

MASTEROPPGAVE

Matematisk argumentasjon og programmering

- Når elever på 7. trinn arbeider i par med Scratch

Mathematical argumentation and programming

- When 7th grade students work in pairs on Scratch

Oda Åbø Pettersen

Master i undervisningsvitenskap med fordypning i matematikk

Fakultet for lærerutdanning, kultur og idrett (FLKI)

Rune Herheim og Andrey Chesnokov

15. mai 2019

Forord

Studietiden på Høgskulen på Vestlandet har vært en utrolig fin tid der jeg har møtt mange flotte mennesker og lært mye. Det å skrive en masteroppgave har vært svært utfordrende, men også veldig spennende og lærerikt. Jeg har gjennom oppgaveskrivingen lært mye som jeg gleder meg til å ta i bruk i arbeidslivet som lærer i grunnskolen.

Det er flere som har hjulpet meg med å skrive masteroppgaven og som jeg vil takke. Først og fremst vil jeg takke veilederne mine Rune Herheim og Andrey Chesnokov, som har hjulpet meg med både små og store problemer. Jeg setter pris på de konkrete og tydelige tilbakemeldingene jeg har fått, og de gode rådene som har gjort oppgaveskrivingen enklere.

Jeg vil også takke elevene som smilende stilte opp på datainnsamlingen og som gjorde sitt beste i å løse oppgavene vi ga dem. Uten dem hadde ikke denne oppgaven vært mulig å gjennomføre.

Takk til familie og venner som har støttet meg gjennom hele studietida. Takk til familien min som har stilt opp med gode og varme ord på tunge dager, og som alltid har hatt troen på meg. En ekstra takk til min mor og søster som i tillegg har hjulpet meg med korrekturlesing.

Til slutt vil jeg også gi en stor takk til alle på lesesalen som har vært en god støtte i hverdagen og som har hjulpet meg gjennom flere problemer med oppgaven. Uten lunsjpausene med dere hadde ikke oppgaveskrivingen vært det samme!

Oda Åbø Pettersen

Bergen, 15. mai 2019

Sammendrag

I 2020 trer en ny læreplan i kraft som inkluderer programmering i matematikkfaget (Kunnskapsdepartementet, 2018a). I den sammenheng har det vært mye debatt på hvordan dette skal gjøres, hvordan programmering skal skrives inn i læreplanene og hvordan lærere kan arbeide med programmering i matematikkundervisningen (Bærland og Gilje, 2017; Flote, 2016; Klovning, 2015; Waage, 2018). Med bakgrunn i den nye læreplanen har jeg i denne oppgaven valgt å forske på følgende problemstilling: *Hvilke kvaliteter ved matematisk argumentasjon kan identifiseres når elever programmerer?*

Studien baserer seg på at matematisk argumentasjon er en sosial prosess som kan være preget av både hverdagslig kommunikasjon og formell bevisføring. For å belyse problemstillingen har det blitt gjort en kvalitativ forskningsstudie der datamateriale av barn som programmerer har blitt analysert og diskutert. Datamaterialet har blitt samlet inn i samarbeid med en annen masterstudent der vi satte oss inn i programmering og matematisk argumentasjon, og utviklet et undervisningsopplegg for datainnsamlingen. Elevene fikk i oppgave å programmere en penn til å tegne geometriske figurer i et programmeringsprogram kalt *Scratch*. Datamaterialet har blitt presentert i form av dialogutdrag og skjermbilder av programmeringen. Med hjelp fra teori om programmering, matematisk argumentasjon og resonnering har kvaliteter ved elevenes matematiske argumentasjon når de programmerer blitt identifisert og diskutert opp mot problemstillingen. I dette arbeidet har særlig arbeidene til Lavy (2006) og Lithner (2000; 2006; 2008) spilt en stor rolle som sentrale analyseverktøy. Begge forskerne skrev i sine artikler om ulike kvaliteter hos elevers argumentasjon, der Lavy forsket på ulike typer argumenter når elever programmerte og Lithner har forsket på ulike former for resonnering.

Analysen og diskusjonen av datamaterialet viser at elever som hadde minst utfordringer i samarbeidet ofte argumenterte matematisk for ulike forslag og løsninger. Programmeringsprogrammet gjorde at arbeidsprosessen ble mer detaljert enn hvis elevene hadde tegnet figurene for hånd. Dette gjorde at det ble skapt flere muligheter for matematisk argumentasjon ettersom det er i arbeidsprosessen disse mulighetene ligger. Studien gir innsikt i hvordan lærere kan arbeide med programmering uten å miste fokuset på matematikk. Samtidig er studien også et bevis på at man kan lage undervisningsopplegg om programmering uten å være ekspert i emnet, og at det, med litt innsats, er mulig for alle lærere å sette seg inn i programmering.

Abstract

In 2020, a new curriculum comes into effect which includes programming in the mathematics subject (Kunnskapsdepartementet, 2018a). Because of this, there has been a great debate on how programming should be written into the curriculum and how teachers can work with programming in mathematics teaching (Bærland og Gilje, 2017; Flote, 2016; Klovning, 2015; Waage, 2018). Based on the new curriculum that comes into effect in 2020, I have chosen to research the following problem: *What qualities of mathematical argumentation can be identified when pupils program?*

This considers mathematical argumentation to be a social process that can be characterized by both everyday communication and formal evidence. To elucidate the issue, a qualitative research study has been made in which data material of children that worked with programming has been analyzed and discussed. The data material has been collected in collaboration with another master's student where we studied programming and mathematical argumentation and developed a teaching program for the data collection. The students were given the task of programming a pen to draw some geometric figures in a programming program called Scratch. The data material has been presented in the form of dialog extracts and screenshots of the computer. With the help from theory on programming, mathematical argumentation and reasoning, qualities of the students' mathematical argumentation have been identified and discussed to answer the problem. In this study, especially the researchers Lavy (2006) and Lithner (2000; 2006; 2008) have played a major role in which they have contributed with analysis tools. Both researchers wrote in their articles on different qualities of student mathematical argumentation, where Lavy researched various types of arguments when pupils programmed and Lithner researched various forms of reasoning.

Analysis and discussion of the data showed that students who had the least challenges in the collaboration often argued mathematically for different proposals and solutions. The programming program made the work process detailed, which opened up to opportunities for mathematical argumentation to be created. Through the process of illuminating the problem, the thesis shows how teachers can work with programming without losing the focus on mathematics. At the same time, the thesis is also a proof that one can create teaching programs for programming without being an expert in the subject, and therefore, with a little effort, it is possible for all teachers to familiarize themselves with programming.

Innholdsfortegnelse

FORORD	II
SAMMENDRAG	III
ABSTRACT	IV
FIGUROVERSIKT	VIII
1. INNLEDNING	1
1.1 BAKGRUNN FOR VALG AV TEMA.....	1
1.2 STUDIENS FORMÅL OG PROBLEMSTILLING.....	2
1.2.1 <i>Kvaliteter</i>	3
1.2.2 <i>Matematisk argumentasjon</i>	4
1.2.3 <i>Programmering</i>	5
1.2.4 <i>Scratch</i>	6
1.3 LÆRINGSPERSPEKTIV.....	7
1.4 DEL AV ET FORSKNINGSPROSJEKT.....	8
1.5 STRUKTUR PÅ OPPGAVEN.....	8
2. TIDLIGERE FORSKNING OG TEORETISK RAMMEVERK	9
2.1 PROGRAMMERING I SKOLEN.....	9
2.1.1 <i>Scratch som læringsverktøy</i>	10
2.1.2 <i>Matematiske muligheter i Scratch</i>	11
2.1.3 <i>Scratch sine rammer</i>	12
2.2 MATEMATISK ARGUMENTASJON I SKOLEN.....	12
2.3 LITHNERS ULIKE FORMER FOR ARGUMENTASJON.....	13
2.3.1 <i>Imitativ argumentasjon</i>	13
2.3.2 <i>Kreativ argumentasjon</i>	14
2.3.3 <i>Lithners tre kvaliteter ved argumentasjon</i>	15
2.4 LAVYS FIRE ARGUMENTASJONSKATEGORIER.....	16
2.4.1 <i>Grunnleggende argumentasjon</i>	16
2.4.2 <i>Sammensatt argumentasjon</i>	16
2.4.3 <i>Utvidet argumentasjon</i>	17
2.4.4 <i>Generelt-presentert-som-spesifikt</i>	17
2.4.5 <i>Modell av Lavys fire argumentasjonsformer</i>	18
2.5 ARGUMENTASJON GJENNOM ULIKE UTTRYKKSFORMER.....	18
2.5.1 <i>Deiktisk språk</i>	19
2.5.2 <i>Argumentasjon gjennom prøving og feiling</i>	19
2.5.3 <i>Eksempelbasert argumentasjon</i>	20

2.5.4 Representasjonsbasert argumentasjonen.....	20
2.5.5 Oppfattet-som-forstått.....	20
3. METODE	22
3.1 VALG AV METODE.....	22
3.2 OBSERVASJON	24
3.2.1 Video- og lydopptak	24
3.2.2 Skjermopptak.....	24
3.2.3 Observatørens rolle.....	25
3.3 UTVALGET	26
3.4 DATAINNSAMLINGSPROESSEN	26
3.4.1 Den opprinnelige planen.....	27
3.4.2 Første møte med elevene.....	27
3.4.3 Scratch-kurs i regi av Kodeklubben Vest.....	27
3.4.4 Det første undervisningsopplegget.....	28
3.4.5 Det endelige undervisningsopplegget.....	30
3.5 BEHANDLING AV DATA	32
3.6 ANALYSEVERKTØY	33
3.7 OPPGAVENS VALIDITET OG RELIABILITET	34
3.8 ETISKE HENSYN	35
4. ANALYSE OG DISKUSJON.....	37
4.1 KNOT OG IDA PROGRAMMERER EN FEMKANT	38
4.1.1 «Er det femkant?»	38
4.1.2 «Vent, hvor mange grader er det i en femkant?».....	40
4.1.3 «Åja, men da kan ikke vi ha 60 grader.»	42
4.2 KNOT OG IDA PROGRAMMERER EN SIRKEL	46
4.2.1 «Det er på grunn av at den vender først 360 grader og så bare går den.».....	46
4.2.2 «Ja, går det an?»	48
4.3 KNOT OG IDA PROGRAMMERER EI STJERNE	51
4.3.1 «Det er ulike vinkler tror jeg.».....	51
4.3.2 «Ja, og så hvor mange grader.»	53
4.4 GJENNOMGÅENDE FUNN FOR HELE ARBEIDSPROESSEN.....	56
4.4.1 Mangel på spørsmål.....	56
4.4.2 Matematisk argumentasjon som prosess.....	57
4.4.3 Argumentasjonskategorier.....	57
4.4.4 Systematisk prøving og feiling	58
4.5 DE ANDRE TRE PARENE.....	59
4.5.1 Gruppe 1, Nils og Anders programmerer en femkant.....	59
4.5.2 Gruppe 3, Silje og Pernille programmerer ei stjerne.....	62
4.5.3 Gruppe 4, Tiril og Line programmerer en sirkel.....	63

5. AVSLUTNING.....	65
5.1 HVA HAR BLITT GJORT?	65
5.1.1 Tidligere forskning og datainnsamling	65
5.1.2 Tidligere forskning, teoretisk rammeverk, analyse og diskusjon.....	66
5.2 OPPGAVENS BESVARELSE PÅ PROBLEMSTILLINGEN.....	66
5.2.1 Gode kvaliteter med elevenes matematiske argumentasjon.....	67
5.2.2 Mindre gode kvaliteter med elevenes matematiske argumentasjon.....	68
5.2.3 Oppgavens aktualitet.....	69
5.3 KRITISK BLIKK.....	70
5.4 VIDERE FORSKNING	72
LITTERATURLISTE.....	73
VEDLEGG 1: OPPGAVEARK TIL DET FØRSTE UNDERVISNINGSSOPPLEGGET	78
VEDLEGG 2: SAMTYKKESKJEMA TIL DE FORESATTE	79
VEDLEGG 3: SAMTYKKESKJEMA TIL ELEVENE.....	82

Figuroversikt

Figur 1. Skjerm bilde av Scratch med begreper.	6
Figur 2. Modell av Lavys fire argumentasjonsformer.....	18
Figur 3. Eksempel av funksjonen «bilde i bilde».	25
Figur 4. Bilde av oppgaven elevene fikk.....	31
Figur 5. Bilde av Knut sin tegning av en femkant.....	38
Figur 6. Eksempel på ytre vinkler.	41
Figur 7. Eksempel på indre vinkler.	41
Figur 8. Skjerm bilde av figuren som ble tegnet.	42
Figur 9. Skjerm bilde av kodelinjen.	42
Figur 10. Skjerm bilde av kodelinje med 80 grader.	43
Figur 11. Skjerm bilde av tegnet figur med 80 grader.	43
Figur 12. Skjerm bilde av tegnet figur med 70 grader.	44
Figur 13. Skjerm bilde av kodelinjen.	46
Figur 14. Skjerm bilde av tegnet figur med vend 360 grader.....	46
Figur 15. Skjerm bilde av kodelinje med gjenta 360 ganger.	48
Figur 16. Bilde av stjerne med navn på de ulike vinklene.	53
Figur 17. Skjerm bilde av kodelinjen til stjerna.	54
Figur 18. Skjerm bilde av stjerna som ble tegnet.....	54
Figur 19. Bilde av koordinatene Anders noterte.	59
Figur 20. Skjerm bilde av kodelinjen.	60
Figur 21. Skjerm bilde av diamanten som ble tegnet.	60
Figur 22. Skjerm bilde av kodelinjen.	62
Figur 23. Skjerm bilde av stjerna som ble tegnet.	62

1. Innledning

Denne oppgaven tar for seg temaene argumentasjon og programmering i matematikk gjennom forskning med barn i skolen. I de følgende delkapitlene presenteres det bakgrunn for hvorfor jeg har valgt å skrive om matematisk argumentasjon og programmering, hva formålet og problemstillingen er for oppgaven, hvilket læringsperspektiv oppgaven baserer seg på, informasjon om det overordnede forskningsprosjektet som denne masteroppgaven er en del av og til slutt struktur på resten av oppgaven.

1.1 Bakgrunn for valg av tema

Både argumentasjon og programmering er felt som skal få et større fokus i matematikkfaget når den nye læreplanen kommer i 2020 (LK20). I en melding fra Kunnskapsdepartementet (2018b) presenteres det kjerneelementer som skal inn i fagene i sammenheng med LK20. Et av kjerneelementene som skal inn i matematikk er «resonnering og argumentasjon» (Kunnskapsdepartementet, 2018b). Her står det blant annet at «Elevene må også lære å utforme sine egne resonnementer både for å løse problemer og for å argumentere for framgangsmåter og løsninger» (s. 15). Denne setningen trekker frem to aspekter ved resonnering, nemlig det å løse problemer i matematikk og argumentasjon i matematikk. Med problemer i matematikk, menes det at øving på resonnering kan bidra til å hjelpe elever med å løse matematiske problemer. Den andre delen av setningen tar for seg det faktumet at gjennom å ha tydelige argumenter for ulike framgangsmåter og løsninger, blir man mer bevisst over de matematiske aspektene i metoden man brukte for å komme frem til en løsning.

IKT har fått et økende fokus i skolen de siste tiårene blant annet ved at digital kompetanse ble én av de grunnleggende ferdighetene elevene skal lære seg gjennom grunnskolen (Utdanningsdirektoratet, 2013). Til tross for dette økende fokuset påpeker en rapport fra Norges offentlige utredninger (NOU 2013: 2, 2013) at «Manglende oppmerksomhet på grunnleggende programmerings- og utviklingsferdigheter gjør at vi utdanner barn og unge til å bruke digitale verktøy, men det etterlater en oppvoksende generasjon konsumenter» (s. 10). Det fremheves i rapporten at i den fremtidige skolen er det ikke nok å arbeide med elevenes brukerkompetanse om digitale hjelpemidler, men at det også må arbeides med å utvikle forståelse og kompetanse om hvordan teknologi fungerer og utvikles. Andre land har allerede tatt valget om å gjøre programmering til en mer utbredt kompetanse blant innbyggerne ved å innføre egne

undervisningsprogram om «computer science» (Grover & Pea, 2013. s. 40). I tillegg har begrepet *computational thinking* blitt introdusert for å beskrive tankeprosessen som er involvert i å formulere problemer og deres løsninger slik at løsningene er representert i en form som kan bli utført av en datamaskin (Wing, referert i Grove & Pea, 2013, s. 39). Computational thinking blir beskrevet som en viktig kompetanse ettersom personer med denne kompetansen vil være bedre rustet til å ta del i en fremtid som er sterkt preget av teknologi (Grove & Pea, 2013). Denne formen for tenking vil også i Norge være en viktig kompetanse i fremtiden. Det har nå blitt bestemt at Norge, slik som mange andre land allerede har gjort, skal direkte programmering i den nye læreplanen som vil tre i kraft i 2020. Det har også blitt bestemt at for å gjøre dette skal programmering bli en del av matematikkfaget (Utdanningsdirektoratet, 2019a).

I rapporten fra Norges offentlige utredninger (NOU 2013: 2, 2013) skrives det også at skolen har «lagt for stor vekt på verktøykompetanse, kommunikasjon i digitale medier og for lite vekt på den kritiske refleksjonen rundt teknologisamfunnet og på det å skape teknologi» (s. 99). Det vil dermed i fremtiden bli et større behov for lærere som har kunnskap om resonnering og argumentasjon og hvordan man kan implementere dette i undervisning om både matematikk og IKT.

1.2 Studiens formål og problemstilling

Et av studiens formål er å vise hvordan man kan arbeide med matematisk argumentasjon når man arbeider med matematikk i skolen. Matematisk argumentasjon er et tema som lærere bør ha fokus på gjennom hele utdanningsløpet til elevene, der elevene regelmessig blir utfordret på å ha gode argumenter og forklaringer på de ulike matematiske handlingene de gjør. For å få til dette er det viktig at lærere skaper situasjoner der det er mulig for elevene å øve seg på å argumentere matematisk for valgene eller påstandene sine. Ved å forske på matematisk argumentasjon og programmering sammen, viser jeg én måte man kan arbeide med matematisk argumentasjon i klasserommet. Oppgavens fokus på matematisk argumentasjon har også som formål å skape mer innsikt om ulike kvaliteter ved elevenes matematiske argumentasjon slik at lærere kan gjenkjenne disse kvalitetene.

Når man nå går mot en fremtid der programmering skal inn i matematikktimene, tror jeg at det sitter mange lærere der ute som tenker at dette er noe de ikke kommer til å klare. Det er mulig at det er flere matematikklærere som ikke har tidligere kunnskap om programmering og kan av

den grunn se på det nye tilskuddet i matematikkfaget som en stor utfordring (Forsström & Kaufmann, 2018, s. 19). Jeg vil derfor med denne oppgaven bidra med innsikt i hvordan en kan implementere programmering i matematikktimene uten å miste fokuset på matematikk. Som lærerstudent som selv ikke hadde noen kunnskaper om programmering, vil jeg vise at med å bruke tid på å sette seg inn i de digitale verktøyene er det mulig for alle lærere å ha en matematikkundervisning som omhandler programmeringsferdigheter uten at en trenger å være ekspert i programmering. Jeg har også som mål at programmering og matematikk skal gå hånd i hånd, og at det ikke skal være slik at elevene mister fokuset på matematikk når programmering brukes i undervisningen.

For å få innsikt i programmering i matematikkundervisning, der det er et tydelig matematikkfaglig fokus, har jeg valgt å kombinere et fokus på matematisk argumentasjon og programmering i matematikkundervisning. Overordnet problemstilling i oppgaven blir da:

Hvilke kvaliteter ved matematisk argumentasjon kan identifiseres når elever programmerer?

Ut ifra problemstillingen rettes det overordnede fokuset i denne oppgaven mot potensielle sammenhenger mellom matematisk argumentasjon og programmering. Det innebærer et særlig fokus på å identifisere matematisk argumentasjon og kjennetegn på dette, når elever programmerer. Det handler om når matematisk argumentasjon kan identifiseres, hvordan elevene argumenterer om matematikk mens de programmerer, samt undersøke eventuelle årsaker til resultatene.

1.2.1 Kvaliteter

Oppgaven skal, gjennom analyse av elever som arbeider i par med å programmere i et program som heter Scratch, belyse hvilke kvaliteter ved matematisk argumentasjon elevene tar i bruk for å løse oppgavene. Med «kvaliteter ved matematisk argumentasjon» henvises det til ulike former og aspekter med argumentasjonen. Blant annet hvordan argumentene er formulert, hvilke matematiske fakta argumentene støtter seg på, i hvilke sammenhenger elevene velger å argumentere for påstandene sine og hvilke muligheter for matematisk argumentasjon som kan identifiseres. Det har også blitt gjort en analyse av hvorfor elevene eventuelt ikke bruker matematisk argumentasjon i arbeidsprosessen, og en analyse av hvordan elevene påvirker hverandres argumentasjon.

1.2.2 Matematisk argumentasjon

Ettersom jeg i oppgaven har sett på ulike kvaliteter ved matematiske argumentasjonen, har argumentasjonsteori hatt en sentral del i oppgaven. Som grunnperspektiv brukes Toulmin (2003) sin teori om argumentasjon. Siden Toulmin sin teori omhandler argumentasjon uten matematikk, har jeg også brukt Krummheuer sitt perspektiv på matematisk argumentasjon. Han bygger videre på Toulmin sin teori ved å se på argumentasjon som omhandler matematikk og som oppstår når individer er i en matematisk diskurs.

Krummheuer og Toulmin har begge et sosialt syn på argumentasjon der «cooperating individuals tried to adjust their intentions and interpretations by verbally presenting the rationale of their actions» (Krummheuer, 1995, s. 229). Argumentasjon oppstår dermed når mennesker som samhandler prøver å forklare og gjøre greie for hvorfor deres handlinger er rasjonelle ovenfor samtalepartnerne. I tillegg til dette er også argumentasjon noe som skjer i den hverdagslige interaksjonen sammen med andre mennesker, og ikke bare når man selv er bevisst over at man argumenterer for noe. I denne oppgaven er matematisk argumentasjon sett på som all argumentasjon som omhandler matematikk, enten om det er konkret bevisføring eller ubevisst hverdagsprat. Siden oppgaven handler om matematisk argumentasjon i en undervisningskontekst vil argumentasjonsbegrepet her, i likhet med Lavy (2006, s. 157), bli utvidet til å inkludere interaksjoner i et læringsmiljø som har som intensjon å tydeliggjøre begrunnelser som støtter opp om en løsning eller en konklusjon. Oppgaven baserer seg dermed på en vid oppfattelse av begrepet argumentasjon og argumenter, og vil inkludere ytringer som med første øyekast gjerne ikke virker argumenterende.

Videre i oppgaven har ikke Toulmin eller Krummheuer blitt brukt som analyseverktøy, men teoriene deres om argumentasjon står som en overordnet definisjon av matematisk argumentasjon. I analysen har Lavy (2006) og Lithner (2000; 2006; 2008) sine forskningsartikler blitt brukt for å hente ut informasjon om elevenes matematiske argumentasjon mens de programmerer i Scratch. Lavy (2006) bygger på Toulmin sin argumentasjonsteori og brukte Toulmin-modellen som analyseverktøy for å komme frem til argumentasjonskategorier. I min oppgave bygger jeg videre på Lavy sine kategorier og bruker dem i analysen. Lithner (2000; 2006; 2008) skriver i flere artikler om *reasoning* (resonnering), men bruker også begrepet argumentasjon inn imellom som en del av resonneringen. I artikkelen *Mathematical reasoning in task solving* (Lithner, 2000) definerer han resonnering som ulike tankeganger som blir brukt for å komme frem til en påstand eller konklusjon. Videre forklarer

han at argumentasjon er sannsynliggjøring av resonnementene, «the part of the reasoning that aims at convincing oneself, or someone else, that the reasoning is appropriate» (Lithner, 2000, s. 166). Argumentasjon er dermed, i følge Lithner, en del av resonneringen. Jeg tolker Lithner som at resonnering og argumentasjon består av mye av det samme, der resonnering er noe du gjør for deg selv for å komme frem til en løsning mens argumentasjon handler om å forklare hvorfor resonnementene dine gir mening både for deg selv og for andre. Ettersom Lithner ser på resonnering og argumentasjon som to handlinger som har mye med hverandre å gjøre, har jeg brukt hans forskning på resonnering til å forske på matematisk argumentasjon der jeg er bevisst over denne forskjellen mellom resonnering og argumentasjon.

Selv om Lithner (2008) sin forskning ikke baserer seg på Toulmin sin teori om argumentasjon, har hans definisjon av resonnering flere likhetstrekk med Toulmin. Begge (Lithner, 2008; Toulmin, 2003) ser på resonnering og argumentasjon som en sosial prosess og som et produkt av disse prosessene. I matematikkens verden har Lithner (2008) og Krummheuer (1995) et syn på resonnering og argumentasjon som rommer mer enn et klassisk bevis. Begge trekker frem at resonneringen og argumentasjonen ikke trenger å være basert på ren logikk og kan til og med være faglig feil så lenge det er en form for grunngiving bak påstandene. Disse likhetene mellom Lithner, Toulmin og Krummheuer gjør at de alle passer inn under min definisjon av matematisk argumentasjon.

1.2.3 Programmering

Forström og Kaufmann (2018) skriver at programmering er «the process related to the development and implementation of instructions for computer programs so the computer can perform specific task, solve problems, and support human interactions» (s. 2). Programmering er altså en prosess som beskriver flere handlinger som er rettet mot å utvikle og vedlikeholde programmer som gir datamaskinen instruksjoner om hva den skal gjøre.

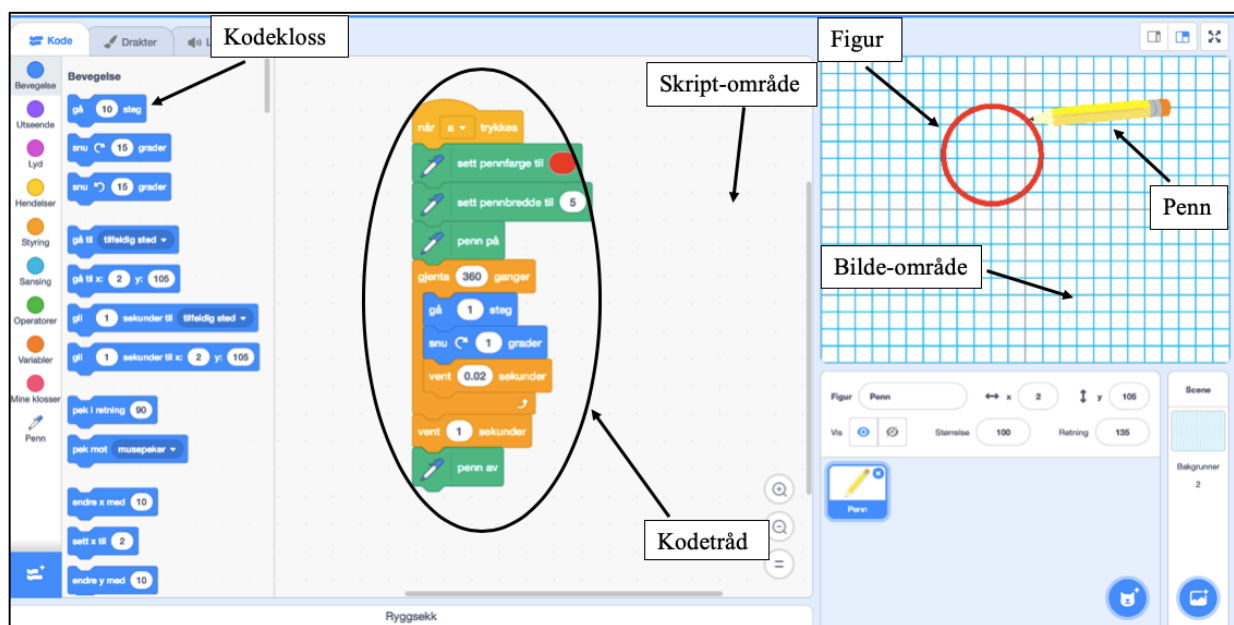
Innenfor programmering er det noe som heter programmeringsspråk. Dette definerer *Store norske leksikon* (Programmeringsspråk, 2017) som en gruppe ulike konstruerte språk man bruker for å gi instruksjoner til datamaskiner og programmer. Noe av det som skiller programmeringsspråk fra språkene mennesker bruker for å kommunisere med hverandre, er at det ikke er rom for tvetydig tolkning da mottakeren (datamaskinen) ikke er i stand til å tolke et slikt innhold. Dette gjør at språkets oppbygning må være entydig og den minste skrivefeil kan

gjøre at dataprogrammet ikke fungerer slik det skal. Programmeringsspråkene er også ofte inspirert fra matematiske og logiske tankeganger.

1.2.4 Scratch

Oppgaven skal som tidligere nevnt, belyse fagområdene matematisk argumentasjon og programmering i matematikk. For å forske på dette ble det valgt et programmeringsprogram som heter Scratch. Scratch finnes gratis på internett (<https://scratch.mit.edu/>), og er et prosjekt av Scratch Foundation utviklet i samarbeid med Lifelong Kindergarten Group ved MIT Media Lab. Nettsiden går ut på at man, ved å bruke Scratch sitt eget programmeringsspråk, programmerer en avatar til å utføre ulike handlinger ved å bygge klosser med ulike kommandoer sammen (se figur 1).

Figur 1. Skjermbilde av Scratch med begreper.



Med *kodekloss* henvises det her til de klossene som fungerer som en erstatning for avanserte kodespråk. En kodekloss inneholder en kommando, og ved å bygge flere kodeklosser sammen får man det som heter *kodetråd*. Det er kodetråden som er det endelige produktet, og som inneholder alle beskjedene datamaskinen skal ta imot. Kodeklossene i Scratch er sortert etter ulike kategorier som bevegelse, utseende, lyd, hendelser, styring mm., der hver kategori har sin farge. For eksempel er alle kodeklossene som inneholder en kommando om bevegelse fargen blå.

Skript-område er den plassen i Scratch der du plasserer kodeklossene og til slutt ender opp med en eller flere kodetråder. *Bilde-område* er det stedet der du kan se resultatet av kodetrådene dine, hvor du har en avatar som du kan programmere til å gjøre ulike ting. I dette prosjektet ble det satt inn en avatar som så ut som en penn. For å gjøre det enklere å forstå hva elevene snakker om i analysen, har jeg valgt å kalle avataren for *penn*. I tillegg har jeg også gitt navnet *figur* som betegnelse på det elevene får pennen til å spore eller tegne når den blir kodet til å bevege seg i bestemte retninger.

1.3 Læringsperspektiv

For å kort posisjonere oppgaven vil jeg presisere at arbeidet med denne masteroppgaven er basert på et sosiokulturelt læringsperspektiv, og bygger på det Barab og Squire (2004) skriver om kognisjon:

A fundamental assumption of many learning scientists is that cognition is not a thing located within the individual thinker but is a process that is distributed across the knower, the environment in which knowing occurs, and the activity in which the learner participates. In other words, learning, cognition, knowing, and context are irreducibly co-constituted and cannot be treated as isolated entities or processes.

(Barab & Squire, 2004, s. 1)

Læring sees i denne oppgaven som noe som er distribuert mellom mennesker, læringsmiljø og læringsaktiviteter. Ved å basere oppgaven på et slikt perspektiv på læring og kognisjon vil både elevene, læringsmiljøet og programmeringsaktiviteten påvirke resultatene. Det vil derfor i analysen være viktig å se på både hva elevene sier og gjør, hvordan læringsmiljøet er og hvilken innflytelse programmeringsoppgavene har på elevenes læring. Ved å bruke Scratch som et læringsverktøy, der jeg med oppgaven prøver å trekke frem det matematiske med programmet ved å oppfordre til samarbeid og utforskning, har Scratch blitt plassert i en sosiokulturell kontekst.

1.4 Del av et forskningsprosjekt

Denne masteroppgaven er en del av forskningsprosjektet *Learning about teaching argumentation for critical mathematics education in multilingual classrooms* (LATACME), som har som overordnet fokus å belyse aspekter ved argumentasjon og kritisk matematikdidaktikk i flerspråklige klasserom. LATACME er videre delt inn i ulike undergrupper, hvor en av gruppene fokuserer på IKT og argumentasjon. Det er denne undergruppen masteroppgaven hører til.

1.5 Struktur på oppgaven

Kapittel 2 presenterer tidligere forskning og teoretisk rammeverk som tar for seg programmering og matematisk argumentasjon i skolen. I delkapitlene om programmering har det blitt lagt et særlig fokus på Scratch ettersom det er dette programmeringsverktøyet som er brukt under datainnsamlingen. Det presenteres tidligere forskning om Scratch som læringsverktøy, Scratch sine matematiske muligheter og hvordan Scratch skaper rammer for undervisningen. Videre i delkapitlene om matematisk argumentasjon presenteres det tidligere forskning og teoretisk rammeverk om hvorfor matematisk argumentasjon er et viktig tema i skolen, hvordan tidligere forskning har kategorisert og definert ulike former for matematisk argumentasjon og til slutt ulike måter argumentasjon kan bli uttrykt på hos barn.

Kapittel 3 er et metodekapittel hvor forskningsmetode og datainnsamlingsprosessen blir introdusert. I dette kapitlet er det gjort rede for valg av metode, informasjon om denne metoden, datainnsamlingsprosessen og de valgene som ble gjort underveis, hvordan datamaterialet har blitt analysert, oppgavens validitet og reliabilitet og etiske hensyn.

I kapittel 4 presenteres deler av datamaterialet kronologisk i forhold til elevenes arbeidsprosess. Datamaterialet vises som dialogutsnitt hvor analyse og diskusjon kommer direkte etter presentasjon av dialogutsnittene. Analysen og diskusjonen av datamaterialet utdypes og nyanseres ved å diskutere materialet opp mot teorien og forskningen presentert i kapittel 2.

Kapittel 5 fungerer som en avslutning på oppgaven. Her er det gjort rede for hva som har blitt gjort i oppgaven og hvordan dette har skapt et grunnlag for å besvare problemstillingen. Videre blir svar på problemstillingen løftet frem og diskutert, oppgavens begrensninger blir presentert og til slutt blir det skissert et forslag til videre forskning.

2. Tidligere forskning og teoretisk rammeverk

I dette kapittelet presenteres tidligere forskning og de ulike teoretiske rammeverkene jeg har brukt i min analyse av datamaterialet og som har hatt en innflytelse på mine refleksjoner av funnene fra analysen opp mot problemstillingen. All form for teori som blir presentert her har som hensikt å bidra til å belyse spørsmålet om hvilke kvaliteter med matematisk argumentasjon som kan identifiseres når elever programmerer i Scratch. Først har jeg gått nærmere inn på tidligere forskning på programmering i skolen og lagt et ekstra fokus på hva andre forskere har sagt om det å bruke Scratch i skolen. Deretter presenteres teori om ulike former for matematisk argumentasjon som har spilt en stor rolle i analysen av datamaterialet.

2.1 Programmering i skolen

I en litteratur gjennomgang (Forsström og Kaufmann, 2018) om programmering i matematikkutdanninga kom det frem at programmering kan øke elevers motivasjon til å lære matematikk og samtidig også forbedre elevenes prestasjoner i faget. Ut ifra Forsström og Kaufmann (2018) sin definisjon av programmering nevnt i 1.2.3, skriver de videre at for å programmere trenger man:

- Kunnskap om programmeringsspråkene.
- Ekspertise i fagene relatert til utviklingen av spesialiserte algoritmer og logikk.
- Evnen til å analysere, forstå og løse problemer ved å verifisere algoritmiske krav og vurdere korrektheten og implementeringen (koding) av algoritmen i et bestemt programmeringsspråk.

Disse tre kompetansene er tankeprosesser som har blitt koblet sammen med matematisk tenkning og da særlig ferdighetene problemløsning, kreativitet og logisk tenkning (Forsström & Kaufmann, 2018, s. 19). Dette er noe Utdanningsdirektoratet (2019a) støtter opp om når de begrunner valget om å ta programmering inn i matematikkfaget med at «Når elevene bruker programmering til å utforske og løse problemer, kan det være et godt verktøy for å utvikle matematisk forståelse». Ved å jobbe med å utvikle elevenes programmeringsferdigheter, kan en derfor også samtidig arbeide med viktige matematiske ferdigheter.

Et annet moment som knytter programmering og matematikk sammen, er algoritmisk tenkning. I utkastet til den nye læreplanen (Utdanningsdirektoratet, 2019b) dukker dette begrepet opp i et av kjerneelementene og går igjen i flere av kompetansemålene:

4. trinn – lage algoritmar og uttrykkje dei ved bruk av variablar, vilkår og lykkjer.
5. trinn – lage algoritmar med bruk av variablar, vilkår og lykkjer og programmere desse.
8. trinn – utforske korleis algoritmar kan skapast, testast og forbeholdt ved hjelp av programmering.

(Utdanningsdirektoratet, 2019b)

Med disse kompetansemålene skjer det en utvikling hvor en gradvis skal kunne lage og utforske algoritmer gjennom programmering. De som har vært med å utvikle kjerneelementene i matematikk (Kjerneelementgruppen for matematikk, 2017), beskriver algoritmisk tenkning og programmering som to emner som går tett sammen. De skriver at det er viktig for elevene å forstå ulike algoritmer i matematikk, og at datamaskiner kan raskere og mer presist utføre flere algoritmer enn mennesker. Videre poengterer de at «For å gi liv til algoritmisk tenkning er det derfor viktig at det kombineres med læring av programmering» (Kjerneelementgruppen for matematikk, 2017). En måte å arbeide med programmering i skolen, kan derfor være å slå sammen algoritmisk tenkning med programmering. Dette viser også til at programmering er mer enn å bare trykke noen koder inn på en datamaskin, men at det også handler om større tankeprosesser som krever ulike matematiske kunnskaper.

2.1.1 Scratch som læringsverktøy

Ved å ha et programmeringsspråk som er basert på blokker med ulike kommandoer, har Scratch erstattet vanskelige koder med bilder av ulike blokker. Grover og Pea (2013) skriver at «Graphical programming environments are relatively easy to use and allow early experiences to focus on designing and creating, avoiding issues of programming syntax» (s. 40). Dette gjør Scratch til et godt læringsverktøy for nybegynnere innenfor programmering, ettersom de får mulighet til å øve seg på logiske ferdigheter som er en stor del av programmeringen uten å måtte sette seg inn i et krevende programmeringsspråk.

Gjennom Scratch sitt brede spekter av ulike kodeklosser kan elever sette sammen, ta fra hverandre og endre på kodeklossene slik som de ønsker. Med denne muligheten i Scratch

fremhever Taylor, Harlow og Forret (2010) at elevene får mulighet til å lære «mathematical and computational concepts that support the development of creative and systematic thinking» (s. 562). Det at Scratch består av et rikt utvalg av kodeklosser, bakgrunner og avatarer som man enkelt kan flytte og bytte på, er dermed noe som kan bidra til å stimulere kreativitet hos elevene ettersom det er lite som begrenser dem. Samtidig er Scratch med på å utfordre elevene til å være strukturerte i arbeidet sitt slik at de riktige kodeklossene havner på riktig sted i kodetråden.

2.1.2. Matematiske muligheter i Scratch

Flere forskere (Calder & Taylor, 2010; Calao, Moreno-Léon, Correa & Robles, 2015; Taylor et al., 2010) skriver om det å bruke Scratch for å utforske de matematiske mulighetene i programmet. Til tross for at Scratch ikke er utviklet for å fremme matematiske tankeganger, skriver Calder og Taylor (2010) at barna i deres prosjekt likevel var engasjert i matematiske idéer. Videre skriver de at «the process the participants undertook more directly facilitated mathematical thinking through the creative problem-solving process it evoked, and the development of logic and reasoning as they responded to the various forms of feedback» (s. 123). Med dette kommer det frem at matematisk tenkning viste seg hos elevene ved at de var i en prosess der de var kreative problemløsere, og at de ulike formene for tilbakemelding fra Scratch ga dem grunnlag for å utvikle logikk og resonnering når de svarte på tilbakemeldingene fra datamaskinen.

I forskningsprosjektet sitt oppdaget Calder og Taylor (2010) også hvordan elevene arbeidet med matematiske emner innenfor geometri når de skulle kode i Scratch. Det å kode ulike bevegelser, vinkler og koordinater, og koble disse handlingene sammen til en momentan handling hos en avatar, skriver Calder og Taylor (2010) åpnet opp for å øke forståelse av disse matematiske emnene.

Calao et al. (2015) utforsket i sitt forskningsprosjekt hvilket utbytte elevene hadde i utvikling av algoritmisk tenkning ved å arbeide i Scratch. Resultatene viste at testgruppa hadde gjennom forskningsperioden forbedret sine matematiske kunnskaper mer enn kontrollgruppa som ikke hadde arbeidet i Scratch. I tillegg fant de ut at utviklingen av algoritmisk tenkning i Scratch bidro til å øke elevenes prestasjoner i matematisk modellering, resonnering, problemløsning og utførelse av prosedyrer og algoritmer.

Det er altså flere forsknings-studier som støtter opp om at elever kan arbeide med matematikk mens de programmerer i Scratch. Dette åpner opp for å forske på kvaliteter ved matematisk argumentasjon blant elever som programmerer i Scratch.

2.1.3 Scratch sine rammer

Innenfor det digitale miljøet, er det de tilgjengelige teknologiske produktene og programmene som skaper rammene for hvilke former for læring som er mulig (Calder & Taylor, 2010). Når man bruker Scratch som digitalt læringsverktøy avgrensner og muliggjør programmet elevenes læringsmuligheter. Som lærer er dette noe man må ta hensyn til når man velger å bruke et digitalt verktøy, der en må vurdere om verktøyet åpner opp for de formene for læring som man er ute etter i den tenkte undervisningen. Scratch vil dermed ikke alltid være hensiktsmessig å bruke. Denne oppgaven ser på om Scratch er aktuell som læringsverktøy i matematikkundervisning om programmering og matematisk argumentasjon.

2.2 Matematisk argumentasjon i skolen

Som beskrevet i 1.2.2 om matematisk argumentasjon, er argumentasjon noe som oppstår i den hverdagslige interaksjonen mellom mennesker. Kuhn (1992) skriver om hvor viktig argumenterende tenkning er for mennesker. Det å tenke på ulike argumenter påvirker valgene vi tar, verdiene vi holder og konklusjonene vi kommer frem til. Argumentasjon tar del i hver eneste viktige avgjørelse mennesker gjør, og det er derfor viktig at skolen legger til rette for å utvikle og øve på slik tenkning. Videre skriver Kuhn (1992) at argumentative dialoger fremhever resonnementer som er argumenterende, og gir mulighet for kontrasterende ideer og praksiser som kan legge til rette for utviklingen av argumenterende tenkning. For at skolen skal legge til rette for en slik utvikling, er det derfor viktig at elevene har mulighet til å delta i dialoger som åpner opp for argumentasjon.

Når det kommer til hvorfor argumentasjon er en viktig del av matematikkundervisningen trekker Jonassen og Kim (2010) frem at argumenterende læringsaktiviteter «promotes productive ways of thinking, conceptual change and problem solving» (s. 439). Ved å fokusere på matematisk argumentasjon kan en med andre ord få elevene til å utforske og oppdage nye aspekter med matematikken. De kan bli bedre til å se nye løsninger og hvordan effektivisere tenkningen sin.

2.3 Lithners ulike former for argumentasjon

Som tidligere nevnt spiller Lithner sin forskning på resonnering en stor rolle i denne oppgaven. Han har gjennom flere artikler (Lithner, 2000; Lithner, 2006; Lithner, 2008) diskutert ulike aspekter og kvaliteter ved matematisk resonnering i skolen. Blant annet har han hatt et særlig fokus på hvordan elever resonnerer når de arbeider med matematikkoppgaver. Her skiller han blant annet mellom imitativ og kreativ resonnering, der han mener at kreativ resonnering er det som er ønsket. Disse formene som han beskriver er noe jeg kommer til å se etter i analysen.

2.3.1 Imitativ argumentasjon

Lithner (2008) skriver om viktigheten av å kunne resonnerer og argumentere for ulike løsninger som en måte å forhindre at elevene lærer gjennom pugging. Han mener at det er et problem at mange elever memorerer ulike løsningsmetoder uten å reflektere over hvorfor en slik løsning er best. En slik form for løsningsmetode kalles for *imitative reasoning* (Lithner, 2008, s. 258), og går ut på at man imiterer en løsning eller et svar fra en tidligere liknende oppgave. Et typisk eksempel på dette er når elever løser flere liknende oppgaver etter hverandre, der den samme løsningsstrategien blir brukt bare med andre tall. En slik måte å lære på kan føre til at elevene ikke får et helhetlig bilde av matematikken, og kan ved senere tidspunkt få problemer med å anvende matematikken til oppgaver som ikke har en klar løsningsmetode. Videre i oppgaven vil imitative reasoning bli omtalt med den norske oversettelsen *imitativ argumentasjon*. Selv om ordet *reasoning* direkte oversatt betyr resonnering, velger jeg som forklart i delkapittel 1.2.2 å bruke begrepet argumentasjon ettersom Lithner sin forskning på reasoning vil bli brukt til å analysere argumentasjon.

Lithner (2008) deler imitativ argumentasjon inn i de to kategoriene *memorised reasoning* (oversatt til memorert argumentasjon) og *algorithmic reasoning* (oversatt til algoritmisk argumentasjon). Her blir begrepet memorert argumentasjon introdusert for å beskrive hvordan elever argumenterer seg frem til et svar ved å huske et likende svar tidligere. Med det andre begrepet, algoritmisk argumentasjon, henviser han til argumentasjon som går ut på at elever identifiserer noen kjennetegn ved oppgaven og anvender en standard algoritme for å løse oppgaven. Lithner (2008, s 259) skriver at ulempen med disse to formene for argumentasjon er at de delene som er vanskelige og oppfordrer til utforskning av matematikk forsvinner ved at elevene gjenkjenner fellestrekk som enten direkte gir dem svaret, eller som ved en standard algoritme løser oppgaven uten videre tenking.

2.3.2 Kreativ argumentasjon

Som det motsatte av imitativ argumentasjon, trekker Lithner (2008) inn *creative reasoning* (oversatt til kreativ argumentasjon). Dette er en form for argumentasjon som oppfordrer til kreativ tenkning, og som ikke består av tidligere kunnskap kun tilegnet gjennom pugging. For å kunne definere kreativ argumentasjon gir Lithner (2008, s. 266) de tre kravene:

1. Nyhet. En ny (for den som argumenterer) argumentasjonssekvens blir skapt, eller en glemt sekvens blir skapt på nytt.
2. Plausibilitet. Det er argumenter som støtter valg av strategi og/eller hvordan utførelse av strategien kan støtte opp om hvorfor konklusjonen er sann eller plausibel.
3. Matematisk grunnlag. Argumentene er forankret i matematiske egenskaper i de komponentene som er involvert i argumentasjonen.

Med *nyhet* referer Lithner (2006) til det å være kreativ i argumentasjonsprosessen. Det å være kreativ i matematikken kan bety flere ting, men Lithner (2006) definerer dette som noe enhver ordinær elev kan være. Kreativitet i matematikk trenger derfor ikke være noe bare genier kan oppnå med sine store matematiske ideer. Så lenge argumentasjonen går videre fra det å følge strenge algoritmiske veier eller å gjenkjenne tidligere ideer skapt av andre, er den kreativ og ny.

For å skille mellom generell kreativ argumentasjon og kreativ *matematisk* argumentasjon har Lithner lagt til de to kravene, plausibilitet og matematisk grunnlag. I den didaktiske kontrakten ligger det at elevene har lov til å gjette, ta sjanser og bruke ideer og argumenter som ikke er fullstendig grunnet. Lithner (2006) skriver at skolen tillater, og gjerne også oppmuntrer, elevene til å bruke former av matematisk argumentasjon som har reduserte krav til logikk. Ettersom Lithner inkluderer argumentasjon som handler om mer enn bare bevis i matematikk, stiller han også lavere krav til plausibiliteten hos argumentene. Pólya (referert i Lithner, 2006) beskriver plausibel argumentasjon som det å skille «a guess from a guess, a more reasonable guess from a less reasonable guess» (s. 8). Med dette menes det at argumentasjonen ikke trenger å være fullstendig sikker i sin konklusjon, men må være basert på argumenter som støtter opp om hvorfor en påstand er mer rimelig enn en annen. Når det kommer til matematisk forankring

menes det ikke bare at påstanden eller konklusjonen handler om matematikk, men at selve argumentasjonsprosessen og argumentene er forankret i relevante matematiske egenskaper.

2.3.3 Lithners tre kvaliteter ved argumentasjon

For å kunne vurdere elevenes argumentasjon skriver Lithner (2008) også at faktorene validitet, evnen til å overtale og konstruktiviteten er med på å avgjøre hvor bra argumentasjonen er. Lithner (2008) sin forskning trekker frem det faktumet at all matematisk resonnering og argumentasjon ikke nødvendigvis trenger å være bra for elevenes læring. Som lærer bør en hele tiden være kritisk og vurdere hvilke former for argumentasjon som fremmer elevenes læring og forståelse av matematikk. For å gjøre dette må en vurdere hvor vidt elevens argumentasjon er riktig i forhold til matematikken (validitet), hvordan elevene presenterer argumentene sine (evnen til å overtale) og til slutt om argumentene baserer seg på konstruktiv fakta som bidrar positivt til argumentet (konstruktivitet).

Selv om validiteten til et argument avgjøres ut ifra matematikken som tas i bruk, er det også en sosial prosess som blir påvirket av sosiomatematiske normer. Med dette menes det at hva som blir godtatt som en god nok matematisk forklaring blir påvirket av de normene som finnes i den matematiske diskursen man tar del i. Et eksempel på dette er et matematikklasserom hvor læreren ofte har mer makt enn elevene til å bestemme hva som er et godt nok argument, samtidig som elevene også lettere godtar lærerens egne argumenter. Slike skjeve maktfordelinger kan man også finne blant elevene, der elever som sjeldent presterer høyt i matematikk kan lettere godta påstander fra høyere presterende elever. Det er derfor viktig å være oppmerksom på denne delen av matematisk argumentasjon, og gjøre elevene bevisst over at det er de matematiske argumentene som bør avgjøre om en velger å tro på en påstand eller ikke.

2.4 Lavys fire argumentasjonskategorier

I et prosjekt om argumentasjon og programmering, undersøkte Lavy (2006) hvordan noen elever argumenterte om matematikk mens de arbeidet med programmeringsprogrammet Logo. For å kunne si noe om elevenes argumentasjon, brukte hun Toulmin sin modell som analyseverktøy. Videre brukte hun funnene fra modellen til å plassere elevenes argumentasjon i fire egenkomponerte kategorier. Disse kategoriene kalte hun for «basic, compound, elaborated and general-presented-as-specific» (Lavy, 2006, s. 158). Videre i oppgaven velger jeg å bruke de norske oversettelsene *grunnleggende*, *sammensatt*, *utvidet* og *generelt-presentert-som-spesifikt*. Lavy sin studie har i tillegg til å bidra med analyseverktøy også fungert som en inspirasjonskilde til dette prosjektet. Det er derfor flere likheter mellom studiene, som for eksempel at vi begge har forsket på programmering og matematisk argumentasjon, gitt elevene oppgaver som åpner opp for utforskning og kreativitet, brukt et programmeringsprogram utviklet for barn og basert studien på Toulmin sin definisjon av argumentasjon.

2.4.1 Grunnleggende argumentasjon

Grunnleggende argumentasjon er den første kategorien for argumentasjon, og henviser til den mest enkle måten å argumentere på. Med denne formen for argumentasjon kommer både konklusjonen og argumentene fra å teste ut tilfeldige løsninger (Lavy, 2006, s. 159). Et eksempel på dette kan være når elever skal finne vinkelstørrelsene i en regulær femkant, og gjør dette med å teste ut mange tilfeldige vinkler frem til de finner den riktige vinkelstørrelsen. Videre blir argumentene for hvorfor denne vinkelstørrelsen er riktig at de tegner mange femkanter med denne vinkelstørrelsen.

2.4.2 Sammensatt argumentasjon

Videre forklarer Lavy (2006, s. 160) at sammensatt argumentasjonen bygger på den grunnleggende argumentasjonen ved at man har både konkrete eksempler og forklarer matematiske egenskaper for å argumentere for konklusjonen. I oppgaven om en regulær femkant, vil et eksempel på sammensatt argumentasjon være at elevene bruker matematikk for å avgrense hvilke vinkelstørrelser de skal teste ut for å finne svaret. Videre vil argumentene for hvorfor vinkelstørrelsen er riktig være basert på både konkrete eksempler som gir en regulær femkant, samt en matematisk forklaring på hvorfor det må være akkurat denne vinkelstørrelsen.

2.4.3 Utvidet argumentasjon

Som begrepet henter til, betyr utvidet argumentasjon at man bygger videre på noe man har oppdaget tidligere ved sammensatt argumentasjon (Lavy, 2006, s. 163). Lavy skriver at en slik form for argumentasjon dukker opp når man oppdager at det er noen matematiske tilfeller som ikke passer inn under en tidligere konklusjon eller at det er noen nye tilfeller som må inkluderes. Ved utvidet argumentasjon justerer man dermed på den tidligere konklusjonen slik at den enten ekskluderer eller inkluderer nye eksempler. I likhet med sammensatt argumentasjon vil man også her bruke matematiske vurderinger for å begrense antall forsøk, mens med grunnleggende argumentasjon gjør man mange tilfeldige forsøk frem til man kommer til den riktige løsningen. Et eksempel på en slik form for argumentasjon kan være når elever tidligere har erfart at for å programmere en geometrisk figur kan man bare veksle mellom å gå og rotere x -antall grader i samme retning, men oppdager ved et senere tidspunkt at denne strategien ikke fungerer for figuren stjerne. Da må elevene justere den tidligere konklusjonen til å ekskludere stjerner.

2.4.4 Generelt-presentert-som-spesifikt

Lavy (2006) definerer den siste kategorien ved setningen, «This kind of argument is based on elaborated and compound conclusions arrived at earlier in the discourse, and its justification is based on arguments which are specific examples of the generalized claim and mathematical considerations relating to number properties» (s. 167). Med andre ord er eksemplene som blir vist i denne kategorien fullverdige argumenter som støtter opp om et nytt argument. Dette argumentet skriver Lavy blir presentert som et spesifikt argument, men som i virkelighet er generelt og omfavner flere eksempler. For å eksemplifisere argumentasjon som er generelt-presentert-som-spesifikt trekker jeg frem eksempelet fra forrige avsnitt om programmering av geometriske figurer. Ved å bruke konklusjoner fra tidligere i oppgaven kan elevene argumentere for hvordan de skal programmere neste figur. Til tross for at de selv argumenterer for et spesifikt tilfelle (hvordan programmere neste figur), argumentere de også for den generelle konklusjonen: kommandoen *gå og roter x -antall grader i samme retning* fungerer på regulære figurer.

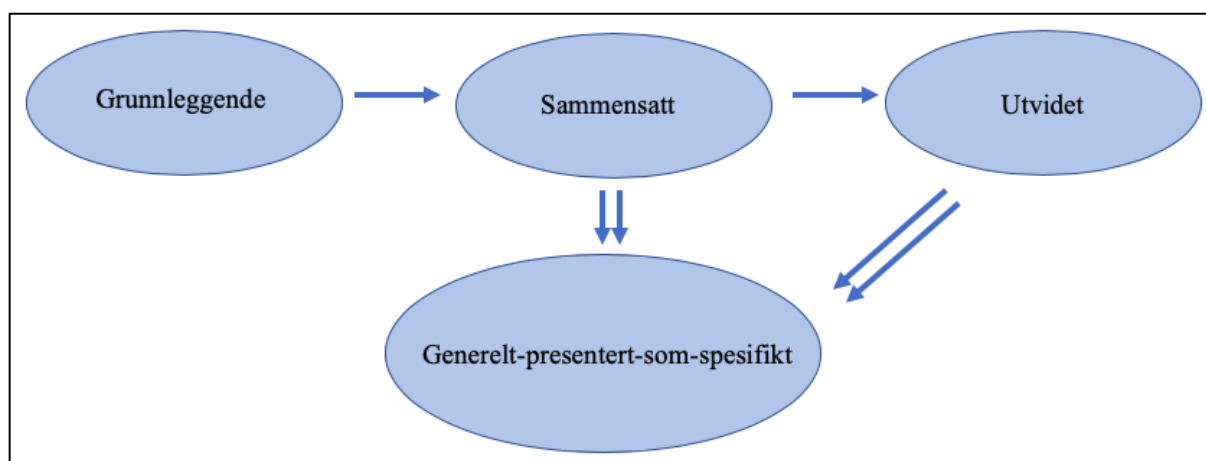
I Lavy sin definisjon av argumentasjon som er generelt-presentert-som-spesifikt skriver hun at argumentene må omhandle matematiske vurderinger om tallegenskaper. Dette kan være fordi oppgavene hun ga elevene sine handlet om nettopp dette. Ettersom det i dette forskningsprosjektet omhandler elever som programmerer frem geometriske figurer, kommer

jeg i analysen til å utvide Lavys kategorier til å inkludere alle former for matematiske vurderinger.

2.4.5 Modell av Lavys fire argumentasjonsformer

For å tydeliggjøre hvordan disse fire formene for matematisk argumentasjon forholder seg til hverandre, har jeg laget en modell av Lavy sine begreper:

Figur 2. Modell av Lavys fire argumentasjonsformer.



Her illustreres det hvordan grunnleggende argumentasjon kan bygges videre på til sammensatt argumentasjon, som igjen kan føre til utvidet argumentasjon. Med generelt-presentert-som-spesifikt, viser pilene hvordan tidligere argumenter fra kategoriene sammensatt og utvidet kan føre til argumenter som er generelt-presentert-som-spesifikt. De doble pilene, er ment for å illustrere at det trengs flere argumenter for å bygge opp om et argument som er generelt-presentert-som-spesifikt.

2.5 Argumentasjon gjennom ulike uttrykksformer

Den muligens mest åpenbare måten å uttrykke argumenter på er gjennom muntlige ytringer. I skolen skal elevene arbeide med og utvikle språket sitt til å kunne samhandle og formidle sine tanker til andre. Selv om muntlige ytringer er en stor del av det å argumentere, kan man også gjøre dette på mange andre ulike måter. Et av Krummheuer (1995, s. 236) sine argumenter til hvorfor matematisk argumentasjon omfatter mer enn det å utføre matematiske bevis, er at barn sjeldent handler og snakker i formen av et logisk bevis. I stedet bærer de matematiske uttrykkene deres preg av erfaringer på ekte matematiske objekter. Dette kan være alt fra fingertelling, tegning, peking eller bygge konklusjoner ut ifra konkrete. Som Johnsen-Høines

(2006) påpeker i boka si, *Begynneropplæringen*, er det viktig at vi voksne «lærer å lytte til barna på deres premisser, lærer å tolke språket de har, og finne ut hva de ønsker å uttrykke» (s. 36). Hensikten med dette delkapittelet er derfor å gi informasjon om hvordan man kan forstå elevenes argumentasjon utover det de uttrykker eksplisitt med muntlige ord. Det som er presentert i dette delkapittelet, er aspekter ved matematisk argumentasjon jeg har sett etter under analysen av datamaterialet i kapittel 4.

2.5.1 Deiktisk språk

I Lavy (2006) sin forskning om matematisk argumentasjon har dataskjermens rolle fått mye oppmerksomhet. Alle kategoriene hennes baserer seg på hvordan skjermbilder spiller ulike roller i elevenes argumentasjon, der bildene på skjermen fungerer som en støtte for elevene i starten av argumentasjonen og blir gradvis mer og mer byttet ut med muntlige matematiske forklaringer på påstandene. To andre forskere som også har sett på hvordan ulike elementer på dataskjermen kan tilby et språk og hjelpe elever som sliter med å matematisk formulere seg er Wyndhamn og Säljö (2009). Her skriver de om hvordan elevene i deres forskningsprosjekt ofte tok i bruk et så kalt *deiktisk språk*. Denne typen språk er ofte preget av muntlige ytringer som: denne, denne her, nå, først, sist, se her osv. For å forstå bruken av disse ordene er man ifølge Wyndhamn og Säljö (2009) «dependent on the co-presence of two actors focussing on the same visual field» (s.37). Dersom en person bruker et deiktisk språk vil det dermed bli vanskelig å forstå hva personen mener hvis man ikke vet hva samtalepartneren ser på. Videre er typiske trekk ved deiktisk språk at ordene som ble nevnt ofte blir ytret, de er veldig preget av konteksten og er ofte etterfulgt av gestikuleringer. Ettersom elevene i dette forskningsprosjektet arbeidet på en datamaskin og fikk et ark å tegne/skrive på, er det mulig at elevene tok i bruk et deiktisk språk som henviser til enten dataskjermen eller noe som har blitt tegnet/skrevet på et ark. Dette er noe som har blitt sett etter i kapittel 4, Analyse og diskusjon.

2.5.2 Argumentasjon gjennom prøving og feiling

Prøving og feiling skriver Hana (2014) er en sentral metode ved problemløsning i matematikk. Videre trekker han frem at denne metoden er mest gunstig dersom den består av *systematisk* prøving og feiling. Dette innebærer at man setter noen krav for hva man vil prøve slik at man slipper å teste ut mange tilfeldige løsninger uten å reflektere over resultatene. Man starter med noen gjennomtenkte forslag til en løsning, tester ut og vurderer svaret opp mot det endelige målet. Deretter prøve man på nytt med en systematisk framgangsmåte som bruker matematisk

innsikt basert på forståelse av hvordan det gikk i det forrige forsøket. På den måten må man ha en viss innsikt i variasjonsmulighetene ved oppgaven slik at man kan presentere noen legitime argumenter for hvorfor noen utfall eller løsninger kan være mer riktige enn andre. Hana (2014) trekker også frem at denne metoden er noe generell, og kan noen ganger være lite effektiv og vanskelig å utføre.

2.5.3 Eksempelbasert argumentasjon

En ofte brukt taktikk for å bevise at en matematisk påstand er korrekt, er å vise eksempler der påstanden stemmer. Russel, Schifter og Bastable (2011) forklarer i boka si, *Connecting Arithmetic to Algebra*, at denne taktikken er vanlig å se hos elever på barneskolen når de skal argumentere for hvorfor en påstand alltid er korrekt. Videre blir det trukket frem at presentasjon av eksempler er et godt steg i veien mot matematisk argumentasjon, og at en kan oppdage matematiske forhold og strukturer ved å se nøye på eksemplene. Ulempen med eksempelbasert argumentasjon er at den aldri vil kunne bevise at påstanden gjelder for uendelige tilfeller, og at det trengs bare et eksempel for å motbevise påstanden.

2.5.4 Representasjonsbasert argumentasjonen

Russel, Schifter og Bastable (2011) skriver også om representasjonsbasert argumentasjon og hvordan dette er en typisk teknikk i barneskolen for å rettferdiggjøre en matematisk påstand. Ved å skape representasjonsbaserte argumenter arbeider elevene med å skape et mentalt bilde over hvordan operasjonen fungerer og hvilken betydning og egenskaper operasjonen har. Et eksempel på dette kan være å tegne penger for å visualisere at svaret i et regnestykke er riktig.

2.5.5 Oppfattet-som-forstått

Yackel (2001, s. 8) skriver i sin artikkel, *Explanation, justification and argumentation in mathematics classrooms*, om når man antar at motparten forstår det underliggende argumentet. Dette blir sett på som *taken-as-shared* eller som jeg velger å kalle det, *oppfattet-som-forstått*. Eksempel på dette kan være når argumentasjonen hos elevene går fra å forklare at: $6+4=10$ fordi 6 er én mer enn 5 og 4 er én mindre enn fem, og jeg vet at $5+5=10$. Til å forklare at $6+4=10$ fordi $5+5=10$. Da blir argumentet *fordi 6 er én mer enn 5 og 4 er én mindre enn fem* oppfattet-som-forstått. Videre forklarer Yackel (2001) at matematiske praksiser som er oppfattet-som-forstått, oppstår når klassen har jobbet såpass mye med det at det etter hvert ikke er hensiktsmessig å bruke tid på å forklare tankegangen. Når dette skjer skriver Yackel (2001)

at de matematiske praksisene ikke trenger å bli begrunnet, og på den måten utvikles behovene for argumenter.

Ettersom oppgaven ikke inkluderer datamateriale som er samlet inn over en lengre periode, vil det være vanskelig å si om hvordan argumentasjonen hos elevene endrer seg. Som konsekvens av dette vil disse *oppfattet-som-forstått*-argumentene vises ovenfor oss som de stedene i dialogen der elevene forstår noe som den andre ikke har forklart. Dette kan være et tegn på at matematikken er enkel og godt etablert hos eleven(e).

3. Metode

I dette kapittelet er det gjort rede for forskningsmetoden som ble brukt under datainnsamlingen, de ulike metodiske valgene som ble tatt underveis i prosessen, utfordringer jeg som forsker møtte på under innsamlingen av datamaterialet, behandling av datamaterialet, analysemetode, oppgavens validitet og reliabilitet og etiske hensyn. Datainnsamlingen har foregått i samarbeid med en annen student, Tonje Lindberg. Det vil derfor i de følgende delkapitlene bli snakket om et «vi», som i dette tilfellet består av Tonje Lindberg og meg selv.

3.1 Valg av metode

Hensikten med oppgaven er å undersøke hvilke kvaliteter med matematisk argumentasjon som kan identifiseres når elever arbeider med programmering i Scratch. Nordahl (2012) skriver at «Systematisk observasjon åpner for muligheten til å studere dynamikken i interaksjon mellom mennesker på en direkte og mer objektiv måte enn ved selvrappport fra deltakerne» (s. 159). Ut ifra dette ble det gjort et valg om å observere et begrenset utvalg av elever mens de arbeidet med noen oppgaver i Scratch som la til rette for matematisk argumentasjon. Vi valgte å dokumentere datamaterialet ved å filme elevene med lyd mens de arbeidet, samtidig som vi også tok skjermopptak av datamaskinen elevene arbeidet på. Dette gjorde vi for å skape et så helhetlig bilde som mulig av elevenes arbeidsprosess og den matematiske argumentasjonen. Forskningsprosjektet er med denne forskningsmetoden en kvalitativ studie, hvor jeg har gått i dybden på datamateriell fra en begrenset mengde informanter (Kvale og Brinkmann, 2015).

Underveis i prosessen har vi brukt mye tid på å sette oss inn i Scratch, observert et Scratch-kurs arrangert av Kodeklubben Vest, pratet med elevene på forhånd av prosjektet og deretter utført to undervisningsopplegg som datainnsamling. Kodeklubben (www.kodeklubben.no) er en undergruppe av organisasjonen Lær Kidsa Kode som har som mål å få flere barn til å begynne å kode. Det første undervisningsopplegget hadde som formål å forberede elevene på det siste undervisningsopplegget, samt hjelpe oss med å utforme et så tilpasset opplegg som mulig. Videre i oppgaven vil dette bli omtalt som undervisningsopplegg 1 og undervisningsopplegg 2. Ettersom vi var interessert i å legge til rette for så mye matematisk argumentasjon som mulig gjorde vi endringer på oppgavene elevene fikk etterhvert som vi lærte mer om elevenes tidligere erfaringer med Scratch og hva de kunne om Scratch før prosjektet begynte.

Ettersom Tonje og jeg har gått gjennom flere faser der vi har hentet inn informasjon og testet ut opplegg, har vi brukt en forenklet versjon av en forskningsmetode under datainnsamlingen som heter design-basert forskning. Dette er en form for forskning som de siste tiårene har vist seg å gi muligheter for å både forske på og utvikle teknologiske læringsmiljø innenfor de pedagogiske fagene (Wang & Hannafin, 2005, s. 5). Wang og Hannafin (2005, s. 6) definerer i sin artikkel design-basert forskning ved å beskrive det som en systematisk, men fleksibel metodologi, som har som mål å forbedre undervisningspraksiser gjennom gjentakende analyse, design, utvikling og utførelse basert på samarbeid mellom forskere og undervisere i virkeligheten. Med andre ord baserer design-basert forskning seg på at man gjennom flere sykluser prøver ut noe i virkeligheten, analyserer, endrer og prøver ut igjen. Man prøver å designe en teori, et opplegg og/eller et redskap som passer til virkeligheten så godt som mulig. Med denne formen for forskningsmetode fremhever Messick (referert i Barab & Squire, 2004, s. 2) de tre fordelene:

1. Forskningsresultater som tar hensyn til rollen om sosial kontekst og som på den måten har større sannsynlighet til å påvirke undervisningspraksiser.
2. Håndgripelige produkter og programmer som kan bli adaptert til andre situasjoner.
3. Forskningsresultater som er validert gjennom å ha blitt testet i virkelige situasjoner.

Selv om forskningsprosessen vår har flere likhetstrekk til design-basert forskning mener jeg at vår forskning ikke har gått gjennom nok sykluser med testing, analyse, endring og ny testing for å kunne kalle forskningen vår for en fullstendig design-basert forskning. På grunn av mangel av tid hadde vi dessverre ikke muligheten til å utføre et opplegg som var fullstendig design-basert forskning. Vi vektla i stedet noen av aspektene ved et design-basert forskningsprosjekt som vi fant både verdifulle for våre prosjekter og som vi hadde mulighet til å utføre. Ved å ha gått gjennom de ulike fasene frem til det endelige undervisningsopplegget har vi tatt hensyn til idéen om at læring ikke er noe som skjer isolert hos den individuelle tenkeren, men at det er en sosial prosess som blir påvirket av både læringsmiljøet og læringsaktiviteten (Barab & Squire, 2004, s. 1).

3.2 Observasjon

Som nevnt i forrige avsnitt brukte vi observasjon av elever som arbeidet med Scratch som metode for å besvare problemstillingen. Næss og Sjøvoll (2018) definerer observasjon som «systematisert innsamling av informasjoner fra omverdenen slik den viser seg for oss, og kan ordnes av oss via våre sanser og med supplerende verktøy» (s. 181). Med begrepene *sanser* og *supplerende verktøy* innebærer det i vår sammenheng at vi var til stede og noterte det vi observerte, samtidig som vi også dokumenterte elevenes arbeidsprosess ved å ta skjermpoptak og video med lyd av elevene og deres dataskjerm. På den måten kunne vi i ettertid både basere analysen på våre egne erfaringer og observasjonsnotater av elevenes arbeidsprosess, samt se mer detaljert på enkelte hendelser ved å se på skjermpoptaket og filmen av elevene.

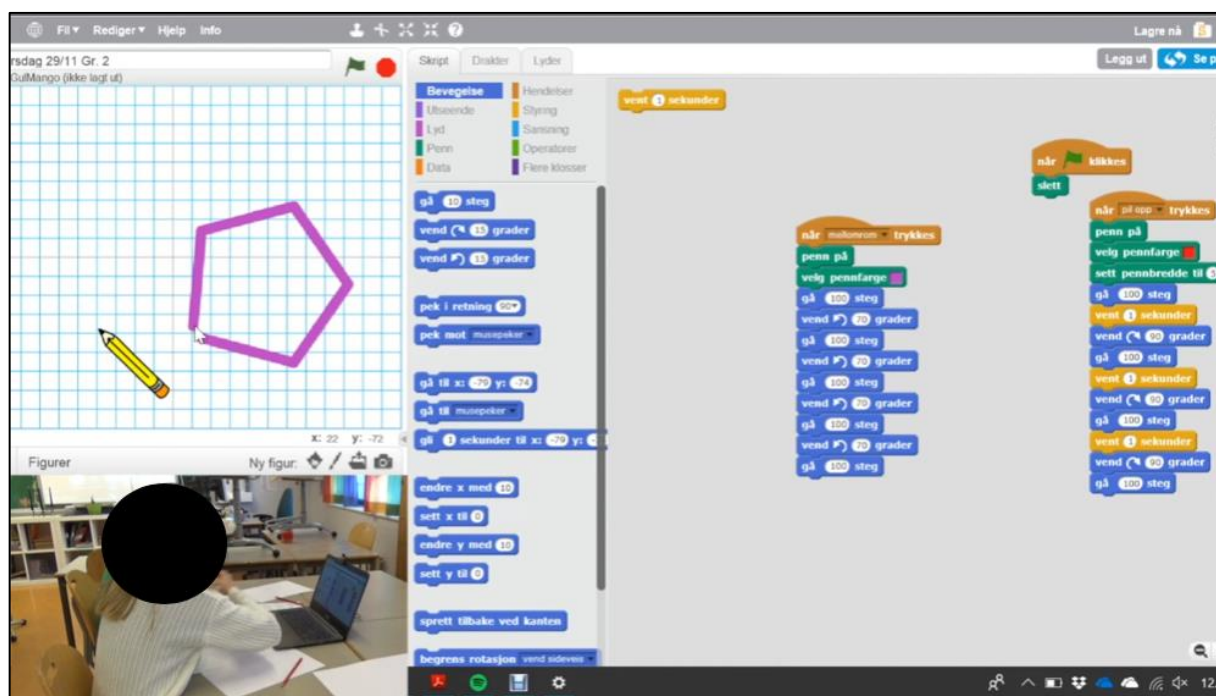
3.2.1 Video- og lydopptak

Etter avtale med elevene, foreldre og NSD-godkjenning (se delkapittel 3.8 om etiske hensyn) ble det gjort video- og lydopptak av elevene mens de arbeidet med oppgavene. Kameraet var plassert ved siden av og litt bak elevene slik at man fikk se hvordan elevene satt i forhold til hverandre og hvor de pekte på dataskjermen. På kameraet var det koblet til en ekstra mikrofon som var plassert på datamaskinens kant slik at lyd og bilde ble synkronisert med hverandre.

3.2.2 Skjermpoptak

For å ta opptak av dataskjermen hadde vi på forhånd lastet ned et program som lot oss ta opptak av alt som skjedde på skjermen. Hensikten med skjermpoptaket var å supplere video- og lydopptaket slik at man lettere kunne se hva som skjedde på skjermen når elevene pratet sammen eller gjorde noe på skjermen. Under analysen ble det brukt funksjonen *bilde i bilde*, hvor vi hadde filmen av elevene i en liten rute nederst på skjermen. Se figur 3. Skjermpoptaket ble synkronisert med video- og lydopptaket.

Figur 3. Eksempel av funksjonen «bilde i bilde».



3.2.3 Observatørens rolle

Når det kommer til observasjon som forskningsmetode, er det vanlig å skille mellom ikke-deltakende og deltakende observasjon. Som begrepene antyder, betyr ikke-deltakende observasjon at observatøren ikke tar del i eller påvirker det som observeres mens deltakende observasjon er det motsatte (Næss & Sjøvoll, 2018, s. 180). Ettersom vi var ute etter å forske på kvaliteter ved elevenes matematiske argumentasjon mens de programmerte i Scratch ble vi enige om at vi skulle være så nøytrale som mulig under arbeidsprosessen til elevene, men at vi kunne gripe inn dersom vi så det som nødvendig. Hensikten med dette var at vi ønsket å se hvilken påvirkning programmeringsdelen med oppgaven hadde for elevenes matematiske argumentasjon, og vi ville ikke ødelegge dette med vår innblanding i elevenes arbeidsprosess. På forhånd skrev vi derfor ned ulike situasjoner vi tenkte kunne oppstå og hvilken type svar vi kunne gi. Dersom elevene hadde spørsmål om eller sto fast på noe som angikk det rent tekniske om Scratch, kunne vi hjelpe elevene med dette. Vi var dermed en blanding av deltakende- og ikke-deltakende observatører. Vi ønsket ikke at elevene skulle bruke mye tid på å forstå det rent tekniske, men at de heller skulle bruke tid på å diskutere matematikken og hvordan de kunne programmere avataren til å gjøre ulike handlinger. Hvis elevene sto fast på en oppgave over lengre tid, uten å gjøre noe, kunne vi i tillegg stille spørsmål som oppfordret til videre tenkning uten å gi elevene et direkte svar. Eksempel på spørsmål vi kunne stille er: *Hva har dere tenkt?* og *Har dere noen idéer til hva som kan hjelpe?*

3.3 Utvalget

Det ble gjort et strategisk valg av hvilken klasse vi spurte om å delta på forskningsprosjektet. Som tidligere nevnt ønsket vi at elevene skulle bruke tida på å diskutere seg frem til en løsning på oppgaven, og ikke på å utforske og prøve ut de tekniske aspektene ved Scratch. Av den grunn var vi interessert i klasser som hadde arbeidet med Scratch tidligere, eller som fortsatt holdt på med Scratch i undervisningen. Etter å ha hørt med ulike skoler, kom vi over en klasse som vi fikk inntrykk av hadde holdt på med Scratch tidligere og som hadde deltatt på et av kodeklubben sine Scratch-kurs. Klassen gikk på 7. trinn, og vi fikk også høre at de skulle være med på et nytt kurs arrangert av Kodeklubben i løpet av høsten 2018. Senere viste det seg derimot at elevenes kunnskaper om Scratch ikke var slik som vi hadde fått inntrykk av. På dette tidspunktet var det for sent til å finne en ny klasse, slik at vi måtte gjøre det beste ut av den klassen vi hadde. Endringene som ble gjort i opplegget som følge av dette vil bli presentert i delkapittel 3.4 om datainnsamlingsprosessen.

Det var totalt åtte elever som deltok på prosjektet, der tre av elevene var gutter og fem var jenter. Ettersom vi ikke var ute etter å forske på én bestemt elevgruppe, ba vi læreren velge ut fire elev-par med både gutter og jenter og med høyt-, middels- og lavt-presterende elever. I tillegg ønsket vi at elevene skulle arbeide i par og samarbeide for å løse oppgavene, og av den grunn ba vi også læreren om å velge elev-par som læreren trodde kom til å samarbeide bra. Elevgruppene vil senere i oppgaven bli omtalt som gruppe 1, 2, 3 og 4. Ut ifra de fire gruppene vi samlet inn datamaterialet av, har det blitt lagt et særlig fokus på gruppe 2 bestående av elevene Knut og Ida. Navnene på elevene er fiktive med hensyn til personvern. Gruppe 1 har fått navnene Nils og Anders. Gruppe 3 er omtalt som Silje og Pernille, og gruppe 4 som Tiril og Line

3.4 Datainnsamlingsprosessen

Gjennom forskningsprosessen ble det gjort ulike tiltak for å sikre at oppgaven vi ga til elevene var tilpasset for deres programmerings-kompetanse så mye som mulig. Det vil i de følgende avsnittene bli presenterte en kronologisk gjenfortelling av vår forskningsprosess under datainnsamlingen og hvordan vi kom frem til det endelige undervisningsopplegget.

3.4.1 Den opprinnelige planen

Opprinnelig var planen at vi skulle være med når klassen deltok på et kurs arrangert av Kodeklubben Vest. Utover høsten gikk vi derimot vekk fra denne planen, da vi fant ut at opplegget til kodeklubben ikke passet så godt til vårt forskningsprosjekt samt at kodeklubben og klassen brukte for lang tid på å avtale tidspunkt. Etter å ha lagt fra oss planen om å samle inn datamateriell fra et av kodeklubben sine kurs, begynte vi å sette oss enda bedre inn i Scratch og hvordan man kunne utvikle et opplegg som la til rette for både programmering og matematisk argumentasjon.

3.4.2 Første møte med elevene

Som nevnt hadde vi før dette møtet fått informasjon om at elevene tidligere hadde hatt kurs i Scratch, og vi antok da at de var trygge på hvordan Scratch fungerte. Vi håpet på å kunne gi elevene helt åpne oppgaver fra starten av, og at elevene fokuserte mest på det matematiske og ikke så mye på det rent tekniske med Scratch. Etter å ha pratet med elevene fikk vi i stedet inntrykk av at de ikke kom til å klare å gjøre så mye med det første hvis ikke de fikk noen føringer og hjelpetips. Det var to år siden de hadde holdt på med Scratch, og da hadde de bare fulgt ulike oppskrifter. Det ble av den grunn laget en plan om å bruke det første undervisningsopplegget for å forberede elevene på det siste undervisningsopplegget. Som undervisningsopplegg 2 ønsket vi at elevene skulle lage en form for animert fortelling der avatarene flyttet på seg og flere bakgrunner ble tatt i bruk, fordi det var ved disse kodeklossene vi mente at elevene kom til å arbeide mest med matematikk. Til undervisningsopplegg 1 bestemte vi oss derfor for å gi oppgaver som i løsningsprosessen ga elevene kunnskaper som de kunne bruke når de skulle lage den animerte fortellingen. På den måten håpet vi på at elevene kunne fokusere mer på det matematiske og kreative, ettersom mye av det rent tekniske var blitt gått gjennom under det første undervisningsopplegget.

3.4.3 Scratch-kurs i regi av Kodeklubben Vest

I tillegg til å prate med elevene på forhånd av forskningsprosjektet, fikk vi også muligheten til lære enda mer om elevenes tidligere erfaringer med Scratch ved at vi fikk bli med på et av kursene til Kodeklubben Vest på en annen skole. Dette var det samme kurset som det elevene våre tidligere hadde vært gjennom.

Kurset skjedde rett etter skolen, på skolen og var frivillig. Elevene gikk på 4.-6. trinn og dette var det fjerde kurset elevene var med på i kurspakken om Scratch. Kurset ble holdt av to ansatte fra Kodeklubben Vest som gikk rundt og hjalp elevene som satt alene på hver sin PC. Opplegget gikk ut på at elevene hadde to nettsider fremme, én nettside for å programmere i Scratch og én nettside der det sto en oppskrift for hvordan man skulle programmere. Oppskriften var ferdig laget og den ligger gratis ute på kodeklubben sin nettside (<https://oppgaver.kidsakoder.no/scratch/flaksefugl/flaksefugl>). Underveis i kurset noterte jeg meg at elevene ofte byttet mellom nettsiden med oppskriften og nettsiden de programmerte i, de snakket ofte til sidemannen og flere av elevene spurte raskt om hjelp.

I en samtale med en av de ansatte, kommenterte han at han var usikker på hvor mye programmering ungene egentlig lærte ettersom de bare fulgte noen instruksjoner. Videre fortalte han at det hadde vært bedre dersom han fikk forklart hvorfor noen av kommandoene gir det utfallet de gir, men at dette var noe de ikke fikk til å gjøre fordi de ikke klarte å holde oppmerksomheten til elevene.

Under kurset klarte jeg ikke å observere at elevene tok eierskap til programmet de laget. Det kan ha sammenheng med at de i liten grad fikk være kreative og finne egne måter å løse oppgaven på. Tonje og jeg ble enige om at det var viktig at elevene skulle få mulighet til kritisk refleksjon, egen tenkning og kreativitet når de skulle arbeide med oppgaver i vår datainnsamling.

Observasjonene fra Scratch-kurset til Kodeklubben ga oss også indikasjoner på at elevene ikke husket så mye om Scratch fordi de aldri hadde fått utforsket fritt i programmet eller fått forklart hvorfor man bruker noen av kommandoene. Dette støttet opp om vår beslutning om at elevene trengte mer føringer under det første undervisningsopplegget for å forberede dem til undervisningsopplegg 2.

3.4.4 Det første undervisningsopplegget

Til det første undervisningsopplegget holdt vi fast på idéen om at elevene skulle lage en animert fortelling under undervisningsopplegg 2. Som forberedelse til dette lagte vi selv ulike animerte fortellinger, og noterte oss ned hvilke funksjoner vi brukte og hvilke aspekter ved oppgaven som kom til å bli mest vanskelig.

Ut ifra disse erfaringene utformet vi et oppgaveark (se vedlegg 1) med seks oppgaver som tok for seg disse tematikkene:

- Utforsking
- Forflytning av avataren
- Bytting av bakgrunn når avataren kom til kanten av bildet
- Endring av drakt på avataren slik at det ser ut som om den går og ikke glir
- Kommunikasjon mellom avatarer
- Hvordan nullstille alt når man starter fortellingen på nytt

Under det første undervisningsopplegget tok vi både skjermopptak og film med lyd av elevene. Ved å gjøre dette sikret vi oss supplerende datamateriell til å se tilbake på når vi skulle ferdigstille opplegget til undervisningsopplegg 2 og mulighet til å bruke datamaterialet fra undervisningsopplegg 1 ved senere tidspunkt dersom det dukket opp noe interessant. I tillegg fikk elevene mer tid til å bli vant med å bli filmet, vi fikk sett om kameraoppsettet fungerte og om det ellers var noen tekniske endringer som måtte gjøres før vi utførte det endelige undervisningsopplegget.

Elevene arbeidet i omtrent 45 minutt, og vi hadde to elev-par inne om gangen. Resten av klassen hadde kunst og håndverk, slik at vi hadde hele klasserommet alene. For å unngå at elevene eller lydopptakeren skulle få med seg hva det andre paret gjorde, plassert vi parene så langt fra hverandre som mulig. Mens elevene arbeidet, vandret vi mellom parene for å få med oss hva begge gjorde.

Før elevene begynte med oppgavene, gikk jeg og Tonje raskt gjennom ulike funksjoner i Scratch for å friske opp elevenes forkunnskaper. Underveis i den første økta ga vi elevene lite hjelp til tross for at de ofte sto fast. Etter den første økta ble jeg og Tonje enige om at oppgavene var for teknisk krevende, og bestemte oss for å gi en enda mer omfattende forklaring av Scratch ved starten samt hjelpe dem mer med det tekniske underveis i arbeidsprosessen. Det ble dermed gjort en stor endring på utførelsen av det første undervisningsopplegget mellom økt 1 og økt 2. I første økt arbeidet gruppe 1 og 2, og i den andre økta arbeidet gruppe 3 og 4. Selv om dette gjorde at gruppe 3 og 4 fikk et annet utgangspunkt for å løse oppgavene i hoveddatainnsamlingen enn gruppe 1 og 2, mener vi at fordelene med å hjelpe gruppe 3 og 4 veide

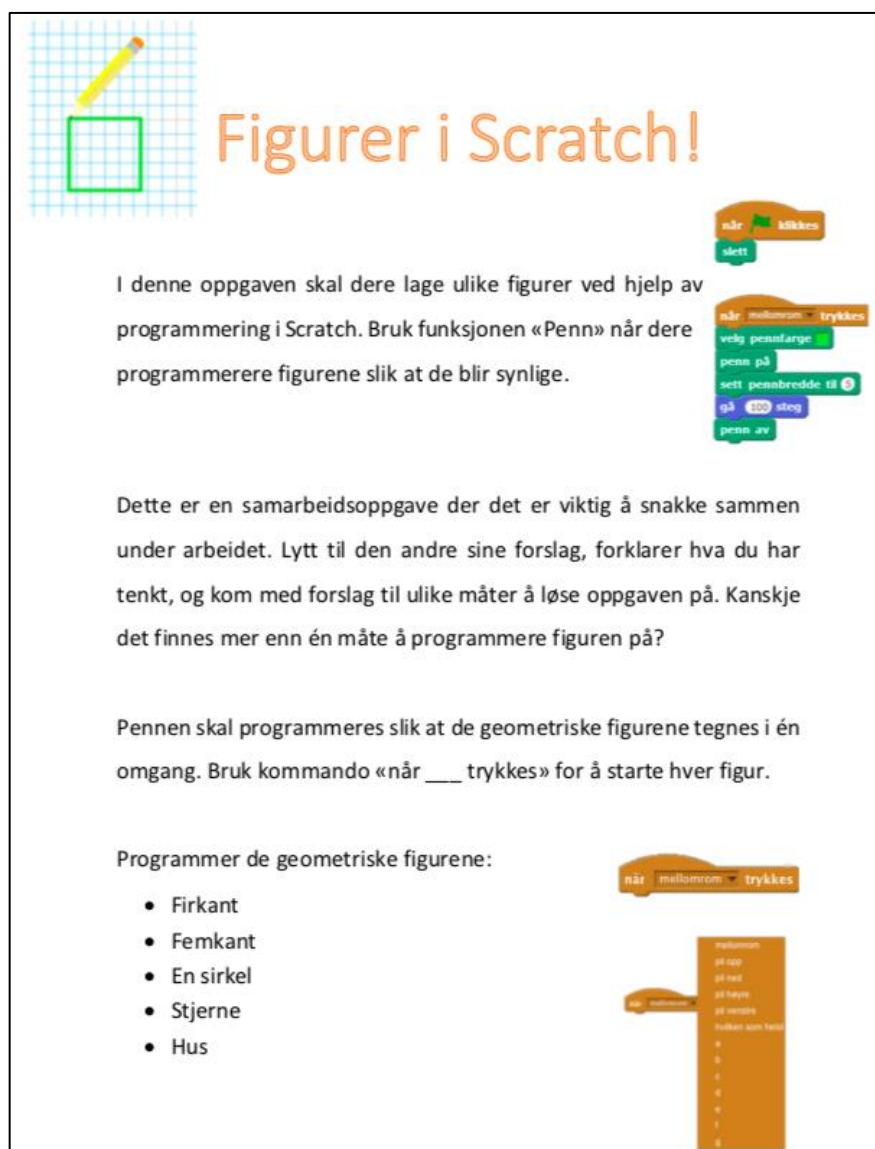
opp for ulempen med å ha to forskjellige utførelser av undervisningsopplegg 1 fordi elevene fikk brukt mer tid på å diskutere det matematiske.

3.4.5 Det endelige undervisningsopplegget

Det første undervisningsopplegget ble utført på en mandag og det endelige undervisningsopplegget på torsdagen samme uke. Vi hadde dermed to dager mellom datainnsamlingene til å gå gjennom resultatene fra undervisningsopplegg 1 og gjøre endringer ut ifra disse. Etter å ha vurdert resultatene fra undervisningsopplegg 1, ble vi enige om at idéen om å gi elevene i oppgave å programmere en animert fortelling var for tidkrevende og inneholdt for mange tekniske aspekter som ikke oppfordret til matematisk argumentasjon. Av den grunn valgte vi å gå vekk fra oppgaven om animert fortelling, til fordel for en mer direkte matematisk rettet oppgave. Selv om det første undervisningsopplegget nå ikke fikk forberede elevene til undervisningsopplegg 2 slik som vi først hadde ment, ga undervisningsopplegg 1 oss likevel verdifull informasjon som jeg mener hjalp oss med å lage en god oppgave som støttet opp om vårt formål med datainnsamlingen.


Oppgaven vi endte opp med å gi elevene, gikk ut på at de skulle programmere en penn til å tegne de geometriske figurene: firkant, femkant, sirkel, stjerne og til slutt et hus (se figur 4). Etter det første undervisningsopplegget oppdaget vi nemlig at det var når elevene arbeidet med forflytting av avataren, de pratet og diskuterte mest om matematikk. Ved å gi elevene denne oppgaven tok vi dermed vekk de tekniske aspektene som ikke oppfordret til så mye matematikk. Dette var tekniske aspekter som å bytte bakgrunn, prate med andre avatarer og være kreativ med fortellingens innhold. Vi ville fortsatt at elevene skulle være kreative, men at de skulle være kreative med ulike matematiske løsninger. På forhånd av undervisningsopplegg 2 hadde Tonje og jeg gjort klart programmeringssiden på Scratch som elevene skulle jobbe på ved å sette inn en bakgrunn og en avatar. Vi la inn en bakgrunn med rutenett, og satt inn en penn som avatar. Vi ordnet det også slik at når avataren skulle spore bevegelsene sine med *penn-funksjonen*, kom streken ut fra tuppen av pennen.

Figur 4. Bilde av oppgaven elevene fikk.



Figurer i Scratch!

I denne oppgaven skal dere lage ulike figurer ved hjelp av programmering i Scratch. Bruk funksjonen «Penn» når dere programmerer figurene slik at de blir synlige.




Dette er en samarbeidsoppgave der det er viktig å snakke sammen under arbeidet. Lytt til den andre sine forslag, forklarer hva du har tenkt, og kom med forslag til ulike måter å løse oppgaven på. Kanskje det finnes mer enn én måte å programmere figuren på?

Pennen skal programmeres slik at de geometriske figurene tegnes i én omgang. Bruk kommando «når ___ trykkes» for å starte hver figur.

Programmer de geometriske figurene:

- Firkant
- Femkant
- En sirkel
- Stjerne
- Hus



Under undervisningsopplegg 2 ble datamaterialet samlet inn på samme måte som ved undervisningsopplegg 1 der elevene satt på de samme plassene og kameraene var plassert på samme sted. Før elevene gikk i gang med oppgavene leste Tonje og jeg opp oppgaven sammen med hvert vårt elev-par og forklarte noen tekniske aspekter. Dette var blant annet hvordan de skulle få pennen til å begynne å tegne og hvordan de kunne skru av pennen hvis de bare skulle flytte pennen. Gruppe 1 og 2 fikk også en nøyere gjennomgang av Scratch generelt, da de ikke hadde fått den samme forklaringen som gruppe 3 og 4 under det første undervisningsopplegget. Det ble også understreket at dette var en samarbeidsoppgave der det var viktig å lytte til hverandre, stille spørsmål og dele det man tenkte. Underveis i arbeidsprosessen observerte Tonje og jeg elevene og vi ga kun hjelp til elevene dersom det handlet om noe teknisk med Scratch.

3.5 Behandling av data

Etter datamaterialet var samlet inn begynte jeg å se gjennom videoene og behandle datamaterialet for å gjøre det enklere å analysere. Som tidligere nevnt ble videoen av elevene satt inn som en liten ramme i videoen av dataskjermen og ble så synkronisert med hverandre. Dette var for å enklere kunne se hva som skjedde på dataskjermen samtidig som elevene sa eller gjorde noe. Videre ble videoene med lyd nøye sett gjennom hvor jeg noterte meg ned tider i videoene som var interessante. Etter å ha fått et overblikk over hva datamaterialet inneholdt valgte jeg meg ut noen elever å se nærmere på.

For å kunne analysere datamaterialet så nøyaktig og detaljert som mulig, transkriberte jeg lydopptakene og videoene. Kvale og Brinkmann (2015) beskriver transkripsjon som en fortolkningsprosess der en oversetter fra talespråk til skriftspråk. Denne oversettelsen krever tolkning av den som transkriberer, hvor tonefall og måten ord blir sagt på kan påvirke aspekter som ironi, usikkerhet, bestemthet, følelser og spørsmål. For å inkludere noen av disse tonefallene og deres betydning har jeg brukt tegn som spørsmålstegn, punktum, komma, ellipse og hakeparentes. I dialogutsnittet under vises det hvordan tegnene har blitt brukt i transkripsjonen:

Ida: Kanskje 60 grader?

Knut: Vent da, den må først vende her. [Knut peker på vinkelen innerst i stjerna.]

Ida: Nei, ikke 60, jeg mener hundre og et eller annet.

Knut: Nei, men sånn, den må vende annerledes grader. For her oppe så vender den, eeh ...

Spørsmålstegn har blitt brukt når en ytring blir stilt spørrende. Punktum og komma har blitt brukt for å avgrense det jeg tolker som en hel setning, og som pauser som blir gjort i setningen. Ellipse har blitt brukt for å vise at elevene slutter en setning som er uferdig og/eller med et tonefall som virker usikkert. Til slutt har hakeparenteser blitt brukt for å beskrive noe elevene gjør, eller for å forklare hva som har skjedd mellom noen ytringer. I transkripsjonen har jeg også oversatt elevenes ytringer fra dialekt til bokmål.

3.6 Analyseverktøy

Analyseverktøy er de midlene jeg har brukt i analysen for å hente ut så mye informasjon som mulig fra hva elevene gjorde og sa under datainnsamlingen. Det første som ble gjort var å skape en oversikt over hva alle elevene sa og gjorde, for så å velge seg ut deler av datamaterialet som var mest interessant å undersøke nærmere. For å gjøre dette brukte jeg teori om hva matematisk argumentasjon er til å identifisere dialogutdrag som kunne gi meg informasjon om ulike kvaliteter ved elevenes matematiske argumentasjon. Videre ble Lithners (2008) ulike former for resonnering og Lavys (2006) kategorier om matematisk argumentasjon brukt til å si noe om elevenes matematiske argumentasjon. I tillegg ble informasjonen om ulike uttrykksformer fra delkapittel 2.5 brukt for å kunne forstå bedre hva elevene mente i argumentasjonen sin og for å kunne kjenne igjen typiske trekk ved elevers matematiske argumentasjon.

For å eksemplifisere hvordan jeg har plassert elevenes matematiske argumenter i Lavys kategorier har jeg, ved kategorien sammensatt argumentasjon, sett om elevenes argumentasjon var todelt med at det inneholdt både et matematisk objekt og en matematisk forklaring på det som er vist. I et dialogutklipp diskuterer to av elevene om en tegning av en figur er en femkant eller ikke. Når den ene eleven sier «En, to, tre, fire, fem, seks. Det er en sekskant», har jeg definert dette som sammensatt argumentasjon fordi tegningen spiller en rolle i elevens argumentasjon. Hvordan tegningen ser ut er et av elevens implisitte argument, samtidig som hun også trekker frem de matematiske egenskapene ved å telle antall kanter på tegningen høyt. Jeg har dermed inkludert skjermbilder, tegninger og gestikuleringer med som argument, selv om elevene ikke eksplisitt forklarte disse som å være et argument. Dette har jeg valgt å gjøre på bakgrunn av Johnsen-Høines (2006) vektlegging av å møte barna i deres språk og prøve å forstå hva de mener. Det betyr å inkludere handlinger og objekt som de ikke eksplisitt nevner, men som likevel har en rolle i elevenes ytringer.

Videre brukte jeg Lithner (2008) sine tre krav om kreativ argumentasjon, for å kunne si noe om elevenes argumentasjon. Dersom argumentene elevene brukte virket som å være en nyhet for elevene, samtidig som de også var matematisk forankret og bidrog til å rettferdiggjør en påstand, har jeg definert argumentene for å være kreative. For å kunne si noe om argumentasjonssekvensen til elevene er en nyhet har jeg sett på hvor sikre de er i formuleringene sine og hvor raskt de kommer frem til argumentene sine. Bruker de litt tid på å formulere seg, er litt nølende og kommer frem til en løsning som ikke er preget av mange innlærte algoritmer, så er dette tolket som tegn på at et argument kan være en nyhet. Presise og

raske argumenter er tolket som et tegn på at de har brukt slike argumenter tidligere og er derfor i mindre grad en nyhet. Argumenter som ikke er kreative har jeg definert som imitative dersom de var basert på tidligere innlærte algoritmer eller tidligere kjente løsninger.

3.7 Oppgavens validitet og reliabilitet

Validitet og *reliabilitet* er begreper som blir brukt for å beskrive et forskningsprosjekts gyldighet og pålitelighet (Kvale & Brinkmann, 2015). Med validitet menes det hvorvidt forskningsprosjektet har undersøkt det som skulle undersøkes, altså hvor gyldig funnene er i forhold til problemstillingen (Krumsvik, 2014). Det rettes også et særlig blikk på prosjektets metode, og hvor egnet metoden er for forskningsprosjektet. I dette forskningsprosjektet har hensikten vært å undersøke ulike kvaliteter ved elevers matematiske argumentasjon mens de programmerer. For å vurdere prosjektets validitet, må en derfor se om metoden i oppgaven gir mulighet til å undersøke dette.

For å kunne forske på hvordan elever matematisk argumenterer når de programmerer, må man først og fremst skape en situasjon der elevene har mulighet til å gjøre dette. Gjennom å gi elevene oppgaver som la til rette for både matematikk (gjennom geometriske oppgaver), programmering (gjennom Scratch) og argumentasjon (gjennom arbeid i par og utfordrende oppgaver), skapte vi nettopp en slik situasjon. Videre ble elevenes arbeidsprosess nøye dokumentert gjennom både video og lyd av elevene samt et skjermopptak av datamaskinen. Alrø og Kristiansen (1997) skriver om video som datainnsamlingsmetode og forklarer at «Hvis man vil fastholde noget, der i princippet forsvinder i det øjeblik det finder sted, så er man henvist til bruke medier – f.eks. video» (s. 75). Ved at Tonje og jeg filmet elevene fikk vi mulighet til å gå tilbake til de ulike situasjonene som utspilte seg og observere i detalj hva som skjedde. På denne måten kunne vi i ettertid analysere arbeidsprosessen til elevene med en nøyaktig gjengivelse av hva elevene hadde sagt og gjort. Ved å bruke begreper og kategorier fra tidligere forskning fikk jeg også hjelp til å beskrive og identifisere de ulike kvalitetene hos elevenes matematiske argumentasjon med en faglig begrunnelse.

Kvale og Brinkmann (2015) skriver at «Reliabilitet har med forskningsresultatenes konsistens og troverdighet å gjøre» (s. 276). I kvantitativ forskning henvises reliabilitet ofte til om forskningsmetoden kan gjennomføres flere ganger og fremdeles gi det samme resultatet. På grunn av at resultatene i en kvalitativ studie ofte vil være sterkt påvirket av personene som

deltar i studien, er dette derimot ikke like enkelt å gjøre ved kvalitativ forskning (Krumsvik, 2014). Ettersom etterprøvbarhet er vanskelig å få til ved kvalitativ forskning, vil transparens om forskningsmetoden være viktig for å styrke prosjektets reliabilitet (Krumsvik, 2014). For å styrke denne oppgavens reliabilitet, har jeg derfor vektlagt å gi grundige beskrivelser av hva som ble gjort frem til datainnsamlingen, bakgrunnen for de ulike valgene som ble tatt, hvordan datainnsamlingen foregikk, informasjon om utvalget av informanter, og hvordan datamaterialet ble behandlet og analysert i ettertid av datainnsamlingen.

Denne oppgavens forskningsresultater er påvirket av elevenes personligheter, forhold til arbeidspartneren og deres matematiske innsikt. Dette er noe som vil være vanskelig å skape på nytt i et annet forskningsprosjekt, og som gjør at et prosjekt med samme forskningsmetode, men med andre elever, kan gi resultater som er annerledes enn hva dette prosjektet har gitt. En kvalitativ studies overførbarhet vil alltid være preget av informantenes påvirkning på funnene. Oppgavens grundige beskrivelse av hva som har blitt gjort og hvorfor, gjør det derimot mulig for andre å ta i bruk ulike aspekter og deler av studien. Leserne av denne oppgaven kan vurdere hva, og i hvilken grad, min metode og mine resultater kan brukes eller overføres til hans eller hennes klasse.

3.8 Etiske hensyn

Backe-Hansen og Frønes (2012) skriver at «Barns situasjon, og de mange sårbare grupper av barn, skaper behov for en rekke etiske så vel som metodiske vurderinger» (s. 17). Med dette faktumet i bakgrunnen ble det gjort ulike valg og tilnærminger for å verne om barnas identitet og egenverdi under mitt forskningsprosjekt.

Gjennom metodekapittelet har jeg presentert alle valgene og vurderingene som ble gjort for å utvikle en så tilpasset oppgave som mulig. Én av grunnene for hvorfor en tilpasset oppgave var viktig for meg, var at jeg ønsket at elevene skulle oppleve mestrings- og skaperglede ved å utføre opplegget. Elevene skulle i ettertid av opplegget være glade for at de hadde deltatt i prosjektet, og ikke sitte igjen med en vond følelse av utilstrekkelighet fordi oppgavene var for vanskelige.

Under analyse og diskusjon av datamaterialet har jeg vektlagt det etiske knyttet til hvordan elevene blir presentert i oppgaven, og hvordan de framstår ut ifra mine analyser av dem. Dersom

det er noe elevene ikke har løst så bra har jeg forklart dette ved å vise til kompleksiteten og vanskelighetsgraden av oppgaven, eller andre momenter som kan ha hindret elevene fra å yte sitt beste. Jeg synes det er viktig å påpeke at alle elevene gjorde en flott jobb under datainnsamlingen, og dersom de gjorde noe feil har jeg prøvd å vise hele bildet på hvorfor denne feilen oppstod.

Backe-Hansen og Frønes (2012) skriver også at «Barns rett til beskyttelse betyr f.eks. at forskere må ha foreldres, og ofte også andre instansers, tillatelse for å studere barn» (s. 17). I mitt tilfelle innbar dette samtykke om barnets deltakelse fra både de foresatte og barnet selv, samt en godkjenning av forskningsprosjektet fra Norsk senter for forskningsdata (NSD). Ettersom vi skulle filme elevene og slik samle inn personidentifiserende materiell, ble forskningsprosjektet meldt inn til NSD for å sikre at personopplysningene kom til å bli behandlet i samsvar med personvernloven. Forskningsprosjektet mitt er som tidligere nevnt en del av det større forskningsprosjektet, LATACME. Av den grunn ble mitt forskningsprosjekt inkludert i LATACME sin søknad til NSD. På den måten kan datamaterialet og funnene mine bli brukt i LATACME sitt forskningsprosjekt.

Under det første møtet med elevene (se 3.4.2) var et av målene å informere elevene om hva forskningsprosjektene våre gikk ut på og hva det ville si for dem dersom de deltok i prosjektet vårt. I den sammenheng ble det også delt ut to samtykkeskjemaer, et til barnas foresatte (se vedlegg 2) og et til barna (se vedlegg 3). I disse skjemaene ble det forklart hva prosjektet gikk ut på, hva det innebar for elevene dersom de deltok og hva som kom til å skje med datamaterialet som ble samlet inn. Samtykkeskjemaet til barna ble skrevet på en slik måte at de forsto hva det innebar for dem å delta, mens samtykkeskjemaet til de foresatte inneholdt mer detaljert informasjon om forskningsprosjektet og de foresattes rettigheter. Underveis i datainnsamlingen ble det påpekt ovenfor elevene at dette var frivillig og at de kunne trekke seg når som helst dersom de ønsket det.

Datamaterialene har i ettertid av datainnsamlingen blitt lagret på eksterne harddisker som ikke er koblet til internett. All gjengivelse av datamaterialet har også blitt anonymisert for å sikre barnas personvern.

4. Analyse og diskusjon

I dette kapitlet presenteres resultat fra gjennomførelsen av undervisningsopplegg 2 i form av transkriberinger av videoene samt noen bilder fra elevenes dataskjerm. Det blir også gjort en analyse av resultatene, der jeg har brukt analyseverktøyene beskrevet i delkapittel 3.6. Formålet med analysen er å hente frem så mye som mulig fra resultatene som kan bidra til å belyse kvaliteter ved matematisk argumentasjon når elever programmerer. Som nevnt i delkapittel 3.3 har det i dette kapitlet blitt lagt et særlig fokus på gruppe 2, Knut og Ida.

Før Knut og Ida startet med å programmere, gikk en av de voksne gjennom oppgaveteksten og forklarte noen av de tekniske aspektene. Oppgaven elevene fikk gikk ut på at de først skulle programmere pennen til å gå i en firkant, deretter en femkant, en sirkel, ei stjerne og til slutt et hus. Det ble ikke gitt noen kriterier for hva som var en godkjent firkant, femkant, osv., men vi ba elevene om å definere disse begrepene selv. På forhånd var det blitt satt inn en avatar som så ut som en penn. For å få denne pennen til å tegne måtte elevene sette inn kodeklossene *penn på*, *sett pennfarge til* og *sett pennbredde til*. Disse kodeklossene gjorde at pennen tegnet en strek der den ble programmert til å gå. For å kunne fjerne det pennen tegnet, brukte de kodeklossen *slett alt* som slettet alle strekene pennen hadde tegnet. Den første oppgaven Knut og Ida løste var å programmere en firkant. Her var de raskt ute med å sette inn kodeklossene *roter 90 grader* og *gå 100 steg* fire ganger etter hverandre. Dette var kodeklosser hvor man kunne endre på gradene og antall steg, og som ble mye brukt gjennom hele arbeidsprosessen. Det er i de følgende delkapitlene blitt gjort rede for arbeidsprosessen til Knut og Ida i kronologisk rekkefølge fra programmeringen av en femkant til ei stjerne.

4.1 Knut og Ida programmerer en femkant

I utklippene som følger har Knut og Ida kommet til oppgaven der de skal programmere pennen til å tegne en femkant. Tidligere har en av de voksne forklart oppgaveteksten til elevene, og de har allerede programmert en firkant.

4.1.1 «Er det femkant?»

I starten av arbeidsprosessen begynner Knut å tegne en femkant (se figur 5), mens Ida setter inn kodeklossene *Penn på* og *Pennfarge* og begynner med kodelinjen for en femkant.

[Knut teller antall kanter i tegningen sin.]

Knut: En, to, tre, fire, fem, seks.

Ida: Og så må vi vende et eller annet.

[Knut viser tegningen til Ida.]

Knut: Er det femkant?

Ida: Ja.

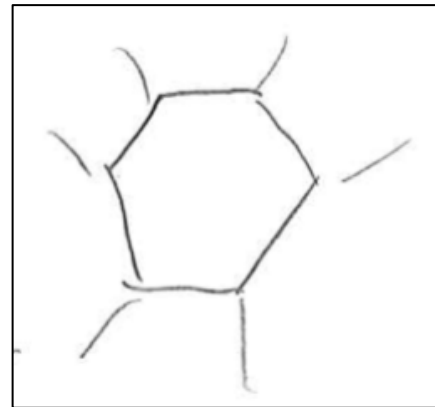
Knut: Den.

Ida: Nei, det er ... Vent litt, få se.

Knut: Jo, det er en, to, tre, fire, fem.

Ida: En, to, tre, fire, fem, seks. Det er en sekskant.

Figur 5. Bilde av Knut sin tegning av en femkant.



Knut ønsker først å klargjøre hvordan en femkant ser ut før han går i gang med kodingen ved at han prøver å tegne en figur som er en femkant. Ida går derimot raskt i gang med å sette inn noen kodeklosser uten å forklare hvorfor hun vil sette dem inn. Det oppstår mulighet for argumentasjon når Knut lurer på om figuren han har tegnet er en femkant. Selv har han først telt antall kanter og kom til seks, mens andre gangen teller han bare fem kanter og mener at dette da må være en femkant. Ida svarer først ja på spørsmålet om figuren er en femkant, men endrer svaret sitt til «Det er en sekskant» etter å ha telt seks kanter i figuren. Til tross for at både Knut og Ida endrer på svarene sine, kommer det tydelig frem at begge to er klar over at det er antall kanter som bestemmer om figuren er en femkant eller ikke. Selv om temaet i dette dialogutsnittet er preget av noe enkel matematikk, er dette likevel et viktig matematisk steg i argumentasjonen deres ved at de får fastsatt en grunnleggende definisjon som påvirker både de videre argumentene deres og hva de vil godta som et riktig svar.

For å argumentere om at tegningen viser en sekskant, bruker Ida sine matematiske kunnskaper om geometriske figurer samt tegningen for å argumentere for svarene sine. Dette beskriver Lavy (2006) som sammensatt argumentasjon, ved at argumentasjonen både er satt sammen av et tegnet eksempel av en figur og en matematisk forklaring som fungerer som brobygger mellom figuren og svaret. Selv om ingen nevner tegningen med ord, ser begge to på tegningen og teller antall kanter som er i figuren. På den måten tar tegningen del i argumentasjonen, og spiller en viktig rolle for å validere argumentene deres.

Ser man etter Lithners (2008) tre kvaliteter som han mener er viktige for å vurdere argumentasjon, bærer argumentasjonen i dette dialogutdraget preg av både konstruktivitet og validitet. Ved at argumentasjonen består av både hva som var blitt tegnet og hvilke matematiske følger tegningen hadde for spørsmålet om det var en femkant eller ikke, validerer Knut og Ida svarene sine. I argumentasjonen sin er Knut og Ida også spisset i den forstand at de kun tar med nyttig fakta og begrunner svarene sine.

Når det kommer til Lithners (2008) kvalitet, evnen til å overtale, er det Ida som får det siste ordet om figuren er en femkant eller ikke. Dette kan være fordi Knut gikk inn i dialogen som en spørrende person som var usikker på svaret. Han viser denne usikkerheten med at han teller flere ganger, mens Ida virker mer overbevisende når hun teller én gang og deretter bestemt sier at det er en sekskant. Med å telle høyt gir hun også Knut en forklaring på hvorfor hun mener figuren ikke er en femkant men en sekskant. Knut stiller ikke spørsmål videre om figuren, noe som kan være et tegn på at han tror på Ida og er enig i at figuren må være en sekskant fordi den har seks kanter.

Til slutt er det også verdt å nevne at selv om det aldri ble uttrykt, av verken Knut eller Ida, er begge to innforstått med at det er antall kanter på figuren som avgjør om den er en femkant eller ikke. Det at de ikke ser det som nødvendig å forklare dette ovenfor den andre, forklarer Yackel (2001) med begrepet oppfattet-som-forstått (se 2.5.5). Dette kan bety at definisjonen på ulike geometriske figurer er noe som har blitt diskutert så pass mye at det har blitt en kjent matematisk praksis som ikke trenger begrunnelse.

4.1.2 «Vent, hvor mange grader er det i en femkant?»

Etter å ha blitt enige om at figuren i tegningen til Knut er en femkant, dukker spørsmålet om vinkelstørrelser opp. Knut og Ida virker usikre på om alle vinklene må være like eller om figuren kan se ut som et hus. En av de voksne svarer på elevenes spørsmål med at de må selv finne ut hva som er kravene for en femkant og arbeide ut ifra hva de tenker er riktig. Ida mener da at vinklene må være like «fordi da kan det se ut som det samme fra alle sidene». Videre fortsetter dialogen som følger:

Knut: Hvordan skal vi gjøre det? Vent, hvor mange grader er det i en femkant?

Ida: Jeg aner ikke. En firkant, det er 90 grader, femkant, det aner jeg ikke. Det er hvert fall over 90 grader.

[Scratch henger seg opp og pennen forsvinner. De bruker litt tid på å hente den frem igjen.]

Ida: Der. Sånn. Ja. Da må vi ha over 90 grader tror jeg. Det må være over 90 grader liksom, for alle er stumpe vinkler.

Knut: 120?

Ida: Ja, kanskje det. Jeg vet ikke.

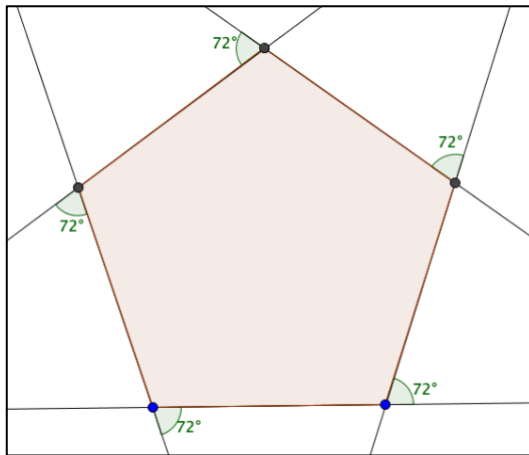
Knut: Vi bare prøver da.

Knut åpner opp for matematisk argumentasjon når han spør om hvor mange grader det er i en femkant. På dette spørsmålet svarer Ida først «Jeg aner ikke», men begynner så å resonnerer seg frem til et forslag. Hun sier at «En firkant, det er 90 grader» og videre at en femkant er «hvert fall over 90 grader». Ida poengterer først at i en firkant er vinklene 90 grader. Det virker som om dette er noe hun føler seg sikker på, og på bakgrunn av vinklene i en firkant, sier hun at vinklene i en femkant i alle fall er over 90 grader og at alle vinklene må være stumpe.

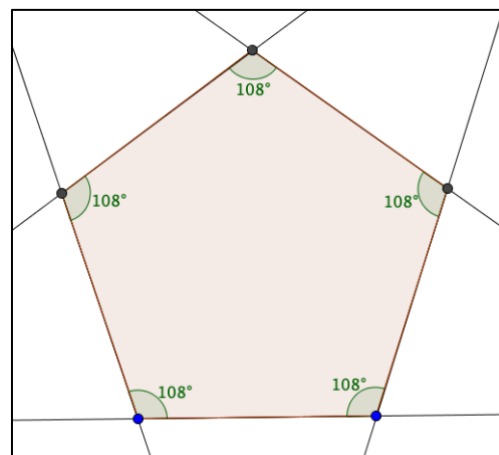
Gjennom Ida sine argumenter viser hun hvordan hun oppfatter at vinkelstørrelser endrer seg ut ifra hvor mange vinkler det er i en figur. Ida har oppdaget at jo flere vinkler det er i en regulær mangekant, dess større blir vinklene. Uheldigvis for henne fungerer Scratch på en slik måte at når man bruker vend-funksjonen, vender pennen om den ytre vinkelen (se figur 6) og ikke de vinkelen som er på innsiden av mangekanten (se figur 7) som man normalt ser på. Det blir da vanskelig å se med det blotte øyet hvilken vinkelstørrelse de bør bruke for å programmere en femkant. For å kunne regne seg frem til hvilken vinkelstørrelse man må bruke for å programmere en femkant, kan man ta $360^\circ : 5 = 72^\circ$. Dette blir det motsatte av hva algoritmen

for vinkelsummer gjør, der man finner vinkelsummen for den innerste vinkelen. Ettersom man finner nabovinkelen til den innerste vinkelen, kan man tenke seg at man trekker vinkelstørrelsen til den innerste vinkelen fra 180 grader, for å finne vinkelstørrelsen man trenger i Scratch. Hos en femkant får man da $180^\circ - 108^\circ = 72^\circ$.

Figur 6. Eksempel på ytre vinkler.



Figur 7. Eksempel på indre vinkler.



Logikken Ida bruker her, er dermed helt riktig da vinklene i en regulær mangekant blir større. Dette gjør at validiteten til Ida sine argumenter er god, til tross for at hun presenter et forslag til en løsning som blir feil i Scratch. Ida er også konstruktiv i sin argumentasjon og selv om forslaget hennes ikke blir riktig i Scratch, blir Knut overbevist om at forslaget til Ida er verdt å prøve. Dette kan sees ved Knut sine ytringer, «120?» og «Vi bare prøver da.», hvor han anerkjenner det Ida har foreslått og er villig til å teste det ut.

Ida bruker resonnering for å komme frem til argumentet om størrelsen på vinklene. I delkapittel 2.3.2 trekkes det frem at det er når elevene er kreative og tenker utenfor algoritmene at det oppstår gode resonnementer og argumenter. I dette tilfellet har Ida gått inn med sine forkunnskaper om vinkelstørrelsen til en regulær firkant og gjort seg noen tanker om hvordan dette påvirker vinkelstørrelsen til en femkant. Ettersom hun ikke finner fort frem til den riktige løsningen eller bruker en algoritme for å løse oppgaven, kan det virke som om hun ikke har møtt på et liknende problem tidligere. Dersom det er slik at Ida her er nytenkende i sin argumentasjon og lager en argumentasjonssekvens som hun ikke har brukt før, fyller hun kravet, nyhet, for kreativ argumentasjon som ble nevnt i delkapittel 2.3.2. Argumentene hun bruker er matematisk forankret ved at de bygger på matematiske egenskaper som er relevante for problemet, samtidig som de også har den funksjonen at de støtter opp om hvorfor den

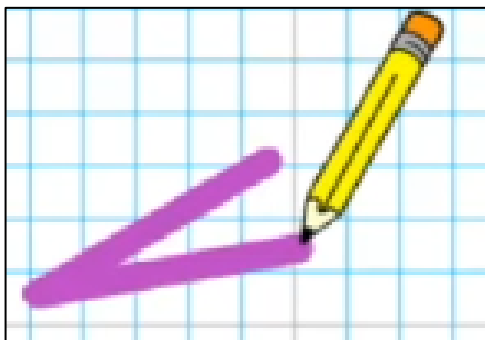
endelige påstanden er rimelig. Med dette har hun også oppfylt de to siste kravene for kreativ argumentasjon, nemlig matematisk forankring og plausibilitet.

I delkapittel 2.4.2 står det beskrevet hvordan Lavys (2006) kategori, sammensatt argumentasjon, kan bli sett på som når elever bruker matematiske vurderinger og testing av ulike løsninger for å komme frem til den endelige løsningen. I dette dialogutsnittet gjør Ida nettopp dette, da hun bruker det hun vet om vinkelstørrelser og geometriske figurer for å avgrense hvilke vinkelstørrelser det er verdt å teste ut. Argumentene hun gjør seg her er at en firkant har 90 grader, og at på tegningen av en femkant ser vinklene ut til å være stumpe vinkler. Dermed må femkanten sin vinkelstørrelse være mer enn 90 grader. Ida kobler tallegenskapene til vinklene sammen med den geometriske formen på mangekanten, og viser gode matematikkferdigheter når hun bruker det hun har av informasjon om vinklene til en femkant til å komme med en påstand om vinkelstørrelsen.

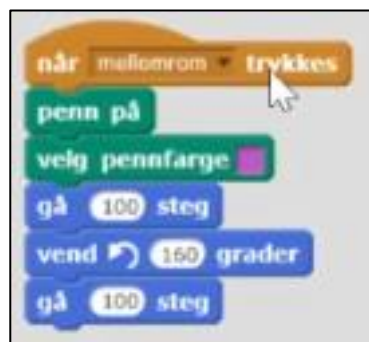
4.1.3 «Åja, men da kan ikke vi ha 60 grader.»

Videre skal Knut og Ida teste ut teorien om å vende 120 grader, men før de får gjort det henger Scratch seg opp. Når de har fått fikset Scratch igjen, bestemmer de seg for at 120 grader er litt lite og trykker inn 160 grader. Ida trykker så på start-knappen og det tegnes en vinkel, se figur 8 og 9.

Figur 8. Skjerm bilde av figuren som ble tegnet.



Figur 9. Skjerm bilde av kodetråden.



Ettersom det ikke vises hvilken vei pennen roterer, kan det se ut som om pennen har tegnet en spiss vinkel. Etter å ha sett strekene som ble tegnet, sier Ida «Nei, da går den jo bare innover» og «Da må vi jo ta mindre grader». Selv om Ida ikke forklarer hva som menes med «innover» eller «mindre grader», virker det som om hun har forstått at pennen tegnet en stump vinkel fordi hun foreslår å ha mindre grader. Uten videre diskusjon skriver Ida inn 60 grader og setter inn flere kodeklosser med *gå 100 steg* og *vend 60 grader* slik at de står fem ganger etter hverandre.

Når de nå trykker på startknappen, tegnes det fem sammenhengende streker som ser ut som en sekskant men med én kant for lite. Videre fortsetter dialogen:

Ida: Ja.

Knut: Vi mangler en. Se.

Ida: Ja, jeg skjønner ikke helt.

Knut: Nei ...

[De trykker start på nytt, og den samme figuren tegnes.]

Ida: Da må vi ta gå.

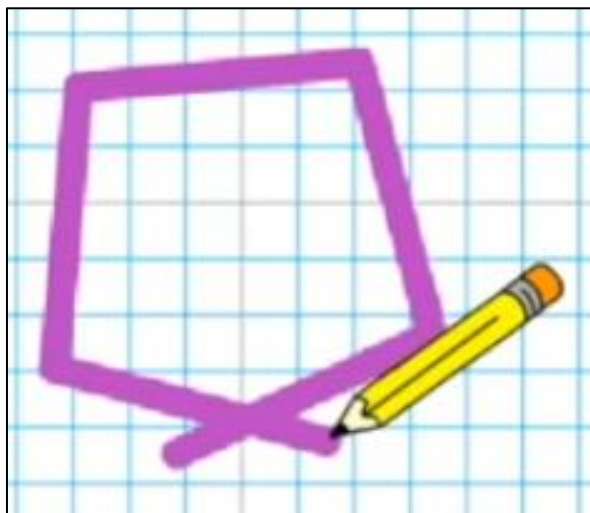
Knut: Nei, men nei, da blir det en sekskant.

Ida: Åja, men da kan ikke vi ha 60 grader. Da må vi ha mindre. Nei mer.

[De endrer gradene til 80 (se figur 10) og trykker start. Det tegnes en figur med fem kanter, men den siste kanten krysser den første (se figur 11).]

Figur 10. Skjerm bilde av kodetråd med 80 grader.

Figur 11. Skjerm bilde av tegnet figur med 80 grader.



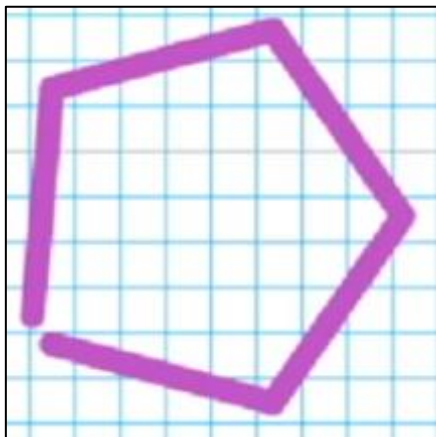
Ida: Hehe, ja da må vi ta.

Knut: Vent da. Er ikke det en sekskant? Femkant.

Ida: Jo, det er det vi skal ha. 70 grader da. 70 er sikkert greit, siden 60 ikke på en måte, da prøver vi 70. Hehe.

[De endrer gradene til 70 og trykker start. Det tegnes en femkant, men den siste kanten stopper rett før den treffer den første kanten. Figuren er ikke lukket. Se figur 12.]

Figur 12. Skjerm bilde av tegnet figur med 70 grader.



I dette dialogutsnittet er det flere ting som skjer. Det kan med første øyekast virke som om de i arbeidsprosessen bare tester ut tilfeldige tall, da de er raske med å trykke inn de ulike gradene uten mye diskusjon mellom prøvene. Ser en derimot nærmere på dialogutsnittet og deler det opp, oppdager man at Knut og Ida bruker matematiske vurderinger for å begrense tallene de tester ut.

Først finner de ut at med kodeklossen *vend 60 grader* får figuren for mange kanter før den er lukket, og at de derfor må ha en større vinkel for å få figuren til å rotere nok innover slik at den blir lukket etter fem roteringer. Fra tidligere av vet de at graden også må være mindre enn 160 grader, og velger å teste ut 80 grader. De gjør her en form for sammensatt argumentasjon og bruker det Hana (2014) definerte som systematisk prøving og feiling, der de ut ifra tidligere resultater gjør både matematiske vurdering samt velger et tilfeldig tall som er større en 60 og mindre enn 160. I denne delen av dialogen bidrar både Knut og Ida sammen til argumentet om at de må bruke andre grader. Knut konstaterer først at de ikke kan ha 60 grader, fordi «da blir det en sekskant». Dette følger Ida videre opp og sier «Åja, men da kan ikke vi ha 60 grader. Da må vi ha mindre. Nei mer». Igjen forklarer ikke Ida hvorfor de må ha en vinkelstørrelse som er høyere enn 60 gradere, og det er dermed uvisst om Knut forstod bakgrunnen til Idas påstand.

I neste del har de testet ut figuren med 80 grader, og ser at figuren roterer så mye at den krysser seg selv. Knut definerer figuren som en femkant, mens det kan tolkes som om Ida fortsatt ikke er helt fornøyd med resultatet. Dette kan sees med ytringen «70 er sikkert greit», hvor hun kommer med et nytt forslag til hvilken grad de kan teste ut. Videre begrunner Ida det å teste ut 70 grader muntlig med å si «siden 60 ikke på en måte», hvor jeg tolker dette som at hun mente at 60 grader ikke var riktig. Tar man skjerm bildet av figuren med 80 grader med som et

argument, har Ida to argumenter for at de skal teste ut 70 grader. Det første er som nevnt den muntlige ytringen om at det var for lite med 60 grader, og det andre argumentet er figuren med 80 grader som viste at denne vinkelstørrelsen ga for store vinkler. I Ida sin argumentasjon om å teste ut 70 grader har hun utvidet argumentasjonen sin fra den forrige påstanden på å teste ut 80 grader, ved å legge til argumentet at vinkelstørrelsen 80 grader ga vinkler som var for store. Ida befinner seg dermed i Lavys (2006) kategori, utvidet argumentasjon, der hun har gått fra det sammensatte argumentet *gradene må være større enn 60 grader* til å utvide argumentet til *gradene må være mellom 60 og 80*.

En annen gruppe (med jentene Tiril og Line) brukte til slutt algoritmen, 360 delt på antall vinkler, for å finne vinkelstørrelsene. Her startet de med den samme strategien som Knut og Ida, nemlig å teste ut og gradvis eliminere ulike vinkelstørrelser. Midt i et forsøk stoppet derimot Tiril plutselig opp, og skrev ned regnestykket $360:5 = 72$. Deretter satte hun inn 72 grader i kodetråden, trykket start og det ble tegnet en likevinklet femkant. Selv om denne løsningen ga et mer nøyaktig og riktig svar enn Knut og Ida sin løsning, forsvant all form for argumentasjon og utforskning om matematikk når de kom på standardalgoritmen. I fremgangsmåten til Knut og Ida bruker de egne argumenter som ikke baserer seg på standardalgoritmer, og er kreative i måten de løser oppgaven på. Arbeidsmetoden til Knut og Ida gjør at de, gjennom argumentene for ulike løsninger, får utforsket det matematiske om femkant på en måte som standardalgoritmer ikke legger like godt til rette for. Dette støtter opp om det Lithner (2008) trekker frem som en av ulempene med argumentasjon og resonnering som baserer seg på algoritmer.

I dette dialogutsnittet er Knut igjen opptatt av å konstatere hvilken type figur det er, og kommer frem til at dette er en femkant. Knut sitt fokus på å definere figurtypene, kan være et tegn på at han er veldig opptatt av å komme i mål med oppgaven. I tillegg til dette var Knut også villig til å slå seg til ro med at figuren de programmerte var en fullstendig femkant, selv om den ikke var lukket. Ida fortsatte derimot videre og la til noen skritt for å få den siste kanten til å treffe den første kanten. Dette viser til et fokus på sluttproduktet og ikke på arbeidsprosessen hos Knut. Når en arbeider med argumentasjon er det arbeidsprosessen som er viktigst da det er her mulighetene for matematisk argumentasjon ligger. Knut sitt fokus på å bli ferdig med oppgaven, kan dermed ha vært en hindring for den matematiske argumentasjonen. Dette kan også være en av grunnene til hvorfor han ofte ikke stiller spørsmål til Ida sin påstander som ikke blir ordentlig argumentert for.

4.2 Knut og Ida programmerer en sirkel

Neste oppgave Knut og Ida skulle løse, var å programmere pennen til å tegne en sirkel. Utfordringen med å programmere en sirkel, er at det ikke er like tydelig hvor vinklene er. Dette gjør det vanskelig å bruke Ida sin tidligere taktikk, der hun ser på vinklene og avgjør hvor store de er ut ifra hvordan de ser ut.

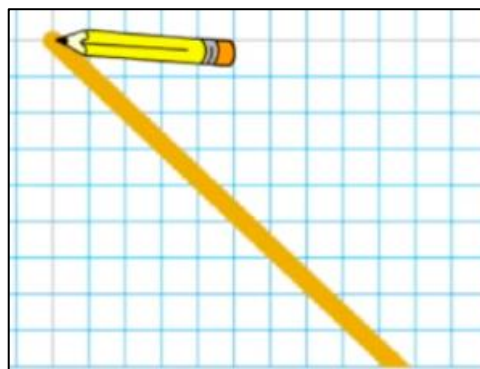
4.2.1 «Det er på grunn av at den vender først 360 grader og så bare går den.»

Når Knut og Ida starter med oppgaven blir det raskt konstatert at løsningen har noe med 360 grader å gjøre. Først prøver de å få pennen til å vende 360 grader med å bruke de to kodeklossene *gå 300 steg* og *vend 360 grader* (se figur 13). Når de så trykker start, tegnes det en lang rett strek (se figur 14). Videre fortsetter dialogen:

Figur 13. Skjerm bilde av kodetråden.



Figur 14. Skjerm bilde av tegnet figur med vend 360 grader.



Knut: Heh, hvordan ...

Ida: Vi kan ikke ha *gå 300 steg* først, vi må ha ...

Knut: Men hvordan ... Det betyr at den vender seg.

Ida: Vend [flytter på klossene slik at «vend 360 grader» står først]. Å, jeg skjønner ikke hvordan vi skal klare sirkel. Vi prøver nå da.

[Har kodene *vend 360 grader* og *gå 300 steg*. Den samme streken tegnes.]

Knut: Det går jo bare i en rett strek. Ja, men det er på grunn av at den vender først 360 grader, og så bare går den.

Ida: Ja, jeg vet det.

I dette dialogutsnittet prøver Knut og Ida å forstå hvorfor pennen tegner en rett strek med kodeklossene *gå 300 steg* og *vend 360 grader*. Først legger Ida opp til at det er på grunn av at kodeklossen *gå 300 steg* står øverst slik at pennen først går rett frem før den roterer 360 grader. Dette er midlertidig feil. Ettersom at *vend 360 grader*-klossen gjør at pennen roterer helt rundt til start igjen, har det ikke noe å si hvor den står i kodetråden. Med Knut sin ytring, «Ja, men det er på grunn av at den vender først 360 grader og så bare går den», er det mulig at han har forstått at rekkefølgen ikke har noe å si fordi kodeklossen *vend 360 grader* gjør at pennen roter rundt til startposisjonen sin uten å gå stegene mellom gradene.

Det endelige målet til Knut og Ida, er å finne ut hvordan de skal programmere en sirkel. Det som vises i dette dialogutsnittet er en del av prosessen mot dette endelige overordnede målet. Ida argumenterer her for løsningen *vend 360 grader* og *gå 300 steg*. Argumentene hun har for denne påstanden er først at en sirkel er 360 grader og at kodeklossen *vend 360 grader* roterer pennen som en sirkel. Disse argumentene følger en logisk tankegang, hvor det gir mening at løsningen må være å vende 360 grader når en sirkel har 360 grader rundt. Validiteten til Idas argumenter er derfor der, nettopp fordi argumentene logisk gir mening til tross for at de er bygget på en misoppfatning av hva kodeklossen *vend 360 grader* gjør.

I neste dialogutsnitt som blir presentert, kommer det frem at Knut opprinnelig hadde en annen idé for å løse oppgaven, men valgte å teste ut Ida sin idé først. Dette kan tyde på at Idas argumenter var overbevisende ovenfor Knut, og at han syntes at de ga mening. En annen mulig forklaring på hvorfor Knut gikk med på Ida sitt forslag kan være at Knut ofte har erfart at Ida har riktig og valgte derfor å stole på henne. Dette er til tross for at hun tar seg lite tid til å forklare resonneringen sin. Hun bruker heller ikke representasjoner i argumentasjonen, noe som gjør det vanskelig å plassere henne i Lavys kategorier da disse har krav om en eller annen form for representasjonsbasert argumentasjon.

Etter å ha sett resultatet av kodeklossene til Ida, argumenterer Knut for hvorfor løsningen til Ida ikke er riktig. Det muntlige argumentet Knut har for denne påstanden er at kodeklossen *vend 360 grader* gjør at pennen roterer rundt til startpunktet, og så går den etterpå. I tillegg til dette argumentet refererer også Knut sin ytring til skjermbildet av pennen som tegner en rett strek. Knut brukte derfor sammensatt argumentasjon ved at skjermbildet fungerer som et bevis på at løsningen til Ida ikke var riktig, og ved den muntlige ytringen som forklarer hvorfor det ikke var riktig.

Knut sin argumentasjon er basert på de samme logiske trekkene som Idas argumentasjon der forskjellen som skiller dem er oppfatningen av hva kodeklossen *vend 360 grader* gjør. Videre er Knut konstruktiv og spesifikk i argumentasjonen sin, ved at det ikke er tvil om hva han mener med ytringen sin «Ja, men det er på grunn av at den vender først 360 grader og så bare går den». Til slutt kan ytringen «Ja, jeg vet det» tyde på at Ida fant Knut sine argumenter som overbevisende, og er enig i Knut sin mening om at det ikke er riktig å bruke kodeklossen *vend 360 grader*.

4.2.2 «Ja, går det an?»

Etter å ha lagt fra seg Ida sitt løsningsforslag går de i gang med å lete etter andre kodeklosser som kan fungere. Etter å ha lett en stund sier Knut:

Knut: Jeg så sirkel med en gang og tenkte hvordan skal vi klare det. For enten må vi gå sånn 15, rundt 1 grad, og så må vi gå sånn 10 skritt, og så 1 grad, og så etter hverandre etter hverandre etter hverandre.

Voksen: Ja, går det an?

Knut: Men det tar veldig lang tid.

[Den voksne spør om det går an å gjøre det i Scratch uten at det tar lang tid.

Knut og Ida finner frem gjenta-klossen.]

Ida: Vi skal vende. *Vend 1 grad, gå 1 steg*. Eller nei, kanskje ...

Knut: Jo jo.

Ida: 3 steg. Eh jeg vet ikke ...

[Ida trekker inn kodeklossen *vend 1 grad* og *gå 3 steg*.]

Knut: Og så ...

Ida: Gjenta 360 grader.

[Ida drar inn kode-klossen «gjenta x ganger» og skriver inn 360 (se figur 15).]

Figur 15. Skjerm bilde av kodetråd med gjenta 360 ganger.



I dette dialogutsnittet går Knut i gang med å forklare hvordan de kan løse oppgaven. Han forklarer ikke eksplisitt hvorfor han mener at løsningen *vend 1 grad og gå noen skritt* flere ganger etter hverandre vil gi en sirkel, men man kan tolke det som om han har forstått at en sirkel består av 360 vinkler som alle roterer 1 grad i samme retning. Det kan også virke som om Ida har forstått det matematiske bak Knut sin ytring, ved at hun senere sier «gjenta 360 ganger» uten at Knut har sagt at de må gjenta det så mange ganger.

Knut sin påstand i dette dialogutsnittet er at de må «gå sånn 10 skritt, og så 1 grad», og så må de gjenta dette flere ganger «etter hverandre». Han forklarer derimot aldri hvorfor denne fremgangsmåten vil gi en sirkel, og argumenterer derfor heller ikke for hvorfor løsningen hans er riktig. Hvorfor han ikke argumenterer for påstanden sin, kan blant annet være på grunn av at en av de voksne rett etter Knut sin ytring sier «Ja, går det an?». Denne ytringen kan ha blitt tolket av Knut som en bekreftelse på påstanden hans, som videre kan ha ført til at han ikke følte behovet for å overbevise de andre om at hans påstand var riktig. Det er også mulig at Knut ikke hadde argumentert for påstanden sin, selv om den voksne ikke hadde bekreftet Knut sin ytring.

Selv om Knut sitt forslag til en løsning på oppgaven gir oss en viss innsikt i Knut sine matematiske kunnskaper om sirkel, kan man ikke med sikkerhet si at Knut har forstått at en sirkel kan sies å bestå av 360 vinkler med 1 grad rotasjon. Hadde Knut argumentert for løsningsforslaget sitt ville det vært mulig å få enda bedre innsikt i Knut sine matematiske kunnskaper om sirkel.

Det at Knut og Ida har fått i oppgave å programmere pennen til å tegne en sirkel, gjør at de må gå inn i detalj på hva man faktisk gjør når man tegner en sirkel. Et eksempel på dette er at de blir bevisste på at for å tegne en sirkel kan man rotere 1 grad mellom 360 forflytninger. Dette er aspekter ved en sirkel som man normalt ikke tenker over når man tegner en sirkel for hånd. Det er dermed mulig at det er selve programmeringsdelen av oppgaven som la til rette for at Knut og Ida produserte disse gode matematiske resonnementene. Dette kan sies å vise et matematisk læringspotensial ved programmering som man ikke på samme måte finner i arbeid med papir og blyant.

Det virker som om de matematiske resonnementene som Knut og Ida gjør seg her er resonneringssekvenser de ikke har tenkt på tidligere ettersom de ikke brukte algoritmer, men heller egne uferdige beskrivelser av løsningen. De matematiske resonnementene som Knut og

Ida gjorde i dette dialogutsnittet oppfatter jeg derfor at bærer preg av det Lithner (2008) beskriver som en *nyhet*. I tillegg bruker de matematiske egenskaper til en sirkel for å finne løsningen til å programmere sirkelen. Resonnementene er derfor også tydelig matematisk forankret ved at de matematiske aspektene er relevante for oppgaven og støtter opp om valget av løsningen. Selv om Knut og Ida ikke har noen matematiske argumenter i denne delen av arbeidsprosessen, gjør de seg likevel opp mange gode matematiske kreative resonnementer på bakgrunn av at de inneholder en nyhet, er matematisk forankret og har plausibilitet.

De andre gruppene som deltok på datainnsamlingen hadde, ved programmering av en sirkel, en liknende arbeidsprosess som Knut og Ida, hvor de først testet ut forskjellige funksjoner i Scratch og endte til slutt opp med å gjenta kodeklossene *gå 1 steg* og *vend 1 grad 360* ganger. Selv om disse gruppene endte opp med den samme løsningen som Knut og Ida, hadde de ikke like mange matematiske argumenter som forklarte hvorfor disse kodene kom til å gi en sirkel. Det er vanskelig å si noe helt sikkert om hvorfor dialogene ble såpass forskjellige mellom gruppene, til tross for at de gjorde det samme i Scratch. En mulighet er at gruppesammensetningen har påvirket hvordan elevene pratet sammen. Som Lithner (2008) påpeker i sin artikkel, vil de sosiomatematiske normene påvirke hvilke matematiske aspekter elevene føler at de må argumentere for. Disse normene kan være forskjellig oppfattet av elevene, og vil på den måten føre til at den matematiske argumentasjonen blir ulik mellom gruppene.

En annen mulig forklaring på denne forskjellen mellom parene er at maktbalansen mellom Knut og Ida var nokså likt fordelt. Med dette menes det at det ikke var noen som hadde tydelig mer makt til å bekrefte eller avkrefte det den andre sa. De stod sammen i å bestemme hvordan de skulle programmere de ulike figurene, og måtte derfor av og til argumentere for å overbevise den andre om påstandene sine. Hos de andre gruppene var det derimot ofte én elev som styrte mest over datamaskinen og som fungerte som en leder i samarbeidet. Særlig hos Tiril og Line kan en ofte se at Line kommer med forslag som Tiril deretter godtar eller forkaster. Dette førte til at Line sjeldent kom med forslag, men når hun gjorde det argumenterte hun godt for seg. Tiril på den andre siden hadde en tendens til å trykke inn kodeklosser og tall på datamaskinen uten å forklare hvorfor.

4.3 Knut og Ida programmerer ei stjerne

Knut og Ida har nå kommet til den delen der de skal programmere pennen til å tegne ei stjerne. Frem til nå har de på alle oppgavene brukt taktikken *gå noen skritt og vend noen grader* for så å gjenta dette like mange ganger som de skal ha vinkler. Med firkanten gikk de og roterte 90 grader fire ganger, og med femkanten gikk de og roterte 70 grader fem ganger. Videre gjorde de det samme med sirkelen der de gikk tre skritt og roterte én grad 360 ganger.

4.3.1 «Det er ulike vinkler tror jeg.»

Det følgende dialogutsnittet starter med at Knut og Ida oppdager at strategien de har brukt på de andre figurene ikke fungerer lenger fordi stjerna roterer med ulike vinkler. Det første begge gjør når de starter på oppgaven, er å tegne hver sin stjerne på arket sitt. De har til nå ikke begynt å kode ei stjerne i Scratch. Videre fortsetter dialogen:

Ida: Det er ulike vinkler tror jeg.

Knut: Nei, men jeg suger på å tegne stjerne.

Ida: Ja, men vi kan bare gjøre sånn her. [Viser til tegningen sin av ei stjerne der hun har tegnet to trekanten oppå hverandre.]

Knut: Da må vi gå sånn 50 steg.

Ida: Femkanta stjerne kan vi ta.

Knut: Ja, og da må vi gå 50 steg og så vende, eh hvilken grad er dette? Jeg tror dette er sånn halvparten, at det er sånn 30, 60 grader.

Ida: Men den har i hvert fall ulike, nei den har helt like grader overalt tror jeg.

Knut: Ja, men what?

Ida: Nei, den har ikke like grader overalt. Nei, den har noen stumpe og noen spisse [ser på tegningen sin]. Se da [Viser tegningen sin].

Knut: Ja, vent [fikser med start-knappen i skriptet]. Når ... nei.

Ida: Fordi der er den litt stumpere enn den [peker på de ulike vinklene på tegningen].

I dette dialogutsnittet fokuserer Ida på å konstatere at stjerna har ulike vinkler, mens Knut vil gå i gang med finne hvor mange skritt pennen skal gå og hvor mange grader den skal rotere. Knut kan dermed virke litt utålmodig og vil gå raskt i gang med kodingen, mens Ida vil drøfte og vurdere hvordan vinklene i stjerna oppfører seg.

Etter litt drøfting kommer Ida frem til påstanden «den har ikke like grader overalt», og argumenterer for dette med de matematiske utsagnene «den har noen stumpe og noen spisse» og «fordi der er den litt stumpere enn den». I tillegg til disse muntlige argumentene, refererer hun aktivt til tegningen både ved å si «Se da» og peker på de ulike vinklene på tegningen mens hun snakker. Ida gjør i dette dialogutsnittet en representasjonsbasert argumentasjon som er sammensatt av muntlige ytringer og flere henvisninger til en tegning av ei stjerne. Tegningen fungerer her som en støtte til Idas muntlige ytringer, der hun peker eller referer til tegningen for å vise at det hun sier stemmer overens med hvordan ei stjerne ser ut. Det at hun har flere argumenter for hvorfor det er ulike vinkler i stjerna, styrker validiteten og konstruktiviteten hos Idas argumentasjon.

I argumentene sine bruker Ida matematiske begreper som spisse og stumpe vinkler for å argumentere for hvorfor stjerna har ulike vinkler. Det at hun bruker disse fagbegrepene er også med på å styrke både validiteten, konstruktiviteten og evnen til å overtale. Ved å bruke fagbegrepene beskriver hun hva som matematisk skjer med de ulike vinklene, og bidrar derfor positivt til argumentasjonen. Hun forklarer derimot ikke hva begrepene betyr, noe som gjør at man må anta at hun har riktig forståelse av disse begrepene. Det at hun ikke forklarer begrepene, kan også bety at begrepene er så godt etablert hos henne at hun tenker at begrepene er forstått også fra Knut sin side.

Det kan virke som om Knut ikke er så interessert i det Ida sier, da han prøver å begynne på en setning om hvilke kodeklosser de skal bruke. En mulig forklaring på dette er at han synes at Idas observasjon om vinklene er en selvfølge, og dermed ikke vits å diskutere. Ved å se på tegningen av ei stjerne er det mulig at Knut lett ser at vinklene er forskjellige. En annen mulig forklaring er at han, som tidligere nevnt, er opptatt av å bli raskt ferdig med oppgaven.

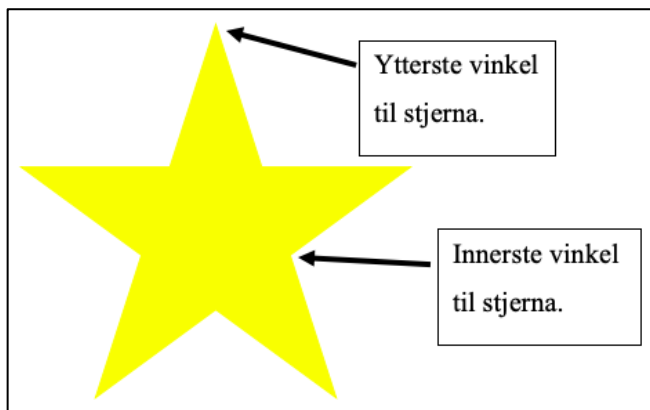
Selv om Idas argumentasjon om vinklene ikke er så matematisk avansert (og kanskje til og med litt for enkel til å være på 7. trinn) er dette en observasjon som for eksempel Tiril og Line ikke oppdaget i sin arbeidsprosess mot ei stjerne. Dette førte til at de ikke fikk til å programmere ei stjerne med taktikken *gå noen steg og vend noen grader*, og skiftet taktikk til å bruke kodeklossen *gå til x-koordinat og y-koordinat*. Når Tiril og Line brukte disse kodeklossene, forsvant nesten all mulighet til å diskutere matematikk ettersom de nå bare måtte notere ned de riktige x- og y-koordinatene. Dialogen bestod dermed av mye oppramsing av tall, og lite diskusjon om de matematiske aspektene ved koordinatsystemet. Til tross for at denne taktikken

i teorien skulle ha fungert, var Tiril og Line for unøyaktige når de noterte ned de ulike koordinatene og endte opp med en figur som verken var lukket eller så ut som ei stjerne. Dette funnet viser at valg av strategi og kodeklosser i Scratch kan være med på å påvirke mulighetene til matematisk argumentasjon, og eventuelt også kvaliteten på argumentene.

4.3.2 «Ja, og så hvor mange grader.»

I neste dialogutsnitt skal Knut og Ida finne ut hvilke kodeklosser de må bruke for å få pennen til å tegne ei stjerne. De har bestemt seg for å bruke kodeklossen *gå 50 steg* for å flytte på pennen. De har også laget en plan på hvor pennen må gå og hvor den må vende. Det som gjenstår er å finne ut hvor mange grader pennen må rotere på de ulike stedene i stjerna. For å lettere kunne forstå hva Knut og Ida mener når de snakker om de ulike vinklene, har jeg i figur 16 satt på navn på de to ulike vinklene det er snakk om. Gjennom dialogen kommer det frem at de ser for seg at stjernas ytterste vinkler er like, og at alle de innerste vinklene er like.

Figur 16. Bilde av stjerne med navn på de ulike vinklene.



Videre fortsetter dialogen:

Knut: Ja, og så hvor mange grader.

Ida: Ja, det er det som er.

Knut: Hvor mange grader er dette? [Knut tegner ei ny stjerne.]

Ida: Kanskje 60 grader?

Knut: Vent da, den må først vende her. [Knut peker på vinkelen innerst i stjerna]

Ida: Nei, ikke 60, jeg mener hundre og et eller annet.

Knut: Nei, men sånn, den må vende annerledes grader. For her opppe så vender den, eeh ...

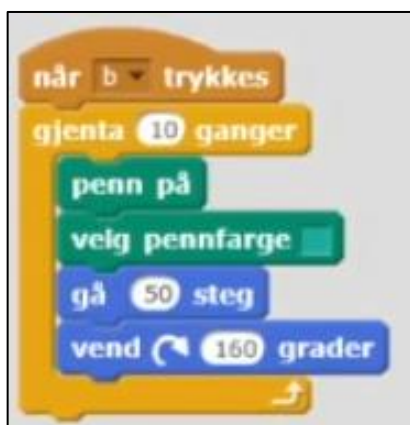
Ida: Jeg vet. Den vender liksom, det er en stumpere vinkel enn ... [Peker først på vinkelen innerst i stjerna, og deretter på den ytterste vinkelen.]

Knut: Ja, men fordi den går, vent da ...

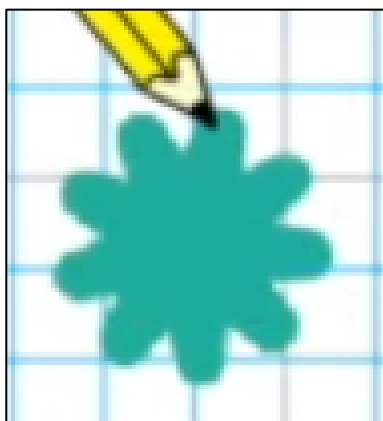
Ida: Den må vende over hundre, jeg tror den må vende 160, vent da ...
[Ida og Knut prater i munnen på hverandre. Ida trykker inn 160 grader.]

Knut: Først her må den vende ... Her [peker på vinkelen innerst i stjerna] må den vende 90 hvert fall. Her må det være 90, sånn at den går ut sant. Så det må hvert fall være annen hver 90, så må det være 50 steg, og så spørsmålstegn, 90, og så 50 steg, og så spørsmålstegn, og så ... Vi må bare finne ut av den graden [peker på vinkelen ute på spissen av stjerna].
[Mens Knut sier det ovenfor, trykker Ida på startknappen ti ganger etter hverandre med kodene *gå 50 steg* og *vend 160 grader* (se figur 17). Det tegnes da en figur som Knut og Ida mener er ei stjerna (se figur 18).]

Figur 17. Skjerm bilde av kodetråden til stjerna.



Figur 18. Skjerm bilde av stjerna som ble tegnet.



Knut starter først med et ønske om å fastslå den innerste vinkelen i stjerna. Ida forslår så 60 grader, men endrer siden svaret til «hundre og et eller annet». Videre kommer Knut med påstanden «den må vende annerledes grader» og prøver å begynne på en forklaring for hvorfor. Ida sier seg enig i Knut sin påstand, og kommer med argumentet at den ene vinkelen er mer stump enn den andre. Det hun sier her er at hun kan se med det blotte øye at disse to vinklene er forskjellige, fordi den ene er mindre spiss enn den andre. Det at hun ser på den ene vinkelen som mer stump enn den andre, kan være grunnen til hvorfor hun foreslår at den innerste vinkelen må være over 100 grader. Dersom dette er en riktig tolkning av Idas ytringer har hun presentert et matematisk argument for hvorfor den innerste vinkelen vender mer enn 100 grader.

Den siste ytringen til Knut inneholder mye informasjon, hvor han på en oversiktlig måte presenterer tankene sine. Han er først opptatt av hvordan den innerste vinkelen vender, og peker på denne vinkelen på tegningen. Han har ingen muntlige argumenter for hvorfor han mener at den innerste vinkelen er 90 grader, annet enn at han peker på vinkelen i tegningen og sier «sånn at den går ut». Dette definerer Lavy som et grunnleggende argument der Knut ikke har en matematisk forklaring på påstanden sin.

Deretter konstaterer Knut at annenhver vinkel må være 90 grader uten å forklare hvorfor. Man kan se på stjerna som en symmetrisk figur der spissene i stjerna roterer om midten. Fra programmeringen sin side, vil dette bety at man kan finne de riktige kodeklossene for én spiss og gradene for spissens rotasjon for så å gjenta dette til man kommer rundt til start. Knut forklarer derimot ikke dette til Ida, og det er derfor mulig at han antar at hun også har oppdaget at stjerna er rotasjonssymmetrisk.

I oppgaven om ei stjerne bestemmer Knut og Ida seg for å løse oppgaven med å programmere ei stjerne med fem spisser. For å gjøre dette velger de taktikken: finne gradene til de ytterste og innerste vinklene, for så å bytte mellom å gå og rotere de ulike gradene. Det er dermed en lignende løsningsmetode som de har brukt med de andre geometriske figurene, men nå må de ta høyde for to forskjellige vinkelstørrelser. De har også tidligere vært i en iterativ prosess der de har begynt med en tilfeldig grad, testet denne graden, og deretter tilpasset ut ifra resultatene. Dette er derimot vanskeligere å gjøre med ei stjerne, nettopp fordi den har to forskjellige grader som roterer i ulik retning. Det at man må programmere vinklene til å rotere i ulik retning, gjør det også vanskeligere å se på riktig vinkel på tegningen for å avgjøre hvor mange grader man må skrive inn i kodeklossen. Dette gjør at Knut og Ida ender opp med å gjette hvilke grader de skal bruke, uten noen matematiske forklaringer på hvorfor de mener at disse gradene er riktige.

Som nevnt valgte Knut og Ida å prøve å programmere ei femkanta stjerne. Når de tegnet stjerner for hånd på arket, tegnet de derimot stjerner med seks spisser som var satt sammen av to trekanter lagt skrått oppå hverandre. Denne fremgangsmåten kunne ha hjulpet Knut og Ida med å programmere ei stjerne, ved at de kunne brukt deres tidligere løsningsmetoder for å programmere en trekant som de så roterte og gjentok over den første trekanten. Dette kunne også ha åpnet opp for bedre og mer utfyllende matematiske argumenter, der de hadde hatt større forutsetninger for å gjøre gjennomtenkte valg og ikke bare gjette. Da kunne de sett at ei sekskanta stjerne består av to trekanter oppå hverandre, og at trekanter i Scratch må

programmeres med vinkler på 120 grader. Når en da setter en trekant midt på den andre trekanten blir vinklene mellom spissene halvert til 60 grader. Slik blir da Knut og Idas såkalte ytterste vinkler 120 grader og de innerste vinklene 60 grader.

4.4 Gjennomgående funn for hele arbeidsprosessen

Knut og Ida var et par som fordelte arbeidet jevnt seg imellom. Under arbeidsprosessen var begge to like involvert i å løse oppgavene, og dialogene bar preg av at Knut og Ida uttrykte annenhver ytring. Knut ville ofte planlegge og tegne ned løsningen, mens Ida var mer direkte og hadde lavere terskel for å prøve og feile. Ida tok også ofte styringen over datamaskinen, og det virket som om Knut stolte på at Ida hadde rett i det meste. Dette kan være en av grunnene til hvorfor Knut stilte få spørsmål til Ida om hennes påstander.

Det var også mye snakk om hvilke grader de ulike vinklene i figurene roterte. En annen måte å beskrive hva Knut og Ida diskuterte, er at de undersøkte hvor langt og i hvilken retning kantene skulle gå. Dette kan igjen trekkes videre til det matematiske begrepet *vektor* som beskriver en fysisk kvantitet som har en bestemt størrelse og retning (Hinna, Rinvold & Gustavsen, 2012). Hadde elevene tegnet geometriske figurer på et ark, hadde de ikke blitt tvunget til å tenke gjennom faktorer som lengde på kantene og retningen på kantene. Knut og Ida har dermed arbeidet med matematiske områder innenfor vektorer uten at de var klar over det. Dette er noe som kan gjøre det lettere for dem når de senere skal lære om vektorer.

4.4.1 Mangel på spørsmål

Yackel (2001) sitt fenomen, oppfattet-som-forstått, dukket opp flere steder i dialogene, hvor Ida ofte lot være å forklare resonnementene sine for en påstand ovenfor Knut. Det er mulig at Ida oppfattet at Knut forstod hva hun mente, fordi han ikke stilte spørsmål til påstandene hennes. Ettersom det ikke ble pratet med verken Knut eller Ida om disse stedene i dialogen, der Ida ikke forklarte ordentlig hvorfor hun mente det hun mente, kan vi ikke vite med sikkerhet om Ida lot være å forklare noe fordi hun trodde at Knut forstod bakgrunnen for påstandene. Vi kan heller ikke vite om Knut forstod hva Ida mente, slik at Ida gjorde rett i å ikke bruke tid på å forklare, eller om han ikke forstod og stilte ikke spørsmål fordi han stolte på Idas matematiske kunnskaper. En annen mulighet er at Knut rett og slett ikke brydde seg nok om oppgaven til å stille spørsmål om ting han ikke forstod. Trekker man inn dataskjermens rolle i dialogene, er det også mulig at de følte at dataskjermen illustrert såpass mye av tankene deres at det ikke var

hensiktsmessig å forklare bakgrunnen for påstandene. Dersom dette stemmer, har det at de programmerte på en dataskjerm vært til hindring for enkelte mulige muntlige matematiske argumenter. En tredje mulig forklaring til hvorfor Knut ikke stilte så mange spørsmål kan være at han var opptatt av å bli raskt ferdig med oppgavene og ville ikke bruke mer tid enn nødvendig.

4.4.2 Matematisk argumentasjon som prosess

Generelt sett gjennom hele arbeidet med å programmere ulike geometriske figurer, var Knut og Ida opptatt av å avgjøre hvordan pennen beveget seg når den skulle tegne en bestemt figur. Dette innebar at de fikk utforsket detaljert hvordan de ulike kantene og vinklene forholdt seg til hverandre hos de ulike geometriske figurene. Jeg vil påstå at dette er betraktninger Knut og Ida ikke ville gjort dersom de kun hadde fått i oppgave å tegne figurene for hånd på et ark. Det er derfor selve programmeringsprogrammet Scratch som gjør at Knut og Ida fikk behov for å utforske og dermed også matematisk argumentere for de ulike løsningene. Som det ble nevnt innledningsvis i 1.2.2 er selve prosessen det sentrale når det kommer til matematisk argumentasjon. Det at Scratch gjør denne prosessen mer detaljert og tidkrevende, gjør at det blir større mulighet til å diskutere og argumentere for ulike matematiske løsninger.

I tillegg til at matematisk argumentasjon er en prosess, er det også en *sosial* prosess. De gangene Knut og Ida argumenterte for noen matematiske påstander, gjorde de det for å forklare ovenfor partneren sin de ulike matematiske resonnementene som var blitt gjort for å komme frem til en konklusjon. Hadde ikke elevene arbeidet i par, hadde de sannsynligvis ikke følt det samme behovet for å overbevise noen om hvorfor deres påstander ga mening. Det å arbeide sammen med noen gjorde at de muntlig ytret sine matematiske resonnementer, som da ble matematiske argumenter.

4.4.3 Argumentasjonskategorier

Argumentene til Knut og Ida var stort sett preget av å være representasjonsbasert ved at enten en tegning eller et bilde på dataskjermen fungerte som en representasjon av det som ble muntlig ytret. Det er også mye sammensatt argumentasjon, fordi elevene ofte argumenterte med en god blanding av representasjoner og muntlige matematiske forklaringer. Som nevnt i delkapittel 2.5.4 er representasjonsbasert argumentasjon en typisk form for matematisk argumentasjon i skolen, og er et viktig steg i det å videreutvikle argumentasjonskompetansen i matematikk. Knut og Ida var dermed på god vei i denne utviklingen, hvor neste steg vil være å utvide og

bearbeide argumentene hvor en til slutt presenterer argumenter som er generelle. Dette er noe Knut og Ida også var i gang med, hvor det i dialogene inn imellom dukket opp argumenter som var utvidet ut ifra resultatene fra en av testene til Knut og Ida. De tok derimot aldri steget til Lavys (2006) siste kategori, generelt-presentert-som-spesifikt.

I tillegg til at den matematiske argumentasjonen hos Knut og Ida bar preg av å være sammensatt av representasjoner og matematiske forklaringer, brukte de lang tid til å komme frem til den endelige løsningen. Dette tyder på at Knut og Ida ikke visste umiddelbart hvordan de skulle løse oppgaven, og på den måten gjorde de sjeldent det Lithner (2008) kaller for imitative resonnementer og argumenter. De var i stedet ofte kreative og presenterte argumenter som ikke var bundet til tidligere innlærte algoritmer, og som oppfylte de tre kravene: nyhet, plausibilitet og matematisk grunnlag. En mulig forklaring til hvorfor Knut og Ida var kreative i sin argumentasjon kan være at Scratch sine kodeklosser gjorde det vanskelig å se hvordan de kunne bruke algoritmer for å løse oppgavene. Det at Scratch tilbyr ulike kodeklosser som kan løse oppgaven på flere måter, kan også ha vært en bidragsyter for den kreative argumentasjonsprosessen. I tillegg kan det at det er lett å teste ut ting gjøre det enklere for elevene å være kreative og prøve ut nye ting. Dersom det er noe som ikke fungerer er det bare å slette. Dette senker terskelen på å teste ut ideer, noe som kan øke elevenes læring ved at de lettere kan utforske flere måter å løse et problem på.

4.4.4 Systematisk prøving og feiling

Setningen, «Vi bare prøver», ble ofte ytret av Knut og Ida. Etersom det er enkelt i Scratch å slette noe som ikke fungerer, ble det å teste ut noe en ofte brukt taktikk. Dersom pennen tegnet noe som kunne likne på sluttresultatet fortsatte de, og dersom det ikke så fornuftig ut ble idéen vraket. Scratch sin enkle brukervennlighet for å teste ut noe, påvirket dermed Knut og Ida sin arbeidsmetode der resultatet fra disse hyppige testene fungerte som et argument for neste test. På den andre siden kan det at det var så enkelt å teste ut noe i Scratch være et hinder for den matematiske argumentasjon, fordi de ikke trengte å være 100% sikker i løsningen sin før de testet den ut.

4.5 De andre tre parene

I dette delkapittelet er de tre andre gruppene som deltok på datainnsamlingen presentert og kort analysert. Dette er gjort for å gi innsyn i hele datamaterialet, for å vise hva det er som skiller Knut og Ida fra de andre gruppene og for å kunne forklare hvorfor Knut og Ida sin arbeidsprosess la opp til mer matematisk argumentasjon enn de andres.

4.5.1 Gruppe 1, Nils og Anders programmerer en femkant

Gruppe 1 bestod av de to guttene Nils og Anders. Under datainnsamlingen hadde de problemer med å programmere en firkant. I motsetning til de andre gruppene, valgte Nils og Anders å programmere en firkant ved å notere ned koordinatene til de ulike vinklene og programmere pennen til å gå til disse punktene. I tillegg var de unøyaktige når de noterte ned koordinatene, noe som førte til at de måtte begynne på nytt flere ganger. Når de skulle gå i gang med å programmere en femkant hadde de allerede brukt nesten 30 minutter på å programmere en firkant. Anders startet oppgaven om en femkant med å spørre Nils om hva han tenkte. Nils gikk så i gang med å tegne en femkant på et ark, og Anders så på tegningen. Videre fortsatte dialogen:

Anders: Det er en diamant. Få se. En, to, tre, fire, fem. Ja det er en diamant.

Nils: Hmm.

Anders: Skal vi ta en diamant da?

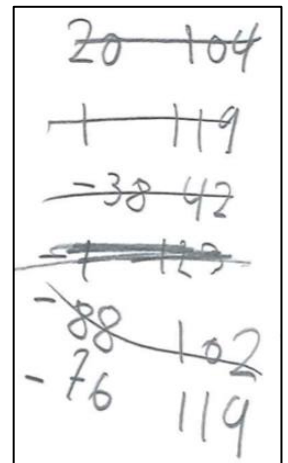
Nils: Ja.

[Anders går så i gang med å skrive ned koordinatene til de ulike hjørnene ved å flytte på musepekeren. Han har seks ytringer på rad der han sier høyt koordinatene han skriver.]

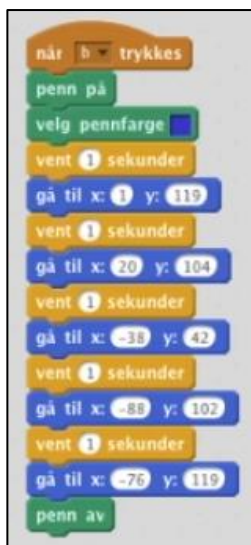
Anders: Der har vi koordinatene, her, en, to, tre, fire, fem [Teller koordinatene han har skrevet ned på arket. Se figur 19].

[Anders begynner så å trykke koordinatene inn i fem kodeklosser som alle sier *gå til x-koordinat og y-koordinat* (se figur 20). Etter å ha trykket inn alle koordinatene blir en figur med diamant-form tegnet (se figur 21).]

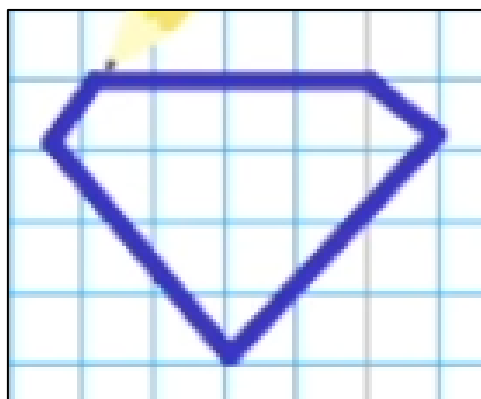
Figur 19. Bilde av koordinatene Anders noterte.



Figur 20. Skjermbilde av kodetråden.



Figur 21. Skjermbilde av diamanten som ble tegnet.



Dette dialogutdraget hvor Nils og Anders programmerer en femkant, gir et representativt bilde av deres arbeidsprosess og tilnærming til programmering. I arbeidet med programmeringen av en firkant har Anders hatt mest kontroll over musepekeren og styrte mye med datamaskinen. Etter spørsmål fra Anders, går Nils i gang med å tegne ned en femkant for å kunne si noe om hvordan figuren kan se ut. Etter å ha tegnet ferdig ser Anders på tegningen til Nils og sier «Det er en diamant». Anders går så i gang med å programmere en femkant ved hjelp av kodeklossen *gå til x-koordinat og y-koordinat*. Etter å ha flyttet musepekeren seks ganger og notert ned alle koordinatene går han også rett i gang med å dra inn de riktige kodeklossene og skrive inn koordinatene uten å forhøre seg med Nils. Når Anders lager kodetråden glemmer han det første punktet han noterte ned, og får ved første test av kodetråden en figur som ikke er lukket og mangler den øverste kanten. Anders flytter litt på pennen for hånd slik at neste gang blir en hel diamant/femkant tegnet uten at han har oppdaget at en kodekloss mangler. Det som skjer når Nils og Anders programmerer en femkant er veldig likt hvordan de programmerte en firkant, men i en forkortet versjon på grunn av at Anders var mer nøyaktig når han noterte ned koordinatene.

Selv om det er lite matematisk argumentasjon å identifisere i Nils og Anders sin arbeidsprosess, dukker det opp noen ytringer som er argumenterende. For å bekrefte om figuren Nils har tegnet er en diamant sier Anders «En, to, tre, fire, fem» for å argumentere for at figuren «er en diamant». Anders bruker her de samme argumentene som Knut og Ida brukte for å bekrefte om Knut sin tegning var en femkant der de også telte antall kanter høyt som et argument for at det var tegnet en sekskant. Anders bruker her, i likhet med Knut og Ida, sammensatt argumentasjon

ved at både tegningen og telling av antall kanter fungerer som argumenter for hvorfor figuren er en diamant. Selv om Anders sine matematiske argumenter er relativt like Knut og Ida sine, er Anders sin argumentasjon ikke like god som Knut og Ida sin. Det er fordi Anders ikke argumenterer for sammenhengen mellom diamant og femkant eller hvorfor en figur er en diamant hvis den har fem kanter. Han har derfor latt være å argumentere for noen ledd i resonneringen sin. Den typen argumentasjon Anders gjør her ble i 4.1.1 beskrevet som litt enkel, men også viktig for å etablere et grunnlag for videre argumentasjon. Det som skiller disse to gruppene fra hverandre her er dermed at der Knut og Ida fortsatte å argumentere for ulike løsninger og påstander, gjorde ikke Nils og Anders det. Videre i arbeidsprosessen går Anders i gang med løsningen uten å samarbeide med Nils.

I dialogutdraget er Anders pådriveren i samtalen samtidig som han også har kontroll over datamaskinen gjennom hele oppgaven om en femkant. Ved et tilfelle strekker Nils seg mot datamaskinen og det virker som om Anders nærmest instinktivt slår bort hånda til Nils. Anders tar seg i å slå vekk hånda til Nils og spør «Hva skulle du gjøre», hvor Nils svarer «Bare trykke på pilene». Ellers under hele datainnsamlingen er Nils svært lite delaktig i å løse oppgavene, og man kan se flere ganger på videoen at han legger hodet i hendene og gjesper. Nils spør i tillegg sjeldent spørsmål til hva Anders gjør, og Anders forklarer også sjeldent hva han tenker eller hvorfor han gjør noe. Dette viser til en samarbeidsdynamikk der Anders sitter med mest makt og kontroll over datamaskinen.

Som forklart i delkapittel 1.2.2 blir matematisk argumentasjon sett på som en prosess der individer samhandler med hverandre. Etersom Nils og Anders i liten grad samarbeider og deler resonnementer og løsninger med hverandre blir det også lite matematisk argumentasjon. I dialogutdraget ovenfor går stort sett hele løsningsprosessen ut på at Anders skriver ned ulike koordinater og plotter dem inn i Scratch. Tidligere i dette kapitlet forklarte jeg hvordan Tiril og Line sin matematiske argumentasjon nærmest forsvant helt når de tok i bruk kodeklossen *gå til x-koordinat og y-koordinat*. Det er dermed mulig at Nils og Anders argumenterer såpass lite fordi de kun har brukt en tilnærming som innebærer å skrive ned ulike koordinater og mister dermed muligheten for å diskutere for eksempel lengder og vinkelstørrelser. Det å notere ned ulike koordinater over lengre tid kan også bli en kjedelig jobb som kan ha ført til at guttene gikk litt lei og ble mindre motivert til å løse oppgavene.

4.5.2 Gruppe 3, Silje og Pernille programmerer ei stjerne

Neste gruppe bestod av de to jentene Silje og Pernille. Dette elev-paret arbeidet godt sammen og ble ferdig med alle oppgavene under datainnsamlingen. Når de programmerte firkanten, femkanten og sirkelen brukte de ved alle oppgavene løsningen *gå og vend noen grader*. Ved programmeringen av ei stjerne valgte de derimot å bruke løsningsstrategien *gå til x-koordinat og y-koordinat*. Løsningen ble formulert av Silje ved at hun sa «Vi tar bare gå til punkt, gå til punkt, gå til punkt, gå til punkt, gå til punkt». Mens hun sier dette ler hun litt og Pernille prøver å få oppmerksomheten til Silje. Etter å ha fått Silje til å slutte å si «gå til punkt», sier Pernille «Hvis vi bare kunne tegnet opp noe. Jeg vet!» og henter frem et bilde av ei stjerne. Hun setter inn bildet av stjerna i skript-området, og samtalen fortsetter som dette:

[Pernille setter inn et bilde av ei stjerne i skript-området.]

Silje: Jaa!

Pernille: Hehe.

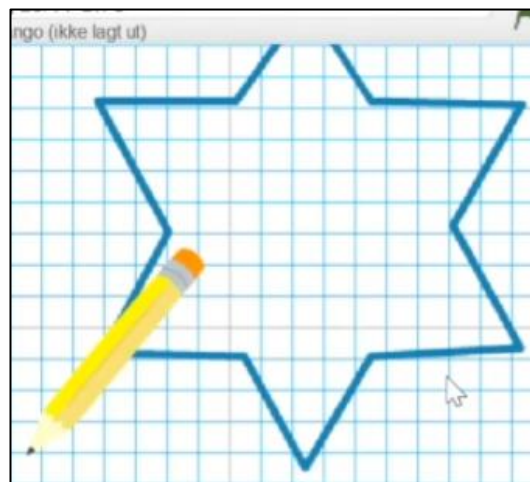
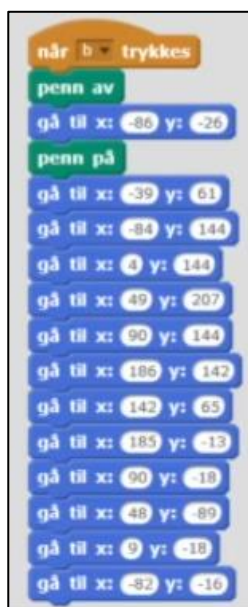
Lærer: Programmerer dere ei stjerne hvis dere legger til ei stjerne?

Pernille: Nei, men vi bare tegner rundt den så har vi en mal.

Lærer: Ja.

[Silje og Pernille arbeider videre sammen om å trekke inn kodeklosser med de riktige koordinatene. Dette gjør de ved å flytte pennen til de ulike vinklene til stjerna og dra inn *gå til x-koordinat og y-koordinat*-kodeklosser med pennens koordinater. Se figur 22 og 23.]

Figur 22. Skjerm bilde av kodetråden. Figur 23. Skjerm bilde av stjerna som ble tegnet.



Ved å sette inn ei stjerne og skrive inn koordinatene til de ulike vinklene i stjerna, unngår Silje og Pernille å måtte diskutere og finne ut hvordan ei stjerne ser ut og hvordan vinklene er i forhold til hverandre. De tar dermed en form for snarvei, noe som de også gjør andre steder under datainnsamlingen. Med oppgaven om å programmere et hus gjorde de det samme der de satte inn et bilde av et hus og noterte ned koordinatene til de ulike vinklene på figuren. Når de skulle programmere en sirkel klarte de først med et uhell å programmere pennen til å tegne en stor prikk ved at de hadde pennebredden veldig stor og roterte pennen 180 grader to ganger. Her var de villig til å si seg ferdig med oppgaven om å programmere en sirkel helt til læreren sa at det var mer en stor prikk enn en sirkel. Disse snarveiene som Silje og Pernille tok gjorde at det ble mindre muligheter for å argumentere matematisk fordi de tok den enkle veien ut der de i liten grad hadde behov for å argumentere for noe.

Det er altså lite matematisk argumentasjon å identifisere hos Silje og Pernille når de tok disse snarveiene. Ellers i arbeidsprosessen argumenterer og forklarer de relativt lite ovenfor hverandre til tross for at begge tar like stor del i arbeidet. Det er mulig at de kjenner hverandre såpass godt at de ikke trenger å forklare så mye. Ved en sirkel trengte Silje å bare si «Vi kan jo bare ta gjenta» så skjønnte Pernille at hun mente at de kunne ta *gå et steg og vend én grad* og gjenta dette 360 ganger for å få en sirkel. En annen mulig forklaring var at oppgavene var for enkle for dem slik at det var nærmest åpenbart for dem hva den andre mente at de måtte gjøre. Dette er noe som skiller Knut og Ida fra Silje og Pernille ved at Knut og Ida strevde mer med å finne en løsning og måtte argumentere mer ovenfor den andre enn det Silje og Pernille gjorde.

4.5.3 Gruppe 4, Tiril og Line programmerer en sirkel

Gruppe 4 bestod av jentene Tiril og Line. Når de kom til oppgaven om en sirkel hadde de allerede programmert en firkant og en femkant ved hjelp av løsningsmetoden *gå og vend noen grader*. Ved programmeringen av en sirkel holder de fast på strategien om å bruke kodeklossene *gå noen steg* og *vend noen grader*, og Tiril kommer med forslaget «Vi kan ha vend 1 grad hver gang så får vi en skikkelig lang rekke». Etter å ha oppdaget kodeklossen *gjenta x ganger* fortsetter dialogen som følgende:

Tiril: Okey, vend da bare prøver vi det. Vend 1 grad.
Line: 1 grad.
Tiril: Og så går den, hvor mange skritt?
Line: 10? 100?
Tiril: For det skal jo gå så lite hver gang.
[Tiril trykker inn *gå 5 steg*]

Dette var et elev-par som virket motiverte og interesserte til å løse oppgavene. De brukte også lang tid på å finne de ulike løsningene og hadde på den måten store muligheter til å argumentere matematisk for påstandene sine. I dette dialogutdraget arbeider Line og Tiril med å finne ut hvor mange grader og skritt pennen skal gå mellom hver gjentakelse. Tiril konstaterer først at pennen må vende 1 grad, men forklarer ikke hvorfor det er slik. Videre stiller hun spørsmål om hvor mange skritt pennen skal gå, og Line forslår 10 eller 100. Mens Line sier «10? 100?» begynner Tiril på setningen «For det skal jo gå så lite hver gang». Tiril er tydelig i hva hun ønsker at pennen skal gjøre, men hun lar være å forklare hvorfor pennen må gå små skritt. Uten å forklare noe mer ovenfor Line, setter Tiril inn kodeklossen *gå 5 steg*.

Det som ofte hindret mulighetene for matematisk argumentasjon hos dette paret var samarbeidsdynamikken. I nesten hele arbeidsprosessen er det Tiril som trykker på datamaskinen og som bestemmer hva de skal gjøre. Hun forklarer relativt ofte hva hun tenker og stiller spørsmål til Line, men hun gjør flere ganger det motsatte av hva Line foreslår. Dette kan sees i dialogutsnittet ovenfor hvor Tiril spør Line om hvor mange skritt figuren skal ta og at Line foreslår 10 eller 100 skritt. Uten å virkelig svare på Line sitt forslag, trykker Tiril derimot inn 5 steg. Dette bidrar til å skape en skeiv maktbalanse mellom jentene hvor Line kan oppleve at hennes argumenter og påstander ikke blir hørt. Line er i likhet med Tiril også flink til å forklare resonnementene sine, men ettersom de sjeldent blir godtatt av Tiril forsvinner de mer og mer utover arbeidsprosessen og Line begynner å foreslå ting med en spørrende stemme. Man kan altså finne matematisk argumentasjon flere steder i datamaterialet av Line og Tiril, men ettersom maktbalansen mellom jentene er såpass skeiv er det vanskelig å si noe om det er samarbeidet, programmeringen eller mangler på kvaliteter som påvirker Line og Tiril sine matematiske argumenter. På grunn av at Knut og Ida sitt samarbeid var såpass bra er det færre utfordringer som kan påvirke argumentasjonen. Det kan sies at det var færre faktorer i bildet, og det er derfor også et noe bedre grunnlag for å kunne si noe om det var programmeringsoppgavene som ga de ulike kvalitetene.

5. Avslutning

Formålet med denne oppgaven har hovedsakelig vært å gi innsikt i ulike kvaliteter ved matematisk argumentasjon når elever arbeider med programmering. Gjennom oppgaven har det også vært et mål å vise hvordan man kan arbeide med matematisk argumentasjon samtidig som elevene også arbeider med programmering. I tillegg har jeg ønsket å vise at man kan lage og utføre undervisningsopplegg med programmering i skolen uten å være ekspert i emnet. Problemstillingen for denne oppgaven har vært å undersøke:

Hvilke kvaliteter ved matematisk argumentasjon kan identifiseres når elever programmerer?

Hensikten med dette kapittelet er å trekke frem de viktigste funnene og se på hvordan disse kan besvare problemstillingen. Kapittelet oppsummerer også hva som har blitt gjort i oppgaven og hvordan dette har påvirket oppgavens funn. I tillegg er oppgaven kommentert med et kritisk blikk, og det er blitt skissert noen tanker om videre forskning på emnet.

5.1 Hva har blitt gjort?

I denne studien har det blitt undersøkt hvordan barn argumenterer matematisk når de programmerer i Scratch. Forskningsprosjektet har samlet inn datamateriale av elever på 7.trinn som arbeidet med matematikkoppgaver i Scratch. Videre har dette datamaterialet blitt behandlet, analysert og diskutert opp mot problemstillingen. Gjennom kapittel 2, «Tidligere forskning og teoretisk rammeverk», har sentral forskning som omhandler oppgavens fokus blitt presentert. Det som ble presentert i dette kapittelet har fungert som et rammeverk for resten av oppgaven ved at det har påvirket både valg og funn som har blitt gjort i denne studien. Man kan se på det som er presentert i kapittel 2 som todelt ved at det har blitt brukt på to ulike måter i resten av oppgaven. Den ene delen har fungert som et hjelpemiddel til datainnsamlingen, og den andre delen har gitt informasjon og verktøy til analyse og diskusjon av datamaterialet.

5.1.1 Tidligere forskning og datainnsamling

Tidligere forskning om programmering, og særlig Scratch, har påvirket dette forskningsprosjektet gjennom valg av undervisningsopplegg som elevene deltok i. Under datainnsamlingen brukte vi programmeringsprogrammet Scratch som andre forskere allerede har forsket på. I tidligere forskning (Calder & Taylor, 2010; Taylor et al., 2010; Forsström & Kaufmann, 2018) har det blant annet blitt funnet ut at det finnes matematiske muligheter i

Scratch, hvor geometri er et av de matematiske emnene som har blitt identifisert. Dette er noe vi brukte i vårt prosjekt ved at vi ga elevene programmeringsoppgaver som omhandlet geometri. I tillegg til at tidligere forskning støtter opp om matematiske muligheter i Scratch, fant Calao et al. (2015) også ut at elevene som arbeidet med Scratch videreutviklet sine ferdigheter i resonnering. For å støtte opp om det sosiale aspektet med matematisk argumentasjon, plasserte vi elevene i par slik at de hadde noen å dele resonnementene sine med og dermed kunne argumentere. Ved å gi elevene slike oppgaver la vi til rette for at elevene kunne argumentere for matematiske problemstillinger. På denne måten åpnet vi opp for muligheten til å undersøke kvaliteter ved matematisk argumentasjon mens elevene arbeidet med programmering.

5.1.2 Tidligere forskning, teoretisk rammeverk, analyse og diskusjon

Den andre rollen innholdet i kapittel 2 har hatt i oppgaven, er rollen som hjelpemiddel til å analysere og diskutere datamaterialet. Toulmin (2003), Krummheuer (1995) og Lithner (2000) sine definisjoner av matematisk argumentasjon og uttrykksformene for argumentasjon som ble presentert i delkapittel 2.5, har i analysen bidratt til å identifisere de ulike delene av elevdialogene som inneholdt matematisk argumentasjon. Informasjon om hvordan tidligere forskning har kategorisert og definert ulike former for matematisk argumentasjon har blitt aktivt brukt i analyse og diskusjon av datamaterialet ved at jeg har brukt disse kategoriene til å identifisere ulike kvaliteter ved elevenes matematiske argumentasjon. Dette har gitt meg et større grunnlag for å kunne si noe om elevenes matematiske argumentasjon, samtidig som det også har hjulpet meg med å utforske og hente ut informasjon fra analysen.

5.2 Oppgavens besvarelse på problemstillingen

Gjennom kapittel 4, «Analyse og diskusjon», har det blitt presentert ulike kvaliteter ved Knut og Ida sin matematiske argumentasjon. Etersom problemstillingen for oppgaven er å identifisere ulike kvaliteter ved matematisk argumentasjon, vil de kvalitetene som er beskrevet i kapittel 4 fungere som svar på problemstillingen. Jeg har derfor i de følgende avsnittene delt funnene inn i to kategorier. Den ene oppsummerer de funnene som ga gode kvaliteter med matematisk argumentasjon, og den andre oppsummerer de funnene som ga mindre gode kvaliteter.

5.2.1 Gode kvaliteter med elevenes matematiske argumentasjon

Først og fremst er det et viktig funn at elevene i det hele tatt argumenterte matematisk for ulike påstander underveis i programmeringen. Dette støtter opp om fremtidige undervisningsopplegg der en både arbeider med programmeringsferdigheter og argumentasjonsferdigheter i matematikk. Ettersom undervisningsopplegget som ble tatt i bruk under datainnsamlingen viste seg å være suksessfullt med tanke på å inkludere både programmering og matematisk argumentasjon, er denne masteroppgaven et eksempel i seg selv på hvordan man kan arbeide med disse to temaene samtidig i matematikkfaget. Det at to lærerstudenter uten tidligere erfaring med programmering har lyktes i å utvikle dette undervisningsopplegget, viser også at lærere kan lage undervisningsopplegg med programmering selv om en ikke har mye kunnskaper om dette fagområdet fra før av. Jeg håper derfor at masteroppgaven min kan brukes som inspirasjon for lærere som føler at deres programmerings-kompetanse ikke er tilstrekkelig i forhold til den nye læreplanen som kommer i 2020.

En av de positive kvalitetene ved elevenes matematiske argumentasjon var at de nesten konstant brukte en form for representasjonsbasert argumentasjon der enten en tegning eller et bilde på datamaskinen fungerte som et av argumentene. Dette forklarte jeg i delkapittel 4.4.3 at er en form for matematisk argumentasjon som er et godt steg i riktig retning mot å utvikle deres argumentasjonsferdigheter i matematikk. I tillegg fungerte bildene fra skjermen ofte som en del av elevenes språk ved at de sa ytringer som «flytt den dit», «den skal lenger opp», «øverst», osv., hvor dataskjermen hadde en sentral rolle for å forstå hva de mente med ytringene sine. En slik måte å prate på ble i delkapittel 2.5.1 beskrevet som deiktisk språk. Et av Scratch sine bidrag under datainnsamlingen har derfor vært å gi elevene bilder som elevene kunne støtte seg på når de skulle programmere og matematisk argumentere for noe. På den måten fikk de mulighet til å bruke det Lavy (2006) beskrev som sammensatt argumentasjon ved at de hadde argumenter som bestod av både skjermbilder og matematiske forklaringer.

Elevene ble ved å programmere i Scratch også tvunget til å se mer detaljert på de ulike stegene i å lage en geometrisk figur. Her fikk de kjenne både på behovet og mulighetene til å utforske og argumentere for ulike vinkelstørrelser og lengder hos de geometriske figurene. Scratch åpnet dermed opp for matematisk argumentasjon ved at programmet gjorde arbeidsprosessen mer detaljert og krevende enn hvis elevene skulle ha tegnet figurene for hånd.

Kreativitet var også et trekk som gikk igjen i Knut og Idas matematiske argumentasjon. En mulig forklaring på dette er at Scratch sitt brede spekter av kodeklosser åpnet opp for ulike måter å løse oppgavene på. I tillegg hadde de aldri laget geometriske figurer gjennom programmering tidligere slik at hele situasjonen var ny for elevene og de hadde derfor ikke så mange erfaringer å basere løsningene sine på.

En ofte brukt strategi for alle gruppene var prøve- og feilemetoden. I delkapittel 2.5.2 om prøving og feiling ble det forklart at denne metoden er mest hensiktsmessig dersom elevene er systematiske i valgene de gjør om hva de vil prøve ut. Dette var noe alle elevene også gjorde. Hos Knut og Ida kom dette ekstra tydelig frem når de skulle programmere en femkant. Her testet de ut en vinkelstørrelse basert på vinkelens utseende på tegningen, registrerte og vurderte resultatet, og testet så ut en ny vinkelstørrelse basert på refleksjonene fra forrige testet. På den måten kom de nærmere og nærmere en riktig løsning hvor de til slutt endte opp med den nesten helt korrekte vinkelstørrelsen, 70 grader. I sammenheng med dette ble det kommentert i 4.1.3 at selv om vinkelstørrelsen ikke var fullstendig korrekt, så fikk de utforsket og argumentert mer om matematikk ved bruk av denne metoden enn hva en annen gruppe fikk når de brukte en annen løsningsmetode.

5.2.2 Mindre gode kvaliteter med elevenes matematiske argumentasjon

Selv om det å programmere i Scratch viste seg å åpne opp for mange muligheter for matematisk argumentasjon med gode kvaliteter, er det også noen litt mindre positive funn om den matematiske argumentasjonen. Et av de er at elevene tidvis ikke argumenterte så grundig og detaljert for påstandene sine. En mulig årsak til dette kan være at Scratch var så beskrivende i bildene sine at de ikke følte for å argumentere for alt, fordi det ble vist på skjermen. Prøve- og feilemetoden som elevene ofte brukte kan også ha hatt en negativ innvirkning på elevenes matematiske argumentasjon, da det kan ha gjort at de senket terskelen for å prøve ut nye ting og trengte derfor ikke å argumentere like mye. Oppgavetypen kan også ha hatt en innvirkning på hvor detaljert og matematisk avansert elevene hadde mulighet for å argumentere.

Ved å se på de ulike gruppene kom det også frem at valg av kodeklosser påvirket mulighetene til å matematisk argumentere. Når elevene fra de andre gruppene brukte kodeklosser som baserte seg på å gå til x- og y-koordinater argumenterte de mindre enn hva Knut og Ida gjorde som holdt seg til *gå*- og *vend*-strategien. Dette viser at argumentasjons-mulighetene i Scratch også blir påvirket av hvilke kodeklosser og løsningsmetode man bruker.

I dette forskningsprosjektet holdt elevene seg som oftest til sammensatt argumentasjon, og det var lite progresjon å se med tanke på Lavy (2006) sine fire argumentasjonskategorier. Det er interessant at resultatene ble såpass annerledes fra Lavy sin studie når studiene våre har så mange likhetstrekk. I Lavy sin studie hadde nemlig elevene en klar progresjon i argumentasjonskategoriene hvor de gikk fra å argumentere helt grunnleggende til å presentere argumenter som var generelt-presentert-som-spesifikt. Selv om min studie er inspirert fra Lavy sitt prosjekt, er det likevel noen forskjeller i valg av programmeringsprogram og oppgaven elevene fikk. Lavy ga elevene sine i oppgave å utforske tallegenskaper hos ulike stjerner og mangekanter, mens Tonje og jeg ga elevene i oppgave å utforske hvordan man kan lage ulike mangekanter. Det er vanskelig å si om det er programmet eller oppgaven som har forårsaket denne forskjellen, eller om funnene hadde vært ulike selv om elevene hadde gjort nøyaktig det samme. Det kan derimot tenkes at oppgaven elevene fikk i Lavy sitt prosjekt var mer utfordrende matematisk enn det oppgavene elevene i dette forskningsprosjektet fikk. Ser man på datamaterialet fra gruppe 3, Silje og Pernille, kan det nemlig tenkes at oppgavene var for enkle slik at arbeidsprosessen ble for kort og de ikke fikk mulighet til å diskutere og argumentere for ulike løsninger.

5.2.3 Oppgavens aktualitet

I sammenheng med den nye læreplanen som kommer i 2020 har det vært mye debatt om hvordan programmering skal implementeres i skolen (Bærland og Gilje, 2017; Flote, 2016; Klovning, 2015; Waage, 2018). Det har vært diskutert mye om hvilke fag programmering skal inn i, i hvor stor grad det skal inn i skolen, når det skal inn i skolen og hvordan det skal implementeres. De siste årene har det kommet flere rapporter og høringer om dette hvor det har skjedd endringer på hva som er best å gjøre og hvordan programmering skal inn i skolen. Ved det første utkastet til LK20 (Utdanningsdirektoratet, 2018) skulle programmering få en større del i skolen enn slik det står i det nyeste utkastet (Utdanningsdirektoratet, 2019b) hvor programmering skal slås sammen med algoritmisk tenkning.

Et av læreplanmålene som står i det nyeste utkastet til LK20 (Utdanningsdirektoratet, 2019b) er at elevene skal etter 6. trinn kunne «bruke variabler, lykkjer, vilkår og funksjonar i programmering til å utforske geometriske figurar og mønster». Dette læreplanmålet beskriver mer eller mindre hva elevene gjorde under datainnsamlingen i dette forskningsprosjektet. Som nevnt i innledinga er et av målene mine med oppgaven å «bidra med innsikt i hvordan en kan

implementere programmering i matematikktimene uten å miste fokuset på matematikk». Det at dette læreplanmålet er såpass likt med hva våre elever gjorde, bidrar til å aktualisere oppgaven min mot den fremtidige skolen som skal få programmering inn i matematikkfaget.

5.3 Kritisk blikk

Selv om oppgaven har som formål å undersøke matematisk argumentasjon mens elever programmerer, må man også ta stilling til at andre faktorer kan ha påvirket kvaliteten på elevenes matematiske argumentasjon. En av de faktorene som kan ha påvirket datamaterialet er parsammensetningen. Det var tydelige forskjeller i dynamikken mellom elevene i de ulike gruppene som deltok på datainnsamlingen. I gruppe 1 og 4 var det et skille mellom elevene der den ene tok kontroll over datamaskinen og bestemte hva som var verdt å prøve. Gruppe 3 sin arbeidsprosess var preget av mye tulling og snarveier for å bli raskt ferdig. Disse gruppene fikk i liten grad vist motivasjon og interesse i å argumentere for arbeidsprosessen. Gruppe 2 (Knut og Ida) var den gruppen som hadde det mest gunstige samarbeidet for å arbeide med matematisk argumentasjon, fordi de var gode på å dele sine tanker med hverandre, lyttet til det den andre sa og virket interessert i å løse oppgavene. Ettersom jeg var ute etter å utforske kvaliteter med matematisk argumentasjon når elever programmerte, og ikke hvordan ulike parsammensetninger påvirket argumentasjonen, valgte jeg å analysere og diskutere Knut og Ida sine dialoger nærmere da disse hadde minst hindringer i samarbeidet. Oppgaven kan derfor ses på som et arbeid for å identifisere muligheter og potensialet for argumentasjon og læring knyttet til matematikk og programmering.

Hvordan oppgavene elevene fikk var utformet har også hatt en påvirkning på elevenes matematiske argumentasjon. Gjennom analyse og diskusjon av datamaterialet kom det frem at noen av elevene kunne virke utålmodig med å bli ferdig med oppgavene. Ettersom argumentasjon er en prosess er det viktig at elevene er interessert i selve arbeidsprosessen og er motivert til å utforske og diskutere ulike mulige løsninger. Dersom oppgaven ikke gjorde elevene motivert og interessert i å løse oppgaven, kan dette ha hatt en negativ innvirkning på argumentasjonen deres.

For at det skal være muligheter til å matematisk argumentere, må elevene også ha en oppgave som er utfordrende og som de ikke umiddelbart finner en løsning for. For å kunne tilpasse matematikkoppgaver så godt som mulig er det å kjenne elevene og vite hva de trenger for å bli

utfordret et viktig moment. En av ulempene med dette forskningsprosjektet var nettopp det at vi ikke kjente elevene fra før av, og hadde kun to møter med dem før vi utførte det endelige undervisningsopplegget. Det kunne dermed ha vært en fordel dersom vi hadde utført datainnsamlingen over et større tidsrom, der vi utførte flere undervisningsopplegg hvor vi ble bedre kjent med elevene og kunne skreddersy oppleggene enda bedre. Dette kunne også ha gjort det enklere å analysere elevene, spesielt med tanke på argumenter som var oppfattet-som-forstått.

I tillegg til å utføre datainnsamlingen over tid, kunne forskningsprosjektet også ha blitt styrket dersom vi hadde brukt tid før det endelige undervisningsopplegget til å snakke med elevene om matematisk argumentasjon og gjøre dem mer bevisst over denne delen av oppgaven. Slik datainnsamlingen ble utført ble det lagt mest fokus ovenfor elevene på programmeringsdelen enn på argumentasjonsdelen. Dette kan ha gjort at vi ikke fikk tilrettelagt godt nok for argumentasjonsdelen av oppgaven slik at elevene ikke fikk vist sitt fulle potensiale innenfor matematisk argumentasjon. På grunn av at det ikke ble fokusert på matematisk argumentasjon når vi pratet med elevene, viser funnene i denne oppgaven kvaliteter ved matematisk argumentasjon som oppsto naturlig for elevene uten påvirkning fra de voksne.

Denne oppgavens overførbarhet er noe svekket av at det kun ble gjort en detaljert analyse og diskusjon av et elev-par. Selv om en oppgaves overførbarhet vil påvirke i hvor stor grad funnene kan bli nyttet i andre sammenhenger, er ikke det å finne *alle* kvalitetene ved matematisk argumentasjon som *alltid* vil oppstå i andre sammenhenger målet med oppgaven. Målet var å identifisere ulike kvaliteter ved matematisk argumentasjon som *kan* oppstå når elever arbeider med programmering. Det vil si at de kvalitetene som har blitt identifisert i denne oppgaven bare er noen av de mange andre mulige kvalitetene som kan oppstå når elever programmerer. Ved å identifisere noen mulige kvaliteter og si noe om hvorfor de oppstod kan andre lærere bruke dette til å gjenkjenne og forstå disse kvalitetene bedre.

5.4 Videre forskning

Som det ble fremhevet i forrige delkapittel, var tid et moment som begrenset hvor godt vi fikk utført datainnsamlingen. Skulle man ha forsket videre på denne oppgavens tematikk hadde det derfor vært interessant å gjøre en lignende datainnsamling, men over lengre tid, med flere undervisningsopplegg og med et større fokus på argumentasjonsdelen. Man kunne da ha forsket mer på om programmering i Scratch åpnet opp for å videreutvikle elevenes argumentasjonsferdigheter i matematikk eller om det finnes et visst nivå hvor progresjonen stopper.

Det hadde også vært interessant å forske på hvilke oppgavetyper med Scratch som åpner mest opp for matematisk argumentasjon. Her kunne man ha testet ut oppgaver som var mer utfordrende og komplekse når det kommer til programmeringsdelen, som å programmere regnefortellinger. Man kunne også ha gitt elevene lignende oppgaver som ble gitt ved Lavy (2006) sin studie, og be elevene om å programmere stjerner med ulikt antall spisser.

Uansett hva det blir så er hvert fall én ting sikkert, og det er at programmering i skolen er et nytt og spennende tema som det trengs mer forskning på og som det blir spennende å følge i fremtiden.

Litteraturliste

- Alrø, H. & Kristiansen, M. (1997). Mediet er ikke budskapet: Video i observation af interpersonel kommunikation. I H. Alrø & L. Dirckinck-Holmfeld (Red.), *Videoobservation* (Vol. Nr. 3, Interpersonel kommunikation i organisationer, s. 73–99). Aalborg: Aalborg Universitetsforlag.
- Backe-Hansen, E. & Frønes, I. (2012). Innledning: Hvordan forske på og med barn og unge? I E. Backe-Hansen & I. Frønes (Red.), *Metoder og perspektiver i barne- og ungdomsforskning* (s. 11–32). Oslo: Gyldendal akademisk.
- Barab, S. & Squire, K. (2004). Design-based research: putting a stake in the ground. *The Journal of the Learning Sciences* 13(1), 1–14. DOI: 10.1207/s15327809jls1301_1.
- Bærland, T. & Gilje, Ø. (2017, 02. juni). Fremmedspråket programmering. *Morgenbladet*. Hentet fra <https://morgenbladet.no/ideer/2017/06/fremmedspraket-programmering>.
- Calder, N. & Taylor, M. (2010). Scratching below the surface: Mathematics through an alternative digital lens? I L. Sparrow, B. Kissane & C. Hurst (Red.). *Shaping the Future of Mathematics Education: Proceedings of the 33rd annual conference of the Mathematics Education Research Group of Australasia* (s. 117–124). Freemantle: MERGA.
- Calao, L. A., Moreno-Léon, J., Correa, H. E. & Robles, G. (2015). Developing mathematical thinking with Scratch. I G. Conole, T. Klobucar, C. Rensing, J. Konert & E. Lavoué (Red.). *Design for Teaching and Learning in a Networked World* (s. 17–27). Lecture notes in computer science, vol. 9307. Springer: Cham.
- Flote, B. (2016, 13. april). No skal elevane lære koding. *Stavanger aftenblad*. Hentet fra <https://www.aftenbladet.no/lokalt/i/QPqRA/No-skal-elevane-lare-koding>.
- Forström, S. E. & Kaufman, O. T. (2018). A literature review exploring the use of programming in mathematics education. *International Journal of Learning, Teaching and Educational Research*, 17(12), 18–32. <https://doi.org/10.26803/ijlter.17.12.2>.

- Grover, S. & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43. DOI: 10.3102/0013189X1246305.
- Hana, G. M. (2014). *Matematiske tenkemåter*. Bergen: Caspar Forlag.
- Hinna, K. R. C., Rinvold, R. A. & Gustavsen, T. S. (2012). *QED 1-7: Matematikk for grunnskolelærerutdanningen bind 1*. Kristiansand: Høyskoleforlaget.
- Johnsen-Høines, M. (2006). *Begynneropplæringen: Fagdidaktikk for barnetrinnets matematikkundervisning* (2. utg.). Bergen: Caspar Forlag.
- Jonassen, G. H. & Kim, B. (2010). Arguing to learn and learning to argue: Design justifications and guidelines. *Educational Technology Research and Development*, 58(3), 439–457. DOI: 10.1007/s11423-009-9143-8.
- Kjerneelementgruppen for matematikk. (2017, 21. September). Kjerneelementer i matematikk, men hvorfor programmering? Hentet fra <https://udirbloggen.no/kjerneelementer-i-matematikk-men-hvorfor-programmering/>.
- Klovning, E. (2015, 04. juni). Programmering bør være allmennkunnskap. *Dagens næringsliv*. Hentet fra <https://www.dn.no/utdanning/hilde-c-bjornland/-programmering-bor-vare-allmennkunnskap/1-1-5391169>.
- Krummheuer, G. (1995). The ethnography of argumentation. I P. Cobb & H. Bauersfeld (Red.), *The emergence of mathematical meaning: Interaction in classroom cultures* (s. 229– 270). Hillsdale, New Jersey: Lawrence Erlbaum Associates.
- Krumsvik, R. J. (2014). *Forskningsdesign og kvalitativ metode: Ei innføring*. Bergen: Fagbokforlaget.
- Kuhn, D. (1992). Thinking as argument. *Harvard Educational Review*, 62(2), 155–178. <https://doi.org/10.17763/haer.62.2.9r424r0113t67011>.

- Kunnskapsdepartementet. (2018a, 26. juni). Fornyer innholdet i skolen. Hentet fra <https://www.regjeringen.no/no/aktuelt/forny-er-innholdet-i-skolen/id2606028/?expand=factbox2606064>.
- Kunnskapsdepartementet. (2018b). *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>.
- Kvale, S. & Brinkmann, S. (2015). *Det kvalitative forskningsintervju* (3. utg.). Oslo: Gyldendal akademisk.
- Lavy, I. (2006). A case study of different types of arguments emerging from explorations in an interactive computerized environment. *Journal of Mathematical Behaviour*, 25(2), 153–169. DOI: 10.1016/j.jmathb.2006.02.006.
- Lithner, J. (2000). Mathematical reasoning in task solving. *Educational Studies in Mathematics*, 41(2), 165–190. Hentet fra https://www.jstor.org/stable/3483188?seq=1#metadata_info_tab_contents.
- Lithner, J. (2006). *A framework for analyzing creative and imitative mathematical reasoning* (Department of Mathematics and Mathematical Statistics, Research reports in Mathematics Education, No.2). Umeå: Umeå universitet.
- Lithner, J. (2008). A research framework for creative and imitative reasoning. *Educational Studies in Mathematics*, 67(3), 255–276. Hentet fra <https://www.jstor.org/stable/40284656>.
- Nordahl, K. B. (2012). Systematisk observasjon av barns samhandling med andre: muligheter og utfordringer. I E. Backe-Hansen & I. Frønes (Red.), *Metoder og perspektiver i barne- og ungdomsforskning* (s. 158–173). Oslo: Gyldendal akademisk.
- NOU 2013: 2. (2013). *Hindre for digital verdiskaping*. Hentet fra <https://www.regjeringen.no/no/dokumenter/nou-2013-2/id711002/>.

Næss, N. G. & Sjøvoll, J. (2018). Observasjon som forskningsmetode. I M. Krogtoft & J. Sjøvoll (Red.), *Masteroppgaven i lærerutdanningen* (2. utg., s. 179–196). Oslo: Cappelen Damm Akademisk.

Programmeringsspråk. (2017, 13. desember). I *Store norske leksikon*. Hentet 19. januar 2019 fra <https://snl.no/programmeringsspr%C3%A5k>.

Russel, S. J., Schifter, D. & Bastable, V. (2011). *Connecting Arithmetic to Algebra*. Portsmouth: TERC.

Taylor, M., Harlow, A. & Forret, M. (2010). Using a computer programming environment and an interactive whiteboard to investigate some mathematical thinking. *Procedia Social and Behavioral Sciences*, 8, 561–570.
<https://doi.org/10.1016/j.sbspro.2010.12.078>.

Toulmin, S. E. (2003). *The uses of argument*. New York: Cambridge University Press.

Utdanningsdirektoratet. (2013). *Læreplan i matematikk (MAT1-04)*. Hentet fra <https://www.udir.no/kl06/MAT1-04>.

Utdanningsdirektoratet. (2018, 18. oktober). *Fagfornyninga – inspelsrunde skisser til læreplanar i matematikk*. Hentet fra https://hoering.udir.no/Hoering/v2/286?notatId=559&fbclid=IwAR1KW5Oay0PU3dQNYO_bFERI2XWu-K_nVxH2NaF2nYNahjrA1athtCSSgGk.

Utdanningsdirektoratet. (2019a, 18. mars). Hva er nytt i fagene? Les våre korte oppsummeringer. Hentet 19. mars 2019 fra <https://www.udir.no/laring-og-trivsel/lareplanverket/fagfornyelsen/hva-er-nytt-fagene-les-vare-korte-oppsummeringer/>.

Utdanningsdirektoratet. (2019b, 18. mars). *Høring: Læreplaner i matematikk*. Hentet 04. april 2019 fra <https://hoering.udir.no/Hoering/v2/343?notatId=656>.

- Waage, T. E. (2018, 12. november). 3,5 millioner til programmering. *Rogalands avis*. Hentet fra <https://www.dagsavisen.no/rogalandsavis/3-5-millioner-til-programmering-1.1232091>.
- Wang, F. & Hannafin, M. J. (2005). Design-based research and technology-enhanced learning environments. *Educational Technology Research and Development*, 53(4), 5–23. Hentet fra <https://link.springer.com/content/pdf/10.1007%2FBF02504682.pdf>
- Wyndhamn, J. & Säljö, R. (2009). Meaning-Making and the Appropriation of Geometric Reasoning: Computer Mediated Support for Understanding the Relationship between Area and Perimeter of Parallelograms. I R. Krumsvik (Red.). *Learning in the Network Society and the Digitized School* (s. 21–40). New York: Nova Science Publishers, Inc.
- Yackel, E. (2001). Explanation, justification and argumentation in mathematics classrooms. I M. van den Heuvel-Panhuizen (Red.), *Proceedings of the 25th International Conference on the Psychology of Mathematics Education, Vol 1* (s. 9–23). Utrecht, Holland: International Group for the Psychology of Mathematics Education.

Vedlegg 1: Oppgaveark til det første undervisningsopplegget

Scratch!



Oppgave 1: Utforsk Scratch

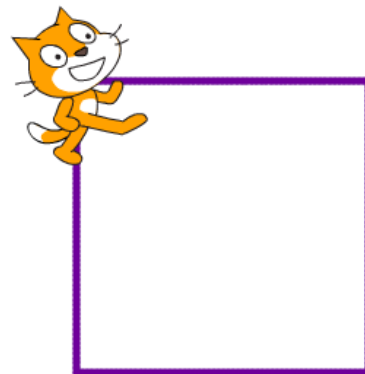
Bruk 5 minutt på å utforske i Scratch.

Gi figuren et navn og legg til en ekstra bakgrunn.

Oppgave 2: Forflytning!

Få figuren til å gå:

- I en firkant
- Skrått (som et kryss)
- Til kanten av bildet



Oppgave 3: Bytt bakgrunn

Gjør slik at bakgrunnen skifter til en ny bakgrunn når figuren kommer til kanten.

Oppgave 4: Drakter

- Prøv å få figuren til å bytte drakt når han går.
 - o Ser det ut som om figuren går og ikke glir?

Oppgave 5: Kommunikasjon

- Sett inn en ny figur.
- Få figurene til å hilse på hverandre.
- Få ny figur til å vise kun på bakgrunn 2



Oppgave 6: Andre funksjoner

- Gjør slik at kun den første figuren vises når grønt flagg klikkes. (Figur 2 er skjult)
- Få alt til å starte i samme posisjon hver gang når grønt flagg klikkes.

Vedlegg 2: Samtykkeskjema til de foresatte

Vil du delta i forskningsprosjektet «Argumentasjon og kritisk matematikkundervisning i flerspråklige klasserom»?

Dette er et spørsmål om å delta i et forskningsprosjekt om argumentasjon og kritisk tenkning i matematikkundervisning i flerspråklige klasserom. I dette skrivet informerer vi kort om innholdet i prosjektet og hva deltakelse innebærer.

Bakgrunn og formål

Prosjektet handler om å fremme lærerstudenters kompetanse i å legge til rette for argumentasjon og kritisk matematikkundervisning for elever i flerspråklige klasserom på barnetrinnet. Dette kan være å kritisk kunne vurdere matematikkforklaringer og å se matematikkens rolle i argumentasjon om aktuelle samfunnsproblemer. Skolene som er med i prosjektet er partnerskoler eller praksisskoler som allerede er en del av et samarbeid mellom Bergen kommune og Høgskulen på Vestlandet (HVL). Prosjektet varer i fire år og er et forskningssamarbeid mellom lærere og elever ved partnerskoler og tilsatte og studenter ved matematikklærerutdanningen ved HVL.

Som en del av dette forskningsprosjektet ønsker to masterstudenter (Oda Åbø Pettersen og Tonje Lindberg) å samle inn datamateriale der elever arbeider med programmering i Scratch. Målet er å få innsikt i hvordan matematisk argumentasjon og programmering kan påvirke hverandre.

Hvem er ansvarlig for forskningsprosjektet?

Høgskolen på Vestlandet er ansvarlig for prosjektet, og det er ledet av Professor Tamsin Meaney. Prosjektet gjennomføres i samarbeid med Bergen Kommune, og det er støttet av Norges forskningsråd.

Hvorfor får du spørsmål om å delta?

Vi spør om å la ditt/deres barn få delta i prosjektet fordi klassen barnet går i har drevet med Scratch tidligere og er dermed kjent med programmering fra før av.

Hva innebærer det for deg å delta?

Det vil bli gjennomført undervisning der elevene arbeider i par/grupper med programmet Scratch. Det er særlig fokus på matematisk argumentasjon der elevene oppfordres til å reflektere, diskutere og dele kunnskap. Deltagelse innebærer at noe av undervisningen barnet deltar i blir observert og filmet. Alle elever får tilbud om samme undervisningen, men kun de som samtykker, blir observert og filmet.

Det er frivillig å delta

Det er frivillig å delta i studien, og du/dere kan uten grunngeving når som helst trekke ditt/deres samtykke. Hvis du/dere trekker barnet fra prosjektet vil alle opplysninger om barnet bli anonymisert. Det vil ikke få negative konsekvenser hvis du/dere ikke ønsker at barnet skal delta, eller senere velger å trekke ditt/deres barn fra prosjektet.

Ditt/deres barns personvern – hvordan vi oppbevarer og bruker barnets opplysninger

Alle personopplysninger blir behandlet konfidensielt og personidentifiserbart materiale lagres på HVL sin forskningsserver, sikret med brukernavn og passord. Kun deltakere i prosjektgruppen og eventuelt transkriberingsfirma har tilgang til materialet. Deltakere vil ikke kunne bli identifisert i publikasjoner.

Prosjektet skal avsluttes 31.12.2023. Etter denne dato vil alle personidentifiserende data slettes og materialet vil ikke lengre være lagret på HVL sin forskningsserver. Videre bruk av dataene blir i presentasjoner, undervisning, eventuelle oppfølgingsstudier og senere forskning basert på transkribert og anonymisert materiale.

Dine/deres rettigheter

Så lenge ditt barn kan identifiseres i datamaterialet, har du/dere rett til:

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

Hva gir oss rett til å behandle personopplysninger om ditt/deres barn?

Vi behandler opplysninger om ditt/deres barn basert på ditt/deres samtykke. På oppdrag fra HVL har Norsk senter for forskningsdata (NSD) vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

Hvor kan jeg finne ut mer?

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

- Prosjektleder Tamsin Meaney på tlf.: 55 58 55 69 eller epost: Tamsin.Jillian.Meaney@hvl.no
- Masterstudentene:
 - Oda Åbø Pettersen på tlf.: 46657720 eller epost: 145522@stud.hvl.no
 - Tonje Lindberg på tlf.: 93417083 eller epost: 145057@stud.hvl.no
- HVL sitt personvernombud: Advokat Halfdan Mellbye, personvernombud@hvl.no
- NSD – Norsk senter for forskningsdata AS, på epost: personverntjenester@nsd.no eller telefon: 55 58 21 17.



Samtykkeerklæring forskningsprosjektet

Jeg/vi har mottatt og forstått informasjon om prosjektet «Argumentasjon og kritisk matematikkundervisning i flerspråklige klasserom» og fått anledning til å stille spørsmål. Jeg/vi samtykker til at barnet mitt/vårt kan:

- delta i videooptak
- delta i lydoptak

Jeg/vi samtykker til at mitt/vårt barns opplysninger behandles frem til prosjektet er avsluttet, 31.12.2023.

(Signert av prosjektdeltakers foresatte, dato)

Samtykkeerklæring for bruk av videoer:

- Jeg samtykker i at videosnutter der mitt/vårt barn er med kan vises i presentasjoner og undervisning.

(Signert av prosjektdeltakers foresatte, dato)

Vedlegg 3: Samtykkeskjema til elevene

Vil du delta i forskningsprosjekt om programmering i Scratch?

Hei! Vi heter Tonje Lindberg og Oda Åbø Pettersen og kommer fra Høgskolen på Vestlandet. Vi ønsker å undersøke hvordan dere arbeider med programmering i Scratch. Da trenger vi å filme noen av dere og ta skjermopptak av datamaskinen. Dette gjør vi fordi vi skal skrive en masteroppgave om programmering og matematikk.

Du kan velge om vi skal få lov til å filme deg. Dersom du ønsker å være med vil det kun være vi som jobber med prosjektet som får se og høre opptakene av deg. Dette er helt frivillig, og du kan ombestemme deg når som helst. Ingen vil kunne gjenkjenne deg når de leser masteroppgavene våre.

Hvis du lurer på noe, ta gjerne kontakt:

- Oda Åbø Pettersen på tlf.: 46657720 eller på epost: 145522@stud.hvl.no
- Tonje Lindberg på tlf.: 93417083 eller på epost: 145057@stud.hvl.no



Samtykkeerklæring forskningsprosjektet

Jeg har mottatt og forstått informasjon om prosjektet og fått anledning til å stille spørsmål. Jeg synes det er greit at jeg:

- deltar i videopptak

Jeg synes det er greit at mine opplysninger behandles frem til prosjektet er avsluttet, 31.12.2021.

(Min signatur (elev), dato)

Samtykkeerklæring for bruk av videoer:

- Jeg synes det er greit at videosnutter der jeg er med kan vises i presentasjoner og undervisning.

(Min signatur (elev), dato)