike fag eller være et eget fag. Siden det har blitt utforsket så lite om programmering i Kunst og håndverk, kan en stille spørsmål ved om faget er modent for et nytt emne. Som Dahlin fremlegger i sitt intervju mangler det en diskusjon om programmering, koding og algoritmisk tankegang i Kunst og håndverk. Som jeg vil komme inn på etterhvert, kan en manglende norm gjør det vanskelig for lærere å gjennomføre og tilrettelegge for en undervisning som inkorporerer programmering. Det kan også gjøre det vanskelig når det kommer til vurderingspraksisene i faget.

En utvidet definisjon

Skal vi følge funnene og rådene til Digitutvalget (2013) og Sanne et al. (2016) om at elever i større grad må være med på å skape teknologi enn å skape med teknologi, så er dette også noe som må gjenspeiles i Kunst og håndverk. Hvis den norske lærerplanen etterhvert vil følge kompetanseområdene til DigComp sitt rammeverk, der problemløsning og programmering også er en del av den digitale kompetansen, vil dette føre til en utvidet definisjon og oppfattelse av hva digitale ferdigheter er innenfor hvert fag. I det reviderte rammeverket for grunnleggende ferdigheter i LK06, kan punktet *produsere og bearbeide* åpne opp for at man skal bruke programmering (Udir, 2016). Selv om målene åpner opp for at man kan bruke programmering, er ikke dette nødvendigvis noe som skjer. Ifølge Liv Merete Nielsen (2009) inspirert og basert på John I. Goodlad (1979) så kan man analysere en læreplan på fem forskjellige nivåer. Disse er *ideologisk nivå*, *vedtatt læreplan*, *tolkningsnivå*, *gjennomført nivå* og *erfart nivå*. En læreplan kan åpne opp for at man skal bruke programmering, som er vedtatt læreplan. Likevel trenger det ikke føre til at lærerne tolker målene på en slik måte og det trenger heller ikke å bli gjennomført i selve undervisningen jf. gjennomført og erfart nivå. Strand (2017) skriver at lærerne ser at det er viktig at elevene lærer om det digitale og at det er nedfelt i LK06, men likevel føler lærerne at de ikke har tid til å gjennomføre det digitale. På *Kunst og håndverkskonferansen 2017* viser Waterhouse (2017) til at det finnes mange kompetansemål som kan bli gjort digitalt eller åpner opp for bruken av programmering i den gjeldende læreplanen LK06, men slik som drøftet ovenfor, er ikke dette noe som alltid blir gjennomført i undervisningen.

I fremlegget av den nye fagfornyelsen ser man at det digitale blir vektlagt på andre måter, enn i den gjeldende læreplanen LK06. I kjerneelementene i Kunst og håndverk, har det digitale både blitt plassert under visuell kommunikasjon og håndverksferdigheter (Kunnskapsdepartementet, 2018a). I det andre utkastet av læreplanen står det under digitale ferdigheter i faget at elevene skal «beherske og forme egne digitale produkter» (Udir, 2019). Mange av kompetansemålene åpner igjen opp for at man kan bruke det digitale og programmering. Det er likevel ikke noe som blir presisert utenom kompetansemålet etter 7. trinn der et av kompetansemålene er «gjenkjenne og programmere mønstersekvenser til visuelle uttrykk» (Udir, 2019). Uten en programmeringsnorm i faget og med et lite utforsket felt kan en videreføring av holdninger som eksisterer i faget gjøre det vanskelig for at programmering blir gjennomført. Selv om det finnes en ideologi om

at det digitale skal ha mer plass i skolen, kan målene tolkes bredt og følgelig kan det digitale bli nedprioritert i tolkningsnivået og gjennomført nivå.

Det finnes mye innen det digitale og teknologiske domenet som enda ikke har vært utforsket innen Kunst og håndverk. Kanskje er det slik at det er andre aspekter ved det teknologiske og digitale som passer bedre inn i Kunst og håndverk enn programmering? Det er viktig at det blir gjort en undersøkelse på hvilke muligheter som ligger innen fagfeltet, og at det blir gjort en vurdering av hva som passer og bør få plass. Som Dahlin er inne på, er det ikke sikkert at programmering er det Kunst og håndverk skal fokusere på innen teknologi.

Rammefaktorer

Det er det ikke nødvendigvis lærerne og fagfeltet som har skyldene for at det har vært lite fokus på digital teknologi og programmering. Ser man på mastergradsavhandlingen til Strand (2017) velger hun å ta frem tre rammefaktorer som kan ha påvirkning på hvorfor det blir brukt lite fagspesifikke digitale verktøy i Kunst og håndverk. Hun nevner tilgang og organisering av utstyr, lærerens digitale kompetanse, og tid til kompetanseheving og undervisning som mulige årsaker. Når det kommer til det tverrfaglig emnet Teknologi og design er det også blitt tillagt argumenter rundt rammefaktorer for hvorfor lærere ikke har prioritert dette emnet. I arbeidet med mastergradsavhandlingen fra naturfagdidaktikk på NTNU til student Alexander A. Dundas (2011) har han intervjuet seks lærere i Naturfag og en i Kunst og håndverk, for å undersøke lærerens implementering av teknologi i undervisningen. I avhandlingen nevner han at lærerne har fått lite kurs i teknologi, og at mye ligger på lærerens faglige bakgrunn og interesse for om emnet blir gjennomført. Dundas nevner også andre rammefaktorer som tid, ledelse og føringer og fysiske rammer og ressurser. Selv om mastergradsavhandlingene til Strand (2017) og Dundas (2011) viser til rammefaktorer for om teknologi og design eller fagspesifikk digitale verktøy blir gjennomført i Kunst og håndverk, kan det tenkes at dette er rammefaktorer som kan påvirke om programmering bør bli en del av faget, da programmering kan komme inn under begge begreper. I min undersøkelse er tid og økonomi blitt nevnt som mulige rammefaktorer for hvorfor det kan være vanskelig at programmering skal bli en del av faget. Jeg vil redegjøre for disse rammefaktorene nedenfor.

Tid

Som nevnt over er tid en faktor som ifølge både Dundas (2011) og Strand (2017) påvirker om teknologi eller det digitale blir vektlagt i undervisningen. Dundas (2011) sier at det å sette seg inn i relevant fagstoff, planlegging av undervisning og gjennomføring av undervisning tar tid i Teknologi og Design. I fagrapporten av Kunst og håndverk, basert på Skolefagsundersøkelsen i 2009, skriver Kjetil Sømoe (2010) at hvis det ikke eksisterer en felles norm for IKT i undervisningen, så vil dette kreve mye av den enkelte lærer. Selv om dette er en eldre kilde, er det fortsatt noe jeg tror er gjeldene i dag. Som diskutert ovenfor er programmering knyttet til Kunst og håndverk fortsatt et lite definert område. Sømoe skriver videre at dette blant annet forutsetter at lærerne må ha oversikt over ulike programmer, oversikt over læringsprosessene elevene går igjennom, veksle mellom teknikkopplæring og fagspesifikk fagkunnskap og muligheten for individuell veiledning. I arbeid med den praktisk-estetiske undersøkelsen med programmering har det stadig dukket opp nye programmeringsplattformer eller kommet videreutvikling av programmeringsspråkene. Scratch har fått en oppgradering og p5.js bygger videre på prinsippene til Processing, men støtter seg heller på JavaScript, enn Java, som igjen kan ha konsekvenser på undervisningsform. Innen p5.js skriver man koden i et utviklingsmiljø som eksisterer i nettleseren, istedenfor å arbeide på et nedlastet utviklingsmiljø. Ved å ha et

utviklingsmiljø i nettleseren, kan dette for eksempel bidra til å gjøre det enklere å dele koden med en klasse. Som jeg har skrevet før finnes det over tusen programmeringsspråk, og det er tidskrevende å sette seg inn i alt dette. Tidsaspektet kan derfor ha konsekvenser på mange ulike faktorer. Jeg velger videre å se på tid til kompetanseheving og planlegging av undervisning, og tid til gjennomføring av programmering.

Tid til kompetanseheving og planlegging av undervisning

Strand (2017) skriver «Når de fleste lærere ikke får mulighet til kompetanseheving gjennom eksterne kurs, blir det opp til dem selv å lære seg å bruke verktøyene» (Strand, 2017, s. 74). Egeberg et al. (2016) viser også til at lærere i hovedsak lærer om IKT ved hjelp av prøving og feiling og veiledning fra kollegaer. Sømoe (2010) nevner tidsressurser som en begrensende faktor til kompetanseheving men også til undervisningen i IKT-verktøyer. Innenfor den gitte tidsrammen til faget fokuserer lærerne i Kunst og håndverk heller på tradisjonelle teknikker. Lærerne som underviser i Kunst og håndverk har høy leseplikt (undervisningstid) på ungdomskolen i motsetning til andre fag (SFS 2213, 2015, s. 11). Dette betyr blant annet at en lærer i Kunst og håndverk i gjennomsnittet må undervise 24,9 skoletimer, à 45 minutters enheter, per uke, og en norsklærer underviser i 21,2 skoletimer per uke (Lie & Nielsen, 2016). En slik ordning gir mindre tid til for- og etterarbeid for en kunst og håndverkslærer, og en kan stille spørsmål om dette er tilstrekkelig med tid for at en lærer skal kunne utvikle nye ferdigheter og kompetanse i et så komplekst emne som programmering, spesielt hvis det ikke blir tilrettelagt for kurs og etterutdanning som undersøkelser viser (Egeberg et al.; 2016, Strand, 2017). Som tidligere påpekt, er både teknologi og programmering felt som forandrer seg raskt. I en hektisk hverdag for lærere kan det dermed bli vanskelig å følge utviklingen om det ikke blir gitt rom for dette. Dette er også vanskelig når det ikke finnes en klar definisjon på programmering, og det ikke eksisterer en gjeldene norm for programmering knyttet til Kunst og håndverk. Erfaringer fra den praktiske-estetiske undersøkelsen viser at tekstbasert programmering kan være vanskelig å lære seg grundig på grunn av innlæring av syntakser. En mulig løsning kan være å lære seg grafisk programmering, selv om dette også kan være utfordrende. Nedenfor vil jeg gå mer inn på økonomiske styringer som også kan ha konsekvenser for denne rammefaktoren.

Duncan, Bell & Tanimoto (2014) skriver at en mulig negativ konsekvens av at elever lærer programmering av lærere som mangler kompetanse og fortrolighet innen programmering, kan føre til at elevene selv får en negativ oppfatning av emnet. De viser også til at dette spesielt kan gjelde for jenter da deres selvtillit innen programmering på barneskolen i større grad vil følge dem i senere utdannings- og yrkesvalg. Varvara Garneli (2014) skriver at mange lærere mener at det er liten interesse for informatikk blant studenter. Hun sier videre at man må ta i betraktning hvordan undervisningen blir lagt opp. Studentene synes det er kjedelig å kun arbeide med tasting og regneark. En slik pedantisk og mekanisk anvendt tilnærming til å lære programmering er også noe Antonsen er bekymret over. Sennett (2008) skriver «It might seem that the more people train and practice in developing a skill, the more practical minded they will become, focusing on the possible and the particular» (Sennett, 2008, s. 46). Som Sennett er inne på her, må man kunne en ferdighet godt nok for også å kunne tilegne det til hva som er mulig. Om man ikke har nok kunnskap, kjennskap eller ferdigheter innen et område kan det være vanskelig å tilrettelegge for riktig bruk. Som lærer er man forpliktet til tilpasset opplæring til den enkelte elev (Opplæringsloven, 1998, § 1-3). Et lavt kunnskapsnivå kan også føre til at det blir vanskelig for en lærer i Kunst og håndverk å tilrettelegge for riktig undervisningsform av programmering, noe Seip også er inne på.

Andre faktorer som kommer frem i Strand sin avhandling viser at læreren har liten tid til å gjennomføre undervisning i det digitale om de skal igjennom alle målene i kunnskapsløftet i Kunst og håndverk, dette kommer også frem i intervjuet med Seip. Seip mener at hun allerede har dårlig tid i faget, så om programmering skal inn, må noe annet ut. Kunst og håndverk skal romme mye, og har mye bredde i ulike kompetanser og ferdigheter (Udir, 2006). En av kommentarene til kjerneelementgruppen i Kunst og håndverk på udirbloggen.no har vært at «Det som er den største utfordringen vår, er å finne hva som skal kuttes eller nedprioriteres» (Kjerneelementgruppen for kunst og haandtverk, 2017b). Antonsen fremlegger likevel at programmering kan ta mindre tid å lære eller gjennomføre, enn andre områder i Kunst og håndverk innen de tradisjonelle materialer. Antonsen nevner at det å lære seg å strikke kan være mer tidskrevende enn å lære seg å programmere.

Likevel er det andre perspektiver når det kommer til tid, som ikke er nevnt av informantene, men som kommer frem i litteraturen som kan ha konsekvenser for om programmering skal inn eller ikke. Kunst og håndverk er et relativt stort fag på barneskolen, som med 477 timer er større enn Naturfag med 366 timer. På ungdomskolen er faget noe mindre med 146 timer og blir dermed mindre enn Naturfag med 249 timer (Udir, 2018i). Dette kan også være en viktig betraktning å ta med videre i diskusjonen om programmeringens plass.

Økonomi

I empirien kommer det fram at økonomi knyttet til utstyr og kompetanse kan være en begrensende faktor for om programmering skal inn i Kunst og håndverk, men også antall elever per lærer. Det kan også være andre faktorer under det økonomiske som ikke vil bli belyst, selv om de eksisterer.

Økonomi til utstyr

Regjeringen bevilget 450 millioner kroner til en 5-årig satsning innen teknologi, som de kaller *Den teknologiske skolesekken*. Av de 450 millionene skal 15 millioner settes av til to av åtte tiltak som styrker programmering i grunnskolen. De to tiltakene er å gi tilskudd til vitensentrene for å styrke arbeid med programmering og til skoleeiere for utstyr til programmering. De seks andre tilskuddene går mer på digitale læremidler (Kunnskapsdepartementet, 2018b). Som relevant litteratur viser, har hovedfokuset vært på realfagsdidaktikk og som diskutert over finnes det liten kompetanse og kunnskap når det kommer til programmering i Kunst og håndverk. Dette kan påvirke skoleeier sine valg når det kommer til innkjøp av utstyr til programmering. Tabell 2 viser områder innen Kunst og håndverk der programmering kan bli benyttet. Utstyr til dette kan være noe annerledes enn utstyr som kjøpes til for eksempel programmering i Matematikk. I Dundas (2011) sin avhandling nevnes også mangel på utstyr som en begrensende faktor for utføring av emnet Teknologi og design. I Strand sin avhandling kommer det også frem at innkjøp og organisering av utstyr på skole- eller kommunenivå har vært en begrensende faktor for om fagspesifikke digitale verktøy blir benyttet.

Programmeringsplattformene som har vært benyttet i den praktisk-estetiske undersøkelsen er alle gratis. Likevel krever BBC micro:bit og Arduino, maskinvare som ledlys, ledninger eller servomotorer som kan bli dyrt. Som tidligere nevnt, skal 100.000 elever i 9 klasse få en BBC micro:bit, og dette kan lette på de økonomiske utfordringene (Teknisk Museum, u.å.b). Likevel kreves det andre maskinvarer for å utføre produktutvikling av fysiske gjenstander, og dette må det bli tilrettelagt for om programmering skal bli en

del av det fysiske domenet i Kunst og håndverk. En manglende kunnskap om programmering i Kunst og håndverk kan også føre til uheldige fremtidskonsekvenser. Dahlin påpeker at det er en eksisterende praksis i faget at elevene får ta med seg hjem de produktene de lager. Dahlin sier videre at en slik praksis kan føre til mye elektronisk avfall. Dette går imot den kommende læreplanen sitt fokus på bærekraft, der ett av tre tverrfaglige temaer er bærekraftig utvikling (Udir, 2018g). Slike problemstillinger er man derfor nødt til å diskutere innenfor fagets premisser. Skal en vanlig praksis være at elevene skal skape innenfor kreativ koding og programvare av fysiske produkter, krever dette at man har oppgaver der maskinvaren kan bli tatt ut av de fysiske rammene. Skal elevene lage fysiske produkter som de senere må demontere? Eller skal fokuset heller ligge på å lage digitale produkter slik at man ivaretar miljømessige problemstillinger?

Økonomi til kurs

Når det kommer til kursing av lærere i programmering har det blitt opprettet nettbasert kurs i programmering som senter for IKT i utdanningen står for (Udir, 2016a). Som belyst ovenfor kan tid likevel være en begrensende faktor for lærerens mulighet til å utvikle kompetanse og kunnskap innen programmering. Det blir derfor viktig at det blir avsatt midler for dette slik at lærere kan følge kursene. Det er flere universiteter som tilbyr kurs for lærere om programmering i skolen. Eksempler er Høgskolen i Østfold som tilbyr studie i *Programmering i skolen 1.-10. trinn*, og NTNU tilbyr *Programmering 8.-13. trinn* (Høgskolen i Østfold, u.å; NTNU, u.å). Disse kursene omhandler i hovedsak grunnleggende kunnskap i programmering på grunnskolen og videregående. De er ikke tilknyttet et spesifikt fag, og følgelig blir det opp til lærerne om hvordan de skal knytte programmering til de spesifikke fagene, og uten en norm i Kunst og håndverk, kan dette bli utfordrende. Om programmering skal inngå i Matematikk, Naturfag, Kunst og håndverk, Musikk og Samfunnsfag vil dette kreve en enorm kompetanseheving, som vil koste. Det bør nevnes at ressursportaler som IKT i praksis, har opplegg i programmering i Kunst og håndverk, som læreren kan støtte seg på (IKT i praksis, 2018). Dette er opplegg i BBC micro:bit, Scratch, Bee-bot og Kodu. Lær kidsa koding har også undervisningsopplegg lærere kan bruke og tilbyr også lærerveiledning og workshops i ulike programmeringsspråk (Bocconi et al., 2018).

Økonomi til mindre klasser

Seip nevner også at programmering kan være vanskelig å forstå for noen, og at det kreves logisk tenkning. Hun sier også at å ha undervisning innen visuelle uttrykk krever mer veiledning av henne som lærer. Hun mener at det kanskje er formålstjenlig med mindre klasser. Undersøkelser viser også at å lære programmering innenfor et kreativt rammeverk, der elevene skal lage noe, er hensiktsmessig (Adams, 2007; Giannakos et al., 2014; Knobelsdorf & Romeike, 2008; Lau et al., 2009). Gjennomføring av praktisk orienterte og skapende elevarbeid krever det ofte mindre klasser for at læreren skal kunne hjelpe alle elevene. Anne Bamford (2012) undersøkte på oppdrag av *Nasjonalt senter for kunst, kultur og kreativitet* situasjonen for kunst og kultur i opplæringen i Norge i 2010-2011. I Bamford sin undersøkelse kommer hun frem til at med større klasser, fokuserer lærerne i større grad på teoretiske måter å lære bort et tema. Gruppestørrelse i undervisning i Teknologi og design har også vært en begrensende faktor, sett fra lærerens perspektiv (Dundas, 2011). I mastergradsavhandlingen til June Oline Kunnikoff (2015) viser hun til at lærerne mener at store elevgrupper og fjerning av delingstimer i Kunst og håndverk er en utfordring, spesielt knyttet til praktiske oppgaver. Programmering kan lett bli en mekanisk øvelse, der det blir fokus på det tekniske. Om programmering skal bli brukt i et skapende rammeverk vil dette kanskje kreve færre elever per lærer. Mindre elevgrupper per lærer passer til fagets innhold og undervisningens behov, men vil også gi økonomiske konsekvenser.

Strukturell utvikling

Som skrevet over finnes det mange faktorer en må ta i betraktning når man skal innføre et nytt emne i skolen, og ovenfor er kun noen av årsakene blitt belyst. Strukturelle betingelser omhandler gjeldende strukturer som kan ha innvirkning på om programmering bør bli en del av Kunst og håndverk i grunnskolen eller ikke. I denne delen ønsker jeg å se på mulige fremtidige scenarioer, som i kritisk realisme blir kalt for strukturell utvikling. Sagt med andre ord: eksisterer det noen argumenter som kan diskuteres og som eventuelt kan føre til at strukturen - kunst og håndverk på grunnskolen - kan utvikle seg.

For å sortere argumentene har jeg valgt å ta utgangspunkt i Britt Ulstrup Engelsen (2003) sin teori for utvikling av innholdet i en læreplan. Hun opererer med utgangspunkt i to kriterier; *indre* og *ytre*. *Indre kriterier* handler om at elevene skal lære det sentrale som ligger i faget, altså her står fagets egenart i sentrum. *Ytre kriterier* vil si «De skal ta hensyn til elevenes situasjon og behov og til det samfunnet – ut fra sine interesser – finner viktig å ta med av faginnhold» (Engelsen, 2003, s. 217). I ytre kriteriene står samfunnsproblemer, samfunnsnytt og elevinteresse sentralt (Engelsen, 2003). I opplæringsloven (1998) § 1-1 *Formålet med opplæringen*, kan ideologien bak ytre kriterier skimtes. Det ene går på utvikling av ferdigheter, kompetanse og kunnskap for at elevene skal mestre sine egne liv. Det andre går ut på at elevene skal kunne delta i arbeid og et fellesskap i samfunnet.

Indre kriterier

Dybdelæring ved bruk av programmering

Et av de mest sentrale begrepene i fagfornyelsen har vært *dybdelæring* (Ryssevik, 2017). I dette ligger det at elevene skal forstå og kunne bruke kunnskapen de har lært i nye situasjoner. Det elevene skal lære må settes inn i en relevant og forståelig sammenheng. Dette betyr at fagstoffet må knyttes opp mot fagets kjerneelementer og kobles til tidligere kunnskaper og erfaringer hos elevene. I fagfornyelsen blir det også siktet til at elevene må lære noe grundig og ikke kun overfladisk (Gilje, Landfald & Ludvigsen, 2018; NOU2015:8). Når Antonsen snakker om det særegne ved programmering i Kunst og håndverk, nevner han antall operasjoner og hastighet. Han sier videre at dette kan brukes til utforskning og skapelse av mønstre, former, generative og prosedurale prosesser. I forskningsintervjuet med Antonsen blir to ulike retninger adressert, det ene er å få en matematisk forståelse av mønstre og former, og det andre er å bruke denne kunnskapen til å skape uttrykk. Denne måten for læring passer til beskrivelsen av dybdelæring. Elevene får en dypere forståelse for mønstre og former, og bruker kunnskapen til å lage former og mønstre. Former og mønstre er noe som er en sentral del av kunst og håndverksfaget. Form blir blant annet nevnt under hovedområdet visuell kommunikasjon i LK06, der det står «Form, farge og komposisjon samt ideutvikling, problemløsning og symbolbehandling er sentrale emner» (Udir, 2006). Antonsens vektlegging på mønstre kan også gjenspeiles i den første skissen for den nye læreplanen. Her er et av kompetansemålene som har vært foreslått etter 7. trinn «Analysere ulike mønstre og programmere en mønsterrapport som lages i myke materialer» (Udir, 2018f). I den andre skissen for den nye læreplanen er dette blitt omgjort til «gjenkjenne og programmere mønstresekvenser til visuelle uttrykk» (Udir, 2019).

Former og mønstre kan gjerne anses som et område innen Matematikk og Kunst og håndverk, bare med forskjellig vektlegging. Andre viktig begreper i den nye fagfornyelsen som er tett forbundet med dybdelæring er tverrfaglig, flerfaglig og tverrfaglige tema (Ryssevik, 2017). På udirbloggen.no i blogginnlegget *Verden*

er ikke delt opp i fag, legger Dag Johannes Sunde (2018) frem argumentet, «Skal elevene lære i dybden, er det viktig å kunne se sammenhenger, kunne overføre forståelse og innsikt i og mellom fag» (Sunde, 2018). Som informantene har lagt frem, har programmering evne til å være et tverrfaglig emne. Dette kan også sees i sammenheng med begrepet STEAM, skapeverkstedet og det tverrfaglige tema teknologi og design. Antonsen bruker matematiske begreper som parametrisere og konstanter når han snakker om mønstre og former, som kan virke ukjent for faget kunst og håndverk. Slik som Dahlin er inne på kan det være vanskelig å innføre fagbegreper som ikke allerede er en del av faget. Men, om en i den fremtidige skolen skal vektlegge dybdelæring og tverrfaglighet, kan programmering være et godt redskap til å lære om mønstre og former. Dette er også noe som kan sees i tabell 2, som viser en oversikt over hvordan programmering kan bli brukt i Kunst og håndverk.

I den praktisk-estetiske undersøkelsen oppdaget jeg også at programmering innen Processing åpnet opp for en større forståelse og anvendelse av blanding av lysfarger. I nåværende læreplanen LK06, er et av kompetansemålene etter 7. trinn «skille mellom blanding av pigmentfarger og lysfarger». Programmering kan brukes som et verktøy til å utforske lysfarger, og slik å få en dypere forståelse av begrepet lysfarger og dermed også kunne skille det fra pigmentfarger. Mønster, former og farger er eksempler på tverrfaglige temaer. Både innen Naturfag og Kunst og håndverk skal elevene lære om farger, men vektlegges ulikt i fagene. I mastergradsavhandlingen til Peter Haakonsen (2012) *Additiv fargeblanding i GIMP* er det mulig å lese mer om samarbeidet mellom fysikkfaget og Kunst og håndverk når det kommer til fargeundervisning.

Det algoritmiske i Kunst og håndverk

Hoem, Antonsen og Dahlin belyser det algoritmiske og den algoritmisk tenkning som allerede ligger i faget kunst og håndverk. De knytter disse begrepene opp mot historiske og eksisterende kunstpraksiser og skapelsesprosesser. Som nevnt tidligere viser Antonsen til mønstre og former, men han viser også blant annet til det algoritmiske i historiske kunstverk som keltiske knuter og islamsk kunst. Både Antonsen og Dahlin snakker om det algoritmiske i veving og strikking, og dette blir også belyst av Sevik (2016) som viser til strikke- og veve-maskiner der mønstrene programmeres.

Hoem og Antonsen snakker også om det algoritmiske når det kommer til å skape. Antonsen snakker om strikking som et eksempel på en algoritmisk handling, der du for eksempel har fem rette og fem vrangle. Hoem viser til den algoritmiske tenkningen når det kommer til designprosessen. Han mener at kreativitet og algoritmisk tankegang ikke er motsetninger, men ulike faser i en prosess. Seip er opptatt av at elevene skal bli gode til problemløsning, og mener dette både er en viktig ferdighet i programmering, men også i kunst og håndverksfaget. Grover og Pea (2013) nevner ulike aspekter som kommer under algoritmisk tankegang. Her nevner de blant annet *abstraksjon og mønster gjenkjenning, systematisk prosessering av informasjon, strukturert problemdekomponering, iterativ, rekursiv og parallell tenkning og betinget logikk*. Mange av de nevnte ferdighetene er også ønskelig i design. Systematisk prosessering av informasjon trenger man for eksempel når man skal finne ut av hva som er behovet til brukeren. Problemdekomponering kan man trenge når man skal finne ut av hvordan man skal skape et designprodukt, og for å løse problemer som oppstår underveis i designprosessen. Iterativ tenkning blir også ofte koblet sammen med design (Waterhouse, 2017). Knochel og Patton (2015) skriver: «We, the authors, believe code block environments are excellent platforms for introducing young art students to learning iterative design habits and algorithmic problem solving associated with computational thinking» (Knochel & Patton, 2015, s. 26).

I kjerneelementene for Kunst og håndverk blir kunst- og designprosesser beskrevet som både en utforskende

og en stegvis prosess (Kunnskapsdepartementet, 2018a). Som algoritmisk tankegang blir beskrevet i denne oppgaven, handler det om muligheten til «å løse et problem gjennom å spesifisere en presis sekvens». Med utgangspunkt i denne vide forklaringen av begrepet, er det dermed mulig å både dra linjer til deler av for eksempel designprosessen og problemløsning. Knochel og Patton (2015) belyser egenskaper som er innebygget i kunstfeltet, slik som å være komfortabel med å arbeide i et komplekst domene, ha pågangsmot, ferdigheter i arbeide med problemer uten et fasitsvar og åpenhet for tvetydighet. De sier at dette også er viktige egenskaper i algoritmisk tankegang. Dermed kan den algoritmiske tankegangen bygge på kvaliteter i faget, men den algoritmiske tankegangen kan også tilby faget ferdigheter. Som belyst av informantene kan programmering åpne opp for en dialog og en dypere forståelse av algoritmiske prinsipper i Kunst og håndverk.

Kjennskap til ulike materialer

Kunst og håndverk er også er fag som vektlegger kjennskap, kunnskap og ferdigheter i ulike materialer og verktøy. Dette kan sees i det ene kjerneelementet, håndverksferdigheter. Her skal elevene utvikle «praktiske ferdigheter og utholdenhet ved å bruke ulike redskaper og materialer» (Kunnskapsdepartementet, 2018a). Som Antonsen er inne på kan programmering bli sett på som et medium til å skape. I delkapittelet om *programmering i kunst og håndverk* beskrives synet om at programmering kan bli forstått som et materiale, der formålet er å skape noe uttrykksfullt istedenfor funksjonelt (Artut, 2017; Knochel & Patton; Maeda, 2000; 2015; Reas & Fry, 2006;). Som Antonsen belyser er det viktig at elevene får kjennskap til det særegne ved hvert materiale. Det at elevene utvikler kjennskap, kunnskap og ferdigheter innen ulike materialer, kan hjelpe elevene til å ta beslutninger om hvilke materialer det er mest hensiktsmessig å bruke til ulike oppgaver. I intervjuene vektlegger Antonsen skapelse av uttrykk som krever mange operasjoner og hastighet, men også skapelse av uttrykk med innebygde strukturer, slik som mønstre. Hoem sikter heller til at programmering er godt egnet til å skape audiovisuelle uttrykk. I kjerneelementet håndverksferdigheter i faget kunst og håndverk, er det ikke definert hvilke materialer elevene skal arbeide med. Det blir nevnt at elevene skal arbeide med harde, myke og plastiske materialer, samt digitale verktøy (Kunnskapsdepartementet, 2018a). En tanke som informantene er inne på, spesielt Hoem og Seip, er å gi elevene læring innen håndverk fra materialer som er relevant, tilgjengelig og viktig for elevenes liv. I denne sammenheng nevner informantene at programmering og teknologi kan være eksempler på slike materialer. Knochel og Patton (2015) belyser også dette å sier « (...) we believe students would derive deeper understanding across disciplines and mediums by learning to code as a part of their artistic practice» (Knochel & Patton, 2015, s. 26). I en verden som i stadig større grad inkorporerer teknologi og er styrt av algoritmer, må elevene også få kjennskap til dette i faget. Det Hoem og Antonsen henviser til, er det jeg har kalt for kreativ koding og programvare av digitale produkter i tabell 2. Dette er noe forskjellig vektlegging av materialkjennskap enn det som diskuteres under neste delkapittel.

Et komplekst designbegrep med interaksjon

Som nevnt før, er det også ønskelig at elevene skal lære innenfor et skapende og kreativt rammeverk. Kunst og håndverksfaget har mulighet til å tilby et skapende rammeverk, med lærere som er kompetente i å undervise i skapende prosesser. Det er likevel viktig å stille spørsmål om det er programmering elevene skal lære i undervisningen, eller om det er andre læringsutbytter i det skapende prosessarbeidet som er viktig. Et viktig argument som har vært brukt i debattene rundt programmering og realfag i grunnskolen, er at programmeringen skal bli lært på fagets premisser og tilføre fagene noe.

Hoem, Seip og Dahlin belyser at programmering kan åpne opp for en mer kompleks designundervisning, der teknologi er integrert. I tabell 2 blir en slik kompleks designundervisning knyttet til programvare av fysiske produkter. Ved å inkorporere maskinvare i designprodukter kan det føre til en mer sammensatt tilnærming til designprosessen. Sevaldson (2011) påpeker at en mer kompleks verden også vil kreve en mer kompleks tilnærming til design og henviser til GIGA-mapping og systemtenkning. Som belyst i delkapittelet om *research by design*, sier Sevaldson (2010) at design er et bredt område som er interdisiplinært og som ofte inkorporerer ulike teknologier. Sanne et al. (2016) påpeker også at teknologi er tett forbundet med design. Som informantene belyser er det viktig med designforståelse, estetisk sans, og produktutvikling når programmering og teknologi er integrerte deler av et fysisk produkt. Det jeg mener programmeringen kan tilby er et større fokus på teknologisk interaksjon. Som jeg har skrevet om i den praktisk-estetiske undersøkelsen er interaksjon et viktig moment i programmeringen. Programmering åpner opp for at man kan bruke sensorer som fanger opp interaksjon slik som posisjon, orientering, bevegelse, berøring, lys og lyd. Dette fører videre til at man må reflektere over hvordan brukeren eller deltageren samhandler med produktet eller over kunsten man skal lage. Dette er allerede en viktig del av design, men med teknologi kan en likevel argumentere for at det åpner opp for et nytt aspekt, der algoritmene er med på å bestemme interaksjonen. Programmering kan også bidra til et økt fokus på interaksjon når det kommer til kunst. En viktig faktor for den moderne tiden og kunstpraksisen er å se på møtet mellom et kunstverk og en betrakter som en deltakelse. Teori rundt estetikken omhandler blant annet at man skal bli aktive betraktere og medskaper, men også at kunstverket skal føre til aktiv og fysisk deltakelse (Bishop, 2006). Ut fra mine refleksjoner kan programmering og teknologi være et redskap til å få til dette i kunst- og designundervisningen. Som vil bli drøftet nedenfor, under ytre kriterier, så vil elevene møte en mer kompleks verden, de må derfor få kjennskap til å arbeide med et komplekst designbegrep der teknologisk interaksjon er en del av design- eller kunstbegrepet. Som belyst over finnes det mange likheter med algoritmisk tankegang og Kunst og håndverk. De to domeneene kan være med på å videreutvikle hverandre og støtte hverandre.

Ytre kriterier

Et kunst- og håndverksfag som relaterer seg til elevenes hverdag

Som Hoem og Seip er inne på er det mulig at programmering kan fange oppmerksomheten til flere elever. Seip mener det kan fange oppmerksomheten til elever som kanskje kjeder seg i faget, ikke får nok utfordringer eller ikke liker tradisjonelle teknikker. Hoem mener også at programmering kan fange oppmerksomheten til flere elever i faget om det blir brukt materialer og verktøy som elevene møter i hverdagen. Knochel og Patton (2015), Artut (2017) og Peppler og Kafai (2009) legger frem argumentet om at kunstundervisningen må følge tendensene som eksisterer i samfunnet, altså må elevene kunne skape med kode i kunst og design. Selv om ICILS 2013-rapporten er litt eldre, viser den at norske elever i hovedsak bruker datamaskinen til kommunikasjon eller underholdning utenfor skolesammenheng (Ottestad et al., 2014). Selv om elevene kanskje ikke skaper med programmering, så er de fleste elever daglige forbrukere og konsumenter av gjenstander som er skapt ved hjelp av programmering, som kommunikasjonsprogramvarer og underholdningsplattformer. Som Digitutvalget (2013) og Sanne et al. (2016) påpeker, må elevene i større grad lære seg ferdigheter innen å skape teknologi og ikke kun skape med teknologi. For å få til en slik praksis, er man nødt til å gi elevene redskapene og verktøyene for å skape teknologi, og programmering kan være et eksempel på dette. Som jeg viser i tabell 2 kan dette være ulike teknologiske produkter, som kreativ koding eller programvare av fysiske gjenstander eller digitale produkter.

Sanne et al. (2016) skriver også at det må være en bredde i teknologifaget for å kunne nå ulike elevgrupper. Som belyst i relevant teori er et gjentakende problem i den vestlige verden at jenter ikke velger studier eller yrker innen informatikk (Corneliussen & Tveranger, 2018; EU-Kommisjonen, 2018; Udir, 2018d). Som Seip påpeker kan programmering innen Kunst og håndverk kanskje åpne opp for at flere jenter blir interessert i programmering, som fører til at jentene velger informatikk videre. Lau et al. (2009) gjorde et forsøk ved å lære studenter om programmering gjennom smarte tekstiler. Forsøket viste at både jenter og gutter ble motiverte til å lære videre om teknologi og programmering. Greenberg et al. (2012) fant også ut at et kreativt rammeverk, der man bruker designprinsipper for å lære om programmering, bidro til at flere jenter ønsket å ta fag innen informatikk etter fullført kurs. I Norge har det også vært forsøkt å bruke et kreativt rammeverk for å lære elever om programmering, og forskningen viser at et slik rammeverk spesielt har en positiv effekt på jenter (Giannikos et al., 2014). Som beskrevet før mener Corneliussen & Tveranger (2018) at man må få et nytt syn på programmering som omfatter et mangfoldig perspektiv, at det må finnes bredde i undervisningen samt inkludering av begge kjønn. Garneli (2014) sier at det er viktig å tenke på eksisterende stereotyper blant elever, foreldre og lærere. Som innledningen viser til, finnes det en oppfattelse av at programmering er et realfag, noe som blir motsagt i teorien og som den praktisk-estetiske undersøkelsen understøtter. Å programmere i Kunst og håndverk kan dermed bidra til å gå bort fra en slik oppfattelse. I Skolefagsundersøkelsen 2011 kommer det også frem at mannlige lærere i Kunst og håndverk i gjennomsnitt prioriterer undervisning som baserer seg på fagspesifikk bruk av digitale verktøy mer enn kvinnelige lærere (Espeland et al., 2011). Dette kan indirekte gi signaler om ulike roller og kjønn, og bidra videre med utviklingen av stereotyper. En større bredde i digital teknologi vil også kanskje åpne opp for at programmering brukes på nye måter, slik Antonsen påpeker.

En ettertraktet ferdighet

Selv om de finnes argumenter for at programmering kan styrke faget og være til fordel for elevene, ligger det mest tyngde i at programmering er viktig for samfunnet, og at samfunnet har nytte av elever som kan programmere, og som har en forståelse for programmering og algoritmisk tankegang. Ett av hovedargumentene for at programmering skal bli en del av grunnskolen, er at næringslivet trenger flere som har kunnskaper og ferdigheter innen programmering (Amiri, 2011; Balanskat et al., 2017; EU-kommisjonen, 2018; Sevik, 2018). Likevel er etterspørselen etter spesialiserte ferdigheter innen IKT lavere i Norge, enn generelt i Europa. Spesialistferdigheter innebærer blant annet bruken av programmeringsspråk og programvareutvikling (NOU 2019:2). Dette er statistikk som gjenspeiler dagens situasjon, og det er vanskelig å spekulere i om det kommer til å bli en større eller mindre etterspørsel i Norge. Likevel legges det frem i strategiplanen *Fremtid, fornyelse og digitalisering - Digitaliseringsstrategi for grunnsopplæringen 2017-2021*, at ferdigheter innen algoritmisk tankegang og programmering kan være en hjelp ved eldrebølgen, et grønt skifte og endringer i arbeidsform (Kunnskapsdepartementet, 2017). Dette er også noe Dahlin og Hoem påpeker i sitt intervju. Dahlin mener at Kunst og håndverk også må være med å bidra til å løse morgendagens utfordringer, og at det er en økende etterspørsel etter designere som kan programmere. Amiri (2011) diskuterer også denne problematikken, og mener at programmering ikke lenger er en disiplin for noen få fagfelt, men for mange. Han mener spesielt at dette kan bli sett i felt som omhandler interaktiv media. Teorien som er brukt, nevner også felt i kunst og design som er skapt ved hjelp av ny teknologi, som blant annet spillutvikling, interaktiv media, multimedia, webutvikling, generativ kunst og newmedia art. Hoem (2017a, 2017b) påpeker at med en gang programvare eller maskinvare skal bli brukt til noe, så må man snakke om det han kaller meningsvare. Den praktisk-estetiske undersøkelsen viser at det er flere felt innen Kunst og håndverk som kan ha nytte av å bruke programmering i utvikling av både kunstneriske uttrykk, men også i produktutvikling.

Det er også en økning i programvarer som bruker kunstig intelligens til kreative formål (Arcas, 2016). To eksempler er blant annet programvaren som heter *Nvidia AI*, som kan lage enkle digitale skisser om til fotorealistiske landskapsbilder på noen få sekunder (Burns, 2019). Et annet eksempel er Google sin *Deep Dream Generator*, der de arbeider med maskinlæring, persepsjon og kreativitet. Basert på algoritmer og enorme databaser kan programvaren skape egne bilder (Arcas, 2016). Slike programvarer kan ha implikasjoner på den utøvende kunst- og designpraksisen om bare noen få år, som fotografi, stål og betong hadde for den moderne kunsten (Edwards & Wood, 2012). Et spørsmål vi må stille oss er om elever skal lære om, eller få en forståelse for slike programvarer. Algoritmer og programmering er en stor del av slik programvarer, og programmering i Kunst og håndverk kan være med i diskusjonen, utviklingen og bruken av slike plattformer.

Som Antonsen legger frem, kan flere brukere også føre til en bedre brukervennlighet. Dette kan man allerede se på utviklingen av blokkbasert programmeringsplattformer som Scratch og BBC:Microbit. Dette er plattformer som er skapt for barn og ungdommer. Et annet eksempel er Processing som er laget for kunstnere og designere. Med disse som eksempler kan vi se at en større brukergruppe også fører til nye programmeringsspråk og en enklere brukervennlighet. Reas & Fry (2006) mener at grunnen til at programmering har blitt ansett som en teknisk disiplin er fordi det gjerne har vært personer med et samme tankesett som har utviklet programmeringsspråkene. Utvikling i programmeringsspråk og programvare som blir mer rettet mot kunstdisipliner, barn og unge er også noe Kunst og håndverksfaget på grunnskolen må ta stilling til.

Kritisk demokratiske skapeferdigheter

Å vokse opp i den digitale tidsalderen, garanterer ikke at man utvikler gode ferdigheter innen det digitale. Det eksistere en oppfatning om at dagens elever som er født inn i den digitale tidsalderen er digitale innfødte og dermed gode til å bruke digitale verktøy. Det finnes en formening om at barn og unge vil lære å bruke digitale verktøy uten opplæring i skolen (Nordvell & Fritze, 2015; Strand, 2017). Forskning viser likevel at elevene ikke nødvendigvis er så flinke til å bruke det digitale på en kreativ, kritisk og problemløsningsorientert måte (NOU2013:2). Programmering kan fungere som en plattform for en slik tilnærming om undervisningen blir lagt opp på riktig måte. Som jeg har fremlagt innledningsvis, blir det mer vanlig å se på programmering som noe kreativt, noe som også kommer frem i den praktisk-estetiske undersøkelsen. Nedenfor vil jeg redegjøre for det kritiske aspektet ved programmering.

Som Antonsen og Seip påpeker, kan programmering åpne opp for at elevene blir mer beviste på gjenstander som er laget ved hjelp av programmering, men også på arbeidet som ligger bak. Slik som den ene generelle årsaken for hvorfor programmering bør bli en del av grunnskolen fremhever at elevene må utvikle en forståelse for et stadig mer digitalisert samfunn (Bocconi et al., 2016; Sevik, 2018). Sanne et al., (2016) peker på verdier som demokrati, likhet, likeverd, likestilling og respekt. Knochel & Patton (2015) argumenterer for at det er viktig med kritisk tenkning over hvordan digitale medier former samfunnet. De henviser blant annet til forskning gjort av dataspillet *Civilization* og *Adobe Photoshop*. Som belyst i teori kapittelet er et viktig punkt innen det digitale i Kunst og håndverk i LK06, at elevene skal tenke kritisk over visuelle medier. Knochel & Patton belyser at man også må tenke over hvordan programvare blant annet er med på å styre våre valg, og måten vi tenker over tingene på. De henviser til programmering som et kritisk språk og mener at kritisk tenkning er viktig for kritisk digital utvikling, også i kunstoffagene. Det har kommet et større søkelys på hvordan algoritmer styrer hverdagen til den enkelte, og ulike foredrag på Ted.com viser til noen aspekter ved dette, både de positive sidene, men også de negative sidene ved teknologi og programmering. På den ene siden kan teknologi og programmering hjelpe mennesket i å ta valg eller løse problemer, på den andre

siden kan programmering og teknologi styre oss og styrke fordommer vi har, som ikke er riktige. Kevin Slavin (2011) snakker blant annet om hvordan algoritmer og programvare styrer verden vår, ved å ta valg for oss eller løse problemer for oss. Han stiller dermed spørsmål om hvor lang tid det vil ta til vi tilslutt vil miste kontroll. Zeynep Tufekci (2017) snakker om hvordan algoritmer brukt av for eksempel Facebook, Google og Amazon kan påvirke den enkeltes tilgang til politisk og sosial informasjon. Andreas Ekström (2015) belyser hvordan søkemotorer er koblet sammen med vår partiskhet, og at bak algoritmen finnes en personlig tro. På grunnlag av dette mener Ekström at teknologi og humaniora må jobbe tettere sammen. Dette er kun et lite utvalg av foredrag på ted.com. Likevel kan det belyse problemstillingen rundt et demokratisk samfunn. Slike problemstillinger påvirker også domener i de praktisk og estetiske fagene, som Knochel & Patton er inne på.

Som jeg har belyst ovenfor omhandler mye av programmeringen om interaksjon. Algoritmene som er bak et produkt påvirker hvordan brukeren interagerer med produktet. Kunst og håndverksfaget omhandler ikke kun trykte medier, men også algoritmer i sosiale medier, slik som spillutvikling, interaktiv media, multimedia og webutvikling. Design- og kunstbegreper omhandler mer enn kun tradisjonelle materialer, det omhandler også møte med teknologi, som påvirker interaksjonen vi har med produktet eller kunsten, som igjen påvirker om vi blir en betrakter, en deltaker eller en forbruker. Douglas Rushkoff (2010) skriver i boken *Program or be programmed* om at alle trenger en forståelse for programmering. Han har opplevd at mange argumenterer mot et slikt syn, med henvisninger til at de fint klarer å kjøre en bil, uten å vite hvordan den fungerer. Rushkoff er ikke uenig i dette, men når det kommer til programmering sier han:

«The difference between a computer programmer and a user is much less like that between a mechanic and a driver than it is like the difference between a driver and a passenger. If you choose to be a passenger, then you must trust that your driver is taking you where you want to go. Or that he's even telling you the truth about what's out there. » (Rushkoff, 2010, s. 9).

Oppsummering

Med utgangspunkt i en økende debatt angående programmering, algoritmisk tankegang og koding tilknyttet den nye fagfornyelsen som skal implementeres i 2020, men også et økende søkelys på at programmering er noe kreativt som også tilhører Kunst og håndverk, har et ønske med oppgaven vært å bidra til denne debatten. Semistrukturert forskningsintervju med Annette Seip, Jon Øivind Hoem, Liv Klakegg Dahlin og Roger Antonsen, *research by design* gjennom en praktisk-estetiske undersøkelse og relevant teori har forsøkt å belyse problemstilling:

Hvilke mulige årsaksforklaringer kan belyse hvorfor programmering skal implementeres i Kunst og håndverk på grunnskolen?

I analyse av intervjuene har årsaksforklaringer som understøtter problemstilling vært innenfor aktualisering av Kunst og håndverk, bevisstgjøring, designprosess med teknologi, forståelse for det særegne ved verktøyet, tverrfaglighet og til fordel for programmering. Det har også blitt belyst argumenter som kan gjøre det vanskelig for at programmering skal bli en del av faget og har vært som følger, fagfelt med et annet fokus og en manglende kompetanse, miljø, økonomi og noe annet må ut. Det som har kommet tydelig frem i oppgaven er at det er en splittelse mellom de eksisterende strukturene i grunnskolen og mulighet til eventuelt en utvikling av programmering som en del av faget kunst og håndverk. I denne oppgaven har funnene vært at fagfeltet i Kunst og håndverk har lite fokus på fagspesifikke digitale verktøy og programmering. Det har vært et større fokus på bruk av tradisjonelle materialer og håndverk, og utvikling av elevenes ferdigheter innen det taktile og motoriske. Som den praktisk-estetiske undersøkelsen viser, trenger det ikke å være en slik motsetning. Rammefaktorer som tid og økonomi kan bidra til at det blir vanskelig å inkorporere programmering i kunst- og håndverksfaget. Dette viser seg i mangel på avsatt tid for lærere til kompetanseheving og planlegging av undervisning, tid til å gjennomføre programmering i klasserommet, mangel på utstyr, tilgjengelige og egnede kurs for lærere, og behovet for å undervise i mindre klasser på grunn av fagets egenart.

Under strukturell utvikling har et av hovedargumentene gått på at programmering har et stort potensial for å skape både digitale produkter og fysiske produkter. Videre kan programmering benyttes for dybdeløring i sentrale begreper innen Kunst og håndverk som mønster, former og blanding av lysfarger, gjerne i et tverrfaglig perspektiv med Matematikk og Naturfag. Programmering kan også bygge videre på det algoritmiske som eksisterer i faget, både i eksisterende og historiske kunst- og håndverkspraksiser, men også i skapelsesprosesser. Programmering blir også ansett som et nytt materiale, og en innføring kan gi elevene kjennskap til hvordan det er mest hensiktsmessig å bruke det nye materiale. Bruk av programvare av fysiske- og digitale produkter kan også føre til at elevene får arbeidet innenfor et komplekst designbegrep som også inkorporerer teknologisk interaksjon. Programmering er allerede en viktig del av kunst og håndverksfaget når det kommer til bruken av programvarer som er programmerte. Programmering i Kunst og håndverk kan være med i diskusjonen, utviklingen og bruken av slike plattformer. I et fremtidsperspektiv er det også mulig å fundere over om programmering blir en viktigere ferdighet for kunst- og designyrker, noe som også bør

gjenspeiles i faget. Til slutt har programmering mulighet til å gi elevene kritisk demokratiske skaperferdigheter, der de kan være med på både å utvikle, men også stille seg kritiske til teknologiske produkter de selv bruker.

Veien videre

At det finnes lite forskning på feltet er noe som har blitt belyst i oppgaven, det er derfor mange muligheter for videre forskning innen feltet. Denne oppgaven er ikke et svar på hvorfor programmering skal bli implementert i Kunst og håndverk på grunnskolen, men er et bidrag til diskusjonen. Det bør bli gjort videre undersøkelser, kartlegging, drøfting eller diskusjoner omkring temaet. I oppgaven har jeg fremlagt ulike skapemuligheter programmering kan ha i Kunst og håndverk, kategoriene som er satt kan bidra til en videre undersøkelse om hva elevene skal lære innen programmering i Kunst og håndverk. Dette må ses i lys av for eksempel bærekraftig utvikling, rammefaktorer rundt læreren slik som tid og tilgang på utstyr. Det bør også undersøkes videre om hvilke programmeringsspråk elevene eventuelt bør bruke, og på hvilket klasstrinn. En videre undersøkelse om rammefaktorer knyttet til digital teknologi bør også bli gjennomført, slik at det blir gitt gode muligheter for at programmering faktisk blir en del av den gjennomførte undervisningen. I min undersøkelse har det kommet tydelig frem at et skapenderammeverk er en god måte å undervise programmering på, videre bør det bli gjort undersøkelser som tydeligere viser hvordan dette skal gjøres, og da tilknyttet til hva elevene skal lære og hvorfor.

Videre er det mulig å undersøke hver enkelt årsaksforklaring som gjør det vanskelig for at programmering skal bli en del av kunst og håndverk på grunnskolen, samt bør-perspektiver som har vært belyst i denne avhandlingen bli undersøkt videre. Hver årsaksforklaring berører kun toppen av problemområde, en grundigere utforskning bør gjøres. Med en utvidet definisjon av digitale ferdigheter, både internasjonalt og nasjonalt, kunne det ha vært interessant å se på konsekvensene av dette i Kunst og håndverk. Videre hadde det også vært spennende å se på hva algoritmisk tankegang, koding og programmering er i forhold til faget, og skape klare definisjoner, som igjen kan gjøre det enklere for vurderingspraksisen.

Jeg synes også en spennende problemstilling ville vært å undersøke det algoritmiske som ligger i kunst og håndverksfaget. Informantene har allerede nevnt noen eksempler, men det eksisterer mange flere, både knyttet til kunst- og håndverkspraksisen men også til skaperprosessen. Videre er det også mulig å undersøke flere områder eller temaer der programmering kan bli benyttet som et verktøy til å lære sentrale begreper i kunst og håndverk, og gjerne i en tverrfaglig sammenheng, slik at det ivaretar dybdelæring og tverrfaglighet. En annen interessant vinkling hadde vært å undersøke om hvordan elevene arbeider i en kunst- eller designprosess der de bruker programmering. Hvilke problemer er det som dukker opp? Hvordan forandrer eventuelt prosessen seg? Er det en forandring i prosessen? Sitter elevene igjen med et annet læringsutbytte, enn om de skulle ha arbeidet med tradisjonelle materialer?

Litteraturliste

- Ackermann, E. (2001) Piaget's constructivism, Papert's constructionism: What's the difference. Hentet fra [https://eclass.gunet.gr/modules/document/file.php/LABGU151/2.-%20Lecturas%20\(readings,%20ENG\)/Ackermann_piaget_papert.pdf](https://eclass.gunet.gr/modules/document/file.php/LABGU151/2.-%20Lecturas%20(readings,%20ENG)/Ackermann_piaget_papert.pdf)
- Adams, J. (2007, Mars) *Alice, Middle-Schoolers, & The Imaginary Worlds Camps*. Innlegg presentert ved the 38th SIGCSE technical symposium on Computer science education (SIGCSE '07). New York, NY, USA, doi: [10.1145/1227504.1227418](https://doi.org/10.1145/1227504.1227418)
- Alvesson, M. & Sköldbeg, K. (2009) *Reflexive methodology: new vistas for qualitative research* (2. utg). London: Sage.
- Amiri, F. (2011) Programming as Design: The Role of Programming in Interactive Media Curriculum in Art and Design. *International Journal of Art & Design Education*, 30(2), 200-210, doi: [10.1111/j.1476-8070.2011.01680.x](https://doi.org/10.1111/j.1476-8070.2011.01680.x)
- Andresen-Borud, N. (2018, 10 september) Innlegg: Kidsa trenger ikke lære koding. *Dagens Næringsliv*. Hentet fra <https://www.dn.no/kommentar/utdannelse/roboter/kultur/innlegg-kidsa-trenger-ikke-lare-koding/2-1-416411>
- Arcas, B. A. (2016, Mai) *How computers are learning to be creative* [Videoklipp]. Hentet fra https://www.ted.com/talks/blaise_aguera_y_arcas_how_computers_are_learning_to_be_creative
- Artut, S. (2017) Incorporation of computational creativity in arts education: Creative coding as an art course. *SHS Web of Conferences*, 37, 01028, doi: [10.1051/shsconf/20173701028](https://doi.org/10.1051/shsconf/20173701028)
- Balanskat, A., Engelhardt, K. & Ferrari, A. (2017) The intergration of computational Thinking (CT) across school curricula in Europa. *European Schoolnet*. Hentet fra http://www.eun.org/documents/411753/665824/Perspective2_april2017_onepage_def.pdf/70b9a30e-73aa-4573-bb38-6dd0c2d15995
- Ball, T., Protzenko, J., Bishop, J., Moskal, M., de Halleux, J. & Braun, M. (2016, mai) *The BBC micro: bit coded by Microsoft Touch Develop*. Innlegg presentert ved 38th International Conference on software engineering companion (ICSE-C), Austin, TX, USA. Hentet fra <https://pdfs.semanticscholar.org/04c4/8be55ed39865458122972d5b829c1bc107a1.pdf>
- Bergersen, T. (2017, 18 oktober) Programmering i Bærumsskolen. *Budstikka*. Hentet fra <https://www.budstikka.no/debatt/barumsskolen/truls-bergersen/programmering-i-barumsskolen/s/5-55-557966>
- Bamford, A. (2012) *Arts and cultural education in Norway 2010/2011: report*. Bodø: Nasjonalt senter for kunst og kultur i opplæringen. Hentet fra <https://kunstkultursenteret.no/wp-content/uploads/2019/01/Arts-and-Cultural-Education-in-Norway-2010-2011.pdf>

- Bishop, C. (2006) *Participation*. London: Whitechapel Gallery
- Bjarnø, V., Giæver, T. H., Johannesen, M. & Øgrim, L. (2017) *DidIKTikk – Fra digital kompetanse til praktisk undervisning*. (3.utg) Bergen: Fagbokforlaget.
- Bjørnevoll, E. (2016) *Hva kan skolen lære av kodeklubbene?* (Mastergradsavhandling). Høgskolen Stord/Haugesund, Stord. Hentet fra <https://brage.bibsys.no/xmlui/bitstream/handle/11250/2427915/EgilBjornevoll.pdf?sequence=1&isAllowed=y>
- Bocconi, S., Chiocciariello, A., Dettori G., Ferrari, A. & Engelhardt, K. (2016) *Developing Computational Thinking in Compulsory Education – Implications for policy and practice* (EUR 28295 EN), doi:10.2791/792158
- Bocconi, S., Chiocciariello, A. and Earp, J. (2018) *The Nordic approach to introducing Computational Thinking and programming in compulsory education*. (Report for the Nordic@BETT2018 Steering Group). Hentet fra <https://www.itd.cnr.it/doc/CompuThinkNordic.pdf>
- Bråten, E. (2015) Spørsmål om «ontologi» - Momenter til en realistisk antropologi. *Norsk antropologisk tidsskrift* 26(2) 162-176. Hentet fra http://dspace.uib.no/bitstream/handle/1956/16708/spoersmaal_om_ontologi_-_momenter_til_en_realistisk_antropo.pdf?sequence=3&isAllowed=y
- Buch-Hansen, H. & Nielsen, P. (2005) *Kritisk realisme*. Frederiksberg: Roskilde Universitetsforlag
- Burke, Q. & Kafai, Y. B. (2010, Juni) *Programming & Storytelling: Opportunities for learning About Coding & Composition*. Innlegg presentert ved 9th International Conference on Interaction Design and Children. Barcelona, Spain, doi: 10.1145/1810543.1810611
- Burns, M. (2019, 18. april) Nvidia Ai turns sketches into photorealistic landscapes in seconds. *Techrunch*. Hentet fra https://techcrunch.com/2019/03/18/nvidia-ai-turns-sketches-into-photorealistic-landscapes-in-seconds/?guccounter=1&guce_referrer_us=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_cs=o_ZupAZhyi0EP3_fJ6HjaA
- Bygstad, B. & Munkvold, B. E. (2011, Juni) *In Search of Mechanisms. Conducting a Critical Realist Data Analysis*. Innlegg presentert ved Thirty Second International Conference on Information Systems. Shanghai, Kina. Hentet fra https://www.researchgate.net/publication/221600158_In_Search_of_Mechanisms_Conducting_a_Critical_Realist_Data_Analysis
- Corneliussen, H. G. & Tveranger, F. (2018) *Programming in Secondary Schools in Norway – a Wasted Opportunity for Inclusion*. Innlegg presentert ved Gender&IT'18, Heilbronn, Germany, May 2018 (Gender&IT'18), New York, NY, USA. Hentet fra <https://doi.org/10.1145/3196839.3196867>
- Dahlin, L. K. & Gjerde, H. (2009) Å være digital i Kunst og håndverk. I H. Otnes (Red.), *Å være digital i alle fag* (s. 189-207). Oslo: Universitetsforlaget.
- Drone How (2015) *Arduino Tutorial: LED Sequential Control- Beginner Project* [Videoklipp]. Hentet fra <https://www.youtube.com/watch?v=e1FVSpkw6q4>

- Duncan, C., Bell, T. & Tanimoto, S. (2014, November) *Should your 8-year-old learn coding?* Innlegg presentert ved 9th Workshop in Primary and Secondary Computing Education. Berlin, Tyskland doi: [10.1145/2670757.2670774](https://doi.org/10.1145/2670757.2670774)
- Dundas, A. A. (2011) *Hva skjedde med teknologi i skolen? En studie av læreres erfaringer med teknologi og design i grunnskolen?* (Mastergradsavhandling). NTNU, Trondheim. Hentet fra https://brage.bibsys.no/xmlui/bitstream/handle/11250/270256/445928_FULLTEXT01.pdf?sequence=1&isAllowed=y
- Edwards, S. & Wood, P. (2012) *Art & Visual Culture – 1850-2010 – Modernity to Globalisation*. London: Tate
- Egeberg, G., Hultin, H. & Berge, O. (2016) *Monitor skole 2016: Skolens digitale tilstand*. Hentet fra https://www.udir.no/globalassets/filer/tall-og-forskning/rapporter/2016/monitor_2016_nn_-_2_utgave_lav.pdf
- Egeberg, G., Guðmundsdóttir, G. B., Hatlevik, O. E., Ottestad, G., Skaug, J. H. & Tømte, K. (2012) *Monitor 2011 Skolens digitale tilstand*. Hentet fra <https://docplayer.me/419721-Monitor-2011-skolens-digitale-tilstand.html>
- Ekström, A. (2015, Januar) *The moral bias behind you search results* [videoklipp]. Hentet fra https://www.ted.com/talks/andreas_ekstrom_the_moral_bias_behind_your_search_results
- Engelsen, B. U. (2003) *ideer som formet vår skole? Lærerplanen som idebærer – et historisk perspektiv*. Oslo: Gyldendal Akademisk
- Espeland, M., Arnesen, T. E., Grønsdal, I. A., Holthe, A., Sømoe, K., Wergedahl, H., & Aadland, H. (2013) *Skolefagsundersøkelsen 2011 Praktiske og estetiske fag på barnesteget i norsk grunnskule* (HSH-rapport 2013/7). Hentet fra <https://brage.bibsys.no/xmlui/bitstream/id/117173/Rapport.pdf>
- EU-kommisjonen (2018) *Communication from the commission to the European parliament, the council, the European economic and social committee and the committee of the regions – on the Digital Education Action Plan* (SWD(2018) 12 final). Hentet fra <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM%3A2018%3A22%3AFIN>
- Garneli, V (2014, Juli) *Instructional Media and Teaching Methods for Engaging Children with Computer Programming*. Innlegg presentert ved [14th International Conference on Advanced Learning Technologies](https://www.researchgate.net/publication/258010206_Teaching_Computer_Science_to_Young_Children_through_Creativity_Lessons_Learned_from_the_Case_of_Norway). Aten, Hellas, doi: [10.1109/ICALT.2014.225](https://doi.org/10.1109/ICALT.2014.225)
- Giannakos, M. N., Jaccheri, L. & Proto, R. (2013, April) *Teaching Computer Science to Young Children through Creativity: Lessons Learned from the Case of Norway*. Innlegg presentert ved Computer Science Education Research Conference (CSERC '13), Arnhem, Nijmegen, Nederland. Hentet fra https://www.researchgate.net/publication/258010206_Teaching_Computer_Science_to_Young_Children_through_Creativity_Lessons_Learned_from_the_Case_of_Norway
- Gilje, Ø., Landfald, Ø. F. & Ludvigsen, S. (2018, 29. november) Det er to grunner til at dybdelæring er

- viktig for elevenes fremtidige kompetanse. *Utdanningsnytt*. Hentet fra <https://www.utdanningsnytt.no/bedre-skole/debatt/2018/november/dybdelaring--historisk-bakgrunn-og-teoretiske-tilnarminger/>
- Godin, D. & Zahedi, M. (2014, Juni) *Aspects of Research through Design*. Innlegg presentert ved Design Research Society (DRS2014). Umeå, Sverige. Hentet fra <http://www.drs2014.org/media/648109/0205-file1.pdf>
- Graham, P (2004) *Hackers & Painters: Big Ideas from the Computer Age*. Sebastopol: O'Reilly Media, Inc.
- Greenberg, I., Kumar, D. & Xu, D. (2012, Februar) *Creative coding and visual portfolios for CS1*. Innlegg presentert ved The 43rd ACM technical symposium on Computer Science Education, Raleigh, North Carolina, USA, doi: 10.1145/2157136.2157214
- Grover, S. & Pea, R. (2013) Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38-43. Hentet fra <http://doi.org/10.3102/0013189X12463051>
- Haakonsen, P. (2012) *Additiv fargeblanding i GIMP* (Mastergradsavhandling). Høgskolen i Oslo og Akershus, Oslo. Hentet fra https://oda.hioa.no/nb/additiv-fargeblanding-i-gimp/asset/dspace:4727/Haakonsen_Peter.pdf
- Haaland, S. & Rosvold, J. S. (2006) *Bruk av Lego Mindstorms i natur- og miljøfag på 9. trinn* (Mastergradsavhandling). Høgskolen Stord/Haugesund, Stord. Hentet fra <https://brage.bibsys.no/xmlui/bitstream/handle/11250/152315/HaalandRosvold.pdf?sequence=1&isAllowed=y>
- Hatlevik O. E. & Throndsen, I. (2015) *Læring av IKT – Elevenes digitale ferdigheter og bruk av IKT i ICILC 2013*. Hentet fra <https://www.duo.uio.no/bitstream/handle/10852/49792/laering-av-ikt.pdf?sequence=1&isAllowed=y>
- Hatlevik, O. E., Egeberg, G. Guðmundsdóttir, G. B., Loftsgarden, M. & Loi, M. (2014) *Monitor skole 2013 Om digital kompetanse og erfaringer med bruk av IKT i skolen*. Hentet fra https://www.udir.no/globalassets/monitor_skole_2013_4des.pdf
- Hiltunen, T. (2016) *Learning and teaching programming skills in Finnish Primary Schools – The potential of games* (Mastergradsavhandling). Universitetet i Oulu, Oulu. Hentet fra <http://jultika.oulu.fi/files/nbnfioulu-201605221873.pdf>
- Hjelmberg, J. (2015) *Nettbrett i kunst og håndverk - muligheter og utfordringer* (Mastergradsavhandling). Høgskolen i Oslo og Akershus, Oslo. Hentet fra <https://oda.hioa.no/nb/nettbrett-i-kunst-og-handverk-muligheter-og-utfordringer>
- Hoem, J. Ø. (2017a, 22. oktober) *Barn og koding – ikke bare realfag* [Blogginlegg]. Hentet fra <https://blogg.hvl.no/iktiutdanning/2017/10/22/barn-koding-realfag/>
- Hoem J. Ø. (2017b, 13. november) *Generativ kunst og tankefullt håndverk* [Blogginlegg]. Hentet fra <https://blogg.hvl.no/iktiutdanning/2017/11/13/generativ-kunst/>

- Hunt, A. & Thomas, D. (2001, september) *The art in Computer Programming*. Innlegg presentert ved Java and Object Oriented Conference, Århus, Danmark. Hentet fra https://www.developerdotstar.com/mag/articles/PDF/DevDotStar_HuntThomas_ArtInProgramming.pdf
- Høgskolen i Østfold (u.å) Programmering i skolen 1.-10. trinn. Hentet den 06. april 2019 fra <https://www.hiof.no/studier/programmer/s870k-programmering-i-skolen-1.-10.-trinn/index.html>
- IEA (u.å) ICILC. International Computer and Information Study. Hentet den 5. mars 2019 fra <https://www.iea.nl/icils>
- IKTipraksis (u.å) Finn ressurser. Hentet 31. Mars 2018 fra https://iktipraksis.iktsenteret.no/sok#fq={!tag=ss_menu_1}ss_menu_1%3Aa_undervisningsoppleggmeny
- Iversen, S. (2015) *Koding som digital grublis. En kvalitativ studie om hvordan elevenes læringsstrategier påvirkes gjennom programmering* (Mastergradsavhandling). UiT Norges arktiske universitet, Tromsø. Hentet fra <https://munin.uit.no/bitstream/handle/10037/8089/thesis.pdf?sequence=2&isAllowed=y>
- Jacobsen, D. I. (2015) *Hvordan gjennomføre undersøkelser? Innføring i samfunnsvitenskapelig metode* (3. Utg). Oslo: Cappelen Damm akademisk.
- Johansson, L. (2018) Kjerneelementer – siste sjanse! *Form – Fagdidaktisk tidsskrift for kunst og design*. 52(2), 9-9. Hentet fra https://docs.wixstatic.com/ugd/75f35c_98c7d855d36a483ead3bdde542fee3db.pdf
- Jåtten, E. (2006) *First Lego League og motivasjon for realfag*. (Mastergradsavhandling). Høgskolen Stord/Haugesund, Stord. Hentet fra <https://brage.bibsys.no/xmlui/bitstream/handle/11250/152236/Jatten.pdf?sequence=1&isAllowed=y>
- KidsaKoder (u.å.a) Straffespark. Hentet den 9. april 2019 fra <https://oppgaver.kidsakoder.no/scratch/straffespark/straffespark>
- KidsaKoder (u.å.b) kodetimen. Hentet den 31. Mars 2019 fra <https://kidsakoder.no/kodetimen/>
- Kjerneelementgruppen for kunst og haandtverk (2017, 22. september) *Tilbake til verkstedene!* [Blogginlegg]. Hentet fra <http://udirbloggen.no/tilbake-til-verkstedene/>
- Kjerneelementgruppen for kunst og haandtverk (2017, 27. november) *Oppsummering av kunst og håndverk – andre innspillrunde* [Blogginlegg]. Hentet fra <https://udirbloggen.no/oppsummering-kunst-handverk-innspillrunde/>
- Kjerneelementgruppen for matematikk (2017, 21. september) *Kjerneelementer i matematikk, men hvorfor programmering?* [Blogginlegg]. Hentet fra <http://udirbloggen.no/kjerneelementer-i-matematikk-men-hvorfor-programmering/>
- Knobelsdorf, M., & Romeike, R. (2008, Juni-Juli) *Creativity as a pathway to computer science*. Innlegg

presentert ved the 13th annual conference on Innovation and technology in computer science education (ITiCSE '08). New York, NY, USA, doi: 10.1145/1384271.1384347

- Knochel, A., D. & Patton, R., M. (2015) *If Art Education Then Critical Digital Making: Computational Thinking and Creative Code*, *Studies in Art Education*, 57(1), 21-38, doi: 10.1080/00393541.2015.11666280
- Knuth, D. (1974) Computer programming as an art. *Communications of the ACM*, 17(12), 667-673. doi: 10.1145/361604.361612
- Ko, A. J. (2017, 12. juli) *Mindstorms: what did Papert argue and what does it mean for learning and education?* [Blogginnlegg]. Hentet fra <https://medium.com/bits-and-behavior/mindstorms-what-did-papert-argue-and-what-does-it-mean-for-learning-and-education-c8324b58aca4>
- Kunnskapsdepartementet (2017, 25. august) *Fremtid, fornyelse og digitalisering -Digitaliseringsstrategi for grunnopplæringen 2017-2021*. Hentet fra <https://www.regjeringen.no/no/dokumenter/framtid-fornyelse-og-digitalisering/id2568347/>
- Kunnskapsdepartementet (2018a, 26. Juni) *Kjerneelementer i fag*. Hentet fra <https://www.regjeringen.no/contentassets/3d659278ae55449f9d8373fff5de4f65/kjerneelementer-i-fag-for-utforming-av-lareplaner-for-fag-i-lk20-og-lk20s-fastsatt-av-kd.pdf>
- Kunnskapsdepartementet (2018b, 12. november) *15 millioner til programmering i skolen* [Pressemelding (Nr: 198-18)]. Hentet fra <https://www.regjeringen.no/no/aktuelt/15-millioner-til-programmering-i-skolen2/id2618641/>
- Kunnikoff, J. O. (2015) *Formell fagkompetanse i Kunst og håndverk: Betydning, Prioritering og konsekvenser*. (Mastergradsavhandling). Høgskolen i Oslo og Akershus, Oslo. Hentet fra <https://oda.hioa.no/en/formell-kompetanse-i-kunst-og-handverk-betydning-prioritering-og-konsekvenser>
- Kvale, S., & Brinkmann, S. (2015) *Det kvalitative forskningsintervju* (3. utg.). Oslo: Gyldendal akademisk
- Kwon, S. & Schroderus, K. (2017) Coding in Schools – Comparing Integration of Programming into Basic Education Curricula of Finland and South Korea. Hentet fra <https://mediakasvatus.com/wp-content/uploads/2017/02/Coding-in-schools-FINAL-2.pdf>
- Lang-Ree, H. L. (2016) *Vi må tenke og ikke bare tegne – en kvalitativ studie om bruk av programmering som verktøy i arbeid med matematikk*. (Mastergradsavhandling). NTNU, Trondheim. Hentet fra https://brage.bibsys.no/xmlui/bitstream/handle/11250/2396148/Lang-Ree_2016.pdf?sequence=1&isAllowed=y
- Lau, W. W. Y., Ngai, G., Chan, S. C. F., & Cheung, J. C. Y. (2009). Learning programming through fashion and design: a pilot summer course in wearable computing for middle school students. *ACM SIGCSE Bulletin*, 41(1), 504-508, doi:10.1145/1539024.1509041
- Lie, C., & Nielsen, L. M. (2016) Lærernes arbeidstid i Kunst og håndverk – grunnlaget for gammeldags leseplikt i grunnskolen. *FormAkademisk - Forskningstidsskrift for Design Og Designdidaktikk*, 9(2).

<https://doi.org/10.7577/formakademisk.1845>

Maeda, J. (2000) *Maeda @media*. London: Thames & Hudson Ltd.

Maeda, J (2004) *Creative Code*. London: Thames & Hudson Ltd.

Makecode.Microbit (u.å) *Acceleration*. Hentet den 9. april 2019 fra <https://makecode.microbit.org/reference/input/acceleration>

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education (TOCE)*, 10(4), 1-15, doi:10.1145/1868358.1868363

Morrison, A., & Sevaldson, B. (2010) 'Getting Going' – Research by Design. *FormAkademisk - Forskningstidsskrift for Design Og Designdidaktikk*, 3(1). Hentet fra <https://doi.org/10.7577/formakademisk.136>

Nielsen, L. M. (2009) *Fagdidaktikk for kunst og håndverk: i går, i dag, i morgen*. Oslo: Universitetsforlaget

Nordkvelle, Y. T. & Fritze, Y. (2015) Digitalt innfødte eller bare medialiserte. I Y. Fritze, G. Haugsbakk & Y. T. Nordkvelle (Red.), *Mediepedagogiske perspektiver* (s. 67- 83). Oslo: Cappelen Damm Akademisk

Norwaymakers (u.å) Hvordan lage et makerspace? Hentet den 10. april 2018 fra <http://norwaymakers.org/kart>

NOU 2013:2 (2013) *Hinder for digital verdiskaping*. Hentet fra <https://www.regjeringen.no/contentassets/e2f0d5676e144305967f21011b715c16/no/pdfs/nou201320130002000dddpdfs.pdf>

NOU 2015:8 (2015) *Fremtidens skole – Fornyelse av fag og kompetanser*. Hentet fra <https://www.regjeringen.no/contentassets/da148fec8c4a4ab88daa8b677a700292/no/pdfs/nou201520150008000dddpdfs.pdf>

NOU 2019:2 (2019) *Fremtidige kompetansebehov II – Utfordringer for kompetansepolitikken*. Hentet fra <https://www.regjeringen.no/no/dokumenter/nou-2019-2/id2627309/>

NTNU (u.å) Programmering 8.-13. trinn. Hentet den 06. april 2019 fra <https://www.ntnu.no/videre/programmering>

Nygård, K. (2018) *Programmering i grunnskolen- hvordan komme i gang?* Oslo: Pedlex

Næss, P. (2012) Kritisk realisme og byplanforskning. *FormAkademisk - Forskningstidsskrift for Design Og Designdidaktikk*, 5(2), 1-17. Hentet fra <https://doi.org/10.7577/formakademisk.493>

Olsen, I. O. (2014) *Digital kompetanse i kunSSst og håndverk - En kvalitativ kasusstudie av fire læreres posisjoneringer* (Mastergradsavhandling). Høgskolen i Oslo og Akershus, Oslo. Hentet fra <https://oda.hioa.no/en/digital-kompetanse-i-kunst-og-handverk-en-kvalitativ-kasusstudie-av-fire-laereres-posisjoneringer>

- Opplæringsloven (1998) Lov om grunnskolen og den vidregående opplæringa. LOV-1998-07-17-61. Hentet fra https://lovdata.no/dokument/NL/lov/1998-07-17-61/KAPITTEL_1
- Ottestad, G., Throndsen, I., Hatlevik, O. & Rohatgi, A. (2014) *Digitale ferdigheter for alle? Norske resultater fra ICILS 2013*. Hentet fra <https://www.uv.uio.no/ils/forskning/aktuelt/aktuelle-saker/2014/icils-rapport-trykk25.11.pdf>
- Pearson, M. (2011) *Generative art – a practical guide using processing*. New York: Manning Publications Co
- Peppler, K. A. & Kafai, Y. B. (2009, Juni) *Coding: Programming for Personal Expression*. Innlegg presentert ved The 8th International Conferende on Computer Supported Collaborative Learning (CSCL). Rodes, Hellas. Hentet fra https://www.researchgate.net/publication/228625777_Creative_coding_Programming_for_personal_expression
- Porko-Hudd, M. (2017) Finsk sløyd: Teknologi, håndverk og design. *Form – Fagdidaktisk tidsskrift for kunst og design*, 51(4), 4-5. Henter fra https://docs.wixstatic.com/ugd/75f35c_c67a9867e83643c88aead61217221d33.pdf
- Price, C. B. (2007) From Kandinsky to Java (The Use of 20th Century Abstract Art in Learning Programming), *Innovation in Teaching and Learning in Information and Computer Sciences*, 6(4), 35-50, doi: 10.11120/ital.2007.06040035
- Rathod, H (2017) *Control Arduino Using GUI (Arduino + Processing)* [Videoklipp]. Hentet fra <https://www.youtube.com/watch?v=5WjEQSMiqMQ>
- Reas, C., & Fry, B. (2006) Processing: programming for the media arts. *AI & SOCIETY*, 20(4), 526-538, doi:10.1007/s00146-006-0050-9
- Statsministerens kontor (2017, 25. august) *Mer koding og teknologi inn i skolen* [Pressemelding (Nr 84/17)]. Hentet fra <https://www.regjeringen.no/no/aktuelt/mer-koding-og-teknologi-inn-i-skolen/id2568375/>
- Statsministerens kontor (2018, 14. januar) *Jeløya-plattformen*. Hentet fra <https://www.regjeringen.no/no/dokumenter/politisk-plattform/id2585544/>
- Statsministerens kontor (2019, 14. januar) *Granavolden-plattformen*. Hentet fra <https://www.regjeringen.no/no/dokumenter/politisk-plattform/id2626036/>
- Rushkoff, D. (2011) *Program or be programmed: ten commands for a digital age*. Berkeley, California: Soft Skull Press.
- Ryssevik, T. (2017, 31. august) Fag og fornying – sentrale idear og begrep. *Utdanningsforbundet*. Hentet fra <https://www.utdanningsforbundet.no/var-politikk/kunnskapsgrunnlag/publikasjoner/2017/fag-og-fornyig--sentrale-idear-og-begrep/>
- Sanne, A., Berge, O., Bungum, B., Jørgensen, E. C., Kluge, A., Kristensen, T. E., Mørken, K. M., Svorkmo, A.-G. & Voll, L. O. (2016) *Teknologi og programmering for alle – En faggjennomgang med forslag til*

endring i grunnopplæringen. Hentet fra <https://www.udir.no/globalassets/filer/tall-og-forskning/forskningsrapporter/teknologi-og-programmering-for-alle.pdf>

Sennett, Richard. (2008) *The craftsman*. New Haven, Conn: Yale University Press

Sevaldson, B. (2010) Discussions & Movements in Design Research. *FormAkademisk - Forskningstidsskrift for Design Og Designdidaktikk*, 3(1). Hentet fra <https://doi.org/10.7577/formakademisk.137>

Sevaldson, B (2011, Mai) *Giga-mapping: Visualisation for complexity and systems thinking in design*. Innlegg presentert ved Nordic Design Research. Helsinki, Finland. Hentet fra <http://www.nordes.org/opj/index.php/n13/article/view/104>

Sevik, K. (2016) *Programmering i skolen*. Hentet fra https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf

Sevik, K. (2018, 27. mars) *Notat om programmering i skolen*. Hentet fra <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/notat-om-programmering-i-skolen/>

Schmidt, A. (2016) Increasing Computer Literacy with the BBC micro:bit. *IEEE Pervasive Computing*, 15(2), 5-7. doi: 10.1109/MPRV.2016.23

Scratch (u.å) About Scratch. Hentet den 31. mars 2019 fra <https://scratch.mit.edu/about>

SFS 2213. (2015). *Særavtale for undervisningspersonalet i kommunal og fylkeskommunal grunnopplæring*. Hentet fra <https://www.utdanningsforbundet.no/globalassets/lonn-og-arbeidsvilkar/tariffavtaler/1-ks/sfs-2213-arbeidstid-skole.pdf>

Shiffmans, D. [TheCodingTrain] (2015) *Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction* [Videoklipp]. Hentet fra https://www.youtube.com/user/shiffman/playlists?view=50&sort=dd&shelf_id=2

Skjong, H (2017, 19 November) Når robotene tar over. *Utdanningsnytt*. Hentet fra <https://www.utdanningsnytt.no/utdanning/artikler/2017/november/nar-robotene-tar-over/>

Slavin, K. (2011, Juli) *How algorithms shape our world* [Videoklipp]. Hentet fra https://www.ted.com/talks/kevin_slavin_how_algorithms_shape_our_world

Strand, I. (2017) *Digitale verktøy i kunst og håndverk – Læreres undervisningspraksis, muligheter og begrensninger* (Mastergradsavhandling). Høgskolen i Oslo og Akershus, Oslo. Hentet fra <https://oda.hioa.no/en/item/digitale-verktoy-i-kunst-og-handverk-laereres-undervisningspraksis-muligheter-og-begrensninger>

Stærk, B. (2016, 21. juni) Programmerere er bare digitale rørleggere. *Aftenposten*. Hentet fra https://www.aftenposten.no/meninger/debatt/i/yEpL2/Barn-bor-kode-fordi-det-er-et-viktig-handverk_ikke-for-a-bli-smartere--Bjorn-Stark

Sunde, D. J (2018, 31 Januar) *Verden er ikke delt opp i fag* [blogginlegg]. Hentet fra <http://udirbloggen.no/>

[verden-delt-fag/](#)

Söderström, F. & Nikka, K. E. (2017) *Hvordan virker Arduino? Lær fra grunnen av og med raskt resultat*. Malmø: Kjell & Company

Sømoie, K. (2010) *Skolefagsundersøkelsen 2009 Fagrapport Kunst og håndverk*. Hentet fra <https://brage.bibsys.no/xmlui/bitstream/handle/11250/152112/Rapport.pdf?sequence=3&isAllowed=y>

TekniskMuseum (u.å.a) Teknoteket makerspace. Hentet den 14. april 2019 fra <https://www.tekniskmuseum.no/utstillinger-2/1064-makerspace-lanseres-i-norge>

TekniskMuseum, (u.å.b) Programmeringsløft for 100 000 grunnskolebarn. Hentet den 14. april 2019 fra https://www.tekniskmuseum.no/nyheter/1707-programmeringsloft-for-100-000-grunnskolebarn?fbclid=IwAR12Nwsi9AIBGZQQBRyTmUTDp9QpOh27Fmz3Perx4DaTzxcJ_6YqqDMdF7U&mc_cid=cdb9c5a897&mc_eid=db07568c2f

That Cat (2016) *How to Read the Temperature Sensor on the BBC MicroBit* [Videoklipp]. Hentet fra <https://www.youtube.com/watch?v=HmqfvISAyIk>

Tufekci, Z. (2017, September) *We're building a dystopia just to make people click on ads* [Videoklipp]. Hentet fra https://www.ted.com/talks/zeynep_tufekci_we_re_building_a_dystopia_just_to_make_people_click_on_ads

Utdanningsdirektoratet (2006) *Læreplan i Kunst og håndverk – KHV1-01*. Hente fra <https://www.udir.no/kl06/khv1-01/>

Utdanningsdirektoratet (2016a) Veiledning til programmering valgfag. Hentet fra <https://www.udir.no/laring-og-trivsel/lareplanverket/veiledning-lp/valgfag-programmering/>

Utdanningsdirektoratet (2016b) Digitale ferdigheter som grunnleggende ferdighet. Hentet fra <https://www.udir.no/laring-og-trivsel/lareplanverket/grunnleggende-ferdigheter/digitale-ferdigheter-rammeverk/>

Utdanningsdirektoratet (2017a) Programmering valgfag på ungdomstrinnet. Hentet fra <https://www.udir.no/kvalitet-og-kompetanse/nasjonale-satsinger/realfagsstrategien/forsok-med-programmering-som-valgfag/>

Utdanningsdirektoratet (2017b) *Første skisse til kjerneelementene i kunst og håndverk/duodji* (Høringsbrev av 06.09.2017) Hentet fra <https://hoering.udir.no/Hoering/v2/151?notatId=227>

Utdanningsdirektoratet (2017c) *Første skisse til kjerneelementer i matematikk fellesfag* (Høringsbrev av 06.09.2017). Hentet fra <https://hoering.udir.no/Hoering/v2/151?notat=213¬atId=222>

Utdanningsdirektoratet (2017d) *Kunst og håndverk/Doudji/Duodje/Duedie* (Høringsbrev av 23.11.2017). Hentet fra <https://hoering.udir.no/Hoering/v2/162?notatId=257>

Utdanningsdirektoratet (2017e) *Matematikk* (Høringsbrev av 23.11.2017). Hentet fra <https://hoering.udir.no/Hoering/v2/162?notatId=257>

[no/Hoering/v2/162?notatId=252](https://www.udir.no/Hoering/v2/162?notatId=252)

Utdanningsdirektoratet (2017f) *Naturfag* (Høringsbrev av 23.11.2017). Hentet fra <https://hoering.udir.no/Hoering/v2/162?notatId=249>

Utdanningsdirektoratet (2017g) Rammeverk for grunnleggende ferdigheter. Hentet fra <https://www.udir.no/laring-og-trivsel/lareplanverket/grunnleggende-ferdigheter/rammeverk-for-grunnleggende-ferdigheter/>

Utdanningsdirektoratet (2018a) *Kunst og håndverk/Doudji/Duodje/Duedie* (Høringsbrev av 05.03.2018). Hentet fra <https://hoering.udir.no/Hoering/v2/197?notatId=356>

Utdanningsdirektoratet (2018b) *Matematikk* (Høringsbrev av 05.03.2018). Hentet fra <https://hoering.udir.no/Hoering/v2/197?notatId=358>

Utdanningsdirektoratet (2018c) *Naturfag / naturfag samisk* (Høringsbrev av 05.03.2018). Hentet fra <https://hoering.udir.no/Hoering/v2/197?notatId=361>

Utdanningsdirektoratet (2018d) *Valgfag på ungdomstrinnet – statistikknotat 1/2018*. Hentet den fra <https://www.udir.no/tall-og-forskning/statistikk/statistikk-grunnskole/valgfag-pa-ungdomstrinnet/>

Utdanningsdirektoratet (2018e) *Kunst og håndverk| Duodji – oppsummering av innspill*. Hentet fra <https://www.udir.no/laring-og-trivsel/lareplanverket/fagfornyelsen/kjerneelementer/kunst-og-handverk--oppsummering-av-innspill/>

Utdanningsdirektoratet (2018f) *Kunst og håndverk* (Høringsbrev av 18.10.2018). Hentet den fra <https://hoering.udir.no/Hoering/v2/283?notatId=546>

Utdanningsdirektoratet (2018g) Overordnet del av læreplanverket. Hentet fra <https://www.udir.no/laring-og-trivsel/lareplanverket/overordnet-del/>

Utdanningsdirektoratet (2018h) *Kunst og håndverk – oppsummering av innspill*. Hentet fra <https://www.udir.no/laring-og-trivsel/lareplanverket/fagfornyelsen/oppsummeringer-av-7000-innspill-til-lareplanskissene/kunst-og-handverk--oppsummering-av-innspill/>

Utdanningsdirektoratet (2018i) *Fag- og timefordeling og tilbudsstruktur for kunnskapsløftet Udir-1-2018*. Hentet fra: <https://www.udir.no/regelverkstolkninger/opplaring/Innhold-i-opplaringen/udir-01-2018/vedlegg-1/2.-grunnskolen/#2.2ordinar-fag-og-timefordeling>

Utdanningsdirektoratet (2019) *Høring – Læreplaner i kunst og håndverk og duodji* (Høringsbrev av 18.03.2019). Hentet fra <https://hoering.udir.no/Hoering/v2/341>

Vasudevan, V. & Kafai, Y. B (2016) Bridging Crafting and Computing in Making: Designing Interactive Touchpads and Board Games With MaKey Makey. I K. Peppler, E. R. Halvorsen & Y. B. Kafai (Red.), *Makeology – Makers as Learners*. (s. 145- 150). New York: Routhledge.

- Verstad, A. L. T. (2017) *Programmering + Matematikk = Sant? En casestudie om overføringsverdien mellom programmering valgfag og matematikkfaget* (Mastergradsavhandling). UiT Norges arktiske universitet, Tromsø. Hentet fra <https://munin.uit.no/bitstream/handle/10037/11601/thesis.pdf?sequence=1&isAllowed=y>
- Vorderman, C (2017) *Spill- og dataprogrammering for barn*. London: Spektrum forlag As.
- Vuorikari, R., Punie, Y., Carretero S. G, & Van Den Brande, G. (2016) *DigComp 2.0: The Digital Competence Framework for Citizens (EUR 27948 En)*. Doi: [10.2791/11517](https://doi.org/10.2791/11517) (online)
- Waterhouse, T (2017) *Teknologi i kunst og håndverk*. [Videoklipp]. Hentet fra <https://film.oslomet.no/teknologi-i-kunst-og-handverk>
- Wing, J (2006) Computational thinking. *Communications of the ACM*. 49(3), 33-35, doi: [10.1145/1118178.1118215](https://doi.org/10.1145/1118178.1118215)
- Yakman, G. (2008) STEAM Education: an overview of creating a model of integrative education. Hentet fra: https://www.researchgate.net/publication/327351326_STEAM_Education_an_overview_of_creating_a_model_of_integrative_education
- Østerud, S. (1998) Relevans av begrepet ”validitet” og ”reliabilitet” i kvalitativ forskning 1: *Norsk pedagogisk tidsskrift*, 1(2) s. 119-129

Figurliste

- Forside bilde: Eget bilde (2018) *Tilfeldige linjer med kode i Processing*.
- Bilde 1: Eget bilde (2018) *Et tegneprogram med enkle linjer*.
- Bilde 2: Eget bilde (2018) *Animasjon av sirkler i ulike farger*.
- Bilde 3: Eget bilde (2018) *Animasjon av cosinus og sinus bølger*.
- Bilde 4: Eget bilde (2018) *Skjerm bilde av frihåndstegning gjort med kode som vist i bilde 5*.
- Bilde 5: Eget bilde (2018) *Skjerm bilde av kode i Processing utviklingsmiljøet. Kode gir et tegneredskap som er brukt i bilde 4 og 1*.
- Bilde 6: Eget bilde (2019) *Skjerm bilde av en tegning som viser en kostyme i Scratch utviklingsmiljøet, brukt til kode som vist i bilde 7*.
- Bilde 7: Eget bilde (2019) *Skjerm bilde av utviklingsmiljøet i Scratch med kode til bilde 6*.
- Bilde 8: Eget foto (2019) *Foto av maskinvare i BBC micro:bit inventorskit og BBC micro:bit programmert som et kompas*.
- Bilde 9: Eget bilde (2018) *Skjerm bilde av en kode i BBC micro:bit utviklingsmiljøet. Kode av temperatursensor og ledlys på ettkortdatamaskinen til BBC micro:bit*.
- Bilde 10: Eget foto (2019) *Foto av maskinvare til Arduino og Arduino Uno kortet som har blitt programmert til ledlys*.

Bilde 11: Eget bilde (2018) *Skjerm bilde av Arduino utviklingsmiljøet, med kode av dioder.*

Bilde 12: Eget bilde (2018) *Utforskning av mønstre.*

Bilde 13: Eget bilde (2018) *Utforskning av former.*

Figur 1: Egen figur (2018) *Samspeilet mellom aktør og struktur basert på Archer (1995) sin modell, gjengitt av Buche-Hansen, H & Nielsen, P. (2005) s. 52.*

Tabell 1: Egen tabell (2018) *Valg av programmeringsplattformer og mulige bruksområder.*

Tabell 2: Egen tabell (2019) *Mulige skapemuligheter med programmering i kunst og håndverk.*

Vedlegg

Vedlegg 1, meldeskjema til NSD

08/04/2019

Meldeskjema for behandling av personopplysninger



Meldeskjema 871930

Sist oppdatert

27.08.2018

Hvilke personopplysninger skal du behandle?

- Navn (også ved signatur/samtykke)
- Fødselsdato
- Adresse eller telefonnummer
- E-postadresse, IP-adresse eller annen nettidentifikator
- Lydopptak av personer
- Bakgrunnsopplysninger som vil kunne identifisere en person
- Andre opplysninger som vil kunne identifisere en fysisk person

Type opplysninger

Du har svart ja til at du skal behandle bakgrunnsopplysninger, beskriv hvilke

Arbeidssted

Du har svart ja til at du behandler andre opplysninger som vil kunne identifisere en person, beskriv hvilke

Navn, kjønn, utdanning, yrke. Jeg vil også kontakte informanten via mail eller telefonnummer.

Skal du behandle særlige kategorier personopplysninger eller personopplysninger om straffedommer eller lovovertridelser?

Nei

Prosjektinformasjon

Prosjekttittel

Programmering i Kunst og håndverk på grunnskolen.

Prosjektbeskrivelse

Masterprosjektet vil forsøke å finne ut hvorfor programmering bør bli en del av Kunst og håndverk på grunnskolen, og hva programmering i faget bør inneholde. Dette betyr at jeg vil finne ut hvorfor programmering er relevant i Kunst og håndverk, hvis det i det hele tatt er det. For å kunne gi en klarere begrunnelse, ønsker jeg også å finne ut av hva med programmering som bør bli lært. Dette vil si hvilket programmeringsspråk som egner seg, og hvilke type oppgaver som bør bli gitt.

Fagfelt

Humaniora, Andre fagfelt

Dersom opplysningene skal behandles til andre formål enn behandlingen for dette prosjektet, beskriv hvilke

Nei

Begrunn behovet for å behandle personopplysningene

Jeg ønsker å ha intervju med et utvalg på fem personer innen fagfeltet programmering og grunnskolen. På grunn av at fagfeltet (programmering i grunnskolen) er så smalt er jeg redd for at det vil være vanskelig å anonymisere personene som skal intervjues. Oppgaven vil også få en mer faglig tyngde om informantene blir nevnt med navn og yrke.

Ekstern finansiering**Type prosjekt**

Studentprosjekt, masterstudium

Kontaktinformasjon, student

Stephanie Corine Antoinette Hoebeke , [REDACTED]

Behandlingsansvar**Behandlingsansvarlig institusjon**

OsloMet - storbyuniversitetet / Fakultet for teknologi, kunst og design / Institutt for estetiske fag

Prosjektansvarlig (vitenskapelig ansatt/veileder eller stipendiat)

Peter Haakonsen, [REDACTED]

Skal behandlingsansvaret deles med andre institusjoner (felles behandlingsansvarlige)?

Nei

Utvalg 1**Beskriv utvalget**

Personer som har kunnskap om programmering og kjennskap til grunnskolen.

Rekruttering eller trekking av utvalget

Informantene vil rekrutteres via mail.

Alder

19 - 70

Inngår det voksne (18 år +) i utvalget som ikke kan samtykke selv?

Nei

Personopplysninger for utvalg 1

- Navn (også ved signatur/samtykke)
- Fødselsdato
- Adresse eller telefonnummer
- E-postadresse, IP-adresse eller annen nettidentifikator
- Lydopptak av personer
- Bakgrunnsopplysninger som vil kunne identifisere en person

Hvordan samler du inn data fra utvalg 1?**Personlig intervju****Grunnlag for å behandle alminnelige kategorier av personopplysninger**

Samtykke (art. 6 nr. 1 bokstav a)

Informasjon for utvalg 1**Informerer du utvalget om behandlingen av opplysningene?**

Ja

Hvordan?

Skriftlig informasjon (papir eller elektronisk)

Tredjepersoner

Skal du behandle personopplysninger om tredjepersoner?

Nei

Dokumentasjon

Hvordan dokumenteres samtykkene?

- Manuelt (papir)

Hvordan kan samtykket trekkes tilbake?

Ønsker informanten å trekke seg kan de sende mail eller ta kontakt via telefonnummer til både veileder eller student. Kontaktinformasjonen står på informasjonsskrivet som informanten får via mail.

Hvordan kan de registrerte få innsyn, rettet eller slettet opplysninger om seg selv?

Ønsker informanten innsyn på sine opplysninger kan de kontakte veileder eller student via mail eller telefonnummer. Informasjonen vil deretter sendes på mail til den gjeldene informanten. Ønsker de å rette det

som står skrevet, kan de kontakte veileder eller student via mail. Blir veileder kontaktet vil veileder gi informasjon til student, studenten vil rette det, sende tilbake revidert informasjon for så at veileder sende den revidert informasjonen til informant. Blir studenten kontaktet vil informasjonen bli rettet og revidert informasjon vil bli sendt tilbake til informant, veileder vil få informasjon om dette. Er det noen opplysninger de ønsker slettet kan de sende mail til student eller veileder. Veileder vil da gi beskjed til student og informasjonen vil bli slettet. Om informanten kontakter studenten vil studenten gå inn i materialet og slette informasjonen og gir beskjed til veileder. Informanten vil også få beskjed om at informasjonen er slettet

Totalt antall registrerte i prosjektet

1-99

Tillatelser

Skal du innhente følgende godkjenninger eller tillatelser for prosjektet?**Behandling**

Hvor behandles opplysningene?

- Private enheter

Hvem behandler/har tilgang til opplysningene?

- Student (studentprosjekt)

Tilgjengeliggjøres opplysningene utenfor EU/EØS til en tredjestat eller internasjonal organisasjon?

Nei

Sikkerhet

Oppbevares personopplysningene atskilt fra øvrige data (kodenøkkel)?

Nei

Begrunn hvorfor personopplysningene oppbevares sammen med de øvrige opplysningene

Fordi de ikke er anonymisert.

Hvilke tekniske og fysiske tiltak sikrer personopplysningene?

- Adgangsbegrensning

Varighet

Prosjektperiode

06.08.2018 - 30.06.2020

Skal data med personopplysninger oppbevares utover prosjektperioden?

Nei, alle data slettes innen prosjektslutt

Vil de registrerte kunne identifiseres (direkte eller indirekte) i oppgave/avhandling/øvrige publikasjoner fra prosjektet?

Ja

Begrunn

Siden fagfeltet "programmering i Kunst og Håndverk på grunnskolen" er så lite, vil det være vanskelig å anonymisere informantene. Informantenes svar vil også få mer troverdighet hvis man vet hvem som har sagt hva, som kan ha mye å si for prosjektets betydning. For prosjektets relevans vil det ha noe å si om en utdannet person innen feltet uttaler seg om problematikken, enn en som ikke har utdanning inne feltet. Siden informantene vil bli nevnt med navn og yrke i avhandlingen kan masterprosjektet bli referert til i vitenskaplige artikler skrevet av undertegnede eller andre.

Tilleggsopplysninger

Vedlegg 2, godkjenning fra NSD

NSD Personvern

08.10.2018 15:13

Det innsendte meldeskjemaet med referansekode 871930 er nå vurdert av NSD.

Følgende vurdering er gitt:

Det er vår vurdering at behandlingen av personopplysninger i prosjektet vil være i samsvar med personvernlovgivningen så fremt den gjennomføres i tråd med det som er dokumentert i meldeskjemaet 8.10.2018 med vedlegg. Behandlingen kan starte.

MELD ENDRINGER

Dersom behandlingen av personopplysninger endrer seg, kan det være nødvendig å melde dette til NSD ved å oppdatere meldeskjemaet. På våre nettsider informerer vi om hvilke endringer som må meldes. Vent på svar før endringer gjennomføres.

TYPE OPPLYSNINGER OG VARIGHET

Prosjektet vil behandle alminnelige kategorier av personopplysninger frem til 30.6.2020.

LOVLIG GRUNNLAG

Prosjektet vil innhente samtykke fra de registrerte til behandlingen av personopplysninger. Vår vurdering er at prosjektet legger opp til et samtykke i samsvar med kravene i art. 4 og 7, ved at det er en frivillig, spesifikk, informert og utvetydig bekreftelse som kan dokumenteres, og som den registrerte kan trekke tilbake. Lovlig grunnlag for behandlingen vil dermed være den registrertes samtykke, jf. personvernforordningen art. 6 nr. 1 bokstav a.

PERSONVERNPRINSIPPER

NSD finner at den planlagte behandlingen av personopplysninger vil følge prinsippene i personvernforordningen om

- lovlighet, rettferdighet og åpenhet (art. 5.1 a), ved at de registrerte får tilfredsstillende informasjon om og samtykker til behandlingen
- formålsbegrensning (art. 5.1 b), ved at personopplysninger samles inn for spesifikke, uttrykkelig angitte og berettigede formål, og ikke behandles til nye, uforenlige formål
- dataminimering (art. 5.1 c), ved at det kun behandles opplysninger som er adekvate, relevante og nødvendige for formålet med prosjektet
- lagringsbegrensning (art. 5.1 e), ved at personopplysningene ikke lagres lengre enn nødvendig for å oppfylle formålet

DE REGISTRERTES RETTIGHETER

De registrerte vil ha følgende rettigheter i prosjektet: åpenhet (art. 12), informasjon (art. 13), innsyn (art. 15), retting (art. 16), sletting (art. 17), begrensning (art. 18), underretning (art. 19), dataportabilitet (art. 20). Rettighetene etter art. 15–20 gjelder så lenge den registrerte er mulig å identifisere i datamaterialet.

NSD vurderer at informasjonen om behandlingen som de registrerte vil motta oppfyller lovens krav til form og innhold, jf. art. 12.1 og art. 13.

Vi minner om at hvis en registrert tar kontakt om sine rettigheter, har behandlingsansvarlig institusjon plikt til å svare innen en måned.

FØLG DIN INSTITUSJONS RETNINGSLINJER

NSD legger til grunn at behandlingen oppfyller kravene i personvernforordningen om riktighet (art. 5.1 d), integritet og konfidensialitet (art. 5.1 f) og sikkerhet (art. 32).

For å forsikre dere om at kravene oppfylles, må dere følge interne retningslinjer og/eller rådføre dere med behandlingsansvarlig institusjon.

OPPFØLGING AV PROSJEKTET

NSD vil følge opp behandlingen ved planlagt avslutning for å avklare om behandlingen av personopplysningene er avsluttet.

Lykke til med prosjektet!

Kontaktperson hos NSD: Lasse André Raa

Tlf. personverntjenester: 55 58 21 17 (tast 1)

Vedlegg 3, Informasjonsskriv

Vil du delta i forskningsprosjektet *”Programmering i Kunst og håndverk”*

Dette er et spørsmål til deg om å delta i et forskningsprosjekt hvor formålet er å finne ut hvorfor programmering bør være en del av Kunst og håndverk i grunnskolen og hva som er relevant innen programmering i Kunst og håndverk. I dette skrevet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

Bakgrunn og formål

Dette forskningsprosjektet gjennomføres i forbindelse med en masteravhandling ved Master i estetiske fag, studieretning i fagdidaktikk: Kunst og design, ved institutt for estetiske fag, OsloMet - Storbyuniversitetet. Formålet er å finne ut om hvorfor programmering bør være en del av faget kunst og håndverk og eventuell hva som kunne ha vært relevant med programmering i faget, både hvilket språk som bør bli benyttet og mulige oppgaver. Min foreløpige problemstilling er:

Hvorfor bør programmering bli en del av kunst og håndverk i grunnskolen, og hva bør bli lært i kunst og håndverk når det kommer til programmering?

Hvorfor får du spørsmål om å delta?

Grunnen til at du får spørsmål om å delta på forskningsprosjektet er fordi det er ønskelig med personer som har kunnskap om programmering og kjennskap til grunnskolen. Det er også ønskelige med personer som er kjent med programmering og har fag som design, kunst eller håndverk. I forskningsprosjektet er det ønskelig med fire informanter med noe ulik bakgrunn men som har kjennskap til problemområdet.

Hva innebærer det for deg å delta?

Hvis du velger å delta i prosjekter, innebærer det et intervjuet som vil ta mellom 40 og 60 minutter. Intervjuet vil bli tatt opp ved hjelp av lydopptak og senere bli omgjort til skriftlig materiale. Dette vil sammen med teori og svar fra flere danne utgangspunkt for en helhetlig masteroppgave. Jeg kommer også til å samle inn personopplysninger (navn, epost, fødselsår, telefonnummer, arbeidssted og utdanning). I selve oppgaven ønsker jeg å bruke navn og yrke, da jeg mener dette vil skape mer troverdighet og bredde rund undersøkelsen. Dette innebærer at du som informant vil gjenkjennes etter publikasjon. Andre personopplysninger som ikke omhandler tematikk tilknyttet oppgaven, vil bli behandlet konfidensielt og vil ikke være tilgjengelig for andre enn meg som student og mine to veiledere. Om du som informant ønsker å lese hvilket material som ønskes brukt i masteroppgaven, kan dette sendes via mail underveis i prosessen.

Spørsmålene i intervjuet vil omhandle den gjeldene problemstillingen, der det vil bli stil spørsmål rundt hvorfor eller eventuelt hvorfor ikke programmering bør være en del av Kunst og håndverk i grunnskolen. Spørsmålene vil også omhandle om hva innen programmering som kan være relevant i Kunst og håndverk.

Det er frivillig å delta

Det er frivillig å delta i prosjektet. Hvis du velger å delta, kan du når som helst trekke samtykke tilbake uten å oppgi noen grunn. Alle opplysninger om deg vil da bli anonymisert. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg. Ønsker du å trekke deg, kan du ta kontakt med student eller veileder via mail eller telefonnummer som står nederst på dette dokumentet.

Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger og hva som skjer med opplysningene dine når vi avslutter forskningsprosjektet?

Alle personopplysninger vil bli behandlet konfidensielt med unntak av navn og yrke. Lydopptaket vil bli oppbevart på egen personlig eksterne harddisk frem til prosjektet avsluttes rundt juni 2020, etter dette vil materialet slettes. Øvrige skriftlige dokumenter i tilknytning til intervjuet som ikke inngår i oppgaven, vil også bli anonymisert og slettet ved prosjektslutt. Den skriftlige masteroppgaven vil etter dette være tilgjengelig for alle som ønsker å lese den.

Dine rettigheter

Så lenge du kan identifiseres i datamaterialet, har du rett til:

- innsyn i hvilke personopplysninger som er registrert om deg,
- å få rettet personopplysninger om deg,
- få slettet personopplysninger om deg,
- få utlevert en kopi av dine personopplysninger (dataportabilitet), og
- å sende klage til personvernombudet eller Datatilsynet om behandlingen av dine personopplysninger.

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag fra OsloMet – Storbyuniversitetet har NSD – Norsk senter for forskningsdata AS vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

Hvor kan jeg finne ut mer?

Hvis du har spørsmål til studien, eller ønsker å benytte deg av dine rettigheter, ta kontakt med:

- Student: Stephanie Hoebeke, [redacted]
Veileder: Peter Haakonsen, [redacted]
- Vårt personvernombud: Ingrid S. Jacobsen, [redacted]
- NSD – Norsk senter for forskningsdata AS, på epost (personverntjenester@nsd.no) eller telefon: 55 58 21 17.

Med vennlig hilsen

Prosjektansvarlig
Peter Haakonsen

Student
Stephanie Hoebeke

Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet programmering i kunst og håndverk og har fått anledning til å stille spørsmål. Jeg samtykker til:

- å delta i intervju
- at opplysninger om meg publiseres slik at jeg kan gjenkjennes med navn, utdanning og yrke – hvis aktuelt

Jeg samtykker til at mine opplysninger behandles frem til prosjektet er avsluttet, ca. Juni 2020.

(Signert av prosjektdeltaker, dato)

Vedlegg 4, intervjuguide

Starter intervjuet med en rask oversikt over hvorfor det blir foretatt.

Mitt navn er Stephanie, og jeg kommer fra OsloMet, jeg tar en master i estetiske fag: fagdidaktikk i kunst og design, og skal dette året skrive masteravhandlingen min. Hensikten med denne undersøkelsen er å finne ut av programmerings rolle i Kunst og håndverk. Som både, naturfag-, matematikk- og kunst og håndverkslærer synes jeg diskusjonen om programmering og grunnskolen er svært interessant og vil gjerne lære mer om det.

Det vi snakker om i dette intervju kan bli brukt i masteravhandlingen min, det er kanskje ikke alt som vil være relevant, men mye. Jeg vil også ha et intervju med 4-5 andre. Din og deres besvarelser vil være grunnlag for undersøkelsen i min master avhandling.

Som du vet og som jeg håper du har fått informasjon om vil du ikke være anonym i avhandlingen. Grunnen for at jeg ønsker det slik, er to grunner, den ene fordi;

- din kunnskap og posisjon vil ha mye å si for oppgavens validitet/relabilitet,
- for den andre er det et fåtall som kan noe om programmering, grunnskolen og kunst og design og det blir dermed vanskelig å anonymisere.

Om du snakker om en tredjepart, vil dette bli anonymisert i avhandlingen.

Jeg kommer også til å ta lydopptak og dette vil bli oppbevart på et trygt sted. Lydopptakene skal gjøres om til skriftlig tekst, og da sender jeg gjerne de tilbake til deg, slik at du eventuell kan gjør om noe eller slette noe, men du må sende en mail å spørre om den skriftlige teksten. Har du noen spørsmål til dette?

Signering av samtykke skjema

Jeg interessert i dine meninger, siden du er en som kan eller har kjennskap til programmering. Alt du sier vil ha en verdi for meg som undersøger, og har kunnskap til enten kunst, design og/eller håndverk.

Informanten, og informantens bakgrunn

1. Hvilken stilling har du nå?
2. Hvilken utdanning har du?

Holdninger til programmering i grunnskolen og faget kunst og håndverk

3. Hva handler programmering om for deg? (hvilken erfaring har du med programmering?)	<ul style="list-style-type: none">- Hvordan bruker personen programmering- Hvordan ser personen på programmering- Hva mener informanten at man bruker programmering til-
4. Hvilket forhold har du til kunst og håndverk?	<ul style="list-style-type: none">- Er personen interessert i kunst, design eller håndverk- Har personen drevet med noen undervisning på feltet- Hva mener de er viktig med faget
5. Hvorfor mener du det er viktig, eventuelt ikke viktig at programmering blir en del av grunnskolen? - Bør programmering være et egent fag, tverrfaglig, eller bli lært i alle fag?	<ul style="list-style-type: none">- Hvordan ser personen generelt på programmering i grunnskolen.- Hvor stor plass bør programmering få- Hvorfor er det viktig at elevene utvikler denne ferdigheten på grunnskolen?- Hvilke kvaliteter utvikler elevene ved å drive med programmering?

	-
6. Mener du at programmering bør bli en del av Kunst og håndverksfaget? - Hvis ja, hvorfor? - Hvis nei, hvorfor?	- Hvorfor mener personen at programmering bør bli en del av kunst og håndverk. - Hvorfor mener personen at programmering ikke bør bli en del av kunst og håndverk.

Så gå vi litt dypere inn i hvorfor informanten mener, eller ikke mener at programmering bør være en del av kunst og håndverk.

7. Hvorfor mener du at det er relevant eller ikke relevant å ha programmering i Kunst og håndverk?	- Hvilke sider av programmering hører hjemme? - Kreativitet? - Håndverk? - Skapelse? - Logisk tekning? - Nytt medium for visuelle uttrykk?
8. Hvilke kvaliteter har kunst og håndverk som kan bidra i programmering?	- Kritisk sans - Kulturelle, samfunnsmessige, politiske kontekst - Formell estetisk sans - Kreativitet - Arbeide med komplekse problemer -
9. Hvilke kvaliteter har programmering som kan bidra i kunst og håndverksfaget?	- Utvikle logisk tenkning? - Algoritmisk tekning? - Digitale ferdigheter? - teknologisk forståelse? - Kritisk tenkning?
10. Om programmering blir den del av faget, ser du noen positive sider for hele faget?	- Mer samfunnsrettet? - Mer fremtidsrettet? -
11. Om programmering blir en del av faget, ser du noen positive sider for programmering?	- Flere jenter som blir interessert? - Generelt flere som blir interessert i det? - Elevene ser andre sider av programmering? - Eventuelt hvilken? -
12. Om programmering blir den del av faget, ser du noen negative sider for hele faget?	- Mindre tid til andre ting? - Mindre fokus på tradisjonelle ting? -
13. Hvilke utfordringer kan det være å implementere programmering i Kunst og håndverk?	- Lærerens kvalifikasjoner - Utstyr -
14. Hva ser du for deg at elevene skal skape ved hjelp av programmering i kunst og håndverk?	- Visuelle uttrykk - Programvare - Roboter - Smarte tekstieler - Hjemmesider - Apper - Generativ kunst - Animasjoner - Spill
** noen av de tingene du har nevnt kan lages ved hjelp av programvarer, hvorfor skal de lære å programmere det?	

15. Bør elevene lære seg å programmere fra bunnen av i kunst og håndverk, eller bruke kunnskap de har lært i andre fag?	<ul style="list-style-type: none"> - Skal det kun bli skapt ting med programmering, eller skal læring av språk også bli en del av kunst og håndverk?
16. Hvilke programmeringsspråk bør elevene lære/bruke i Kunst og håndverk?	<ul style="list-style-type: none"> - Blokkbasert? - Scratch (ser du noen problemer med å bruke dette språket på ungdomskolen?) - Tekstbasert? - Processing? - Html? - Python? - Hvorfor dette språket?
17. Ut fra begrepene: koding, programmering eller algoritmisk tankegang, hvilke begrep synes du passer best inn i faget og hvorfor?	<ul style="list-style-type: none"> - Algoritmisk tankegang litt for omfattende? - Koding, kun kopiering? - Programmering, mangler essensielle deler innen algoritmisk tankegang?
18. hva bør hensikten med programmering være i kunst og håndverk?	<ul style="list-style-type: none"> - Er det noen spesifikke mål? - På fagets premisser?
19. Kan du komme med noen spesifikke mål som elevene skal nå, når det kommer til programmering i kunst og håndverk?	

Avslutning

20. Er det noen vi ikke har snakket om, eller noe som du synes er relevant for temaet, som du ikke har fått snakket om?

Slå av lydopptaker

Takke for intervjuet og oppsummere sammen.


```

linje l1;
linje l2;

void setup () {
    fullScreen ();
    background (255);
    l1 = new linje ();
    l2 = new linje ();
}

void draw () {
    l1.visLinje ();
    l1.bevegLinje ();
    l1.checkEdges ();

    l2.visLinje ();
    l2.bevegLinje ();
    l2.checkEdges ();

    if (keyPressed) {
        if (key == 'n') {
            background (0);
        }
    }
}

class linje {
    float x1 = random (50,
450);
    float y1 = random (50,
450);
    float x2 = random (50,
450);
    float y2 = random (50,
450);
    float xspeed = (random
(0.5, 4));
    float yspeed = (random
(0.5, 4));
    float x2speed =
(random (0.5, 4));
    float y2speed =
(random (0.5, 4));

void visLinje () {
    strokeWeight (1);
    stroke (50, 70);
    line (x1, y1, x2, y2);
}

void visLinjeSvart () {
    strokeWeight (1);
    stroke (0, 50);
    line (x1, y1, x2, y2);
}

void bevegLinje () {
    x1 = x1 + xspeed;
    y1 = y1 + yspeed;
    x2 = x2 + x2speed;
    y2 = y2 + y2speed;
}

void checkEdges () {
    if (x1 > (width-50) ||
x1 < 0 + 50) {
        xspeed = xspeed * -1;
    }
    if (y1 > height-20 ||
y1 < 0 + 20) {
        yspeed = yspeed * -1;
    }
    if (x2 > width-50 || x2
< 0+50) {
        x2speed = x2speed
* -1;
    }
    if (y2 > height-20 ||
y2 < 0+20) {
        y2speed = y2speed
* -1;
    }
}
}

```