

## **Problemløsningsoppgaver i Scratch**

En studie av elevers algoritmiske tenkning

BENTE BLOCH-JOHNSEN LOHNE

REBEKKA HELEN KOLSTAD FRANKRIG

VEILEDER

Kjetil Damsgaard

**Universitetet i Agder, 2023**

Fakultet for teknologi og realfag

Institutt for matematiske fag

Master



# Forord

Denne oppgaven setter et etterlengtet punktum etter 5 år på lærerutdanningen ved Universitetet i Agder. Det har vært 5 lærerike år og vi føler oss godt rustet for arbeidslivet i skolen. Når vi ser tilbake på prosessen med masteroppgaven sitter vi igjen med både glede og frustrasjon. Det har vært mange lange dager, flere vendepunkter og en stressende periode. Vi har lært mye, både om oss selv og det vi har forsket på. Vi opplever at studien har gitt oss kunnskap om hvordan algoritmisk tenkning kommer til uttrykk hos disse elevene, og håper at studien kan bidra til å berike feltet innenfor algoritmisk tenkning.

Vi vil benytte anledningen til å takke vår veileder, Kjetil Damsgaard, som har gitt oss gode råd og støttet oss under hele prosessen. Han har stilt kritiske spørsmål, og veiledet oss slik at oppgaven vår er blitt et produkt vi er stolte av. Vi vil også takke læreren og elevene som var villige til å bidra i denne studien. Vi ønsker også å takke familie, kjærester og venner for å ha støttet oss gjennom studiet, spesielt i denne krevende prosessen. Vi føler oss heldige som har så mange fine mennesker rundt oss.

En avgjørende faktor for en god opplevelse under utdanningen er vennskapet vi to har hatt. Vi ønsker å takke hverandre for et godt samarbeid, både i arbeid med oppgaven, men også gjennom hele studiet. Vi kunne ikke ha tenkt oss å skrive masteroppgaven med noen andre. Det har vært en positiv opplevelse å kunne le og ha det morsomt på egen bekostning, og de utallige latterkrampene i Vrimlehallen ville vi aldri vært foruten.

Stavanger/Tysvær, mai 2023

*Bente Bloch-Johnsen Lohne og Rebekka Helen Kolstad Frankrig*

## Sammendrag

Gjennom et utvalg på fire elever på 6. trinn, ser denne studien på hvordan algoritmisk tenkning kommer til uttrykk i arbeid med problemløsningsoppgaver i Scratch. Studien er en kvalitativ case-studie, hvor elevene har gjennomført et undervisningsopplegg i par. Undervisningsopplegget inneholder problemløsningsoppgaver og legger til rette for algoritmisk tenkning. Datamaterialet ble samlet inn ved skjerm- og lydopptak, samt intervju. Datamaterialet ble videre analysert ut ifra forhånds definerte nøkkelbegreper for algoritmisk tenkning, og videre forklart og diskutert. Studien baseres på teori om Computational Thinking (CT), og bygger videre på tidligere forskning om hvordan algoritmisk tenkning vises hos elever. Ettersom vår studie innebærer bruk av teknologi og programmering, har vi også inkludert to modeller som beskriver bruk av teknologiske verktøy i skolen.

Resultatene viser stor variasjon i hvordan algoritmisk tenkning kommer til uttrykk hos elevene i denne studien. Det kommer frem at elevene i stor grad har evne til algoritmisk tenkning, men vi ser også at det er enkelte elementer innenfor nøkkelbegrepene som elevene ikke mestrer. Funnene viser at det er en utbredt bruk av logikk ved at elevene utformer hypoteser. Elevene bruker også dekomposisjon i stor grad ved at de deler opp en sammensatt figur, og programmerer de ulike figurene hver for seg. Vi ser også at elevene mestrer å forenkle løsningene sine ved bruk av løkker, noe som er et aspekt innen evaluering og abstraksjon. En mulig implikasjon fra denne studien er å undersøke om digitale arbeidsmåter gir bedre effekt på utviklingen av elevers algoritmiske tenkning, sammenliknet med analoge arbeidsmåter.

## Summary

Through a sample of four 6th-grade students, this study investigates how computational thinking is expressed in problem-solving tasks using Scratch. The study is a qualitative case study, where the students have completed a teaching module in pairs. The teaching module contains problem-solving tasks and facilitates computational thinking. The data was collected through screen and audio recordings, as well as interviews. The data was analyzed based on pre-defined key concepts for computational thinking, and further explained and discussed. The study is based on theory of Computational Thinking (CT) and builds on previous research on how computational thinking is manifested in students. As our study involves the use of technology and programming, we have also included two models that describe the use of technological tools in schools.

The results show significant variation in how computational thinking is expressed among the students in this study. It is evident that students have a high level of ability in computational thinking, but we also see that there are certain elements within the key concepts that they do not master. The findings show that there is a widespread use of logic, where students formulate hypotheses. Students also use decomposition to a great extent, by breaking down a complex figure and programming the different figures separately. We also see that the students are able to simplify their solutions by using loops, which is an aspect of evaluation and abstraction. An implication of this study is to investigate whether digital ways of working have a greater impact on the development of students' computational thinking, compared to analog ways of working.

# Innholdsfortegnelse

Forord.....	ii
Sammendrag .....	iii
Summary .....	iv
1.0 Innledning .....	1
2.0 Teori.....	3
2.1 Digitalisering i skolen .....	3
2.2 Algoritmisk tenkning.....	6
2.2.3 Nøkkelbegreper.....	7
2.3 Scratch sin relevans i problemløsning.....	9
3.0 Metode .....	13
3.1 Valg av metode.....	13
3.2 Casestudie.....	14
3.3 Utvalg.....	14
3.4 Pilot .....	15
3.5 Gjennomføring .....	15
3.6 Datainnsamlingsmetode .....	16
3.6.1 Observasjon.....	16
3.6.2 Intervju .....	17
3.7 Oppgavedesign .....	18
3.8 Analysestrategi.....	23
3.8.1 Transkripsjon .....	23
3.8.2 Analyse .....	24
3.9 Kvalitetskriterier.....	25
3.9.1 Gyldighet.....	25
3.9.2 Pålitelighet .....	26
3.9.3 Ethiske betraktninger .....	27
4.0 Resultat og analyse .....	29
4.1 Case 1: Anna og Amanda.....	29
4.2 Case 2: Bendik og Beate .....	40
5.0 Diskusjon .....	51
6.0 Konklusjon og implikasjoner .....	59

7.0	Litteraturliste.....	63
8.0	Vedlegg.....	67
8.1	Informasjonsskriv og samtykkeskjema.....	67
8.2	Vurdering fra Sikts personverntjenester.....	71
8.3	Intervjuguide.....	74
8.4	Oppgavene.....	75
8.5	Transkripsjon case 1.....	79
8.5.1	Oppgaveløsning.....	79
8.5.2	Intervju.....	103
8.6	Transkripsjon case 2.....	112
8.6.1	Oppgaveløsning.....	112
8.6.2	Intervju.....	137





## 1.0 Innledning

Ny teknologi er med på å endre samfunnet vi lever i og vil påvirke arbeidslivet i årene som kommer (Meld. St. 28 (2015-2016), s. 54). Barr og Stephenson (2011, s. 49) poengterer at alle elever i dagens samfunn kommer til å leve et liv som er under stor påvirkning av teknologi, og at flere av jobbene de kommer til å få, også vil påvirkes av dette. Videre konstaterer de at elever må få mulighet til å jobbe med algoritmisk problemløsning, og databehandlingsmetoder og verktøy i grunnskolen (Barr og Stephenson, 2011, s. 49). Med bakgrunn i dette, valgte vi å gjennomføre en studie av elevers algoritmiske tenkning i arbeid med det digitale verktøyet, Scratch.

I et av kjerneelementene i læreplanen i matematikk trekkes det frem at algoritmisk tenkning er en viktig prosess innen problemløsning (Kunnskapsdepartementet, 2019, s. 2). Vi tolker læreplanen slik at algoritmisk tenkning og problemløsning går hånd i hånd. Wing (2006, s. 33) poengterer også at algoritmisk tenkning er en fundamental ferdighet som alle bør ha. Hun sier videre at det bør legges til rette for at elever får utvikle sin evne til algoritmisk tenkning, og det er det vi har forsøkt å gjøre i vår studie.

På bakgrunn av den teknologiske utviklingen i samfunnet, samt vår egen interesse for fagfeltet, ønsket vi å se nærmere på hvordan elevene faktisk viser sin algoritmiske tenkning. Kunnskapen vi får fra studien vil vi bruke til å legge til rette for utviklingen av ferdighetene som kreves for at våre fremtidige elever skal bli algoritmiske tenkere. Derfor utviklet vi følgende forskningsspørsmål:

Hvordan kommer elevenes algoritmiske tenkning til uttrykk i arbeid med problemløsningsoppgaver i Scratch?

Algoritmisk tenkning er en kognitiv prosess og Utdanningsdirektoratet (2019) legger frem seks nøkkelbegreper som inngår i denne prosessen, og viser til en modell (se illustrasjon 1).



Illustrasjon 1: "Den algoritmiske tenkeren" utviklet av Utdanningsdirektoratet (2019).

Denne modellen har vi brukt som inspirasjon og vil ved hjelp av tidligere forskning definere nøkkelbegrepene i modellen. Nøkkelbegrepene vil videre danne grunnlag for analysen og vi vil studere elevenes arbeid med fokus på de ulike aspektene innenfor algoritmisk tenkning. På denne måten vil vi fange opp hvordan elevene bruker algoritmisk tenkning i arbeid med problemløsningsoppgaver.

Videre i studien vil vi presentere to modeller som omhandler digitale verktøy i skolen, samt tidligere forskning knyttet til algoritmisk tenkning og problemløsningen. Deretter vil vi i metodekapittelet fremlegge de metodiske valgene vi har gjort, beskrive hvordan undervisningsopplegget og datainnsamlingen foregikk, samt presentere oppgavene elevene arbeidet med. I resultat- og analysekapittelet vil vi legge frem elevenes arbeid med oppgavene og presentere funn. Avslutningsvis vil vi diskutere funnene og koble dem til teori som er presentert, og videre runde av med en konklusjon og implikasjoner.

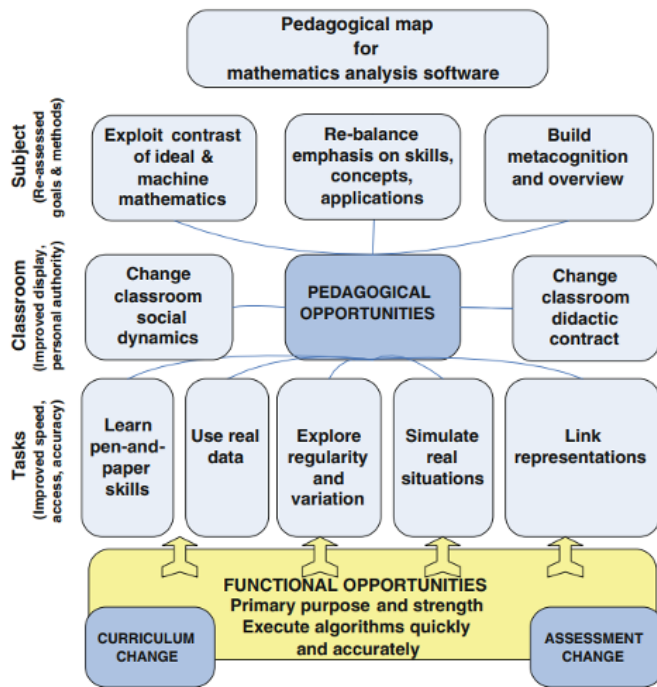
## 2.0 Teori

I dette kapitlet vil vi presentere relevant teori og tidligere forskning. Vi vil først presentere to modeller som omhandler didaktisk bruk av digitale verktøy i skolen. Deretter vil vi redegjøre for ulike definisjoner av algoritmisk tenkning, og videre presentere en kategorisering av algoritmisk tenkning gjennom seks nøkkelbegreper. Til slutt vil vi legge frem en kobling mellom problemløsning, algoritmisk tenkning og programmering ved hjelp av Scratch. Dette gjøres ved å først definere begrepet problemløsning. Videre vil vi se på et av kjerneelementene i læreplanen i matematikk, hvor problemløsning og algoritmisk tenkning er representert. Deretter vil vi presentere tidligere forskning hvor utviklingen av elevers problemløsningsferdigheter knyttes opp mot programmering.

### 2.1 Digitalisering i skolen

Digitalisering i skolen er noe som er uunngåelig. Samfunnet endrer seg raskt og skolen må gjøre grep for å endres i takt med dette. I dette kapitlet vil vi presentere to modeller som beskriver funksjonen til digitale verktøy i skolen.

Pierce og Stacey (2010) har utviklet et “pedagogisk kart” der de oppsummerer fordeler med å bruke teknologi i undervisning. De mener dette kan være til hjelp både for lærere og forskere. Modellen inneholder ti pedagogiske dimensjoner, og disse er delt inn i tre nivåer. De tre nivåene er oppgavedimensjonen, klasseromsdimensjonen og fagdimensjonen, og presenteres som tre rader i modellen.



Illustrasjon 2: "Pedagogical map for mathematics analysis software", utviklet av Pierce og Stacey (2010).

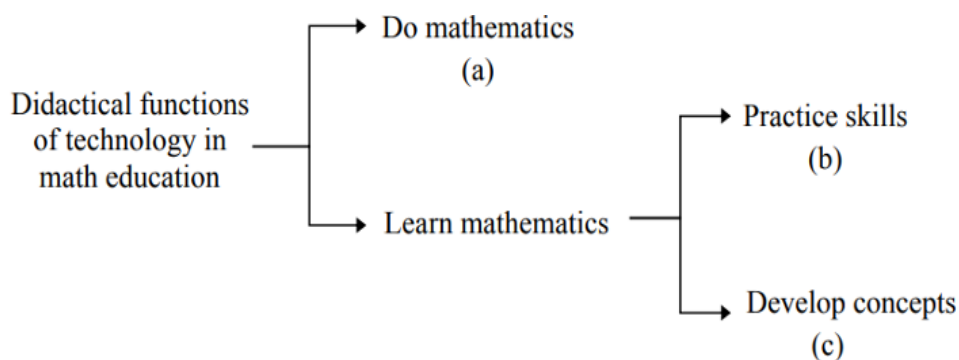
Opgavedimensjonen i modellen handler om hvordan teknologi kan endre oppgavene i matematikken (Pierce & Stacey, 2010, s. 8). Leung og Bolite-Frant (2015, sitert i Santos-Trigo, 2016, s. 23) trekker frem at ved å ta i bruk dynamiske geometriprogrammer, som for eksempel Geogebra eller Scratch, vil elevene få mulighet til å utforske nye figurer og videreutvikle argumentasjonen. Laborde (2000, sitert i Pierce & Stacey, 2010, s. 8) legger vekt på at slike programmer gir elevene mulighet til å utforske variasjon og regelmessighet, og videre komme med antagelser.

Klasseromsdimensjonen handler om hvordan bruk av teknologi er med på å endre rollene i klasserommet (Pierce & Stacey, 2010, s. 9). Programvarer kan brukes til å gi pålitelige svar på komplekse problemer og dermed er ikke læreren lenger den eneste med faglig autoritet (Pierce & Stacey, 2010, s. 9). Ved at det introduseres en ny autoritet i klasserommet kan elevene få en følelse av personlig autoritet (Pierce & Stacey, 2010, s. 9). Dette fører videre til at elevene får mulighet til å ta et større ansvar for egen læring, noe som resulterer i at den didaktiske kontrakten endres (Brousseau, 1997, sitert i Pierce & Stacey, 2010, s. 9).

Fagdimensjonen handler om i hvor stor grad mulighetene for teknologi støtter nye mål eller undervisningsmetoder i matematikkundervisning (Pierce & Stacey, 2010, s. 10). Oppgaver

som starter med en annen inngangsvinkel, for eksempel å starte med en oversikt eller å bruke teknologi til å generere resultater, og deretter gå tilbake til detaljene, hører til fagdimensjonen av modellen (Pierce & Stacey, 2010, s. 12). Å bruke dynamiske undersøkelser inngår også i fagdimensjonen. Det kan fungere som en start for mer abstrakt geometri og gjør at balansen i geometri undervisningen flyttes mer bort fra å lære fakta, og mer mot å støtte elevenes argumentasjon og koblinger til beviser (Pierce & Stacey, 2010, s. 12).

Drijvers (2013) har utviklet en illustrasjon der det beskrives hvordan en vurderer den didaktiske funksjonen av teknologiske verktøy i matematikkundervisningen (Drijvers, 2013, s. 3). Illustrasjonen deles først inn i om elevene bruker verktøyet til å gjøre matematikk som ellers kunne blitt gjort for hånd (Do mathematics, a), eller om de bruker verktøyet til å lære matematikk (Drijvers, 2013, s. 3). Deretter deles sistnevnte inn i hvorvidt elevene øver på ferdigheter (Practice skills, b) eller om de utvikler konseptuell forståelse (Develop concepts, c) (Drijvers, 2013, s. 3).



Illustrasjon 3: “Didactical functions of technology in mathematics education”, utviklet av Drijvers (2013).

I tillegg nevner Drijvers tre faktorer som har betydning for en vellykket integrering av teknologi i klasserommet (Drijvers, 2013, s.14). 1) *Designet*, som omhandler både designet av Scratch, verktøyet vi bruker, designet av oppgavene i undervisningsopplegget, samt undervisning generelt (Drijvers, 2013, s. 14-15). 2) *Lærerens rolle*, og hvordan denne rollen endres i takt med den didaktiske kontrakten. Denne faktoren går altså ikke ut på å minske lærerens rolle, men heller endre den. Slik at lærerens oppgave er å legge til rette for læring innenfor det programmet som brukes (Drijvers, 2013, s. 15). 3) *Den pedagogiske konteksten*, knyttes til elementene i det pedagogiske kartet av Pierce og Stacey (2010) (Drijvers, 2013, s.

15), som nevnt over. Den pedagogiske konteksten inneholder også flere andre viktige faktorer, som for eksempel elevenes motivasjon og engasjement (Drijvers, 2013, s. 15).

## 2.2 Algoritmisk tenkning

I dette kapitlet vil vi forklare nærmere hva som inngår i algoritmisk tenkning. Den engelske oversettelsen av algoritmisk tenkning er computational thinking (Utdanningsdirektoratet, 2019), samtidig er forholdet mellom de to begrepene omdiskutert. I denne studien har vi, i likhet med Utdanningsdirektoratet (2019), valgt å se på computational thinking som en direkte oversettelse av det norske begrepet, og vi fokuserer derfor på forskning omkring Computational Thinking (CT).

Den teknologiske utviklingen i samfunnet og i skolen har ført til en diskusjon om hvordan teknologi skal undervises. Til tross for at algoritmisk tenkning har blitt en større del av denne diskusjonen, er det ikke enighet blant forskere om hva algoritmisk tenkning innebærer (García-Peñalvo & Mendes, 2018, s. 408; Barr & Stephenson, 2011, s. 49). En av forskerne på dette temaet er Jeanette Wing, som har vært en stor bidragsyter innenfor algoritmisk tenkning i nyere tid (Nouri et al., 2019, s. 2). Wing la, i 2006, frem en definisjon på begrepet der hun vektlegger problemløsning, design av systemer og det å forstå menneskelig oppførsel (Wing, 2006, s. 33). I en senere definisjon vektlegger hun tankeprosessen som er involvert i formuleringen av et problem, og hvordan løsningen blir uttrykt (Wing, 2017, s. 8).

Det finnes også flere definisjoner av CT, for eksempel Aho (2012, s. 832) som definerte algoritmisk tenkning som tankeprosessen som er involvert i å formulere et problem, der løsningen kan bli representert trinnvis og ved hjelp av algoritmer. Denne definisjonen er med på å bygge under den senere definisjonen til Wing (2017), ettersom fokuset er på tankeprosessen for å løse et problem. Aho (2012, s. 832) legger stor vekt på algoritmer i sin definisjon, og understreker at løsninger blir representert trinnvis.

Computer Science Teachers Association (CSTA) og International Society for Technology in Education (ISTE) har lagt frem en operasjonell definisjon av begrepet (2011, sitert i Román-González et al., 2017, s. 679). I denne definisjonen trekkes det frem at CT er en problemløsningsprosess som inkluderer, men ikke begrenser seg til, spesifikke kjennetegn

(CSTA & ISTE, 2011, sitert i Román-González et al., 2017, s. 679). I disse kjennetegnene legges det vekt på å kunne formulere problemer på en slik måte at vi kan bruke et dataprogram til å løse dem, samt å organisere og analysere data og å representere data gjennom abstraksjon. Videre trekkes det også frem å automatisere løsninger gjennom trinnvise steg og å identifisere og analysere mulige løsninger for å komme frem til den mest effektive kombinasjonen. Kjennetegnene inkluderer også det å generalisere og overføre tidligere løsninger til mer komplekse problemer (CSTA & ISTE, 2011, sitert i Román-González et al., 2017, s. 679).

I likhet med definisjonen over, trekker også García-Peñalvo og Mendes (2018, s. 408) frem at CT er en metode for å løse problemer der elevene tar i bruk ulike konsepter som for eksempel abstraksjon, eller å finne likheter, slik at de kan prosessere og analysere data (García-Peñalvo & Mendes, 2018, s. 408).

Wing (2006, s. 33) påpeker at CT også går ut på å omformulere tilsynelatende vanskelige problemer, til problemer som en vet hvordan en skal løse. Med dette menes å dele opp problemet og velge en passende representasjon slik at problemet bli håndterbart, altså abstraksjon og dekomposisjon (Wing, 2006, s. 33). Dette kan gjøres ved å redusere, transformere og simulere problemet (Wing, 2006, s. 33). Det handler ikke bare om å transformere problemet du står overfor, men også å kunne evaluere om løsningen din er korrekt og effektiv (Wing, 2006, s. 33). Innenfor evalueringen er det også faktorer som går på programmets estetikk, enkelhet og eleganse (Wing, 2006, s. 33).

### 2.2.3 Nøkkelpbegreper

For å vite hvordan vi skal gå frem for å kartlegge våre elevers algoritmiske tenkning, har det vært nødvendig å studere ulike forskningsartikler. Vi har vært på jakt etter hvordan kompetanse knyttet til algoritmisk tenkning blir presentert. Målet har vært å lage kategorier for algoritmisk tenkning som vi kan bruke i analysen av elevenes tenkning. I dette delkapittelet vil vi presentere en kategorisering av algoritmisk tenkning i noen nøkkelpbegreper.

Nøkkelpbegrepene er identisk med innholdet i modellen “den algoritmiske tenkeren” fra Utdanningsdirektoratet (2019), og bygger på artikkelen til Selby og Woollard (2013). Denne kategoriseringen inneholder begreper som har bred støtte i forskning innen computational thinking. Og vi vil i presentasjonen av nøkkelpbegrepene vise hvordan ulike studier støtter opp under relevansen til disse begrepene.

## ① Logikk

Logikk går ut på at elevene organiserer og analyserer data (CSTA & ISTE, 2011, sitert i Román-González et al., 2017, s. 679). Det lar elevene bruke deres egen kunnskap og interne modeller til å lage og bekrefte hypoteser (L'Heureux et al., 2012, sitert i Selby & Woollard, 2013, s. 3). I logikk inngår altså å fikse kodene, finne ut hvorfor noe ikke fungerer og systematisk teste kodene. I dette nøkkelbegrepet inngår også å forstå og modifisere koder som er skrevet av andre, eller å lage nye programmer (Weintrop et al., 2016, s.139).

## ② Algoritmer

Algoritmer beskrives som en trinnvis prosedyre for å kunne utføre oppgaver (Selby & Woollard, 2013, s. 3). Det er også evnen til å presentere løsningen i form av sekvenser (Korkmaz et al., 2017, s. 561). Barr og Stephenson (2011, s. 52) trekker frem at det å implementere en algoritme for å løse et problem er et aspekt innen CT. Å bruke algoritmer til å løse et problem er altså å forstå, bruke, vurdere og lage egne algoritmer (Brown, 2015, sitert i Korkmaz et al., 2017, s. 560).

## ③ Dekomposisjon

Dekomposisjon kan forstås, ifølge Wing (2006, s. 33), som å reformulere et tilsynelatende vanskelig problem til et problem som vi vet hvordan vi løser. Det er en måte å tenke på et problem som en samling av ulike deler. Det handler om å dekomponere problemer til delproblemer, og å simplificere komplekse problemer og gjøre de mer oppnåelige (Weintrop et al., 2016, s. 139). En del av dekomposisjonen er også å definere de ulike delene og metodene som kreves for å løse dem (Barr & Stephenson, 2011, s. 52). Elever som mestrer dette har muligheten til å reformulere problemer slik at de kan løses, eller i det minste gjøre fremgang (Weintrop et al., 2016, s. 139).

## ④ Mønstre

Mønstre blir av Selby og Woollard (2013, s. 4), forklart gjennom ordet generalisering, og bygger videre på dekomposisjon. Generalisering går ut på å identifisere mønstre, likheter og forbindelser (Selby & Woollard, 2013, s. 4). For eksempel å forstå hvordan en tegner et kvadrat, ved å definere de indre vinklene, for så å bruke den samme algoritmen til å kunne produsere en sirkel (Selby & Woollard, 2013, s. 4). På denne måten kan en generalisere en



problemløsningsprosess, for å kunne bruke den til å løse en bredere variasjon av problemer (CSTA & ISTE, 2011, sitert i Román-González et al., 2017, s. 679).

### ⑤ Abstraksjon

Abstraksjon handler om at elevene danner seg nye begreper og representerer en ide eller prosess, på en mer generell form (Weintrop et al., 2016, s. 139). Abstraksjon er essensielt for at elevene skal kunne løse flere problemer, som har lik struktur, men annerledes detaljer (Weintrop et al., 2016, s. 139). Et annet aspekt innen abstraksjon er å se bort fra de delene av problemet eller koden som ikke er viktig (Weintrop et al., 2016, s. 139). Elevene trenger altså ikke å forstå alle detaljene i et komplekst problem for å kunne løse det (Wing, 2006, s. 33). Videre er det å kunne forenkle et problem en del av abstraksjon (Barr & Stephenson, 2011, s. 52).

### ⑥ Evaluering

Å analysere noe ut ifra løsningen en har kommet frem til kan, ifølge Selby og Woollard (2013, s. 3), forstås som å evaluere noe. Wing (2006, s. 33) trekker frem at evaluering er å bedømme et program/kode etter både effektivitet og riktighet, samt enkelhet, altså om koden er skrevet på enklest mulig vis. Selv om to forskjellige løsninger produserer samme resultat, må elevene også ta andre faktorer i betraktning, som for eksempel tid, fleksibilitet og at de kan bruke løsningen sin videre (Weintrop et al., 2016, s. 139). Elever som mestrer å evaluere viser at de kan vurdere ulike løsninger for et problem, basert på hva de skal finne ut, og hvilke mulige verktøy de har for å løse det (Weintrop et al., 2016, s. 139). Altså er det ikke bare resultatet som har betydning, men også alle detaljene i løsningen.

## 2.3 Scratch sin relevans i problemløsning

Problemløsning går åpenbart ut på å løse et problem. Det å løse et problem er, ifølge Pólya (1962, s. 117), å lete bevisst etter en *handling* som kan passe for å oppnå et mål som er klart utformet, men som ikke er umiddelbart oppnåelig. Når du har løst et problem har du funnet denne handlingen. For at noe skal være et problem, må det være noe som ikke er kjent. Dersom du kjenner til alt på forhånd er det ikke noen problemer som skal løses (Pólya, 1962, s. 3). Samtidig må noe være gitt, altså elevene må få noe data for å kunne løse problemet (Pólya, 1962, s. 3). Det må også være en kobling mellom dataene elevene får, og det som er

ukjent (Pólya, 1962, s. 4). Matematiske problemer er sentralt for å kunne fremme elevers læring (Pólya 1945; Halmos 1994, s. 19).

Problemløsning og utforskning er et av kjerneelementene i læreplanen i matematikk (Kunnskapsdepartementet, 2019, s. 2). I kjerneelementene står det at algoritmisk tenkning er viktig innenfor problemløsning og det blir nevnt som en måte å utvikle strategier og fremgangsmåter for å løse et problem (Kunnskapsdepartementet, 2019, s. 2). Altså går algoritmisk tenkning og problemløsning hånd i hånd.

I studien til Kalelioğlu & Gülbahar (2014, s. 47) viste resultatene at programmet Scratch ikke hadde en signifikant betydning for utviklingen av problemløsningsferdigheter. De understreker sannsynligheten for at resultatene kun kan knyttes til egen forskning, samt at lignende forskning kan gi et annet resultat (Kalelioğlu & Gülbahar, 2014, s. 47). Samtidig viser de til at det var en liten forbedring av elevenes selvtillit knyttet til egne problemløsningsferdigheter. På bakgrunn av dette påpeker de muligheten for at programmering kan påvirke elevenes ferdigheter knyttet til problemløsning (Kalelioğlu & Gülbahar, 2014, s. 47).

Resultatene fra Lai og Yangs (2011, s. 6943) studie viser derimot at programmering i Scratch hadde en signifikant positiv påvirkning på elevenes problemløsningsferdigheter. I denne studien deltok elevene i et kurs som gikk over et helt semester (Lai & Yang, 2011, s. 6940), i motsetning til Kalelioğlu og Gülbahars (2014) studie som kun gikk over en kort periode. Funnene viser at når elevene bruker Scratch får de mulighet til å bruke variabler og løkke konsepter, samt mulighet til utforme og teste hypoteser, og å endre på kodene sine (Lai & Yang, 2011, s. 6943).

Basert på tidligere forskning konstaterer også Kalelioğlu og Gülbahar (2014, s. 48-49) at det er åpenbart at programmering kan forbedre problemløsningsferdighetene for elever i alle aldre. De vektlegger derfor at lærere må utvikle et miljø for å bruke programmering i undervisningen, spesielt hos yngre elever (Kalelioğlu & Gülbahar, 2014, s. 48-49). Fessakis et al. (2013, s. 96) påpeker også viktigheten av at det må utvikles et miljø for å kunne programmere. Videre trekker de frem at det må utvikles nye undervisningsmetoder og læringsressurser, samt at programmering må inn i læreplanen (Fessakis et al., 2013, s. 96). I den nye læreplanen i matematikk, har programmering fått en større plass, og det er

kompetansemål i matematikk som går direkte på programmering (Kunnskapsdepartementet, 2019, s. 5). I den norske skolen ligger det derfor mer, enn tidligere, til rette for å kunne utvikle dette miljøet som Kalelioğlu og Gülbahar (2014, s. 48-49) og Fessakis et al. (2013, s. 96) trekker frem.



## 3.0 Metode

I denne delen av oppgaven vil vi redegjøre for forskningsmetoden vi har valgt, samt de metodiske valgene vi har gjort i forkant av og underveis i studien. Vi vil presentere utvalget som har deltatt i studien, gjennomføringen av undervisningsopplegget og hvordan datainnsamlingen foregikk. Videre vil vi presentere hvordan oppgavene er utviklet og hvordan datamaterialet ble analysert. Til slutt vil vi drøfte ulike kvalitetskriterier knyttet til gyldighet og pålitelighet, samt etiske betraktninger.

### 3.1 Valg av metode

På bakgrunn av vår problemstilling har vi valgt å bruke en kvalitativ metode. Som i alle forskningsmetoder, finnes det styrker og svakheter med kvalitativ metode (Postholm & Jacobsen, 2018, s. 89). Vi vil gjøre rede for begrensningene og mulighetene vi har på bakgrunn av valgt metode.

Målet med vår forskning er å få inngående kunnskap om hvordan algoritmisk tenkning kommer til uttrykk hos elever. For å oppnå inngående kunnskap er vi avhengige av å kunne studere elevenes arbeid nøye og å kunne gå i dybden. Dermed ble det naturlig for oss å velge kvalitativ metode, ettersom en slik metode tar utgangspunkt i et mindre utvalg og egner seg godt til å få dyp innsikt i en konkret tematikk (Postholm & Jacobsen, 2018, s. 165). Ved å bruke et mindre utvalg vil ikke resultatene kunne generaliseres, men det kan være anvendbart i kontekster som likner studien (Postholm & Jacobsen, 2018, s. 141). Hensikten er altså å kunne bruke deler av kunnskapen vi skaper, og overføre dette inn i andre liknende situasjoner.

Kvantitativ metode tar utgangspunkt i et større utvalg og er mer fokusert sammenlignet med kvalitativ metode. Fokuseringen og bruk av et større utvalg gjør det mulig å danne et representativt bilde av flertallet, og foreta en generalisering (Postholm & Jacobsen, 2018, s. 165). Denne metoden vil ikke gi oss like gode muligheter til å gå i dybden, sammenlignet med kvalitativ metode (Postholm & Jacobsen, 2018, s. 165), og vil dermed ikke være hensiktsmessig å bruke i denne studien.

## 3.2 Casestudie

Det sentrale i casestudier er at en studerer en case der tid og sted er avgrenset (Postholm & Jacobsen, 2018, s. 63). I vår studie arbeider elevene i par, i inntil tre skoletimer. Studien tar sted i elevenes klasserom, som er et kjent miljø for elevene. Bakgrunnen for dette valget er at vi ønsker å skape et trygt læringsmiljø.

For at studien skal kunne karakteriseres som en casestudie må den unike konteksten være sentral. Konteksten i vår studie er sentral ettersom elevene arbeider med problemløsningsoppgaver, som har til hensikt å få frem deres algoritmiske tenkning. Samtidig arbeider elevene på en måte som de ikke har gjort mye av tidligere. Elevene har ikke i tidligere sammenhenger arbeidet med å prate sammen om oppgaver samtidig som de løser dem, og de har heller ikke i stor grad arbeidet med Scratch tidligere. Grunnen til at vår kontekst er sentral, er fordi vi mener at vi ikke hadde fått akkurat samme resultat dersom vi hadde gjennomført studien i en annen kontekst. For eksempel at elevene hadde arbeidet med andre type oppgaver, med blyant og ark eller dersom vi hadde gjennomført studien et annet sted enn i klasserommet.

I en casestudie kan en studere ett eller flere individ, en gruppe eller en skole (Postholm & Jacobsen, 2018, s. 63). Vi har valgt å studere enkeltindivider i par, slik at hvert par er én case. Det vil si at vi har en enkel casestudie. Vi studerer hvordan den algoritmiske tenkningen hos elevene i hvert par kommer til uttrykk i arbeid med problemløsningsoppgaver.

## 3.3 Utvalg

For å besvare forskningsspørsmålet vårt var vi avhengige av noen informanter som vi kunne observere og intervju. Vi ble fort enige om at informantene skulle være fra en skole der vi ikke kjente elevene fra før, dette var for å minske vår egen subjektivitet, slik at våre meninger eller opplevelser av elevene fra tidligere møter ikke skulle påvirke resultatene.

Vi hadde begge vært i praksis sammen på en barneskole tidligere, hvor vi kjente en av matematikklærerne, men ikke elevene. Vi tok kontakt med læreren og informerte om prosjektet. Læreren syntes dette virket som et interessant prosjekt som hun, og 6. klassen hun

underviste, gjerne ønsket å delta i. Vi fikk også godkjenning av rektor, på den aktuelle skolen, til å gjennomføre prosjektet.

Læreren fikk tilsendt et samtykkeskjema (vedlegg 8.1), som vi skrev ut ifra retningslinjene til Sikt. Læreren leverte ut skjemaet i klassen, for å få tillatelse fra foresatte. Til sammen samtykket tolv elever til å bli med i prosjektet, og alt det innebar. Ettersom vi kun skulle studere fire elever, ba vi læreren om å gjøre utvalget for oss, slik at vi fremdeles kunne være så objektive som mulig.

### 3.4 Pilot

I forkant av datainnsamlingen gjennomførte vi en pilot i en 9.klasse. Dette var i utgangspunktet for å se om oppgavene var for utfordrende, samt om de var forståelige. Det viste seg å være nødvendig å gjennomføre denne piloten, ettersom vi gjorde flere justeringer på både oppgavene og gjennomføringen i etterkant. Vi fant blant annet ut at vi burde skrive ned noen spørsmål på arket som gjorde at elevene pratet mer underveis, da vi i piloten avdekket at elevene ikke pratet med hverandre i stor grad.

Vi fant også ut at vi måtte ta oss tid til å legge inn "penn" funksjonen på iPadene i forkant da dette ikke var lagt inn automatisk. Vi diskuterte også om vi skulle sette språket til norsk, da det automatisk var satt til engelsk. Vi endte opp med å glemme dette på selve dagen vi gjennomførte, men vi gikk fort rundt til elevene og spurte om de heller ville ha det på norsk.

Den viktigste oppdagelsen vi gjorde under piloten var at vi ønsket at de skulle arbeide sammen i par, i stedet for en gruppe på tre. Under piloten var det ofte en tredjepart som meldte seg ut av opplegget, fordi det var kun plass til to rundt en skjerm. Dette gjorde at ikke alle deltok i like stor grad. Vi endret også på noen av oppgaveformuleringene, da vi fant ut at noen spørsmål/oppgaver var litt ledende.

### 3.5 Gjennomføring

I dette kapitlet vil vi presentere en oversikt over hvordan gjennomføringen av undervisningsopplegget og intervjuene foregikk. Selve dagen var fra 08:30-14:00. Elevene

har “lesekvart” hver dag, altså at de bruker et kvarter på å lese en skjønnlitterær bok, uavhengig av hvilket fag de starter med. Dette valgte vi også å gjøre på datainnsamlingsdagen, for å opprettholde rutinene til elevene. Da elevene var ferdig å lese, tok vi dem med til et annet klasserom, der de skulle arbeide med oppgavene. Elevene arbeidet i intervaller på 20 minutter, ettersom det var så lenge Screen Capture kunne ta opp av gangen. Etter tre intervaller tok elevene friminutt. Etter friminutt gjennomførte vi to nye intervaller. Elevene arbeidet for det meste på egenhånd, men hadde mulighet til å stille oss spørsmål dersom de hadde behov for det. Under oppgaveløsningen vekslet vi mellom å gå rundt i klasserommet og å sitte i midten av rommet for å observere og notere ned aktuelle spørsmål til intervjuet. I etterkant av oppgaveløsningen satt vi oss sammen og diskuterte aktuelle spørsmål nærmere før vi hentet inn elevene til intervju. Dette var etter lunsj samme dag. Vi hentet inn ett par av gangen, og intervjuene varte mellom 20 og 30 minutter.

## 3.6 Datainnsamlingsmetode

Vi valgte å bruke en kombinasjon av intervju og observasjon fordi de på flere måter utfyller hverandre (Postholm & Jacobsen, 2018, s. 114). I kvalitativ forskning regnes ikke observasjon som en tilstrekkelig datainnsamlingsmetode når den står alene (Postholm & Jacobsen, 2018, s. 114). Observasjonene av elevenes oppgaveløsning ble i hovedsak gjort ved hjelp av skjerm- og lydopptak. Intervjuene ble brukt til å underbygge observasjonene slik at vi kunne gjengi en så nøyaktig beskrivelse av virkeligheten som mulig.

### 3.6.1 Observasjon

Ved hjelp av observasjon kan vi som forskere se og høre hva som skjer (Corbin & Strauss, 2015, s. 41). Vi ønsket å gjennomføre observasjon fordi elevene kan kommunisere at de gjør noe på en måte, og ha en opplevelse av det, men i realiteten gjør de noe annet. Ved å ta i bruk observasjon kan vi avdekke dette (Corbin & Strauss, 2015, s. 41). Ettersom vi tok skjerm- og lydopptak av elevenes oppgaveløsning, har vi også mulighet til å gå tilbake og reflektere over dette på en systematisk måte.

Innenfor kvalitativ forskning brukes observasjon til å fange opp det som skjer i en naturlig situasjon (Angrosino & Pérez, 2000, referert i Postholm & Jacobsen, 2018, s. 113). På denne



måten får forskeren mulighet til å få et innblikk i elevenes aktivitet og den fysiske situasjonen som observeres (Postholm & Jacobsen, 2018, s. 113). Vi ønsket ikke å skape en ekstraordinær situasjon for elevene. Til tross for at de arbeidet med nye typer oppgaver og på en annen måte, ønsket vi å holde miljøet og rutineene kjent for dem. Dette gjorde vi ved å beholde “lesekvarteren”, samt å følge den normale timeplanen, i form av friminutt og pause. Ved å opprettholde disse rutineene håper vi å styrke observasjonen vår ved å skape en mest mulig naturlig kontekst, som avgrenses i kjent tid og sted.

En fundamental forskjell i ulike observasjonsstrategier er i hvor stor grad forskeren deltar i situasjonen som observeres, og hvor mye forskeren samhandler med de som blir observert (Patton, 1990, s. 206). Vi har valgt rollen “observatør som deltaker”, hvor forskeren har mer fokus på observasjon kontra deltakelse (Johnson & Christensen, 2020, s. 200). Ettersom vi tok denne rollen førte det til at elevene i stor grad arbeidet alene og fikk muligheten til å utforske på egenhånd, uten at vi deltok nevneverdig. Elevene fikk med dette mulighet til å ta større ansvar for egen læring, og som nevnt i delkapittel 2.2.1, vil dermed den didaktiske kontrakten endres (Brousseau, 1997, sitert i Pierce & Stacey, 2010, s. 10). En fordel med denne rollen er at den begrenser subjektivitet og det er enklere for forskeren å holde seg nøytral (Johnson & Christensen, 2020, s. 200).

Observasjon er et arbeid som krever mye konsentrasjon, og det kan være vanskelig å få med seg alt (Postholm & Jacobsen, 2018, s. 131). For å sikre at vi fikk med oss all informasjon, valgte vi å ta i bruk verktøyene skjerm- og lydopptak. Dersom vår forskning skal kunne brukes i andre lignende kontekster er det gunstig å ha med seg helheten av både resultatene og konteksten. I tillegg ønsket vi å ha muligheten til å se oppgaveløsningene flere ganger, slik at vi kunne gå i dybden og gjøre detaljerte analyser av situasjonene (Postholm & Jacobsen, 2018, s. 131). Ifølge Postholm og Jacobsen (2018, s. 131) viser forskning at bruk av video kan påvirke elevene negativt, ved at det blir et forstyrrende element. Med bakgrunn i denne kunnskapen valgte vi å bruke skjerm- og lydopptak istedenfor video, slik at vi kunne se elevenes arbeid uten at elevene selv ble filmet.

### 3.6.2 Intervju

Vi har valgt å gjennomføre intervjuer i tillegg til observasjon fordi det, som nevnt tidligere, kan bidra til å utfylle en observasjon slik at det kan skapes kontekstuell informasjon mellom forsker og forskningsdeltaker (Postholm & Jacobsen, 2018, s. 115). Vi valgte å gjennomføre

intervjuene samme dag som observasjonen. Bakgrunnen for dette valget var at alle parter da vil ha friskt i minne hva som har foregått, noe som vi mener er en fordel i arbeidet med å skape kontekstuell informasjon.

I forkant av datainnsamlings-dagen utformet vi en rekke spørsmål som vi ønsket å stille under intervjuene (vedlegg 8.3), og disse ble stilt dersom det var naturlig. Som nevnt var også observasjonene med på å forme enkelte spørsmål og vi var også åpne for at elevene kunne bringe frem noe interessant som vi ikke hadde sett for oss på forhånd (Postholm & Jacobsen, 2018, s. 121). Spørsmålene som ble stilt var derfor en kombinasjon av planlagte og uplanlagte spørsmål, og vil si at vi gjennomførte et semi-strukturert intervju (Postholm & Jacobsen, 2018, s. 121).

### 3.7 Oppgavedesign

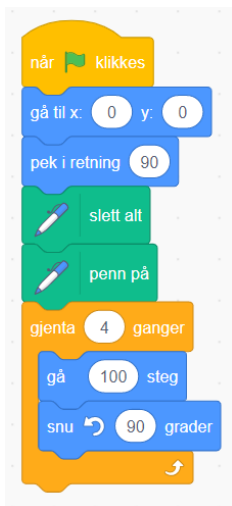
Designet av oppgavene er en av de viktige faktorene for å implementere digitale verktøy i undervisningen (Drijvers, 2013, s. 14-15). Vi har valgt å bruke problemløsningsoppgaver i et programmeringsverktøy fordi det henger tett sammen med algoritmisk tenkning. Da vi designet problemløsningsoppgavene var målet å legge til rette for at elevene skulle bruke algoritmisk tenkning.

Elevene fikk gjennom hele oppgaveheftet mulighet til å evaluere kodene sine, både ved hjelp av Sprite og ved å prate sammen. Sprite er katten i Scratch som utfører operasjonene som blir gitt ved kodene. Sprite viser om de har fått det til eller ikke, og elevene kan evaluere hva som har gått galt i koden. Oppgavene er utformet slik at vanskelighetsgraden øker. De første oppgavene handler om å bruke koder de får oppgitt, deretter skal de bruke disse kodene til å løse nye problemer, og til slutt får de mulighet til å lage egne koder. Vi vil nå presentere hver oppgave, og hva som var formålet med oppgaven.

#### Oppgave 1

Her er en kode.

- a) Klarer dere å se hvilken figur vi har laget, bare ved å se på koden?
- b) Hvordan ser dere, ut ifra koden, at det må være akkurat denne figuren?



I oppgave 1 valgte vi å gi elevene en ferdig kode for et kvadrat. Bakgrunnen for dette var at de skulle bli kjent med de ulike blokkene, da de ikke har arbeidet mye med programmet tidligere. Målet med oppgaven var at elevene skulle analysere hvert ledd i koden og videre finne ut hvilken figur koden representerer. Denne oppgaven legger opp til at elevene må organisere og analysere data, samt forstå koder som er skrevet av andre, og legger med det til rette for at elevene skal ta i bruk logikk (CSTA & ISTE, 2011, sitert i Román-González et al., 2017, s. 679; L’Heureux et al., 2012, sitert i Selby & Woollard, 2013, s. 3; Weintrop et al., 2016, s. 139-140). Vi forventet også at elevene skulle legge inn koden i Scratch for å teste om hypotesen deres stemte, som også er et aspekt innen logikk (L’Heureux et al., 2012, sitert i Selby & Woollard, 2013, s. 3; Weintrop et al., 2016, s. 140).

## Oppgave 2

Bruk koden fra oppgave 1, og prøv å endre denne koden slik at Sprite tegner et rektangel, der to og to sider er like lange.

**Før dere begynner på oppgaven, snakk sammen om hvordan dere vil løse den.**

I denne oppgaven skulle elevene arbeide videre med koden fra oppgave 1. De skulle endre denne koden for å programmere en ny figur. Dermed legger denne oppgaven til rette for logikk, ved at elevene skal forstå og modifisere koder som er skrevet av andre (Weintrop et al., 2016, s.139). Vi ønsket også at de skulle knytte forbindelser til oppgave 1, for å løse oppgaven, altså at de skulle bruke mønstre (Selby & Woollard, 2013, s. 4). I denne oppgaven skulle elevene løse et problem som har lik struktur som oppgave 1, men ulike detaljer. Dermed legger oppgaven også til rette for abstraksjon (Weintrop et al., 2016, s. 139).

### Oppgave 3

Vi har prøvd å få Sprite til å lage en trekant, men det er noe som ikke stemmer helt i koden vår. Kan dere hjelpe oss å gjøre slik at Sprite bare går akkurat én trekant?

- Gå igjennom koden sammen, og snakk sammen om de ulike blokkene og finn ut hva som ikke stemmer.
- Legg inn koden i Scratch og fiks problemet.



I oppgave 3 skulle elevene analysere koden de har fått og finne ut hva som ikke stemmer. Koden får Sprite til å tegne en trekant, men det er to elementer som er feil; “gå 100 steg” før løkken og “gjenta 4 ganger”. Målet med denne oppgaven var i utgangspunktet å få elevene til å fjerne unødvendige detaljer, som er et aspekt innen abstraksjon. I senere tid oppdaget vi at det å fjerne detaljer som ikke er korrekte, nødvendigvis ikke er en del av abstraksjon. Abstraksjon handler om å fjerne de rette detaljene slik at løsningen blir enklere, uten å miste noe som er viktig (Weintrop et al., 2016, s. 139). I denne oppgaven vil det dermed ikke være et tegn på abstraksjon å fjerne de to elementene som ikke er riktige. Målet vårt med denne oppgaven ble derfor endret fra å fjerne unødvendige detaljer, til å teste, sjekke og fikse koden, som er et aspekt innen logikk (Weintrop et al., 2016, s. 140).

## Oppgave 4

Dere skal nå være arkitekter og designe et hus. Lag først en skisse av huset dere vil designe. Når dere har designet huset skal dere lage en kode slik at Sprite tegner det.

- a) Snakk sammen om hvordan dere vil at huset skal se ut og tegn huset deres her:



- b) Lag koden i Scratch som tegner deres hus.

I oppgave 4 var målet at elevene skulle bruke en bredere anvendelse av sin tidligere kunnskap og løsninger, altså mønstre (CSTA & ISTE, 2011, sitert i Román-González et al., 2017, s. 679). I tillegg la vi til rette for at elevene kunne dele opp problemet inn i mindre figurer, og løse dem hver for seg. Altså at de kunne bruke dekomposisjon ved å se på hele huset som en sammensatt figur, som består av flere deler (Wing, 2006, s. 33; Weintrop et al., 2016, s. 139). Elevene har tidligere programmert både en trekant og et kvadrat, og kan bruke disse algoritmene til å løse et nytt problem, noe som er et aspekt innen dekomposisjon (Barr og Stephenson, 2011, s. 52). Dette er altså dersom de designet huset sitt som et “tradisjonelt” hus, som består av et kvadrat med en trekant på toppen.

## Oppgave 5

Her ser dere ulike mønstre og figurer. Velg en av dem og lag en kode slik at Sprite tegner den samme figuren. Får dere tid, gjerne lag flere, eller design deres eget mønster.

**Når dere har valgt dere en figur eller et mønster snakk sammen om hvordan dere skal lage koden på best mulig måte.**

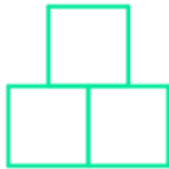
**Sikksakk**



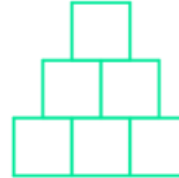
**Rutenett**



**Pyramide kvadrat 1**



**Pyramide kvadrat 2**



**Kvadrat på rad**



I oppgave 5 fikk elevene i oppgave å lage en kode slik at Sprite tegnet en av de fem figurene vi hadde gitt dem på arket. Alle figurene inneholder kvadrater eller trekanteder, som de har programmert i de tidligere oppgavene. Målet med denne oppgaven var å legge til rette for alle de seks nøkkelbegrepene innen algoritmisk tenkning. Logikk ved at de organiserer og analyserer data (CSTA & ISTE, 2011, sitert i Román-González et al., 2017, s. 679) og videre lager og tester hypoteser (L'Heureux et al., 2012, sitert i Selby & Woollard, 2013, s. 3). Algoritmer ved at de lager sine egne algoritmer (Brown, 2015, sitert i Korkmaz et al., 2017, s. 560). Dekomposisjon ved at en figur består av flere figurer. Dermed kan elevene dele opp det komplekse problemet ved å løse de ulike delene hver for seg (Weintrop et al., 2016, s. 139). Mønstre ved at de knytter forbindelser til tidligere oppgaver (Selby & Woollard, 2013, s. 4). Abstraksjon ved at de oppdager at figurene inneholder repetisjoner og dermed kan bruke løkker for å gjøre koden enklere å forstå (Barr & Stephenson, 2011, s. 52). Og til slutt evaluering ved at de vurderer om løsningen deres er riktig i forhold til bildet de tar utgangspunkt i, samt om koden er skrevet på enklest mulig vis (Wing, 2006, s. 33).

## 3.8 Analysestrategi

I dette kapitlet vil vi gjøre rede for hvordan datamaterialet er behandlet og hvordan vi analyserer materialet.

### 3.8.1 Transkripsjon

I etterkant av datainnsamlingen transkriberte vi både skjempptakene og intervjuene (se vedlegg 8.5 og 8.6). Dette arbeidet innebar å skrive ned hva hver elev sa, samt å notere hva elevene gjorde på skjermen under opptakene, dersom det var relevant. Vi ga elevene fiktive navn, slik at elevene ble anonymiserte underveis. All data som er samlet inn er blitt transkribert. I analysedelen vil det bli presentert utdrag fra transkripsjonene.

Transkripsjonen er gjort på følgende måte:

Operasjon	Forklaring	Eksempel
<i>Kursiv</i>	Det elevene sier, er i analysen skrevet i kursiv, men i transkripsjonen er det ikke det. Dette var for å tydeliggjøre hva som er utdrag.	Analysen: <i>Anna: Vi må få den til å lage et rektangel.</i>  Transkripsjon: Anna: Vi må få den til å lage et rektangel.
*...*	Det elevene gjør på skjermen.	*Legger til en «penn på» *.
( )	Det som vises på skjermen, er skrevet i en parentes.	(Sprite tegner en figur som minner om en trekant).
<i>Kursiv</i>	Det elevene gjør som ikke vises på skjermen er, i transkripsjonen, skrevet i kursiv.	<i>Leser oppgave 2 i kor.</i>

( )	Når elevene snakker om en spesifikk blokk, skriver vi hvilken blokk det er snakk om i parentes og med fet skrift.	Nei, vi må endre på den ( <b>blokken med snu</b> ) fordi den må gå mer innover.
(...)	Fraser som ikke er relevante i analysen og derfor utelates.	

### 3.8.2 Analyse

I etterkant av transkripsjonen studerte vi datamaterialet og kategoriserte elementene etter nøkkelbegrepene i algoritmisk tenkning. Dette gjorde vi ved å markere teksten med fargene som korresponderte med de ulike nøkkelbegrepene (se illustrasjon 4).

1:L	Logikk
2:AL	Algoritmer
3:D	Dekomposisjon
4:M	Mønstre
5:AB	Abstraksjon
5:E	Evaluering
7:U	Udefinert

*Illustrasjon 4: Nøkkelbegrepene med korresponderende fargekoder.*

I resultatdelen vil det bli lagt frem en tabell fra hver case som viser hvor mange ganger vi har oppfattet at elevene har brukt de ulike nøkkelbegrepene underveis i oppgaveløsningen. Disse tallene gjelder akkurat denne gruppen elever og er ikke absolutte tall. Tabellene viser tendenser, og det er ikke selve antallet som er poenget, men å få en oversikt over funnene hos akkurat disse elevene.



## 3.9 Kvalitetskriterier

I dette kapitlet vil vi ta for oss tre aspekter som er sentrale for studiens kvalitet. Vi vil først kommentere studiens gyldighet. Deretter vil vi kommentere studiens pålitelighet, med fokus på i hvilken grad datamaterialet og funnene som blir presentert er til å stole på. Til slutt vil vi kommentere etiske betraktninger vi har gjort i studien, med fokus på forskningsdeltakerne, undersøkelsen og forskeren selv.

### 3.9.1 Gyldighet

Forskingens gyldighet handler om hvilke begrensninger forskningen har, altså hva en har dekning for å si basert på data som er samlet inn (Postholm & Jacobsen, 2018, s. 222).

Gyldighet deles i to kategorier; ytre og indre gyldighet. Den ytre gyldigheten relateres til i hvor stor grad funnene er overførbare til andre kontekster (Postholm & Jacobsen, 2018, s. 223). Den indre gyldigheten relateres til om funnene er gyldige for de vi har studert (Postholm & Jacobsen, 2018, s. 223).

En utfordring med vår studie er at utvalget er lite og funnene kan derfor ikke generaliseres. Samtidig vil funnene være preget av elevenes forkunnskaper, oppgavetyper, relasjonen mellom informantene, og relasjonen mellom informantene og forskeren. Summen av dette fører til at den ytre gyldigheten i studien er lav. Til tross for dette ønsker vi at studien skal kunne brukes i andre kontekster og være av interesse for andre. Det er derfor en sentral del i vårt arbeid å beskrive hva som kjennetegner konteksten i studien, for å konkretisere hva som kan skille den fra andre, eller hva som er likt. Ved å reflektere over dette kan den ytre gyldigheten styrkes (Postholm & Jacobsen, 2018, s. 64-65).

Fordelen ved å bruke en kvalitativ casestudie er at vi får gode muligheter til å kunne si noe nøyaktig og detaljert om hvordan akkurat disse elevene arbeider med problemløsningsoppgaver. Dersom vi skulle gjennomført undersøkelsen igjen, med den samme elevgruppen eller andre elever fra samme klasse, kan vi være ganske sikre på at resultatene hadde blitt de samme. Dermed vil den indre gyldigheten i studien være stor (Postholm & Jacobsen, 2018, s. 64).

### 3.9.2 Pålitelighet

Forskningens pålitelighet handler om i hvor stor grad vi kan stole på funnene som blir presentert (Postholm & Jacobsen, 2018, s. 222). I dette inngår forskerens subjektivitet, altså at forskeren må reflektere over hvordan hen kan ha påvirket resultatene, samt forskningsprosessens synlighet, altså at forskeren er åpen om hvordan forskningen har foregått slik at andre kan reflektere over den (Postholm & Jacobsen, 2018, s. 224).

For at funnene vi presenterer skal være pålitelige er det en avgjørende del i vår studie å få inn tilstrekkelig data som danner et godt grunnlag for analysen. For å sikre dette var vi blant annet avhengige av at oppgavene var utformet slik at de inviterte til algoritmisk tenkning, samt at de var forståelige for elevene. Vi valgte derfor å gjennomføre en pilot, som beskrevet i kapittel 3.4, slik at vi kunne foreta justeringer og forbedringer knyttet til oppgavene, samt planen for gjennomføringen.

Observatørrollen vår vil være preget av hva vi ønsker å finne ut, og kan ha farget analysen vår. Skjerm- og lydopptakene er en stor del av observasjonen, og gir oss mulighet til å observere elevene gjentatte ganger slik at vi får fanget opp mest mulig. Dette er også med på å minske subjektiviteten i observasjonen, da datamaterialet ikke tar utgangspunkt i vår tolking av situasjonen, men viser nøyaktig hva som har skjedd og hva elevene og vi har kommunisert underveis. Under intervjuet stilte vi elevene åpne spørsmål, der målet var å få dem til å reflektere over eget arbeid. Intervjuet ble brukt til å underbygge observasjonene våre, og med det styrke kvaliteten til datamaterialet.

Resultatene og funnene baseres på datamaterialet fra observasjonene og intervjuene. Datamaterialet er transkribert og er vedlagt i sin helhet, slik at leseren selv kan studere materialet. I analysekapittelet vil vi presentere utdrag fra datamaterialet og plassere dem i kontekst, slik at leseren får mulighet til å sette seg inn i hva som har skjedd i forkant og etterkant av utdraget. På denne måten kan leseren gjøre egne refleksjoner knyttet til situasjonen. Videre vil vi forklare disse funnene i lys av teori og plassere dem i de gitte kategoriene.

Vi valgte å definere kategoriene for algoritmisk tenkning på forhånd av datainnsamlingen. Dette valget medfører en risiko for at disse kategoriene påvirker funnene og fører til at andre

typer av algoritmisk tenkning ikke ble oppdaget. I tillegg er vi avhengige av at elevene kommuniserer eller viser algoritmisk tenkning for å kunne måle den. Dermed er det en mulighet for at det har foregått mer algoritmisk tenkning enn det vi har registrert. Valget med å definere kategoriene på forhånd ble gjort på bakgrunn av oppgavedesignet, slik at oppgavene inviterte til algoritmisk tenkning. Dersom vi skulle gjennomført prosjektet igjen ville vi sannsynligvis tatt de samme valgene. Dette er på bakgrunn av at begge casene tok utgangspunkt i samme teori og dermed er det lite rom for å tolke casene ulikt.

### 3.9.3 Ethiske betraktninger

I denne studien har vi tatt hensyn til flere etiske betraktninger, blant annet at det var frivillig å delta. Etersom elevene vi har studert er under 16 år og var det de foresatte som måtte gi samtykke til deltakelse (Personopplysningsloven, 2018, Europaparlaments- og rådsforordning (EU) 2016/679 av 27. april 2016, kapittel II prinsipper, artikkel 8). Vi innhentet samtykke ved hjelp av et samtykkeskjema (se vedlegg 8.1). For oss var det også viktig at elevene selv forsto at de deltok i et forskningsprosjekt og hva det innebar. For å sikre oss dette fikk vi klassens lærer til å kommunisere dette til elevene, da samtykkeskjemaene ble delt ut. Vi gjentok også denne informasjonen på starten av selve undersøkelsesdagen. Elevene hadde også til hvilken som helst tid mulighet for å trekke seg fra prosjektet.

For å opprettholde anonymiteten til elevene bruker vi pseudonymer, slik at det ikke vil være mulig å identifisere elevene i etterkant av undersøkelsen. I tillegg brukte vi universitetets iPader slik at aktiviteten ikke kunne spores tilbake på elevenes egne enheter. Når det kommer til lagring av personlig data bruker vi universitetets database, som krever en totrinnsverifisering for å få tilgang til informasjonen. Dette sikrer at informasjonen ikke havner på avveie. Datamaterialet blir anonymisert, og videre slettet ved prosjektslutt.



## 4.0 Resultat og analyse

I dette kapittelet vil vi presentere utdrag fra datamaterialet vi har samlet inn. Data fra både arbeid med oppgavene og intervjuene vil bli presentert. Av hensyn til oppgavens omfang har vi valgt å kun presentere noen få utdrag fra datamaterialet. Vi har også forklart noen av dialogene mellom elevene, i stedet for å sette inn utklipp. Dette er fordi vi mener det er mer hensiktsmessig å forklare enkelte dialoger, istedenfor å presentere dem i sin helhet.

Utdragene vil bli analysert og plassert innenfor nøkkelbegrepene for algoritmisk tenkning som vi har gjort rede for i kapittel 2.2.3. Vi deler analysen inn i de to casene, og presenterer oppgaveløsningen kronologisk. Vi har valgt å separere analysen av de to casene fordi vi mener dette gir en bedre oversikt over hver case, slik at leseren enklere kan følge progresjonen og den algoritmiske tenkningen hos elevene. Vi har valgt å studere to caser fordi det gir oss mulighet til å undersøke flere elevers algoritmiske tenkning enn om vi kun hadde hatt en. Dette vil videre gi oss et bedre grunnlag for å svare på forskningsspørsmålet: Hvordan kommer elevenes algoritmiske tenkning til uttrykk i arbeid med problemløsningsoppgaver i Scratch?

### 4.1 Case 1: Anna og Amanda

Elevene arbeider med rektangel (Opptak 1.1)

Anna og Amanda leser oppgave 1 i kor, og snakker seg deretter gjennom koden. Videre blir de enige om å legge inn koden i Scratch, og ser at Sprite tegner et kvadrat. Elevene diskuterer og forsikrer seg om at koden de har laget er riktig og stemmer overens med koden på arket. Her bruker de *evaluering* ved at de bedømmer kodens riktighet (Wing, 2006, s. 33) og forsikrer seg om at løsningen deres er god. Her ser vi derimot at elevene ikke har forstått oppgaven. I oppgaven skulle de prøve å finne ut hvilken figur koden representerer bare ved å se på den. Gjennom den lange dialogen mellom elevene ser vi at de ikke gjør dette. De snakker heller ikke om hvilken figur som vises på skjermen. De går videre og leser oppgave 2, hvor de skal programmere et rektangel.

*Anna: Vi må få den til å lage et rektangel.*

*Amanda: Ja, der to og to sider er like lange.*

*Anna: Ja, to og to like sider, ikke alle er like.*

Anna og Amanda snakker sammen om hva oppgaven krever. Vi ser at de har forstått at de skal lage et rektangel der to og to sider er like lange. Anna legger frem en hypotese om at ikke alle sidene skal være like lange, selv om det ikke står noe om det i oppgaven. Det vil si at hun, ut ifra den informasjonen som er gitt i oppgaven, bruker *logikk* til å konstatere at alle sidene i rektangelet ikke skal være like lange. I nøkkelbegrepet inngår det å lage hypoteser ut ifra deres egen kunnskap, og videre forstå konsepter ved hjelp av fakta som de har fått oppgitt (L'Heureux et al., 2012, sitert i Selby & Woollard, 2013, s. 3).

Videre i arbeid med oppgaven presenterer de ulike forslag, og Anna beslutter at de skal skrive ned hva de velger å gjøre først.

*Anna: Okei, men hvis vi skriver ned hva vi velger å gjøre først.*

Vi ser at elevene ikke gjennomfører dette. Likevel kan dette tyde på at Anna ønsker å lage en trinnvis prosedyre for hvordan de skal løse problemet. Hun viser med dette evnen til å tenke i form av *algoritmer* (Selby & Woollard, 2013, s. 3). Anna og Amanda tester ulike blokker for å prøve å løse oppgaven, men møter motgang. Vi velger å stoppe opp på et utdrag hvor de legger til flere blokker med "penn på", for at figuren skal bli større.

*Anna: Vi må bare ta enten en mer av de der.*

\*Legger til en «penn på» \*

*Anna: Se, nå ble den større.*

*Amanda: Ja, det er den blokken som gjør den større.*

*Anna: Hvis vi tar litt flere, så blir den større.*

*Amanda: Jeg tror ikke den blir noe større egentlig.*

*Anna: Den ble større enn når vi bare hadde en.*

\*Fjerner «penn på» slik at de bare har en av disse\*

*Anna: Nei, den ble ikke større.*

Her har Amanda og Anna en hypotese om at rektangelet blir større dersom de legger til flere blokker “penn på”. De tester hypotesen og evaluerer algoritmen de har brukt for å avkrefte dens riktighet og bruker med det *logikk* og *evaluering* (L’Heureux et al., 2012, sitert i Selby & Woollard, 2013, s.3; Wing, 2006, s. 33). Vi observerer, i skjerm- og lydopptakene, at Anna og Amanda bruker denne formen for logikk og evaluering gjentatte ganger i oppgaveløsningen. Anna og Amanda fullfører ikke oppgave 2 ved denne anledningen, og går videre på oppgave 3.

### Elevene arbeider med trekant og rektangel (Opptak 1.2)

I neste opptak har Amanda og Anna akkurat startet på oppgave 3, hvor de skal fikse koden som lager en trekant. De går ut ifra koden for kvadrat, istedenfor å bruke koden de har fått oppgitt. De følger altså ikke instruksjonene i oppgaveteksten. Anna og Amanda diskuterer litt frem og tilbake, og kommer frem til at trekanter har noe med spisse og stumpe vinkler å gjøre.

*Amanda: Det har noe med 90 grader. Pek i retning 90 eller noe sånn. Det har noe med den å gjøre.*

*Amanda: Den skal være stump hvis vi skal lage trekant.*

*Anna: Og da må gradene være mindre enn 90.*

*Amanda: Ja, men hva er gradene da?*

*\*Endrer «Pek i retning 90 til 120» \**

*Amanda: Det er stump.*

I utdraget ser vi at Amanda og Anna lager en forbindelse og identifiserer likheter mellom det de jobber med, og noe de har lært i matematikken tidligere, som vil si at de har brukt *mønstre* (Selby & Woollard, 2013, s. 4). Det kan virke som at når de har arbeidet med trekanter og vinkler så har de vært inne på spisse og stumpe vinkler. Amanda og Anna prøver å bruke dette videre til å løse oppgaven.

*Amanda: Nei, prøv spiss. Ja, 60.*

*Anna: Et rektangel er ikke 60.*

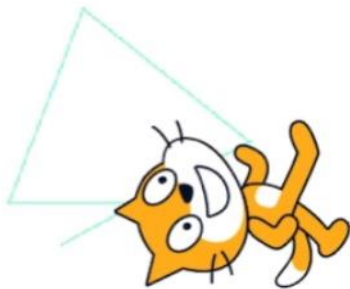
*Amanda: Nei, en trekant tenker jeg på nå.*

*Amanda: En trekant skal alltid være 180 grader.*

Amanda sier at vinkelen må være spiss, og 60 grader. Og forteller videre at en trekant skal alltid være 180 grader. Dette tolker vi som at de tidligere har lært at vinkelsummen i en trekant er 180 grader. Videre tolker vi det slik at hun bruker vinkelsummen, som tidligere erfaring, og konstaterer dermed at vinklene i trekanten må være 60 grader. Her går de fra en spesifikk anvendelse av vinkelsummer, til en bredere en, noe som er en del av nøkkelbegrepet *mønstre* (Selby & Woollard, 2013, s. 4). Ut ifra vår tolkning vet altså Anna og Amanda at vinkelsummen skal være 180 grader, og de kan videre finne ut, ved hjelp av denne kunnskapen, at vinklene inne i trekanten må være 60 grader. De er inne på noe her, men de justerer feil blokk. Dersom de hadde endret gradene Sprite snur, istedenfor hvilken vei den peker, hadde de kommet nærmere en løsning.

Videre i oppgave 3 endrer Anna løkken på koden som lager et kvadrat, slik at den gjentas tre ganger, i stedet for fire. Hun sier videre at de kun trenger å gjenta to ganger, for da mangler de kun hypotenusen, eller som hun sier: *den siste streken*. De finner ut at for å få dette til må de gjøre noe med gradene Sprite snur seg med.

De har nå endret løkken tilbake til å gjenta fire ganger. Anna og Amanda prøver å endre gradene Sprite skal snu, til 110 grader. Sprite tegner noe som minner om trekant (se illustrasjon 5) og Amanda konstaterer at de fikk en trekant og Anna sier seg enig.



Illustrasjon 5: Anna og Amandas forsøk på å programmere en trekant.

Etter en evaluering finner Amanda ut at, “*nei dette er ikke en ordentlig trekant*”. Anna påpeker videre at det er et rektangel Sprite prøver å lage, det tolker vi som at hun ser at figuren har fire sider og ikke tre. Elevene bedømmer kodens riktighet, og finner ut at det er noe som ikke stemmer, og bruker med det *evaluering* (Wing, 2006, s. 33). De har også, etter vår tolkning, fokusert på detaljer. Altså ikke bare hvordan figuren ser ut, men hvor mange



sider den har. Vi tolker det slik at elevene fant ut at figuren Sprite tegnet hadde fire sider og ikke tre, og de endret derfor løkken til gjenta tre ganger. Under intervjuet sier Anna “*Ofte hvis du skal lage firkant så må den gjenta 4 ganger. Men hvis du skal lage en trekant eller femkant så må du gjenta det flere ganger*”. Hun sier her at du må gjenta flere ganger dersom du skal lage en trekant istedenfor en firkant, vi ser derimot i oppgaveløsningen at de endrer løkken slik at den gjentas færre ganger. Med bakgrunn i dette tolker vi det slik at de har forstått konseptet med løkken, og at den representerer antall sider i figuren. Dette er en form for generalisering, som hører til både *mønstre* og *abstraksjon* (Selby & Woollard, 2013, s. 4; Weintrop et al., 2016, s. 139). De foretar en generalisering der antall gjentakelser blir antall sider i figuren.

Videre presenterer Anna en hypotese om at den ytre vinkelen må være større, altså at Sprite må snu mer innover. Vi tolker det slik at de bruker sine interne modeller på hvordan en trekant ser ut, og forstår dermed konseptet med trekantenes vinkler. Dette er et aspekt innenfor *logikk* (L’Heureux et al., 2012, sitert i Selby & Woollard, 2013, s. 3). Elevene endrer derfor på gradene slik at Sprite snur 120 grader, og får da en trekant.

*Anna: Hvis vi endrer på “gå steg”. Nei, vi må endre på den (blokken med snu) fordi den må gå mer innover.*

\*Endrer til snu 120 grader\*

*Amanda: Der, vi fikk trekant!*

Når de har løst oppgave 3, går Anna og Amanda tilbake til oppgave 2, for å gjøre et nytt forsøk på å programmere rektangelet. De endrer løkken slik at den gjentas fire ganger, fordi at de skal lage en firkant. De prøver å endre på både grader, og pek i retning. Videre diskuterer de å endre på antall steg, før de til slutt spør om hjelp. Bente går først gjennom oppgave 3 med dem. Deretter stiller hun dem spørsmål som hjelper dem å dele opp problemet i oppgave 2.

I dialog med Bente viser Anna og Amanda at de forstår at to av strekene i rektangelet må være lengre, men vet ikke hvordan de skal løse dette. Anna og Amanda kommer, etter litt tid og veiledning, frem til at det er antall steg Sprite går som avgjør lengdene på sidene i figuren. De går frem og tilbake med hvor mange ganger Sprite skal gjenta, hvor lange sidene er og

hvor mange grader han må snu. Opptak 1.3 starter her, og Anna og Amanda finner ut, med veiledning, at de må legge til én steg blokk og én snu blokk inne i løkka, og løser dermed problemet. De studerer figuren og Amanda sier “*Der, der er den!*”, noe som tyder på at hun bruker Sprite til å bekrefte kodens riktighet, som hører til nøkkelbegrepet *evaluering* (Wing, 2006, s. 33).

### Elevene programmer et hus (Opptak 1.3)

Anna og Amanda går i gang med oppgave 4, hvor de skal designe et hus. Vi ser at de planlegger og definerer klare steg for hvordan de skal løse oppgaven. De lager en slags trinnvis prosedyre, og bruker da *algoritmer* (Selby & Woollard, 2013, s. 3). De planlegger å sette huset sammen av figurene de har programmert tidligere.

*Anna: Okei, hvis vi lager en dør som er et rektangel.*

*Amanda: Ja, for det vet vi hvordan vi lager.*

*Anna: Så har vi et vindu som er en firkant. Det vet vi hvordan vi lager.*

*Amanda: Ja.*

*Anna: Vi kan også ha et vindu som er en trekant.*

Her kommer det frem at de bruker tidligere kunnskap, og ser likheter mellom figurene i huset og figurene de har programmert tidligere (Selby & Woollard, 2013, s. 4). De har anvendt programmering av både kvadrat og trekant på en spesifikk måte, og anvender nå denne kunnskapen på en bredere og mer sammensatt måte, de bruker altså nøkkelbegrepet *mønstre* (Selby & Woollard, 2013, s. 4). Videre ser vi at Anna og Amanda er usikre på hvordan de skal programmere huset, dersom de skal ha med vinduer og dører. De tegner en skisse av hvordan huset deres skal se ut (se illustrasjon 6). Under intervjuet kommer det frem at de brukte denne skissen aktivt under oppgaveløsningen. Anna sier “*vi tegnet liksom hvor vi måtte få den der figuren til å gå for å tegne den*”. Her viser elevene at de bruker tegning for å organisere problemet, og viser med det tegn på bruk av *logikk* (CSTA & ISTE, 2011, sitert i Román-González et al., 2017, s. 679).

*Anna: Så vi kan lage en trekant oppå firkanten.*

*Og for å lage trekanten kan vi bare ta gjenta.*

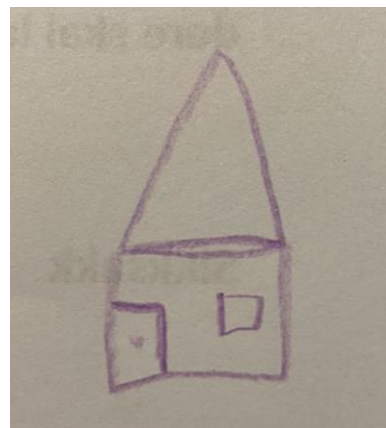
*Anna: Her kan døren være, og det er et rektangel.*

*Vi må liksom prøve å lage alt.*

*Anna: Skal vi prøve å lage det først.*

*Amanda: Ja, vi prøver det først.*

*Anna: Da må vi først lage firkanten.*



*Illustrasjon 6: Anna og Amandas skisse av huset.*

Skissen elevene har tegnet viser et komplekst problem. Dialogen viser at de velger å dele opp problemet slik at de lager de ulike figurene hver for seg. Videre velger de å lage firkanten først, for så å plassere trekanten på toppen av firkanten. De planlegger å løse de ulike delene av problemet hver for seg, som gjør at dette komplekse problemet blir lettere å løse, altså å *dekomponere* problemet (Wing, 2006, s. 33; Weintrop et al., 2016, s. 139).

Problemet de støter på videre i oppgaveløsningen er at de ikke klarer å plassere trekanten på toppen av kvadratet. De prøver mye forskjellig, og endrer på så å si alle blokkene. De evaluerer løsningene sine underveis, slik som i utdraget under.

*Amanda: Okei, vi prøver.*

*\*Tester – gir ikke riktig løsning\**

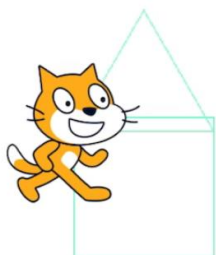
*Amanda: Nei.*

*Anna: Jo, det er nesten.*

Anna og Amanda tester koden og Sprite tegner noe annet enn det de prøvde å programmere. Her ser vi at Sprite er med å evaluerer for dem, ettersom de tar evalueringen videre, og sier at de ikke er langt unna en løsning, havner dette under *evaluering* (Selby & Woollard, 2013, s. 3). De evaluerer altså kodens riktighet (Wing, 2006, s. 33). Etter at de har arbeidet med oppgaven en stund, uten å komme frem til en løsning, velger de å spørre om hjelp. I slutten av opptaket får de hjelp av Bente til å dele opp problemet. Elevene prøver litt forskjellig, og blir ikke ferdige med oppgaven før pausen.

Elevene ferdigstiller huset og programmerer fem kvadrater på rad (Opptak 1.4)

Anna og Amanda forsøker fremdeles å programmere et hus, og får etter hvert, med veiledning, plassert trekanten på toppen av kvadratet, men taket skjærer toppen av kvadratet (se illustrasjon 7).



*Illustrasjon 7: Huset Anna og Amanda har programmert, der taket skjærer kvadratet.*

Amanda studerer huset og sier *“Der, da ble den litt sånn rar. Men det går fint, det kan bare være et ekstra sånn vindu det”*. Dette tyder på at hun *evaluerer* en tilsynelatende uriktig løsning, som en god løsning likevel. Vi tolker det slik at Anna og Amanda vurderer resultatet som riktig, da det ligner på et hus, men at de ikke har vurdert detaljene i løsningen sin (Weintrop et al., 2016, s. 139). Bente poengterer at antall skritt Sprite går før han tegner trekanten oppå huset, ikke stemmer overens med sidelengden i kvadratet. I intervjuet snakker Anna og Amanda også om dette og Anna sier at *“trekanten gikk litt ned i firkanten”*, men når de fikk justert dette lyktes de med oppgaven.

Da Anna og Amanda starter på oppgave 5 fikk de mulighet til å velge hvilken figur de ønsket å programmere. Anna og Amanda bestemmer seg for å programmere sikksakk. Vi tolker det slik at Amanda ser likheter mellom figuren de har programmert tidligere og figuren de nå skal programmere. De bruker den tidligere løsningen til å løse et nytt problem og bruker med det *mønstre* og *abstraksjon* for å løse oppgaven (Selby & Woollard, 2013, s. 4; CSTA & ISTE, 2011, sitert i Román-González et al., 2017, s. 679; Weintrop et al., 2016, s. 139).

*Amanda: Så det er litt som trekanten da. Bare uten streken under.*

*Anna: Så du skal ikke gjenta 3 ganger, du skal gjenta 2.*

*Amanda: Ja.*

*Amanda: Husker du i stad da vi lagde den firkanten, så tok vi to, så gikk den opp sånn. Det er litt det samme vi skal gjøre her bare at den skal gå opp sånn.*

Vi ser at de tenker å bruke samme algoritme som de brukte tidligere, men de må endre på løkken. Ettersom de implementerer en algoritme de har brukt tidligere, til en ny oppgave, kan vi si at de også bruker *algoritmer* i denne sekvensen (Barr & Stephenson, 2011, s. 52).

Anna og Amanda prøver seg litt frem med hvor mange steg Sprite skal gå, og i hvilken retning han skal snu. Anna kommer med et forslag om at de heller skal velge en annen figur, og forklarer hvordan de kan programmere den ved hjelp av *algoritmer*, altså en trinnvis prosedyre (Selby & Woollard, 2013, s. 3). Elevene fortsetter å arbeide med siksakk mønsteret, men møter motgang og finner ikke en løsning. Amanda foreslår at de skal prøve å programmere rutenettet heller og Anna blir med på det. De snakker sammen om figuren, men oppdager at det er enda en figur de kan lage; “Kvadrat på rad”, og velger å prøve seg på denne.

Anna og Amanda går tilbake til oppgave 1 og legger inn koden for kvadratet, her bruker de både *mønstre* og *algoritmer*, fordi de bruker en algoritme de har brukt tidligere til å løse en ny oppgave (Selby & Woollard, 2013, s. 4; Barr og Stephenson, 2011, s. 52). De knytter forbindelser til tidligere oppgaver, og bruker en eksisterende algoritme. De prøver litt forskjellig for å få til mellomrommet mellom kvadratene. I dette utdraget tolker vi det slik at Anna og Amanda bruker sine interne modeller til å løse problemet.

*Anna: Vi må få ham til å gå dit.*

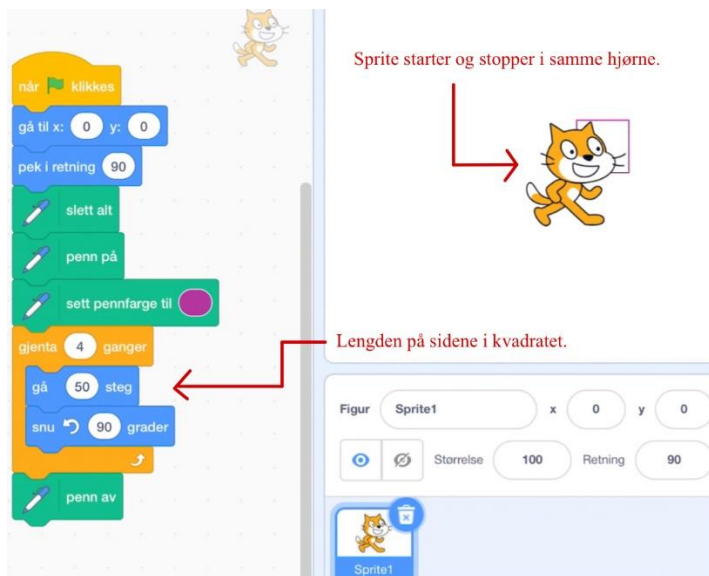
*Amanda: Ja, jeg skal det. Så tar vi gå 20 steg.*

*Amanda: Nei, vi må ta flere enn det. Vi må ta 60 steg. Fordi ruta er jo 50, ikke sant, så hvis vi skal komme litt unna sånn.*

*Anna: Ja, men den må gå lengre unna. Den må gå hundre.*

*Amanda: 100?*

*Anna: Ja, for da går den lengre bort. Så må vi lage firkanten på nytt.*



Illustrasjon 8: Anna og Amanda programmerer et kvadrat.



Illustrasjon 9: Figur fra oppgaveark, oppgave 5.

Elevene ser at det er en avstand mellom kvadratene på oppgavearket (se illustrasjon 9). Etersom Sprite stopper på samme plass han startet (se illustrasjon 8), må Sprite gå lengre enn lengden på kvadratet for å skape dette mellomrommet. Vi tolker det slik at hun bruker *logikk*, ved hjelp av sine interne modeller (L'Heureux et al., 2012, sitert i Selby & Woollard, 2013, s. 3), til å få Sprite til å gå flere steg enn han gjør på sidene av kvadratet.

Videre skal Amanda og Anna programmere flere kvadrater. De *dekomponerer* oppgaven ved å se på mønsteret som en figur sammensatt av flere deler. De velger å programmere et kvadrat og gjenta det flere ganger, slik at det blir lettere å løse problemet (Wing, 2006, s. 33; Weintrop et al., 2016, s. 139). De definerer de ulike delene som inngår i figuren, som også inngår i *dekomposisjon* (Barr & Stephenson, 2011, s. 52).

Elevene har programmert tre kvadrater på rad og spør oss om de må lage alle fem kvadratene. De får til svar at de skal prøve å lage alle. De diskuterer med Bente hvordan de har løst oppgaven, og elevene finner selv ut at det er en lettere måte å gjøre det på. De oppdager at de skriver det samme flere ganger og velger å legge til en løkke på hele koden. Elevene prøver å forenkle løsningen sin ved hjelp av løkken og bruker med dette *abstraksjon* (Barr & Stephenson, 2011, s. 52).

## Elevene endrer på løkken (Opptak 1.5)

Anna og Amanda har tatt i bruk løkken og er inne på riktig tankegang. De møter et problem fordi de legger til løkken på hele koden. Da tegner Sprite tre kvadrater før disse blir slettet og tegner tre nye kvadrater. Bente står allerede ved elevene når de skal prøve å programmere ferdig koden. Hun stiller elevene kun spørsmål om hva de ulike delene i koden betyr. Amanda forklarer at de lager en firkant en gang, deretter tar de av pennen og lager et mellomrom, før de tar på pennen igjen, og gjør det samme igjen. Elevene får deretter hjelp til å plassere løkken på riktig plass, samt at innholdet i løkken er korrekt. Selv om elevene får hjelp til dette, viser de at de har forstått at de kan forenkle koden. Dermed viser elevene at de bruker *abstraksjon*, fordi de forstår at koden gjentas flere ganger og de kan da legge til en løkke og fjerne flere av blokkene i koden. Her velger de å representere koden på en forenklet måte slik at den lettere kan forstås, som er et aspekt innen nøkkelbegrepet *abstraksjon* (Barr & Stephenson, 2011, s. 52). Her bruker Anna og Amanda også evaluering i form av å vurdere kodens enkelhet og effektivitet (Wing, 2006, s. 33). Dette bekrefter de også i intervjuet og poengterer at de hadde *“gjenta blokken på feil sted”*. Anna sier at *“når vi satt den inn på nytt et annet sted så kunne vi få laget fem med en gang”*. Her viser de at de har evaluert kodens riktighet og bruker med det også *evaluering* i intervjuet (Wing, 2006, s. 33). Anna og Amanda har løst et av mønstrene i oppgave 5, og avslutter med det arbeidet.

Tabellen under viser en oppsummering av hvor mange ganger vi har funnet tegn på at Anna og Amanda bruker de ulike nøkkelbegrepene i arbeidet med oppgavene. Vi ser at logikk og evaluering er nøkkelbegrepene som kommer til uttrykk i størst grad i denne casen. Vi ser også at abstraksjon og dekomposisjon ikke kommer like ofte til uttrykk.

Tabell 1: Antall forekomster av nøkkelbegrepene i case 1.

Kategori	Antall funn
Logikk	31
Algoritmer	11
Dekomposisjon	9
Mønstre	22
Abstraksjon	6
Evaluering	34

## 4.2 Case 2: Bendik og Beate

Elevene arbeider med å lage et rektangel (Opptak 2.1)

Bendik og Beate starter med å lese oppgave 1. De bestemmer seg for å legge inn koden de har fått og tester den. Sprite tegner et kvadrat, men Bendik sier *“den går bare i ring”*. Beate leser oppgaven på ny og de konkluderer med at de har løst oppgaven og går videre til neste. Under intervjuet spurte vi dem om hvordan de løste oppgaven, og da kom det frem at de ikke helt hadde forstått oppgaven. De trodde det var Sprite som var figuren de skulle studere, ikke figuren Sprite hadde tegnet. Bendik uttaler at de etter hvert forsto at det var *“hvilken form den lagde, ikke hvilken figur”*. De har altså brukt *evaluering* i etterkant og vurdert løsningen som uriktig (Wing, 2006, s. 33).

Elevene leser oppgave 2 hvor de skal programmere et rektangel. Bendik og Beate prøver seg frem ved å endre gradene Sprite skal snu. Etter en stund prøver de å endre på antall steg Sprite skal ta, og ser at kvadratet blir større, altså at alle sidene blir lengre. Bendik kommer med en hypotese om at de må legge til flere blokker, men Beate er uenig. Vi tolker det slik at Beate tror de kun skal endre på verdiene i de eksisterende blokkene og ikke kan legge til flere. De fortsetter å endre på gradene, men lykkes ikke. Videre endrer de på løkken, men vi



tolker det slik at de ikke har forstått konseptet med den og får derfor ikke riktig løsning. De prøver seg frem og tilbake en god stund og endrer på flere deler i koden, men tar aldri bort eller legger til blokker. Bendik sier “*Det er ikke mulig å lage et rektangel uten å legge på mer*”. Her viser Bendik evnen til å *evaluere* ved å analysere koden ut ifra løsningene de har kommet frem til (Selby & Woollard, 2013, s. 3). Han konstaterer med dette at en løsning er uoppnåelig med de forutsetningene de har nå, de er altså nødt til å legge til flere blokker. Elevene leser oppgaven på ny og Beate mener fremdeles at de kun må bruke koden fra oppgave 1. Bendik er enig i det, men mener samtidig at de kan legge til flere blokker. Beate og Bendik blir ikke enige, og velger å spørre om hjelp.

Elevene fullfører rektangelet og programmerer et hus (Opptak 2.2)

Rebekka spør elevene om hva som er forskjellen på et kvadrat og et rektangel. Beate forklarer forskjellen, og Rebekka spør om det går an å gjenta noe fire ganger, dersom to og to sider skal være like. Beate og Bendik konkluderer med at dette ikke går og fortsetter å diskutere hvordan de skal løse oppgaven.

*Bendik: Jeg vet det. Det er derfor denne må være to. Og så må vi bare ha en til av de her.*

*Beate: Kanskje vi må lage en til sånn. Hvis vi prøver det.*

\*Legger til en lik løkke. Får da et kvadrat. \*

*Bendik: Nei.*

*Beate: Nei, det funket ikke.*

\*Fjerner løkke nr. 2. \*

Her bruker de både *logikk* og *evaluering*. I de første to linjene ser vi at Beate og Bendik kommer med en hypotese og tester denne og bruker dermed logikk for å løse oppgaven (L’Heureux et al., 2012, sitert i Selby & Woollard, 2013, s.3; Weintrop et al., 2016, s. 139). I de to siste linjene bruker de Sprite til å *evaluere* løsningen sin og konkluderer med at hypotesen deres ikke fungerte. Her bruker de *evaluering* for å bedømme dens riktighet (Wing, 2006, s. 33). Videre presenterer Bendik en annen løsning på problemet.

*Bendik: Den er ganske unødvendig.*

*\*Drar bort løkken. \**

*Bendik: Vi trenger ikke den egentlig.*

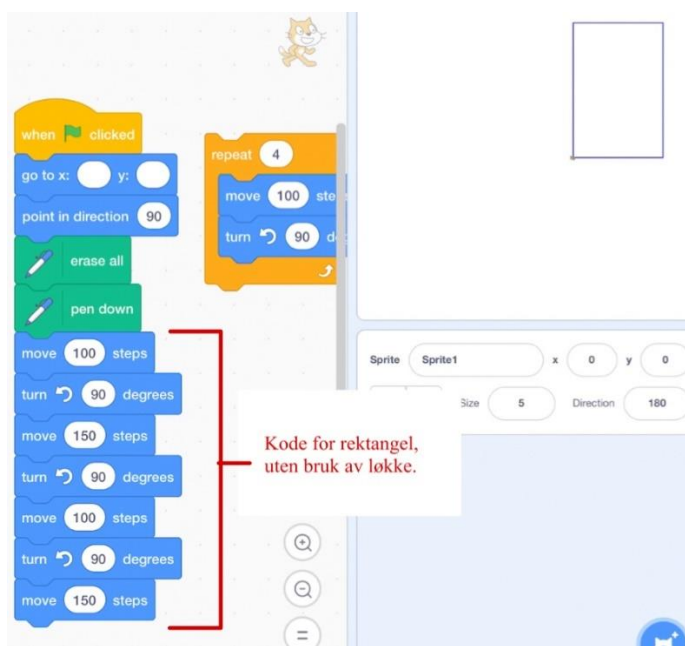
*\*Flytter på det som står over løkken. \**

*Bendik: Men jeg beholder den der. Men da trenger vi egentlig bare denne.*

(...)

*Bendik: Sånn.*

*Beate: Ja, nå fikk vi rektangel. Endelig.*



*Illustrasjon 10: Bendik og Beate programmerer et rektangel uten bruk av løkke.*

Bendik oppdager at de kan løse oppgaven uten å bruke løkken. Han velger å fjerne løkken, og legger inn en individuell blokk for hver handling Sprite gjør (se illustrasjon 10). Her tolker vi det slik at han har *dekomponert* problemet slik at det blir enklere å løse (Wing, 2006, s. 33; Weintrop et al., 2016, s. 139). Vi forstår det slik at Bendik ikke har forstått konseptet med løkken og velger derfor å fjerne den. Løkken er en detalj som ikke er nødvendig for å løse oppgaven, og ved å fjerne denne tolker vi det slik at Bendik bruker *abstraksjon* (Weintrop et al., 2016, s. 139). På denne måten klarer han å løse det komplekse problemet, uten å forstå

konseptet med løkken (Wing, 2006, s. 33). Samtidig viser dette at Bendik ikke klarer å representere koden på en mer generell form, som er et annet aspekt innen *abstraksjon* (Weintrop et al., 2016, s. 139), eller at han klarer å generalisere løsningsprosessen som er et aspekt innen *mønstre* (CSTA & ISTE, 2011, sitert i Román-González et al., 2017, s. 679). Ettersom Bendik i dette eksempelet ikke mestrer å generalisere løsningen, kan det bli utfordrende å bruke den videre. Bendik evaluerer løsningens riktighet, men han har ikke tatt i betraktning tidsbruken og det å kunne bruke løsningen sin videre som er aspekter innen *evaluering* (Weintrop et al., 2016, s. 139). Han viser dermed at han kun evaluerer resultatet, men ikke detaljene i løsningen (Weintrop et al., 2016, s. 139).

Etter at Beate og Bendik har løst oppgave 2, går de videre til oppgave 4 hvor de skal programmere et hus. Under arbeidet med oppgave 2 programmerte de en trekant, og mente dermed at de allerede hadde løst oppgave 3. Elevene leste altså ikke oppgaveteksten til oppgave 3. Dette så vi også da de begynte på oppgave 4. Elevene valgte å programmere huset uten å lage en skisse først. De har dermed ikke fulgt instruksene i oppgaven.

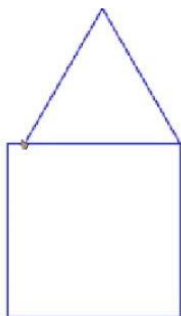
Bendik og Beate velger å lage et hus bestående av et kvadrat og en trekant. De legger inn koden for kvadrat som de har brukt tidligere.

*Beate: Ja, for vi har jo allerede en firkant, og så må vi bare ha en trekant oppå. Og da blir det et hus.*

Her ser vi at de løser et nytt problem ved å bruke tidligere erfaringer og knytter forbindelser til en tidligere oppgave. Samtidig løser de et nytt problem med lik struktur som et tidligere problem. Elevene bruker altså både *mønstre* og *abstraksjon* (Selby & Woollard, 2013, s. 4; Weintrop et al., 2016, s. 139). De bruker også *dekomposisjon* ettersom de deler huset inn i ulike figurer (Weintrop et al., 2016, s. 139), og definerer disse delene (Barr & Stephenson, 2011, s. 52). Vi tolker det slik at de velger å løse de ulike delene hver for seg, for å gjøre det komplekse problemet lettere å løse (Weintrop et al., 2015, s. 139; Wing, 2006, s. 33).

Videre prøver Bendik og Beate å programmere trekanten slik at de lager et tak, men taket treffer ikke det ene hjørnet i kvadratet (se illustrasjon 11). De evaluerer løsningen sin og finner ut at de må endre på gradene Sprite skal snu. De endrer litt og litt på gradene og taket

treffer til slutt hjørnet i kvadratet. Her bruker elevene *logikk* ved at de tester, fikser og systematisk sjekker koden (Weintrop et al., 2016, s. 140).



*Illustrasjon 11: Beate og Bendik programmerer et hus, men taket treffer ikke hjørnet.*

Da de har løst oppgaven leser de oppgaveteksten og tegner et hus på arket, før de går videre til oppgave 5. Bendik og Beate velger å lage sikksakk mønsteret og planlegger hvordan de skal løse dette.

*Beate: Da må vi ta sånn 1 grad og så 1 og så 1.*

*Bendik: 1 grad?*

*Beate: Ikke 1 grad, men liksom.*

*Bendik: 70 grader mener du.*

*Beate: Ja, 70 grader.*

*\*Legger til steg og snu. \**

(Koden viser gå, snu mot høyre, gå, snu mot venstre, gå, snu mot høyre, gå, snu mot venstre).

Her planlegger elevene en trinnvis prosedyre for hvordan de skal løse sikksakk mønsteret. De definerer de ulike stegene og bruker med dette *algoritmer* (Selby & Woollard, 2013, s. 3) Sprite beveger seg på skjermen, men elevene har glemt å legge til “penn på”, noe som resulterer i at Sprite ikke tegner mønsteret.

Elevene lager sikksakk mønster og rutenett (Opptak 2.3)

Bendik og Beate endrer gradene Sprite snur til 50 grader. Deretter oppdager de at de må legge til penn funksjonen og Sprite tegner et sikksakk mønster. I eksempelet under har Beate og Bendik en dialog om hvordan løsningen deres ble.

*Beate: Nå har vi sikksakk. Er ikke det sikksakk?*

*Bendik: Jo.*

*Beate: Det er ikke ordentlig sikksakk.*

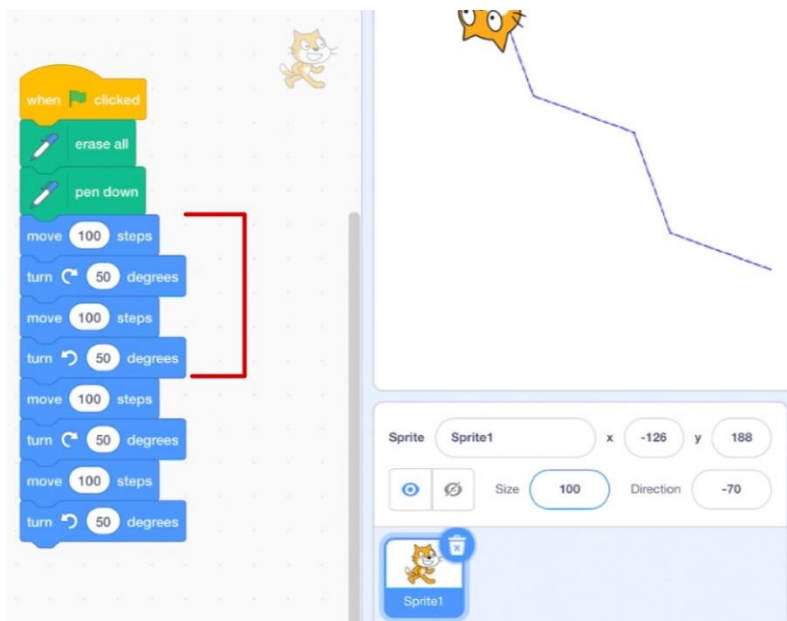
*Bendik: Jo, det er sikksakk, bare at vi ikke ser den helt.*

*Beate: Ja, men det er ikke helt ordentlig sikksakk.*

*\*Flytter på katten – deler av streken viser nå på skjermen\**

*Beate: Sånn, det er sikksakk. Vi ser det er sikksakk.*

*Bendik: Ja, det er sikksakk nå.*



Illustrasjon 12: Beate og Bendik programmerer sikksakk mønster.



Illustrasjon 13: Figur fra oppgavetekst, oppgave 5.

Her ser Beate og Bendik at de har klart å programmere et sikksakk mønster, men at ikke hele mønsteret viser på skjermen (se illustrasjon 12). De forsikrer seg om at løsningen sin er god, fordi de ser at Sprite tegner sikksakk, selv om de ikke får med hele figuren på skjermen. Grunnen til at de ikke ser figuren er fordi de ikke har satt inn en blokk som gjør at Sprite

starter på et bestemt sted. Her bruker de *evaluering* for å bedømme kodens riktighet (Wing, 2006, s. 33). Beate sier at det ikke er ordentlig sikksakk, noe som vi tolker at henger sammen med at mønsteret de har kommet frem til, ikke er likt mønsteret i oppgaven (se illustrasjon 13). Vi ser samtidig at de ikke har skrevet koden på enklest mulig vis (Wing, 2006, s. 33), og har derfor ikke brukt alle aspektene innenfor *evaluering*. Illustrasjon 12 viser at koden de har laget er lang. Ved å plassere de fire første blå blokkene inne i en løkke (markert med rødt) ville koden vært skrevet på enklere vis. På tross av dette vurderer de løsningen som god og går videre.

I oppgaveteksten i oppgave 5 står det at elevene skal velge en figur eller et mønster de ønsker å lage. Dersom de har mer tid kan de lage flere. Beate og Bendik har programmert et av mønstrene og velger deretter å programmere et rutenett. Utdraget under viser at Bendik og Beate diskuterer hvordan de skal løse oppgaven. Beate starter med å analysere figuren og oppdager at hun kan dele opp rutenettet i fire kvadrater, og bruker med det, *dekomposisjon* (Weintrop et al., 2016, s. 139).

*Beate: Vi må jo på en måte lage 4 firkanter.*

*Bendik: Her starter vi. Og vanligvis, når vi lagte firkanten så gikk vi bare sånn. Men nå må vi gå sånn, sånn, sånn, sånn.*

*Beate: Eller så kan vi gjøre sånn her, at vi starter her.*

*Bendik: Ja, det er jo det vi gjør.*

*Beate: Og så gjør vi sånn, sånn, sånn, og så sånn, sånn, sånn.*

*Beate: Hvis vi starter i det ene hjørnet, det har ikke så mye å si, men hvis vi starter der.*

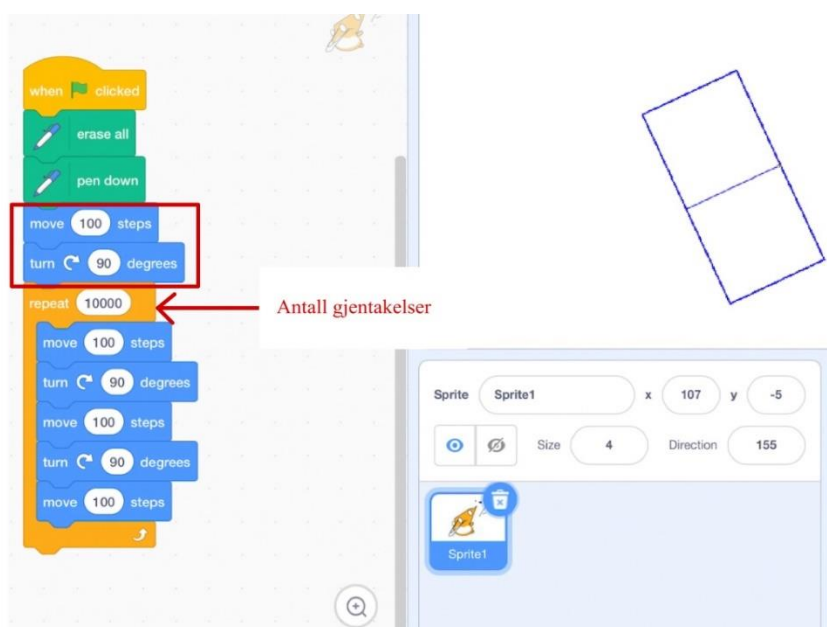
*Bendik: Jo, det har ganske mye å si.*

*Beate: Okei, vi starter der og så opp, til siden, ned, til siden, og så fortsetter vi til siden, opp der, bam bam bam. Sånn.*

Dialogen eleven har tyder på at de legger en plan for hvordan de skal løse problemet. Vi kan ikke se hva de viser til hverandre, men når vi ser på dialogen tolker vi det slik at de utformer en slags trinnvis prosedyre, og definerer de ulike stegene som kreves. På grunnlag av dette tolker vi det slik at elevene bruker *algoritmer* for å løse problemet (Selby & Woollard, 2013,

s. 3-4). Etter at Beate og Bendik har lagt en plan, legger de inn koden for kvadrat som de har programmert tidligere. Her viser de at de mestrer å anvende en tidligere løsning på et nytt problem, og bruker med det *mønstre* og *abstraksjon* (CSTA & ISTE, 2011, sitert i Román-González et al., 2017, s. 679; Weintrop et al., 2016, s. 139).

Beate og Bendik arbeider videre med rutenettet og prøver å bruke samme kode fire ganger for å få til oppgaven. De ender opp med to kvadrater etter hverandre, og prøver i en god stund å endre på dette. De endrer på antall gjentakelser i løkken og hvor løkken skal stå.



Illustrasjon 14: Beate og Bendik arbeider med å programmere et rutenett.

Etter at de har gjort disse endringene oppdager Bendik at det er to blokker som mangler inne i løkken (se illustrasjon 14, markert med rødt). Han flytter disse inn og lykkes da med å programmere et rutenett. Illustrasjon 14 viser at løkken gjentas 10 000 ganger, men Bendik mener den kun trenger å gjentas fire ganger og endrer derfor på dette. Her viser Bendik tegn på *logikk* ved at han sjekker og fikser koden (Weintrop et al., 2016, s. 140). I dette eksempelet ser vi at Bendik bruker *evaluering* ved å bedømme kodens effektivitet og enkelhet (Wing, 2006, s. 33), fordi han får samme resultat ved å gjenta fire ganger, altså forenkler han koden. Vi tolker det også slik at han viser tegn på *abstraksjon* ved å fjerne de gjentakelsene som ikke er nødvendige, og gjør med det koden enklere å forstå (Weintrop et al., 2016, s. 139; Wing, 2006, s. 33).

Etter at de er ferdige med rutenettet bestemmer de seg for å prøve på “pyramide kvadrat 1”. Her ser vi at Beate bruker interne modeller for å løse problemet:

*Beate: Så vi må bare gjøre akkurat det samme som vi har gjort alle andre gangene egentlig.*

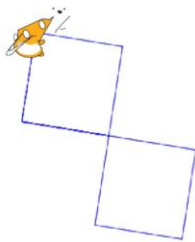
\*Endrer til gjenta 4\*

*Beate: Må det ikke være 3, for det er bare 3 sånne. Se, vi må ha 3. Det står «repeat» 4 fordi der måtte vi ha 4 ruter. Men her er det 3 ruter.*

Her har Beate og Bendik gått fra å arbeide med rutenett, som har fire kvadrater, til pyramide kvadrat 1, som består av tre kvadrater. Elevene legger til koden for kvadrat og Amanda foreslår at løkken skal gjentas tre ganger. Vi tolker det slik at Beate ser for seg rutenettet, og ser at det er ett mindre kvadrat i den figuren de nå skal lage. Derfor lager hun seg en hypotese om at løkken skal gjentas tre ganger, i stedet for fire. Her bruker hun *logikk* i arbeid med oppgaven, med fokus på sine interne modeller (L’Heureux et al., 2012, sitert i Selby & Woollard, 2013, s. 3). Vi tolker det også slik at hun forstår at gjenta funksjonen henger sammen med hvor mange figurer de skal programmere. Her kan det tolkes slik at hun bruker *mønstre*, altså at hun gjør en slags generalisering (Selby & Woollard, 2013, s. 4). Til tross for at Beate har en hypotese, tester de ikke denne og velger å la løkken gjentas fire ganger.

Elevene prøver å lage pyramide (Opptak 2.4)

Videre legger elevene inn to løkker for kvadrat og får to kvadrater som møtes i hjørnene (se illustrasjon 15). Elevene studerer figuren Sprite har laget og diskuterer løsningen.



*Illustrasjon 15: Beate og Bendik forsøker å programmere en pyramide.*



Elevene fortsetter med pyramiden. De legger til enda en løkke for kvadrat og får da to kvadrater oppå hverandre og en på siden, altså formet som en "L". Videre legger de til flere løkker for kvadrat slik at de har fem løkker. Sprite tegner da et rutenett. Det kan virke som elevene har gått bort fra å lage en pyramide og bestemmer seg nå for å lage et rutenett bestående av 3 x 3 kvadrater.

Mot slutten av oppgaveløsningen prøver de forskjellige lengder på sidene, og hver gang de tester koden blir figuren annerledes. De prøver frem og tilbake, men ettersom koden ikke stemmer helt, kommer de aldri i mål. Utdraget under er hentet fra intervjuet og her trekker Beate frem at de startet på en avansert måte, som førte til at de ikke lyktes med oppgaven.

*Beate: Vi startet på en veldig avansert måte, vi startet med liksom firkant, firkant, firkant, ja det ble helt sånn feil. Så vi måtte slette noen ting og så ble det helt feil. Så vi startet på en veldig vanskelig måte for å finne ut hvordan vi skulle gjøre det.*

Her bruker Beate *evaluering*, fordi hun bedømmer kodens riktighet, og spesielt enkelhet (Wing, 2006, s. 33). De får til slutt to kvadrater på siden av hverandre og bestemmer seg for å lage pyramiden igjen. Elevene spør om hjelp og Rebekka snakker med dem. I starten på opptak 2.5 forstår derimot Rebekka at Beate er veldig sliten, og hun uttrykker at hun kunne tenke seg å gi seg. Bendik ønsker å fortsette, men ut av hensyn til Beate sier Rebekka at Bendik kan arbeide alene i klasserommet, og avslutter opptaket.

Tabellen under viser en oppsummering av hvor mange ganger vi har funnet tegn på at Bendik og Beate bruker de ulike nøkkelbegrepene i arbeidet med oppgavene. I denne casen er mønstre det nøkkelbegrepet som har kommet mest til uttrykk. Vi ser at algoritmer og dekomposisjon ikke kommer like mye til uttrykk her som de andre nøkkelbegrepene.

Tabell 2: Antall forekomster av nøkkelbegrepene i case 2.

<b>Kategori</b>	<b>Antall funn</b>
Logikk	7
Algoritmer	5
Dekomposisjon	5
Mønstre	11
Abstraksjon	7
Evaluering	9

## 5.0 Diskusjon

I forrige kapittel presenterte vi resultatene og analysene fra datainnsamlingen. I dette kapitlet vil vi diskutere resultatene, knyttet opp mot teori og tidligere forskning i et forsøk på å besvare forskningsspørsmålet: Hvordan kommer elevenes algoritmiske tenkning til uttrykk i arbeid med problemløsningsoppgaver i Scratch?

Funnene viser at elevene tok i bruk samtlige av nøkkelbegrepene vi hadde utformet på forhånd, og at disse kom til uttrykk på forskjellige måter. Samtidig har det, som nevnt i kapittel 3.8, oppstått algoritmisk tenkning som vi ikke har registrert. Vi har valgt å plassere funnene fra resultatkapitlet i tabeller. Vi mener at tabellene gir en oversikt over hvordan den algoritmiske tenkningen kommer frem hos elevene. Oppgavene elevene arbeidet med er problemløsningsoppgaver, og som nevnt i kapittel 2.3, går problemløsningsoppgaver og algoritmisk tenkning hånd i hånd, og er en viktig del av det første kjerneelementet i læreplanen (Kunnskapsdepartementet, 2019, s. 2). Ifølge Pólya (1962, s. 117) er hovedpoenget i problemløsningsoppgaver at det ikke finnes en umiddelbar løsning og at du må lete bevisst etter en handling for å oppnå et mål som er klart utformet. Vi ønsker å trekke linjer mellom problemløsningsoppgaver og nøkkelbegrepene i algoritmisk tenkning, og hvordan disse nøkkelbegrepene kan komme til uttrykk når elevene løser problemer.

Tabellene og funnene som er presentert viser at begge gruppene bruker logikk gjennom hele arbeidet. Et aspekt innen logikk er å lage og bekrefte hypoteser ved hjelp av egen kunnskap og interne modeller (L'Heureux et al., 2012, sitert i Selby & Woollard, 2013, s. 3).

Resultatene viser at begge gruppene brukte logikk ved å legge frem hypoteser for å løse oppgavene. Når elevene i denne studien legger frem hypoteser sier de noe om hva de “tror” kommer til å skje eller hva de “tror” oppgaven betyr. Selv om elevene ikke bruker ordet hypotese, eller legger mye vekt på det i oppgaveløsningen, vil det å ha en antakelse om hva som kommer til å skje, anses som en hypotese. I problemløsningsoppgaver er det alltid noe som ikke er kjent (Pólya, 1962, s. 3), og for at elevene skal kunne løse oppgaven må de komme med et forslag til en løsning, som i disse tilfellene blir en hypotese.

Funnene viser at elevene bruker denne formen for logikk gjentatte ganger i oppgaveløsningen. I arbeid med oppgave 2, har Beate og Bendik en hypotese om at de trenger to løkker for å lage et rektangel. De tester hypotesen og avkrefter den. Videre lager de

nye hypoteser og tester dem og løser til slutt oppgaven. I arbeid med oppgave 3, der elevene skulle programmere en trekant, tolker vi det slik at Anna og Amanda bruker sine interne modeller og lager videre en hypotese der de må øke den ytre vinkelen. De tester hypotesen og bekrefter dens riktighet og løser oppgaven. Til tross for at vi ikke har muligheten til å si med sikkerhet hvordan de interne modellene ser ut, kan vi gjennom tolkning si noe om hvordan vi tror de har brukt disse interne modellene.

I arbeid med oppgave 4, har Beate og Bendik problemer med å programmere taket til huset. De velger å endre litt og litt på gradene og løser til slutt problemet. I dette eksempelet ser vi at elevene ikke nødvendigvis finner ut hvorfor koden ikke stemmer. Samtidig ser vi at de tester koden ved å systematisk endre litt og litt på gradene, og dette er dermed et eksempel på hvordan elevene bruker logikk i form av å sjekke og å teste koden systematisk (Weintrop et al., 2016, s. 140). Ettersom Scratch er et dynamisk program, får elevene mulighet til å utforske variasjon og regelmessighet i figurene og komme med antakelser. I disse oppgavene fikk elevene mulighet til å utforske detaljene i kodene, og med dette, studere de ulike blokkenes funksjon for de ulike figurene. Dette gjorde at de kunne bruke sin logikk og systematisk teste kodene (Weintrop et al., 2016, s. 140) på en effektiv og mindre tidkrevende måte. På denne måten er teknologien med på å endre oppgaven og dermed blir oppgavedimensjonen i det pedagogiske kartet (Pierce & Stacey, 2010) endret.

Et annet aspekt innen logikk er å forstå og modifisere koder som er skrevet av andre (Weintrop et al., 2016, s.139). I oppgave 1 og oppgave 3 la vi til rette for at elevene skulle forstå og modifisere koder vi hadde laget. I analysen trakk vi frem at ingen av gruppene løste disse oppgavene slik de var tenkt. På den ene siden kan en derfor si at elevene ikke mestrer dette aspektet ettersom de ikke viser at de har forstått koden eller klarer å modifisere den. På den andre siden kan en si at oppgaven ikke var presis nok og at elevene ikke oppfattet hva de skulle gjøre. Vi må også ta i betraktning at elevene har i liten grad arbeidet med programmering tidligere. Samtidig la vi også til rette for dette aspektet i oppgave 2, hvor elevene skulle bruke koden fra oppgave 1 til å lage et rektangel. Funnene viser at Bendik og Beate prøvde å modifisere koden vi gav dem, men da de ikke lyktes valgte de å programmere et rektangel uten bruk av løkke. Dette tyder på at de ikke har forstått koden vi gav dem og kan derfor tolkes som at de ikke mestrer dette aspektet. Anna og Amanda klarer derimot å bruke koden fra oppgave 1 til å lage et rektangel. Som nevnt hadde begge gruppene behov for veiledning for å løse oppgaven, noe som tyder på at elevene ikke helt mestrer dette aspektet.

Ettersom begge gruppene hadde behov for veiledning på oppgave 2, er det naturlig å reflektere over vanskelighetsgraden på oppgave 1 og 2. Pólya (1962, s. 3-4) legger vekt på at noe data må være kjent for elevene, og det må være en kobling mellom problemet og dataen. Vår refleksjon i etterkant er at koblingen mellom dataen de har fått, og oppgaven de skal løse, ikke er tydelig nok for elevene på dette tidspunktet. En løsning kunne ha vært å unngå å bruke løkke i oppgave 1, slik at de lettere kunne modifisert koden.

Elevene har gjentatte ganger tatt i bruk algoritmer for å løse problemene. Dette kan knyttes til problemløsning ved at de bevisst leter etter en handling for å oppnå en løsning (Pólya, 1962, s. 117). Dette så vi blant annet da Bendik og Beate arbeidet med oppgave 5. De utformet en trinnvis prosedyre for å programmere rutenettet, der de forklarer hvordan Sprite må tegne for å lage hver firkant. Dette viser hvordan elevene bruker et av aspektene innen algoritmer, altså det å lage en trinnvis prosedyre for å utføre oppgaver (Selby & Woollard, 2013, s. 3). I arbeid med å programmere huset viser også Anna og Amanda evnen til å utforme en trinnvis prosedyre for å løse problemet. De velger først å programmere et kvadrat og deretter en trekant som tak. I denne oppgaven velger begge gruppene å bruke kodene for kvadrat og trekant og viser dermed at de mestrer å implementere en algoritme for å løse et problem, som er et annet aspekt innen algoritmer (Barr og Stephenson, 2011, s. 52). Anna og Amanda gjør også dette da de arbeider med sikksakk mønsteret i oppgave 5. De velger å implementere algoritmen for trekant og endre løkken til å gjenta to ganger. Et tredje aspekt innen algoritmer er å presentere løsningen i form av sekvenser (Korkmaz et al., 2017, s. 561). I denne studien bruker elevene blokkprogrammering, og løsningen vil da bli vist i sekvenser, i form av blokkene i koden. Ettersom programmet gjør dette for dem, får ikke elevene mulighet til å vise at de mestrer dette.

Anna og Amanda brukte dekomposisjon i arbeid med oppgave 5, da de skulle programmere fem kvadrater på rad. De valgte å programmere kvadratene hver for seg slik at de enklere kunne løse problemet. Dette er et eksempel på hvordan elevene brukte dekomposisjon ved at de reformulerer et tilsynelatende komplekst problem til et problem de vet hvordan de løser (Wing, 2006, s. 33). Samtidig som de delte opp problemet, definerte de også de ulike delene i problemet og metodene som krevdes for å løse dem, noe som også er en del av dekomposisjon ifølge Barr & Stephenson (2011, s. 52). Vi så også at begge gruppene brukte dekomposisjon da de skulle programmere et hus. Gruppene valgte å dele problemet opp ved å

først programmere et kvadrat og deretter en trekant. Elevene viste med det at de evner å dele opp det komplekse problemet og samtidig definere de ulike delene og metodene som kreves.

Når elevene bruker dekomposisjon, får de mulighet til å bruke ferdighetene sine til å utvikle nye figurer. De bruker dermed ferdighetene sine til noe mer nyttig. Dette kommenteres i modellen “Didactical functions of technology in mathematics education”, som er utviklet av Drijvers (2013). Det er altså en forskjell på å øve på ferdigheter og å kunne bruke disse ferdighetene til å utvikle nye konsepter, altså lære noe nytt. Målet vårt var at elevene skulle kunne utvikle nye konsepter, og vi mener at vi har flyttet fokuset bort fra å øve på ferdigheter, til å bruke disse ferdighetene til å utvikle noe nytt. Vi plasserer derfor den didaktiske funksjonen av undervisningsopplegget vårt i “develop concepts (c)” i modellen. Med dette mener vi at vi designet oppgavene slik at de fikk mulighet til å utforske detaljene i figurene, og kunne videreføre disse til mer komplekse oppgaver, slik som elevene har gjort i oppgave 4. De sitter altså ikke å programmerer et kvadrat flere ganger for å kunne lære seg koden utenat, men arbeider med å videreutvikle kodene og bruker dem til noe nyttig.

Oppgavene er designet slik at de bygger på hverandre, og legger dermed til rette for at elevene gjennom hele arbeidet kunne identifisere likheter og forbindelser, som ifølge Selby & Woollard (2013, s. 4) er et aspekt innen mønstre. Dette så vi at elevene gjorde gjentatte ganger, blant annet når de skulle programmere et hus og brukte kodene for kvadrat og trekant. Elevene identifiserte altså likheter og forbindelser mellom oppgavene og brukte dermed tidligere løsninger til å løse nye problemer. Det å generalisere en løsning slik at den kan brukes for å løse en bredere variasjon av problemer, er også et aspekt innen mønstre (Selby & Woollard, 2013, s. 4; CSTA & ISTE, 2011, sitert i Román-González et al., 2017, s. 679). Dette ser vi at Anna og Amanda gjør da de skulle programmere siksakk mønsteret. De ser likheter mellom trekanten og mønsteret, og bruker den tidligere løsningen til å løse et nytt problem. Til tross for at begge gruppene mestrer å bruke mønstre og generalisering i oppgaveløsningen, ser vi at elevene kunne tatt i bruk dette ved flere anledninger enn de har gjort. Et eksempel på dette er da Beate og Bendik skulle programmere et rektangel. De valgte å fjerne løkken og fjernet dermed muligheten for å generalisere løsningen sin. Som følge av dette ville det blitt utfordrende å bruke samme kode videre i oppgavesettet. Samtidig som oppgavene bygger på hverandre og med det legger til rette for mønstre, kunne vi på den andre siden i større grad lagt til rette for at elevene kunne bruke erfaringer de hadde fra før

prosjektet. Dette var en utfordring for oss ettersom vi hadde lite kunnskap om hva elevene hadde av tidligere erfaringer.

I tabellen som tilhører case 1 ser vi at det var en markant forskjell i antall ganger nøkkelbegrepet abstraksjon kom til uttrykk, sammenlignet med flere av de andre nøkkelbegrepene. Dette åpner for muligheten til å drøfte om oppgavene la godt nok til rette for at elevene skulle ta i bruk abstraksjon under oppgaveløsningen. Vårt opprinnelige mål var å legge spesielt til rette for abstraksjon i oppgave 3. Etter en ny gjennomgang av nøkkelbegrepene oppdaget vi at målet med oppgave 3 måtte endres og ble mer rettet mot logikk, som nevnt i kapittel 3.7. Denne endringen ble foretatt etter datainnsamlingen, og kan dermed ha ført til at kun en oppgave, oppgave 5, la spesielt til rette for å bruke abstraksjon. Samtidig viser funnene at elevene har brukt abstraksjon under oppgaveløsningen.

Anna og Amanda brukte abstraksjon i arbeid med oppgave 5, da de skulle lage fem kvadrater på rad. De valgte å bruke løkke for å gjøre koden deres kortere og med det forenkle løsningen, som er et aspekt innen abstraksjon ifølge Barr og Stephenson (2011, s. 52). Vi så også at Anna og Amanda representerte en ide på en generell form, da de oppdaget at antall gjentakelser kan representere antall sider i figuren. De foretok altså en generalisering ved bruk av løkke. Bendik og Beate valgte derimot å fjerne løkken i arbeid med rektangelet, og fjerner dermed detaljer som ikke er nødvendige for å løse problemet, som ifølge Weintrop et al. (2016, s. 139) er et annet aspekt innen abstraksjon. Vi ser at elevene klarer å fjerne unødvendige detaljer, og å forenkle problemer. Vi har derimot ikke i stor grad sett at Bendik og Beate danner seg nye begreper eller representerer en ide på en generell form. Et annet aspekt innenfor abstraksjon er å løse flere problemer som har lik struktur, men ulike detaljer (Weintrop et al., 2016, s. 139). Dette ser vi når elevene bruker mønstre i oppgaveløsningen sin, da de knytter forbindelser til problemer som har like strukturer og kan derfor bruke mye av de samme strategiene (Selby & Woollard, 2013, s. 4).

Funnene som er presentert viser at evaluering ble i stor grad brukt i begge casene. Evaluering handler om å analysere løsningen sin (Selby & Woollard, 2013, s. 3), både ved å bedømme dens riktighet, effektivitet og enkelhet (Wing, 2006, s. 33). Evaluering innebærer også å vurdere tidsbruk, fleksibilitet, samt om løsningen kan brukes videre (Weintrop et al., 2016, s. 139). I arbeid med oppgave 2 valgte Bendik og Beate å programmere et rektangel uten bruk av løkke. De vurderte løsningen som riktig, da resultatet ble et rektangel. Dersom elevene

hadde brukt løkke ville løsningen vært mer effektiv og skrevet på enklere vis. Dette vitner om at elevene mestrer å bedømme kodens riktighet, men har problemer med å bedømme kodens effektivitet og enkelhet. Eksempelen viser også at elevene valgte en tidkrevende løsning ved at de la til flere blokker enn nødvendig, noe som tyder på at de ikke har evaluert løsningen sin i henhold til tidsbruken. Ettersom løsningen ikke var skrevet på enklest mulig vis, vil det være vanskeligere å bruke den videre. Koden de har laget inneholder mange blokker, noe som kan føre til at de mister oversikt dersom de skulle kombinere denne koden med andre. Det samme ser vi i oppgave 5 da elevene skulle lage sikksakk mønster. Da valgte de også å lage en individuell blokk for hver handling, altså uten bruk av løkke. Dermed viste de også her at de kun evaluerte kodens riktighet og ingen av de andre faktorene. Samtidig ser vi at da elevene skulle programmere et rutenett, valgte de å ta i bruk løkke. De satte først inn at den skulle gjentas 10 000 ganger, men endret dette da Bendik mente den kun trengte å gjentas fire ganger. I denne oppgaven viste elevene at de mestrer å evaluere løsningens riktighet, effektivitet og enkelhet. De har også fokusert på detaljene i løsningen ved at de studerer antall gjentakelser, noe som er et annet aspekt innen evaluering (Weintrop et al., 2016, s. 139).

Når det gjelder Anna og Amanda ser vi at også de mestrer å bedømme kodens riktighet. Da elevene skulle programmere et hus, så vi derimot at elevene evaluerte en tilsynelatende uriktig løsning som god. Taket på huset skar kvadratet, og figuren ble dermed ikke helt riktig. Dette eksempelet tyder på at elevene ikke har evaluert detaljene i løsningen og med det ikke mestrer dette aspektet innen evaluering. I arbeid med å programmere fem kvadrater på rad ser vi at Anna og Amanda mestrer å evaluere kodens enkelhet og effektivitet. Elevene programmerte tre kvadrater på rad uten bruk av løkke, og oppdaget at det var tidkrevende og lite effektivt. De valgte da å legge til en løkke slik at løsningen deres ble mer effektiv, samt at den ble skrevet på en enklere måte. Dette vitner om at Anna og Amanda mestrer å evaluere flere faktorer enn kun løsningens riktighet.

Evalueringen foregikk både ved at elevene evaluerte egne løsninger, og ved at de brukte Scratch til å evaluere løsningene. Elevene laget koder og Sprite tegnet for dem, og ved å studere tegningen Sprite har laget, kunne elevene fastslå om løsningen deres var korrekt. På denne måten brukte de programmet som et verktøy for å evaluere. Dette er med på å endre den didaktiske kontrakten, som nevnt i kapittel 2.1. I vår studie betyr det i praksis at det ikke lenger var læreren som var den eneste med faglig autoritet (Pierce & Stacey, 2010, s. 9). Dette mener vi førte til at elevene fikk ta større ansvar for egen læring, og at de ikke var



avhengig av oss for å sjekke løsningene sine (Brousseau, 1997, sitert i Pierce & Stacey, 2010, s. 9). Det er dette vi tar opp i resultatdelen, hvor vi konstaterer at Scratch evaluerer for dem til en viss grad. Et annet aspekt innenfor dette er at ettersom vi valgte å bruke Scratch, er det en risiko for at vi fjernet noe av elevenes muligheter for å evaluere løsningene sine selv, ettersom Sprite viser “svaret” til elevene, før de har mulighet til å diskutere hvorfor denne løsningen er god, eller ikke. Dersom vi trekker linjer til Drijvers (2013, s. 14) faktorer for en vellykket integrering av teknologi i klasserommet, ser vi at designet av verktøyene er sentralt. Scratch, og designet av det, har altså påvirket hvordan elevene fikk brukt sin algoritmiske tenkning. Det vil også derfor ha noe å si for hvordan det integreres i klasserommet.



## 6.0 Konklusjon og implikasjoner

Som vist i både analysen og diskusjonen kommer algoritmisk tenkning til uttrykk på ulike måter hos elevene vi har studert. Vi vil nå konkludere med noen av hovedfunnene i vår studie. *Logikk* og *evaluering* var de nøkkelbegrepene vi så mest av hos Anna og Amanda, og de fremtrådte også i stor grad hos Beate og Bendik. Den mest hyppige måten elevene i denne studien bruker og viser nøkkelbegrepet *logikk* er ved å utforme hypoteser, som er en del av begrepet ifølge L'Heureux et al. (2012, sitert i Selby & Woollard, 2013, s. 3). Dette kan kjennes igjen ved at elevene for eksempel sier: "Jeg tror at ..." eller "Hvis vi gjør det slik". Det å teste disse hypotesene er også en del av nøkkelbegrepet, og elevene i denne studien gjør dette kontinuerlig. Forekomsten av *evaluering*, som er presentert i tabellene, inkluderer både når de evaluerer på egen hånd og når de bruker Scratch til å evaluere. Elevene viser algoritmisk tenkning i form av *evaluering* ved å bedømme kodens riktighet, slik som Wing (2006, s. 33) trekker frem. Dette ser vi når elevene vurderer figuren som Sprite har tegnet for dem, altså om de "godtar" løsningen eller ikke. Elevene viste også under intervjuet at de evner å evaluere løsningene sine i etterkant, og viser med det at de mestrer å evaluere prosessen sin også. Elevene i denne studien prøvde også å gjøre løsningene lettere for seg selv. Vi så at elevene valgte å bruke løkke, slik at de kunne legge til færre blokker og med det ble løsningen forenklet, samt mer effektiv som begge er aspekter innen *evaluering* (Wing, 2006, s. 33).

Det å forenkles et problem er også et aspekt innenfor *abstraksjon*, ifølge Barr og Stephenson (2011, s. 52), og funnene viser at elevene bruker dette i noe grad. Det vi har sett av abstraksjoner hos elevene i denne studien er hovedsakelig bruk av løkker, og et konsekvent valg om å ikke bruke løkke. Beate og Bendik så på løkken som en unødvendig detalj og fjernet den, og på denne måten viste de evnen *abstraksjon*. Løkken ble også brukt av begge gruppene til å generalisere løsninger. Anna og Amanda generaliserte blant annet at antall sider i figuren samsvarte med antall gjentakelser i løkken. Ved generaliseringen viser de også evnen til å bruke *mønstre*, som samsvarer med beskrivelsen av nøkkelbegrepet (CSTA & ISTE, 2011, sitert i Román-González et al., 2017, s. 679). *Mønstre* viser seg også når elevene bruker tidligere erfaringer til å løse oppgavene. Dette så vi hos elevene når de gikk bakover i løsningene sine og fant en løsning de kunne bruke i oppgaven de arbeidet med.

I denne studien så vi at elevene brukte *algoritmer* på forskjellige måter. Noe som forekom hyppig var at elevene implementerte en algoritme for å løse en oppgave, og Barr og Stephenson (2011, s. 52) trekker frem at dette er et aspekt innen algoritmisk tenkning. De brukte kodene for kvadrat og trekant, for å løse de andre oppgavene. Elevene i denne studien laget flere ganger en trinnvis prosedyre for å løse oppgaven, som også er et aspekt innen algoritmer ifølge Selby og Woollard (2013, s. 3). Dette kom til uttrykk ved at de laget en slags plan for å løse oppgaven. Altså, “først må vi gjøre sånn, deretter må vi gjøre sånn”. I forhold til nøkkelbegrepet *dekomposisjon* er det spesielt en måte elevene i denne studien viser dette. Det er ved at de programmerer de ulike figurene i en sammensatt figur hver for seg, før de setter dem sammen. De deler altså opp et komplekst problem i deler de vet hvordan de kan løse (Weintrop et al., 2016, s. 139; Wing, 2006, s. 33).

Vi håper at resultatene fra vår studie kan gi lærere et overblikk på hvordan en kan kjenne igjen algoritmisk tenkning hos elevene, ettersom vi selv syntes det var utfordrende i forkant av prosjektet. Resultatene, sett i sammenheng med tidligere forskning, er ment for å kunne brukes i undervisningen, når en arbeider med å utvikle elevers ferdigheter i algoritmisk tenkning. Vi har erfart at elevene responderer positivt på å arbeide med problemløsningsoppgaver, med tanke på algoritmisk tenkning. Vi opplever også at bruk av teknologi har hatt en positiv effekt. På bakgrunn av våre erfaringer ønsker vi å bruke kunnskapen vi har fått fra denne studien inn i vår egen lærerhverdag. Vi har opplevd at dette er en fin måte å variere undervisningen, la elevene ta mer ansvar for egen læring og at temaet er mer fremtidsrettet. Vi vil derfor ta i bruk denne typen undervisningsopplegg videre, og bruke det i våre fremtidige klasser. Vi ønsker å bruke vår kompetanse fra denne studien til å bidra i utviklingen av et miljø for å kunne programmere, bruke nye undervisningsmetoder og flere læringsressurser, slik som Fessakis et al. (2013, s. 96) påpeker. Vi ønsker også å videre legge til rette for at teknologien kan støtte nye mål og undervisningsmetoder, slik at fagdimensjonen endres (Pierce & Stacey, 2010, s. 10).

En mulig implikasjon fra denne studien er å se videre på Lai og Yangs (2011) og Kalelioğlu og Gülbahars (2014) sin forskning, for å kunne bestemme hvorvidt Scratch er med på å forbedre problemløsningsferdighetene til elever på mellomtrinnet. Et annet aspekt som kan være interessant å undersøke er om elever blir bedre på algoritmisk tenkning ved å arbeide med problemløsningsoppgaver over en lengre periode. Altså om en kan trekke konklusjoner som sier at det er problemløsningsoppgaver som bidrar til å utvikle disse ferdighetene. Det vil

også være interessant å studere hvordan motivasjonen til elevene har betydning for deres arbeid. Vi har, gjennom denne studien, opplevd hendelser som kan indikere at motivasjonen har betydning, og skulle gjerne ha studert dette nærmere.



## 7.0 Litteraturliste

- Aho, A. V. (2012). Computation and Computational Thinking. *The Computer Journal*, 55(7), 832-835. <https://doi.org/10.1093/comjnl/bxs074>
- Barr, V. & Stephenson, C. (2011). Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community? *ACM Inroads*, 2(1), 48-54. <https://doi.org/10.1145/1929887.1929905>
- Corbin, J. & Strauss, A. (2015). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory* (4.utg.). Sage Publications, Inc.
- Drijvers, P. (2013). Digital technology in mathematics education: Why it works (or doesn't). *PNA*, 8(1), 1-20. <https://doi.org/10.30827/pna.v8i1.6120>
- Fessakis, G., Gouli, E. & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers and education*, 63, 87-97. <https://doi.org/10.1016/j.compedu.2012.11.016>
- García-Peñalvo, F. J. & Mendes, A. J. (2018). Exploring the computational thinking effects in pre-university education. *Computers in Human Behavior*, 80, 407-411. <https://doi.org/10.1016/j.chb.2017.12.005>
- Johnson R. B. & Christensen, L. (2020). *Educational Research: Quantitative, Qualitative, and Mixed Approaches* (7.utg.). Sage Publications, Inc.
- Kalelioğlu, F. & Gülbahar, Y. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in Education*, 13(1), 33–50. <https://doi.org/10.15388/infedu.2014.03>

- Korkmaz, O., Çakir, R. & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, 72, 558-569  
<http://dx.doi.org/10.1016/j.chb.2017.01.005>
- Kunnskapsdepartementet. (2017). *Overordnet del – verdier og prinsipper for grunnopplæringen*. Fastsatt som forskrift ved kongelig resolusjon. Læreplanverket for Kunnskapsløftet 2020. <https://www.regjeringen.no/no/dokumenter/verdier-og-prinsipper-for-grunnopplaringen/id2570003/>
- Kunnskapsdepartementet. (2019). *Læreplan i matematikk 1. - 10. trinn (MAT01-05)*. Fastsatt som forskrift. Læreplanverket for Kunnskapsløftet 2020.  
<https://www.udir.no/lk20/mat01-05>
- Lai, A.-F. & Yang, S. -M. (2011). The Learning Effect of Visualized Programming Learning on 6th Graders' Problem Solving and Logical Reasoning Abilities. *International Conference on Electrical and Control Engineering*, 6940–6944.  
<https://doi.org/10.1109/ICECENG.2011.6056908>
- Meld. St. 28 (2015-2016). *Fag – Fordypning – Forståelse: En fornyelse av Kunnskapsløftet*. Kunnskapsdepartementet. <https://www.regjeringen.no/no/dokumenter/meld.-st.-28-20152016/id2483955/>
- Nouri, J., Zhang, L., Mannila, L. & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, 11(1), 1-17. <https://doi.org/10.1080/20004508.2019.1627844>
- Patton, M. Q. (1990). *Qualitative Evaluation and Research Methods* (2. utg.). Sage Publications, Inc.
- Personopplysningsloven. (2018). *Lov om behandling av personopplysninger* (LOV-2018-06 15-38). Lovdata. <https://lovdata.no/lov/2018-06-15-38>



- Pierce, R. & Stacey, K. (2010). Mapping Pedagogical Opportunities Provided by Mathematics Analysis Software. *International Journal of Computers for Mathematical Learning*, 15(1), 1-20. <https://doi.org/10.1007/s10758-010-9158-6>
- Pólya, G. (1962). *Mathematical Discovery: On understanding, learning, and teaching problem solving: 1.* (Vol. 1). John Wiley & Sons, Inc.
- Postholm, M. B. & Jacobsen, D. I. (2018). *Forskningsmetode for masterstudenter i lærerutdanning*. Cappelen Damm Akademisk.
- Román-González, M., Pérez-González, J. -C. & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 72, 678-691. <https://doi.org/10.1016/j.chb.2016.08.047>
- Santos-Trigo, L. M. (2016). Digital Technologies and Mathematical Problem Solving. I P. Liljedahl, L. M. Santos-Trigo, U. Malaspina & R. Bruder. *Problem Solving in Mathematics Education* (s. 19-31). Springer Open. <https://doi.org/10.1007/978-3-319-40730-2>
- Selby, C. & Woollard, J. (2013). Computational thinking: the developing definition. *University of Southampton*. <http://eprints.soton.ac.uk/356481/>
- Utdanningsdirektoratet. (2019, 27. mars). *Algoritmisk tenkning*. <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L. & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 25, 127-147. <https://doi.org/10.1007/s10956-015-9581-5>

Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33-35.  
<https://doi.org/10.1145/1118178.1118215>

Wing, J. M. (2017). Computational thinking's influence on research and education for all.  
*Italian Journal of Educational Technology*, 25(2), 7-14.  
<https://doi.org/10.17471/2499-4324/922>

## 8.0 Vedlegg

### 8.1 Informasjonsskriv og samtykkeskjema

#### **Vil ditt barn delta i forskningsprosjektet «Utforskende undervisning i Scratch - en studie om algoritmisk tenkning og elevenes motivasjon»?**

Dette er et spørsmål til deg om å delta i et forskningsprosjekt hvor formålet er å forske på elevenes algoritmiske tenkning og motivasjon i arbeid med programmeringsverktøyet Scratch. I dette skrivet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

#### **Formål**

Vi ønsker i vår masteroppgave å forske på algoritmisk tenkning og motivasjon knyttet til et utforskende opplegg i Scratch. Vi vil undersøke hvordan et utforskende opplegg kan påvirke elevenes motivasjon, og hvordan algoritmisk tenkning kommer til uttrykk hos elevene.

#### **Hvem er ansvarlig for forskningsprosjektet?**

Vår veileder, universitetslektor Kjetil Damsgaard, ved Universitetet i Agder er ansvarlig for prosjektet.

#### **Hvorfor får du spørsmål om å delta?**

Vi har tidligere vært i praksis på denne skolen, i regi av Universitetet i Agder. Vi kontaktet derfor vår tidligere praksislærer, for å gjøre datainnsamling til vårt prosjekt. Lærer, og rektor, har fra skolens side gitt samtykke til å gjennomføre undersøkelsen i denne klassen. I tillegg ber vi her om ditt samtykke som foresatt for at ditt barn kan delta i undersøkelsen.

#### **Hva innebærer det for deg å delta?**

Masterstudentene Bente Bloch-Johnsen Lohne og Rebekka Helen Kolstad Frankrig utvikler i samarbeid med lærer, et undervisningsopplegg. Læreren gjennomfører undervisningen i klassen, mens Bente og Rebekka observerer og foretar datainnsamling til forskningsprosjektet sitt. Elevene skal jobbe i grupper med et utforskende opplegg i Scratch. Underveis i opplegget vil elevene jobbe på iPad som lånes ved Universitetet i Agder. Arbeidet vil foregå i inntil to skoletimer (2x 45 min). I arbeidet med oppgavene vil vi ta skjermpoptak på iPaden og lydopptak av elevene for å sikre data. Skjermpoptakene vil kun

vise det som skjer på skjermen. Det vil ikke bli tatt video av elevene kun opptak av lyd. I etterkant vil noen elever velges ut til et intervju. For å sikre data fra intervjuet vil vi ta notater og lydopptak. Vi vil velge ut inntil 10 elever til å delta i prosjektet.

Intervjuguide kan bli tilsendt på e-post ved å kontakte oss (bblohn18@uia.no).

### **Det er frivillig å delta**

Det er frivillig å delta i prosjektet, men alle elever følger normal undervisning. Skjermopptak og lydopptak vil kun gjøres i grupper som har gitt samtykke. Hvis du velger å la ditt barn delta, kan du når som helst trekke samtykket tilbake uten å oppgi noen grunn. Alle dine personopplysninger vil da bli slettet. Det vil ikke ha noen negative konsekvenser for ditt barn hvis du ikke tillater opptak/intervju eller senere velger å trekke tillatelsen. Elevene vil delta i undervisningsøkten sammen med klassen, men vil ikke tas opptak av eller trukket ut til intervju.

### **Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger**

Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket.

- Elevene vil ikke kunne bli gjenkjent i oppgaven og vil behandles anonymt. Navn vil bli erstattet med fiktive navn, som lagres adskilt fra øvrige data.
- Datainnsamling vil være anonymisert og beskyttet i form av adgangsbegrensning.
- Det vil kun være oss og vår veileder som har tilgang til datainnsamlingene.

### **Hva skjer med personopplysningene dine når forskningsprosjektet avsluttes?**

Prosjektet vil etter planen avsluttes når oppgaven blir godkjent, innen utgangen av 2023. Etter prosjektslutt vil alle orginaldata, lyd- og skjermopptak bli slettet.

### **Hva gir oss rett til å behandle personopplysninger om deg?**

Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag fra Universitet i Agder har Personverntjenester vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

### **Dine rettigheter**

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

- innsyn i hvilke opplysninger vi behandler om deg, og å få utlevert en kopi av opplysningene

- å få rettet opplysninger om deg som er feil eller misvisende
- å få slettet personopplysninger om deg
- å sende klage til Datatilsynet om behandlingen av dine personopplysninger

### **Hvor kan jeg finne ut mer?**

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

- Masterstudent: Bente Bloch-Johnsen Lohne      [bblohn18@uia.no](mailto:bblohn18@uia.no)
- Mesterstudent: Rebekka Helen Kolstad Frankrig      [rhfran18@uia.no](mailto:rhfran18@uia.no)
- Veileder: Kjetil Damsgaard      [kjetil.damsgaard@uia.no](mailto:kjetil.damsgaard@uia.no)
- Personvernombud UiA: Trond Hauso      [personvernombud@uia.no](mailto:personvernombud@uia.no)

Hvis du har spørsmål knyttet til Personverntjenester sin vurdering av prosjektet, kan du ta kontakt med:

- Personverntjenester på epost ([personverntjenester@sikt.no](mailto:personverntjenester@sikt.no)) eller på telefon: 53 21 15 00.

Med vennlig hilsen

*Kjetil Damsgaard*

(Veileder)

*Bente Bloch-Johnsen Lohne og  
Rebekka Helen Kolstad Frankrig*

(Masterstudenter)

## Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet «*Utforskende undervisning i Scratch - en studie om algoritmisk tenkning og elevenes motivasjon*», og har fått anledning til å stille spørsmål. Jeg samtykker til:

- at mitt barn deltar i forskningsprosjektet
- at mitt barn kan tas lydopptak av
- at mitt barns arbeid tas skjermopptak av
- at mitt barn kan velges ut til intervju

Jeg samtykker til at mine opplysninger behandles frem til prosjektet er avsluttet

-----  
(Signert av foresatt til prosjektdeltaker, dato)

## 8.2 Vurdering fra Sikts personverntjenester



[Meldeskjema / Utforskende undervisningsopplegg i Scratch. En studie av elevers algor...](#) / Vurdering

### Vurdering av behandling av personopplysninger

Referansenummer	Vurderingstype	Dato
584340	Standard	16.11.2022

#### Prosjekttittel

Utforskende undervisningsopplegg i Scratch. En studie av elevers algoritmiske tenkning og motivasjon på mellomtrinnet.

#### Behandlingsansvarlig institusjon

Universitetet i Agder / Fakultet for teknologi og realfag / Institutt for matematiske fag

#### Prosjektansvarlig

Kjetil Damsgaard

#### Student

Bente Bloch-Johnsen Lohne

#### Prosjektperiode

01.11.2022 - 01.11.2023

#### Kategorier personopplysninger

Alminnelige

#### Lovlig grunnlag

Samtykke (Personvernforordningen art. 6 nr. 1 bokstav a)

Behandlingen av personopplysningene er lovlig så fremt den gjennomføres som oppgitt i meldeskjemaet. Det lovlige grunnlaget gjelder til 01.11.2023.

[Meldeskjema](#) 

## **Kommentar**

### **OM VURDERINGEN**

Personverntjenester har en avtale med institusjonen du forsker eller studerer ved. Denne avtalen innebærer at vi skal gi deg råd slik at behandlingen av personopplysninger i prosjektet ditt er lovlig etter personvernregelverket.

Personverntjenester har nå vurdert den planlagte behandlingen av personopplysninger. Vår vurdering er at behandlingen er lovlig, hvis den gjennomføres slik den er beskrevet i meldeskjemaet med dialog og vedlegg.

### **VIKTIG INFORMASJON TIL DEG**

Du må lagre, sende og sikre dataene i tråd med retningslinjene til din institusjon. Dette betyr at du må bruke leverandører for spørreskjema, skylagring, videosamtale o.l. som institusjonen din har avtale med. Vi gir generelle råd rundt dette, men det er institusjonens egne retningslinjer for informasjonssikkerhet som gjelder.

### **TYPE OPPLYSNINGER OG VARIGHET**

Prosjektet vil behandle alminnelige personopplysninger frem til 01.11.2023.

### **LOVLIG GRUNNLAG**

Prosjektet vil innhente samtykke fra de foresatte til behandlingen av personopplysninger om barna. Vår vurdering er at prosjektet legger opp til et samtykke i samsvar med kravene i art. 4 og 7, ved at det er en frivillig, spesifikk, informert og utvetydig bekreftelse som kan dokumenteres, og som den registrerte/foresatte kan trekke tilbake.

Lovlig grunnlag for behandlingen vil dermed være de foresattes samtykke, jf. personvernforordningen art. 6 nr. 1 bokstav a.

### **PERSONVERNPRINSIPPER**

Vi vurderer at den planlagte behandlingen av personopplysninger vil følge prinsippene i personvernforordningen om:

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



## DE REGISTRERTES RETTIGHETER

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

Så lenge de registrerte kan identifiseres i datamaterialet vil de ha følgende rettigheter: innsyn (art. 15), retting (art. 16), sletting (art. 17), begrensning (art. 18) og dataportabilitet (art. 20).

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

## FØLG DIN INSTITUSJONS RETNINGSLINJER

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

Ved bruk av databehandler (spørreskjemaleverandør, skylagring eller videosamtale) må behandlingen oppfylle kravene til bruk av databehandler, jf. art 28 og 29. Bruk leverandører som din institusjon har avtale med.

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

## MELD VESENTLIGE ENDRINGER

Dersom det skjer vesentlige endringer i behandlingen av personopplysninger, kan det være nødvendig å melde dette til oss ved å oppdatere meldeskjemaet. Før du melder inn en endring, oppfordrer vi deg til å lese om hvilke typer endringer det er nødvendig å melde:

<https://www.nsd.no/personverntjenester/fylle-ut-meldeskjema-for-personopplysninger/melde-endringer-i-meldeskjema>. Du må vente på svar fra oss før endringen gjennomføres.

## OPPFØLGING AV PROSJEKTET

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

Kontaktperson hos oss: Sturla Herfindal

Lykke til med prosjektet!

## 8.3 Intervjuguide

Aktuelle spørsmålsformuleringer:

- Kan dere fortelle litt om hvordan dere løste oppgaven?
- Hvorfor valgte dere å løse denne delen av oppgaven slik?
- Hvis dere skulle løst oppgaven igjen, på hvilken måte ville dere da løst den?
- Hvordan kan dere være sikre på at løsningen din vil gi riktig svar?
- Er det noe dere har lært i matten som du kunne bruke når du løste oppgaven?
- Er det noen deler av denne oppgaven som ligner på en oppgave dere har løst tidligere?
- Kunne dere løse den på samme måte, eller måtte dere endre litt på stegene?
- Var det noe informasjon i denne oppgaven som var unødvendig?
- Hvordan synes dere det var å jobbe med slike oppgaver?
- Har dere lyst til å jobbe med flere slike oppgaver? Hvorfor/hvorfor ikke?
- Var det en plass dere stoppet opp i oppgaven? Hva gjorde dere da?
- Er det noe forskjell på å jobbe med disse oppgavene i forhold til hva dere pleier å jobbe med i matten?

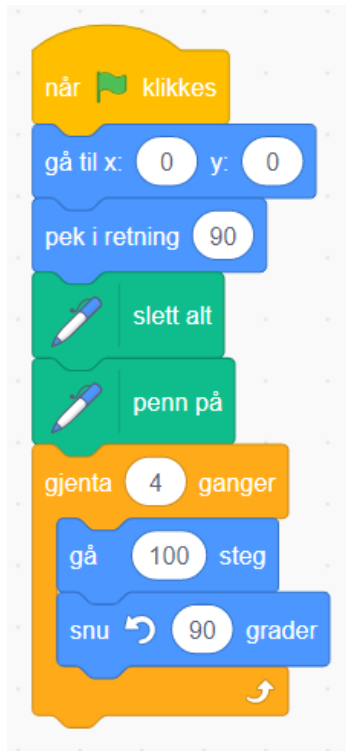
## 8.4 Oppgavene

### Figur programmering

#### Oppgave 1

Her er en kode.

- Klarer dere å se hvilken figur vi har laget, bare ved å se på koden?
- Hvordan ser dere, ut ifra koden, at det må være akkurat denne figuren?



#### Oppgave 2

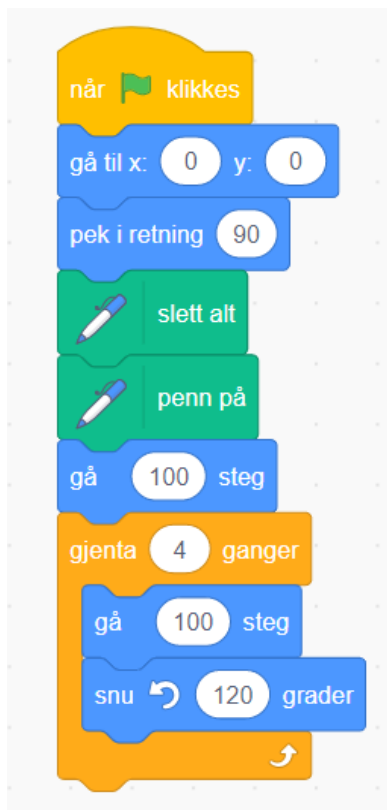
Bruk koden fra oppgave 1, og prøv å endre denne koden slik at Sprite tegner et rektangel, der to og to sider er like lange.

**Før dere begynner på oppgaven, snakk sammen om hvordan dere vil løse den.**

### Oppgave 3

Vi har prøvd å få Sprite til å lage en trekant, men det er noe som ikke stemmer helt i koden vår. Kan dere hjelpe oss å gjøre slik at Sprite bare går akkurat én trekant?

- Gå igjennom koden sammen, og snakk sammen om de ulike blokkene og finn ut hva som ikke stemmer.
- Legg inn koden i Scratch og fiks problemet.



## Oppgave 4

Dere skal nå være arkitekter og designe et hus. Lag først en skisse av huset dere vil designe. Når dere har designet huset skal dere lage en kode slik at Sprite tegner det.

- a) Snakk sammen om hvordan dere vil at huset skal se ut og tegn huset deres her:



- b) Lag koden i Scratch som tegner deres hus.

## Oppgave 5

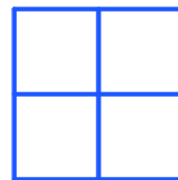
Her ser dere ulike mønstre og figurer. Velg en av dem og lag en kode slik at Sprite tegner den samme figuren. Får dere tid, gjerne lag flere, eller design deres eget mønster.

Når dere har valgt dere en figur eller et mønster snakk sammen om hvordan dere skal lage koden på best mulig måte.

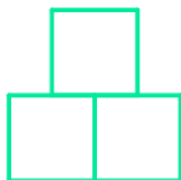
**Sikksakk**



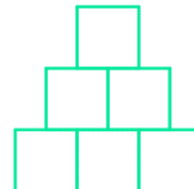
**Rutenett**



**Pyramide kvadrat 1**



**Pyramide kvadrat 2**



**Kvadrat på rad**



## 8.5 Transkripsjon case 1

### 8.5.1 Oppgaveløsning

#### **Opptak 1.1**

- Amanda: Oppgave 1, her er en kode. Klarer dere å se hvilken figur vi har laget, bare ved å se på koden?
- Anna: De har lagd den, vi må også lage den.
- Amanda: Ja.
- Anna: Da er det først det flagget.
- Amanda: Når flagget klikkes.
- Anna: Nei, vent jeg tror ikke vi skal gjøre det.
- Amanda: Jo, vi skal det.
- Anna: Okei, greit.

*Anna og Amanda snakker seg gjennom koden sammen.*

\*Legger inn koden fra oppgave 1\*

- Amanda: Leser spørsmålet på arket. Hvordan ser dere ut ifra koden at det må være akkurat denne figuren?
- Anna: Den er helt lik.
- Amanda: Ja, den er helt lik, det er sant det.
- Anna: Men klikk flagget da.

(Klikker på flagget – Sprite tegner et kvadrat.)

- Amanda: Hm.
- Anna: Det er riktig.
- Amanda: Men det føles ikke riktig ut.
- Anna: Jo, det er riktig.
- Amanda: Ja, okei.
- Anna: Neste.

*Leser oppgave 2 i kor.*

Anna: Vi må få den til å lage et rektangel.  
Amanda: Ja, der to og to sider er like lange.  
Anna: Ja, to og to like sider, ikke alle er like.  
Anna: Hvordan skal vi gjøre det?  
Amanda: Jeg tror hvis vi tar penn og tar forskjellige farger.  
Anna: Vi kan ta samme farge på de to sidene som skal være like lange.  
Amanda: Ja, det var det jeg også tenkte.  
Anna: Okei, hvis vi skal bruke denne koden til å lage et rektangel.

*\*Leter gjennom de ulike blokkene. \**

Amanda: Styring, jeg tror det er her.  
Anna: Nei, jeg tror det er skritt. Eller gjenta noe.

*Sitter lenge og tenker. Anna finner frem papir.*

Anna: Okei, men hvis vi skriver ned hva vi velger å gjøre først.  
Amanda: Kanskje vi skal ta bort slett alt.

*\*Tar bort «slett alt», «gå 100 steg», «snu 90 grader» og løkken. \**

Anna: Vi må ha flere slike «penn på» eller noe sånn.  
Amanda: Fire «penn på» kanskje  
Anna: Prøv det. Nei, det går ikke.  
Amanda: Hva gjør egentlig «penn på»? Ingenting,  
Anna: Slett alt av de der penn på.

*\*Tar bort alle blokkene unntatt «når flagg klikkes» \**

*\*Legger til tre «penn på», en «penn av» og en «slett alt» \**

Anna: Skjønnte du det?  
Amanda: Nei. Det var vanskelig.



Anna: Skal vi sjekke oppgave 3 og se om den kan hjelpe oss noe?

Amanda: Lag en kloss. Det er garantert denne vi må bruke.

\*Prøver å lage en ny kloss, men går bort fra det. \*

Anna: Her «stemple avtrykk», den må vi bruke.

Anna: Nei det funka ikke.

\*Legger til «gå til x: 0 og y: 0», «stemple avtrykk» og «sett pennfarge til grønn» \*

Anna: Vi klarte jo å lage en firkant i stad. Hvis vi går tilbake og legger inn den så kanskje den kan hjelpe oss.

\*Legger inn koden for kvadrat \*

Amanda: Der fikk vi firkanten tilbake.

Anna: Vi må bare ta enten en mer av de der.

\*Legger til en «penn på» \*

Anna: Se nå ble den større.

Amanda: Ja, det er den blokken som gjør den større.

Anna: Hvis vi tar litt flere så blir den større.

Amanda: Jeg tror ikke den blir noe større egentlig.

Anna: Den ble større enn når vi bare hadde en.

\*Fjerner «penn på» slik at de bare har en av disse \*

Anna: Nei den ble ikke større.

Anna: Okei hvis vi endrer på gjenta.

\*Endrer til 5 ganger \*

Amanda: Nei, men da stopper den bare der.

Anna: Ja, men den går en ekstra.

\*Endrer til 6 ganger \*

Anna: Den går lengre og lengre. Vi må ta flere penner og så må vi gjenta flere ganger.

Amanda: Gjenta 4 ganger.

Anna: Nei, gjenta 5 ganger så tar du flere «penn på».

(Prøver – gir ikke riktig løsning)

Amanda: Vi må få den til å være et rektangel.

Amanda: Skal vi prøve den klosse greia igjen.

\*Lager en ny kloss, «Rektangel 4» \*

\*Setter den nye klossen sammen med tre stykker «penn på». Prøver å koble den sammen med «når flagg klikkes» \*

Amanda: Men den går ikke på den.

Anna: Den kan heller ikke gå i noen andre. Det går ikke.

Anna: Skal vi ikke gå til oppgave 3 så ser vi.

Amanda: Nei, jeg vet ikke.

Anna: Jeg leser oppgave 3.

*Leser oppgave 3.*

## **Opptak 1.2**

Anna: For å lage en trekant, det vet jeg.

\*Går inn for å lage ny kloss igjen. Lager kloss med navn rektangel. \*

Anna: Kan jeg prøve en ting.

\*Endrer til gjenta 2 ganger. \*

Anna: Vi burde spør om det er et rektangel eller om det er en firkant.

Amanda: Det har noe med 90 grader. Pek i retning 90 eller noe sånn. Det har noe med den å gjøre.

Amanda: Den skal være stump hvis vi skal lage trekant.

Anna: Og da må gradene være mindre enn 90.

Amanda: Ja, men hva er gradene da?

\*Endrer «pek i retning 90 til 120» \*

Amanda: Det er stump.

(Prøver koden. Sprite tegner fremdeles et kvadrat, men den starter en annen plass enn før.)

Amanda: Hæ?

Anna: Vent kan jeg prøve en ting. Hvis den er helt rundt.

\*Snurrer på hjulet på «pek i retning» \*

Amanda: Nei prøv spiss. Ja 60.

Anna: Et rektangel er ikke 60.

Amanda: Nei, en trekant tenker jeg på nå.

Amanda: En trekant skal alltid være 180 grader.

Anna: Ja, men rektangelet er også 90 grader.

\*Prøver «pek i retning 180» \*

Amanda: Det fungerte ikke.

Anna: Rektangel skal også være 90 grader.

Amanda: Det har noe med den går til x (blokken «gå til x: og y:»).

Anna: Det må være forskjellig hvis den ene linja er x og den andre er y.

Amanda: Ja, nettopp.

\*Endrer «gå til x: 0 og y: 2». Prøver koden. \*

Amanda: Nei, det var ikke det.

Anna: Vent kan jeg prøve en ting.

\*Endrer gjenta 4 ganger til 3 ganger. \*

Anna: Nei, den må gå to.

Amanda: Hvis vi skal lage en trekant?

Anna: Ja, da må den gå to. For hvis den går to ganger så er den sånn, og da trenger vi bare den streken (hypotenusen). Og da må vi endre på noe nå.

Anna: Det er gradene som er feil. Den skal ikke snu 90 grader for at det skal bli et rektangel.

Amanda: Nei, den skal ikke det. Nei det skal den jo ikke. Hvor mange grader må den snu? Nå trenger vi matte. Åh jeg trodde dette ikke var matte.

Anna: Okei vi kan kladde. Jeg skriver firkant er lik 90 grader.

Amanda: Hva er rektangel da?

Anna: Rektangel må være mer. Prøv 110.

Amanda: Da fikk vi en rett strek.

Anna: Okei vi tar den gjenta fire ganger. Så tar vi den snu 110 grader.

(Sprite tegner noe som minner om en trekant)

Amanda: Vi fikk en trekant.

Anna: Trekant!

Amanda: Nei, det er ikke en ordentlig trekant.

Anna: Men det er rektangel han prøver på der. Vi må ta den ned til 100.

\*Endrer til snu 100 grader. \*

Amanda: Hæ? Nei, jeg vet hvordan vi lager trekant nå.

\*Endrer til gjenta tre ganger. \*

Amanda: Nei hæ?

Anna: Hvis vi endrer på gå steg. Nei vi må endre på den (blokken med snu) fordi den må gå mer innover.

\*Endrer til snu 120. \*

Amanda: Der, vi fikk trekant!

Anna: Yes.

*Elevene prøver nå å lage rektangel.*

\*Endrer til gjenta fire ganger. \*

\*Endrer til snu 100 grader. \*

Anna: Det blir ikke rektangel. Det er på grunn av stegene.

Amanda: Hvis vi prøver 80 grader. Nå er vi i nærheten til å få en femkant, ikke et kvadrat.

Anna: Men du ser at strekene er lengre. Det betyr at vi er i nærheten.

\*Prøver å endre på «pek i retning». \*

Anna: Ta den på 90.

Amanda: Vi prøver 105.

\*Fortsetter å prøve og endre retningen. \*

Amanda: Dette var vanskelig.

Anna: Kanskje vi skal spørre?

*Elevene spør om hjelp.*

Anna: Vi har greid trekanten, men vi greier ikke rektangelet, vi greier bare firkant.

Bente: Ja, på oppgave 2 tenker du?

Anna: Ja, vi greier trekanten, men ikke rektangelet det blir bare en firkant.

Bente: Ja, er det den dere har lagt inn nå?

Amanda: Ja, det er trekanten.

Bente: Okei, hvis vi først ser på trekanten. Har dere sett på alle blokkene og tenkt på at den skal gå akkurat en trekant. Dere har vel fjernet noe, men går den akkurat en trekant nå?

Amanda: Nei.

Anna: Prøv å sett inn gå 100 steg.

\*Legger inn koden slik den ser ut på arket. \*

Amanda: Nå går de liksom en ekstra.

Bente: Det var sånn dere begynte var det ikke?

Amanda: Jo.

Bente: Så dere har gjort noe riktig, når dere fjernet den (100 steg). Men hvor begynner Sprite å gå når han tegner?

*Elevene peker.*

Bente: Ja, og hvor stopper han nå?

*Elevene peker.*

Anna: Det er fordi vi har gjenta fire ganger.

Amanda: Ja, så hvis vi endrer til tre så blir det sånn.

Bente: Bra.

Bente: Så var det rektangelet. Kan dere legge inn det dere prøvde på.

Anna: Ja, okei. Så vi lagte firkanten, så prøvde vi oss litt frem på hvordan vi kunne lage de to strekene lengre.

\*Legger inn kode for kvadrat. \*

Amanda: Da får vi firkant.

Anna: Ja og de to strekene må jo være lengre. For det er jo en firkant sant.

Bente: Ja en firkant, eller et kvadrat.

Anna: Og vi skal ha et rektangel og da må jo to av strekene være lengre. Men vi finner ikke ut hvordan de to strekene kan bli lengre.

Bente: Hvilken blokk er det her som bestemmer hvor langt den går?

Anna: Stegene.

Bente: Ja, og hva betyr stegene, hvis dere ser på figuren?

Anna: At den lager lengre strek.

Bente: Ja, så hvis den går 100 steg ...

Anna: Så vi må egentlig bare få den til å gå flere steg.

\*Endrer antall steg fra 100 til 120. \*

(Sprite tegner et kvadrat der sidene er lengre enn i stad)

Amanda: Da bli firkanten større. Så vi trenger jo den andre stegene også.

\*Legger til «gå 100 steg» før løkken. \*

Anna: Nei, det funker ikke.

Bente: Hvis dere tegner et rektangel her.

*Elevene tegner.*

Bente: Siden dere har skrevet her at den skal gjenta fire ganger, så betyr det at den går like langt fire ganger.

\*Endrer til fem, så til tre, og så til fire igjen. \*

Anna: Han skal snu når han har gått 90 grader så snur han. Så vi må ta den høyere sånn at han går lengre før han snur. Det skal ikke være 90 grader, det skal være 120. Nei trekanten er 120, så den må være mindre.

\*Prøver 110. \*

Anna: Nei det skal være mindre enn 90.

Bente: Hvis dere tegner et kvadrat på arket, der alle sidene er like lange.

*Eleven tegner.*

Bente: Hva er forskjellen på de to tegningene?

Anna: De to sidene der er lengre.

Amanda: Det er to og to like sider.

Bente: Ja. Så hvis dere tenker at Sprite begynner her, så må den gå sånn, så må den gå en annen lengre, så må den gå den samme lengden som der nede, og så den korte igjen.

Anna: Det betyr at vi må sette inn noe mer. Så den skal først gå 90 steg. Så skal den gjenta den en gang.

Amanda: Ja, men vi må flytte den (90 steg) inni.

Anna: Okei, nå har vi laget en lengde.

Amanda: Ja.

Anna: Så skal vi lage ny lengde. Det betyr at vi må sette på penn igjen.

Anna: Så må vi si hvor den skal gå. Den skal gå opp, så da må den snu 90 grader.

Anna: Så skal den gå igjen. Men den kan ikke gå 90 for det er det samme som den andre.

Amanda: Ja, så 100 steg.

Anna: Nei, den må gå 80.

Amanda: Ja, okei.

Amanda: Skal den peke i retning 90?

\*Endre på blokken «pek i retning 90». \*

Bente: Det jeg kan si er at det som står frem til «penn på» trenger dere ikke å endre på. Det er bare innstillingen for å få den til å stå på riktig plass. Så det er det under dere må tenke på.

Anna: Det er det her vi må tenke på, så det er det som er feil.



\*Endrer på koden slik at inni løkken står det «gå 100 steg», «snu 90 grader», «gå 80 steg». \*

\*Endrer til gjenta to ganger. \*

Anna: Jajaja.

Amanda: Da ble den lengre der.

Anna: Ja, men den må bare være kortere der oppe.

\*Endrer til gjenta fire ganger. \*

Anna: Nei, nå ble det firkant.

Amanda: Kvadrat.

Anna: Men vi gjorde riktig i stad.

\*Prøver å endre steg. \*

Amanda: Nei, det er noe med den 90 grader greia.

\*Fjerner «snu blokken». \*

Amanda: Det funket ikke.

Anna: Den må snu, så vi må ha med 90 grader.

\*Legger til «snu», men endrer til «snu 105 grader». \*

Anna: Nei, du må ta i 90.

Amanda: Men det har vi jo prøvd.

### **Opptak 1.3**

Anna: Vi var nærmest i stad, når den gikk opp der for da var det et rektangel.

Bente: Ja, hva var det dere endret på da?

Amanda: Men da var den for høy.

Anna: Men det var den (første blokken med «gå antall steg») vi endret på. Var det ikke det?

Amanda: Nei.

Anna: Nei, det var den (løkken). Da hadde vi den på to.  
Anna: Vi må bare få den til å gå en til. Men det er to tallet som er riktig?  
Bente: Ja.  
Anna: Så må vi ta den på nytt her nede.  
Amanda: Ja, sikkert.

\*Legger til en ny løkke, med gjenta to ganger. \*

Anna: Men vi må sette inn hva den skal gjenta, fordi den skal jo ikke gå likt.  
Bente: Men hvis den gjentar to ganger, da betyr det at hvis dere legger inn den siden og den siden, så vil den jo gjenta det to ganger. Da får dere den siden og den siden. (*Peker på figuren de har tegnet på arket*).  
Bente: Så dere trenger ikke den (løkke nr. 2). Men dere må legge inn riktig i denne (løkke nr. 1).

(Koden viser nå en løkke som gjentas to ganger. Inni løkka står det «gå 90 steg», «snu 90 grader», «gå 70 steg»).

Anna: Nå mangler vi bare den siste streken.  
Amanda: Ja, vi mangler den streken rett ned.  
Anna: Skal vi endre den til tre ganger? Nei, da blir det en firkant.  
Bente: Nå har den gått den lengden, så har den snudd 90 grader, så har den gått 70 steg, så mangler vi en til slik at den snur seg igjen her.  
Anna: Den skal snu 90 grader. Du må sette inn snu 90 grader.

\*Setter inn «snu 90 grader». \*

Amanda: Der, der er den!  
Anna: Yes. Vi greide det.  
Bente: Bra jobba!  
Anna: Takk.  
Anna: Så har vi greid trekant.

Amanda: Så vi er ferdig med den og den. Så vi er på oppgave 4.

### **Opptak 1.3**

*Leser oppgave 4.*

Anna: Jeg har et forslag. Vi må ha vinduer.

Amanda: Ja, selvfølgelig. Men hvordan skal han gjøre alt? Altså legge inn vinduer og dører og alt må være fra hverandre.

Anna: Vi må legge det inn i riktig rekkefølge.

\*Sletter koden fra forrige oppgave. \*

Amanda: Så hvis vi tenker vinduer her.

Anna: Nei, vi kan ikke ha de for det greier han ikke.

Amanda: Nei, det er sant.

Anna: Okei, hvis vi lager en dør som er et rektangel.

Amanda: Ja, for det vet vi hvordan vi lager.

Anna: Så har vi et vindu som er en firkant. Det vet vi hvordan vi lager.

Amanda: Ja.

Anna: Vi kan også ha et vindu som er en trekant.

Amanda: Ja, det kan vi.

Anna: Så vi kan lage en trekant oppå firkanten. Og for å lage trekanten kan vi bare ta gjenta.

Anna: Her kan døra være, og det er et rektangel. Vi må liksom prøve å lage alt.

Anna: Skal vi prøve å lage det først.

Amanda: Ja, vi prøver det først.

Anna: Da må vi først lage firkanten.

Amanda: Først firkant. Da går vi bare til denne oppgaven (oppgave 1).

*Amanda leser koden på arket, og Anna legger den inn i Scratch.*

Anna: Så må vi endre på gjenta.

Amanda: Neineinei.

Anna: Jo, fordi vi skal lage trekant på toppen og da må vi ha gjenta seks ganger.

Amanda: Men ja se hva som skjer da. Da står den bare der.

Anna: Ja. Så etter det, må vi ta hvordan vi lagte trekanten.

\*Anna og Amanda ser på oppgave 3 og legger inn i koden. Her fjerner ikke elevene den unødvendige blokken med «gå 100 steg». \*

(Elevene tester koden. Da viser et kvadrat med en trekant til venstre som vender nedover)

Amanda: Men trekanten må være oppå huset.

Anna: Kanskje vi må endre den (løkken til kvadratet).

\*Endrer fra gjenta seks til fem ganger. \*

Anna: Men vi har det nesten.

Amanda: Prøv syv ganger.

\*Endrer til syv. \*

Anna: Den lager den bare under, siden den går jo rundt.

Amanda: Ja.

Anna: Så den må gjenta den fem ganger.

Amanda: Så må vi prøve å vippe den.

Anna: Men den skal ikke gå 100 steg.

\*Endrer den unødvendige blokken til 90 steg. \*

Amanda: Nei.

Anna: Nei, okei. Den har ikke noe å si.

\*Endrer steg i trekanten fra 100 til 90. \*

Anna: Den skal ikke snu 120.

\*Endrer til 110. \*

Anna: Nei, den må ta 120.  
Amanda: Ja, hvis ikke blir det ikke en trekant.  
Anna: Men vi må ha noe under der.  
Amanda: Ja, men hvis vi vipper den.  
Anna: Vi kan jo ikke vippe den.  
Amanda: Jeg vet ikke.  
Anna: Men det har noe med gradene å gjøre.  
Amanda: Jeg vet.  
Anna: Det skal ikke gjentas tre ganger, det skal gjentas fem ganger. Hvis den gjentas tre ganger så går den 1, 2, 3. Men hvis den gjentas fem ganger går den 1,2,3, ... (Blir avbrutt).  
Amanda: Okei vi prøver.

(Prøver – gir ikke riktig løsning)

Amanda: Nei.  
Anna: Jo, det er nesten.

\*Prøver litt forskjellig uten å snakke. \*

Anna: Men hvis vi skal vippe den ned ...

\*Tar bort den unødvendige blokken. \*

(Trekanten er nå inni kvadratet).

Amanda: Vi fikk den inni i hvert fall.  
Anna: Det blir ikke et hus.  
Amanda: Nei, jeg blir litt irritert kjenner jeg.  
Amanda: Skal vi spørre om hjelp?  
Anna: Det velger du.  
Anna: Kanskje de har en ting for å få den til å vippe ned.  
Anna: Okei vi prøver, mens du rekker opp hånda. Så kan de se hva vi har greid.

*Får hjelp.*

- Anna: Nå skal vi lage et hus. Vi har prøvd å få tak på selve huset.
- Amanda: Det har ikke fungert så veldig bra for oss.
- Anna: Det eneste er at vi ikke vet hvordan vi skal få den streken ned og så opp.
- Bente: Nei, skjønner. Så nå går den først en ...
- Anna: Firkant.
- Bente: Ja.
- Anna: Og da har vi tatt den gjenta fem ganger, fordi da går den opp dit.
- Anna: Det eneste er at vi har en rett strek der, men den skal jo være skrå.
- Bente: Ja. Hvis dere tenker at dere først lager den (kvadratet) som dere har gjort nå, så får dere Sprite til å stoppe der. Kan dere gjøre det først?
- Anna: Okei, da må vi ta gjenta fire ganger.

*\*Endrer løkken til kvadratet fra gjenta fem ganger til fire. \**

- Bente: Ja, nå har dere jo begge delene sant. Og ingen ekstra streker som dere ikke trenger. Men vi kan jo ikke ha tak på bakken, det funker jo ikke.
- Begge: Nei, hehe.
- Bente: Da må dere finne ut hvordan dere kan få Sprite til å begynne å tegne der.
- Anna: Men hvis den starter å gå der da, så går den der.
- Bente: Ja, hvordan får du til det da?
- Anna: Da må du ta gjenta seks ganger. Nei da lager han den på siden.
- Bente: Ja, men hvis vi lar den stå på fire ganger den du endret på nå.
- Anna: Er det noe med de to?
- Bente: Jeg synes egentlig alt dette ser bra ut nå.
- Amanda: Vi prøvde å ta bort den «gå 90 steg». Så hadde vi på gjenta fem ganger. Og da havnet den inni firkanten.
- Anna: Det eneste er at vi må få den på utsiden.
- Bente: Ja, det jeg tenker er at vi først tegner den (kvadratet). Så får vi Sprite til å flytte seg opp her uten å tegne noe. Klarer vi det?
- Anna: Da må vi ...

*\*Legger til gjenta fem ganger. \**

Amanda: Men uten å tegne noe.  
Bente: Uten å tegne noe ja.  
Anna: Så før vi tar den (løkken) må vi ta «penn av».  
Bente: Hvordan skal dere få han til å snu seg og gå opp der?

\*Legger til snu 90 grader til høyre. \*

Anna: Nå går han uten å tegne i hvert fall.  
Amanda: Ja vi har klart det.  
Anna: Men vi må få han til å gå den veien.  
Amanda: Ja.  
Anna: Vi har den feil vei.

\*Bytter blokk til snu mot venstre. \*

Amanda: Vi trenger ikke den gjenta.  
Anna: Jo.  
Amanda: Nei vi trenger ikke den.  
Anna: Ja okei, vent litt.  
Anna: Men nei, den går feil vei.  
Bente: Hvor mange grader har dere den på nå?  
Amanda: 15 grader

\*Endrer til 90 grader. \*

Amanda: Der!  
Bente: Ja okei, så da er vi her. Så da må den gå opp her og ned her. Hvilken figur er dette da?  
Anna: Når vi lagte trekanten så fikk vi.. hvis vi går tilbake.  
Amanda: Da trenger vi «penn på» først.  
Anna: Ja, sånn at den får tegnet.  
Amanda: Så trenger vi «gå 100 steg», så gjenta fire ganger.  
Anna: Og inni skal det være steg.  
Amanda: 100 steg.

Amanda: Så snu, den, 120 grader.

*Prøver koden.*

Amanda: Da har vi trekanten der.

Anna: Nei den skal ikke gå steg der. Vi må ta den bort.

*Prøver koden igjen.*

Amanda: Åh nei, nå har vi fått den samme igjen.

Anna: Vi har den samme.

Amanda: Ja, vi har den samme figuren igjen.

Bente: Havnet dere på akkurat samme måte?

Anna: Men det er noe med den fire ganger som er feil. Den skal bare gjenta tre.

Bente: Ja.

Amanda: Det hjelper ikke.

----- Opptak stopper – elevene tar pause -----

#### **Opptak 1.4**

Amanda: Vi holdt på med huset ja. Det vi ikke helt fikk til.

--- Brudd i opptak ---

Amanda: Så vi har klart den firkanten da.

Anna: Men det er noe feil.

Amanda: Så må vi få den opp sånn, men nå har vi den sånn.

Bente: Men hvis dere så på den her, har dere lagt inn den?

Amanda: Ja, der har vi den.

Bente: Inni der (løkken), hva tenker vi må stå der, hvis det skal være en trekant.

Amanda: Tre, men da blir den sånn.

Anna: Han går for mange ganger, han må bort der.

Amanda: Men det gjorde vi jo egentlig her.

Bente: Okei, hvis vi ser på firkanten først. Hvor begynner han å tegne?



Amanda: Han begynner der. For her står det fire så han begynner der og så slutter han der også.

Bente: Ja, fordi han snur 90 grader først. Hvis dere prøver å bytte plass på blokkene inni der (løkken til kvadratet).

Bente: Da har vi firkanten, og så står vi her. Hvis vi får han til å sånn.

Amanda: Ja, som en trekant.

Bente: Ja, hvordan skal vi få det til?

Anna: Hvis vi da tar at han først skal gå en gang.

Bente: Hvis han begynner å gå nå, hvilken retning går han i da?

Anna: Da går han rett opp.

Bente: Ja, vil vi det?

Anna: Nei, han må snu 90 grader.

Bente: Ja.

Amanda: Ja, da kan han gå oppover i sånn stump eller spiss.

Bente: Ja, eller så får vi han til å gå langs den linja der. Det går også an. Men da må vi vite hvor mange grader han må snu.

Anna: 90 grader.

Bente: Ja, prøv å se hvor han havner da.

Anna: Så skal vi få han til å gå dit og da skal han gå ...

Amanda: Prøv med denne.

\*Legger til koden for trekant. \*

Amanda: Der, da ble den litt sånn rar. Men det går fint, det kan bare være et ekstra sånn vindu det.

Bente: Hvorfor ble den sånn? Hvis dere ser på den, hvor mange steg går den der?

Amanda: 90.

Bente: Går den helt til enden der, eller går den bare litt.

\*Prøver å fjerne blokken «gå antall steg» som er før trekanten. \*

Amanda: Nei, da ble den inni.

Bente: Ja, så den skal være der. Men hva er målet med den?

Anna: 90.

Bente: Ja, men poenget er jo at vi vil at den skal bevege seg opp der. Hvor mange skritt må han gå for å komme til hjørne der? Hvor lang er den siden?

Amanda: 100 steg.

Bente: Ja.

\*Endrer den til 100 steg. \*

Amanda: Ja! Da greide vi det!

Anna: Yes, vi klarte det!

Amanda: Da går vi til neste oppgave.

Anna: Nei, b.

Amanda: Ja, b. Lag koden ... Men det har vi gjort, vi har laget den.

Anna: Ja, neste neste. Det her er siste!

Amanda: Ja, det er bra!

Amanda: Wow!

#### *Leser oppgave 5*

Anna: Skal prøve å ta en av de litt lette?

Amanda: Ja.

Anna: Jeg har en følelse av at den, den eller den er litt lett. Kanskje den er den letteste?

Amanda: Vi kan lage sikksakk.

Anna: Eller den, for der er det bare å gjøre det samme. Eller så kan vi lage den for der er det bare å lage firkant sånn, sånn, sånn, så skal den snu og gå ned sånn, sånn, sånn. Så skal den opp.

Amanda: Jeg tror vi prøver oss på sikksakk. Hvis vi får til den, så kan vi gå over på den så den.

Anna: Ja, vi gjør det.

Anna: Når den starter så skal den gå opp.

Amanda: Så det er litt som trekanten da. Bare uten streken under.

Anna: Så du skal ikke gjenta tre ganger du skal gjenta to.

Amanda: Ja.

Amanda: Husker du i stad da vi lagte den firkanten, så tok vi to, så gikk den opp sånn. Det er litt det samme vi skal gjøre her bare at den skal gå opp sånn.

Anna: Vil du prøve.

Amanda: Ja, okei, så hvis vi em ... Vi trenger vertfall «penn på». Så hvis vi klikker nå så er pennen på.

Anna: Men han må gå streken opp. Gjenta en gang.

Amanda: Men hvis den gjentar en gang?

Anna: Ja, da lager han den ene streken.

Amanda: Vi må ha noe inni her.

Anna: Ja, vi må sette inni der at han skal gå 90, em 20 eller 120 steg.

Amanda: Vi kan ta 10 steg da også.

Anna: Men den går ikke oppover.

Amanda: Nå går han 10 steg, ikke sant.

Anna: Vent da kan vi bare lage den. Så er det bare der, strek opp, og en strek der og en strek der. Inni der lager du en strek der.

Amanda: Jeg vil helst lage sikksakk.

Anna: Ja, okei.

Amanda: Men nå går den jo rett frem ikke sant.

Anna: Ja.

Amanda: Vi kan ikke slette noe av dette da.

(Elevene har ikke slettet det som allerede er tegnet, så huset deres viser på skjermen.)

Amanda: Vi burde ha litt mer enn 10 steg.

Anna: Men den skal ikke gå rett frem, det er det vi må endre på, før vi tar stegene.

\*Blar gjennom de ulike kodene. \*

Amanda: Dette her blir vanskelig. Skal vi prøve rutenettet?

Anna: Okei, greit. Da har vi en firkant der. Hvis vi bare får den til å lage en firkant der.

Amanda: Hvis vi flytter han dit.

\*Flytter Sprite til den ene siden på huset. \*

Anna: Så må vi lage den streken du har laget der.

Amanda: Hvis vi skriver 10 steg. Nei da går han der.

Anna: Du må flytte streken lengre ned.

*Blar i oppgavesettet og ser den siste figuren de kan lage.*

Anna: Jeg tror det også er en vi kan lage.

Amanda: Ja, det er det.

Anna: Skal vi lage den?

Amanda: Ja, okei, vi kan prøve.

Anna: Firkant, firkant, firkant, firkant

Amanda: Det tror jeg blir enkelt.

Anna: Ja, vi prøver.

*Går tilbake til oppgave 1 der de lagte firkant.*

\*Legger inn koden for kvadrat. \*

Amanda: Da har vi en firkant. Vi trenger den mindre. Hvis vi putter på mindre steg. 80 steg. Da blir den litt mindre.

Anna: Men vi må lage den her.

\*Prøver å flytte Sprite slik at han skal tegne der, men de har satt inn at han skal gå til x: 0 og y:0. \*

Anna: Okei, vi tar den på 50 steg, så ser vi.

Amanda: Ja, det er passe.

Anna: Så må vi få han til å gå der.

Amanda: Da må vi ta av.

Anna: Gjenta fem ganger.

Amanda: Neineinei, vi må ikke det. Vi må ta penn av.

Anna: Men først må vi gå bort dit, så må vi ta penn av.

Amanda: Nei, vi må ta penn av hvis ikke så tegner han.

Anna: Ja han tegner bare bort dit, men der er det allerede strek.

Amanda: Vi tar bare penn av, det er enklere.

Anna: Så gjenta.

Amanda: Nei, ikke gjenta.  
Anna: Jo, for han skal gå bort dit.  
Amanda: Ja, men vi trenger ikke gjenta.  
Anna: Vi må få han til å gå dit.  
Amanda: Ja, jeg skal det. Så tar vi gå 20 steg.  
Amanda: Nei, vi må ta flere enn det. Vi må ta 60 steg. Fordi ruta er jo 50, ikke sant, så hvis vi skal komme litt unna sånn.  
Anna: Ja, men den må gå lengre unna. Den må gå hundre.  
Amanda: 100?  
Anna: Ja, for da går den lengre bort. Så må vi lage firkanten på nytt.  
Amanda: Okei, men hvor mange firkanter er det? Sjekk.  
Anna: Det er fem stykker.  
Anna: Så da tar vi bare å lager det på nytt.  
Amanda: Ja, men vi må først ta penn på. Gjenta fire ganger. Gå 50 steg og snu 90 grader. Så får vi en ny en.  
Anna: Så må vi ta penn av.  
Amanda: Men jeg føler vi må gjøre de stegene litt mindre, for nå er det ganske mye avstand.  
Anna: Nei, vi kan ikke gjøre det for da blir det ujevnt.  
Amanda: Hvor mange skal vi lage? Skal vi lage tre?  
Anna: Ja, vi lager tre.

\*Legger inn samme kode igjen for å lage nytt kvadrat. \*

Amanda: Jaa!

*Elevene spør oss om de må lage alle fem.*

Amanda: Vi har laget tre, men må vi lage fem?  
Bente: Ja, gjerne prøv om dere får til alle fem.  
Amanda: Da blir den så forferdelig lang.  
Anna: Vi skriver bare det samme på nytt.  
Bente: Hvordan har dere gjort det?

Anna: Vi har bare skrevet hvordan vi lager en firkant, og så skrevet det flere ganger, med penn av imellom.

Amanda: Men går det ikke an å gjøre sånn at.. Sånn ja.

\*Legger til en løkke på alt de har skrevet. \*

### **Opptak 1.5**

Amanda: Okei, så tar vi gjenta en, eller gjenta to.

(Sprite tegner tre kvadrater på rad, så slettes de før det tegnes tre nye)

Anna: Det er fordi du har penn av. Det går ikke.

Amanda: Den lager bare tre fortsatt.

Bente: Men kan dere gjøre noe med koden slik at når dere har laget en firkant og flyttet dere bort her så kan dere gjenta det? Så slipper dere å legge det inn så mange ganger.

Amanda: Vi trenger egentlig bare dette. Men da gjentar han ikke det. Så hvis du ser så lager han bare to.

Amanda: Vi får den bare til å gjøre sånn.

Anna: I stad klarte vi å gjøre det tre ganger, men den ble så veldig lang.

Bente: Ja, men nå mangler vi egentlig bare to. Men hva betyr denne?

Amanda: At den gjør det fire ganger.

Bente: Ja, hva lager den da?

Anna: En firkant.

Amanda: En firkant, en gang

Bente: Ja, så det blir en firkant.

Amanda: Ja.

Bente: Så tar dere penn av og gjør hva?

Amanda: 80 steg slik at det blir mellomrom.

Bente: Ja, så hva gjør dere her?

Anna: Penn på, sånn at den kan tegne igjen.

Amanda: Og så er det det samme.

Bente: Ja, hvis dere prøver å fjerne den da. Hva skjer da?

Amanda: Men skal vi fortsatt ha den?

Bente: Ja, den kan bare være der. Så prøver dere å kjøre den.

Amanda: Åja, fem! Ta på fem.

\*Endrer løkken til gjenta fem ganger. \*

Amanda: Jaa, vi klarte det!

Anna: Wiho! Yeey.

## 8.5.2 Intervju

Rebekka: Da tenkte vi å bare ha et lite intervju med dere i forhold til oppgavene og litt i forhold til deres tanker og sånn. Veldig fint hvis dere har lyst til å dele, og så tenker jeg at dette her er ikke noe skummelt i det hele tatt. Det er bare for vår del og for å kunne høre litt hva dere tenker. Og hvis Bente skriver ned et eller annet pc-en så er det ikke fordi at det er noe skummelt, det er bare fordi at hun skriver noe som kanskje er interessant eller sånt.

Bente: Kan dere sette dere litt nærmere sånn at det er helt sikkert at vi får med lyden.

Rebekka: Så tenker jeg at det er veldig fint hvis vi får svar fra begge to. At begge svarer på hvert spørsmål, men hvis dere ikke har noe å komme med, er det selvfølgelig helt greit. Og hvis dere har noen spørsmål underveis må dere bare spørre. Okei?

Begge: Ja.

Rebekka: Nå har vi begynt lydopptaket bare sånn at dere vet det.

---

Rebekka: Hvordan syns dere det var å jobbe med slike oppgaver som dette?

Begge: Gøy!

Rebekka: Ja, hvorfor var det gøy?

Anna: Det var spennende å gjøre noe nytt. Jeg har ikke helt gjort det før, eller vi har koda før, men ikke akkurat helt sånn. Gøy å gjøre noe nytt.

Rebekka: Hva var forskjellen på slik dere har koda før, og hvordan dere koda nå?

Amanda: Vi har koda mest med Lego.

Rebekka: Jaa okei, men var det noe i selve kodingen som var forskjellig eller?

Anna: På Legoen er det lettere programmering. Men her er det annerledes for med Lego skal du bare bygge en figur og så trykke på en knapp liksom og så skal du velge hva den skal gjøre. Men her kan du gjøre litt mer og sånn.

Rebekka: Du kan gjøre litt mer ja. Hva kan du gjøre mer av?

Amanda: Du kan få figuren til å tegne sånn dere hadde. Em og den andre kan du få til å kjøre.

Anna: Så kan du lage vifte sånn at den snurrer rundt liksom.

Amanda: Du kan lage et lite helikopter, men den flyr ikke.

Anna: Du kan lage mer gøy ting her.

Rebekka: Så det va mer gøy ting?

Anna: Ja!

Rebekka: Kunne dere gjøre flere ting i dette programmet, enn dere kunne i det andre?

Begge: Ja.

Anna: Og så er det av og til gøyere å gjøre noe der det er litt mer sånn at du må tenke litt.

Rebekka: Ja, du må tenke litt.

Anna: Istedenfor Lego Wedo som du kanskje syns er litt lett og da er det gøyere å gjøre noe som du må jobbe litt med.

Rebekka: Jaa så bra! Kom du på noe som var annerledes?

Amanda: Det var annerledes for når du bygde med Lego så kunne du lage figuren selv. Sånn at den kjørte rundt i klasserommet og sånn. Mens den andre, den dere hadde, så kunne vi tegne og det kunne vi ikke gjøre med Legoen.

Rebekka: Så dere mener det var flere muligheter med den vi brukte?

Begge: Ja.

Rebekka: At dere fikk bruke hode litt mer og tenke litt?

Begge: Ja.

Rebekka: Har dere lyst å jobbe med flere sånne oppgaver?

Begge: Ja.

Rebekka: Hvorfor det?

Amanda: Det er spennende å se hva vi kan få til og å oppleve ny måte å programmere på.

Anna: Det er spennende å se hva vi kan få til. Og det er også gøy å bruke hodet og tenke litt. Og det er ofte gøy å tenke litt og at det tar litt tid.



Rebekka: Det va greit for dere at det tok litt tid å løse oppgavene og det var egentlig litt gøy?

Begge: Ja.

Rebekka: Var er det en plass dere liksom stoppet opp når dere jobbet med oppgavene?

Begge: Ja, mer enn en.

Rebekka: Ja, hva gjorde dere da for å komme dere videre?

Anna: Vi spurte om hjelp. Vi så også tilbake på de oppgavene vi hadde greid for å se om det var noe der vi kunne bruke til å se om det er den ene tingen vi må forandre på liksom.

Rebekka: Skjønner, så dere brukte på en måte de oppgavene dere hadde gjort før til å løse de oppgavene dere slet med da?

Begge: Ja.

Rebekka: Ja så bra. Var det noe annet dere gjorde når dere stoppet opp?

Anna: Vi spurte om hjelp, men vi tegnet også på arket.

Rebekka: Dere tegnet på arket ja, det var lurt. Hva var det dere tegnet der da?

Amanda: Vi tegnet litt forskjellige figurer.

Anna: Så tegnet vi liksom hvordan vi kunne gjøre det. For eksempel når vi skulle lage et hus så skulle vi lage taket. Så tegnet vi først firkanten og den greide vi. Så skulle vi lage trekanten oppå der og da tegnet vi liksom hvor vi måtte få den der figuren til å gå for å tegne den.

Rebekka: Ja okei, det var jo lurt da. Så dere tegnet og lagde en plan for hvordan dere skulle gjøre det?

Begge: Ja.

Rebekka: Fikk dere det til da eller måtte dere jobbe litt?

Begge: Vi måtte jobbe litt (i kor)

Rebekka: Dere klarte det til slutt?

Begge: Jaa

Rebekka: Så bra da!

Rebekka: Hvordan var det når dere klarte det da?

Begge: Gøy

Rebekka: Fikk dere lyst å jobbe videre da når dere fikk det til?

Begge: Ja.

Rebekka: Det va gøy å få litt nye utfordringer?

Begge: Ja.

Rebekka: Så bra.

Rebekka: Okei, på oppgave 1 her så står det: Klarer dere å se hvilken figur vi har laget bare med å se på koden. Hva gjorde dere da?

Anna: Em, når vi skulle se på figuren så måtte vi se på grader, vinkler og hva den skulle gjenta. Ofte hvis du skal lage firkant så må den gjenta 4 ganger. Men hvis du skal lage en trekant eller femkant så må du gjenta det flere ganger.

Rebekka: Ja okei. Så dere så på hvor mange ganger dere gjorde det

Anna: Ja, så så vi på vinklene. For eksempel 90.

Rebekka: Ja, så når det er 90 grader, hva vil det si?

Amanda: Da kan det være et kvadrat eller et rektangel.

Rebekka: Ja, så det va det dere så på da?

Anna: Ja.

Rebekka: Men så dere hvilken type firkant det var?

Amanda: Em, likesidet

Anna: Likesidet siden alle sidene var like.

Rebekka: Ja, så dere så at alle sidene var like lange og at det var 90 grader. Og da så dere at det va en firkant. Men kan dere navnet på den firkanten som har helt like sider?

Anna: Nei.

Amanda: Kvadrat.

Rebekka: Kvadrat ja, veldig bra.

Rebekka: Så tenkte jeg litt på den oppgave 4 her. Dere sa jo litt om hvordan dere løste den. Dere lagte en plan og dere tegnet ikke sant. Men når dere skulle kode, hva gjorde dere da?

Anna: Først lagde vi en firkant sånn vi hadde laget tidligere. Så gikk vi tilbake til den firkanten. Så når vi skulle lage trekanten ...

Amanda: Vi måtte liksom først skru av sånn at den ikke tegnet for å gå opp for at vi kunne lage taket.

Anna: Vi lagde først en firkant så med en trekant på toppen som tak.

Rebekka: Så da skrudde dere av pennen?

Begge: Ja.

Rebekka: Tok det litt tid før dere forsto at dere måtte skru av pennen, eller kom dere på det med en gang?

Amanda: Em, det tok litt tid.

Anna: Ja, litt tid. Så vi skjønte ikke med en gang, men etter hvert skjønte vi at hvis vi skrur av pennen så kommer det ikke flere streker.

Rebekka: Ja, så bra. Skrudde dere den på igjen etterpå da?

Begge: Ja.

Rebekka: Så bra.

Bente: Men var det ikke dere som hadde trekanten litt på skeiva?

Begge: Jo.

Anna: Trekanten var liksom skeiv med en gang siden vi starta i feil side. Og av og te gikk den litt nedenfor siden vi ikke hadde gått helt opp på toppen av firkanten. Så da gikk trekanten litt nedi firkanten. Så når vi da tok og gikk opp på nytt så greide vi det til slutt.

Rebekka: Ja, da klarte dere det til slutt. Så dere jobba litt, så jobba dere videre så løste dere oppgaven etter hvert?

Begge: Ja.

Rebekka: I stad sa dere jo at dere hadde brukt noen av kodene dere hadde brukt tidligere?

Amanda: Ja, oppgavene vi hadde gjort tidligere.

Rebekka: Ja, så bra. Så tenkte jeg på oppgave 5, fikk dere begynt noe her?

Amanda: Vi begynte på den der. Den med fem små firkanter.

Rebekka: Ja, fem små firkanter ja. Så bra. Hvordan begynte dere når dere skulle starte på denne oppgaven?

Amanda: Vi lagte først en firkant. En helt vanlig en.

Anna: Så så vi på størrelsen, og så tok vi å gjorde den mindre, siden den var litt stor.

Amanda: Så tok vi den til 80 skritt hver linje.

Anna: Etter det tok vi av pennen, så tok vi å gjorde sånn at den går litt sånn at det ble litt mellomrom. Så tok vi på pennen igjen, så lagde vi en firkant til. Så spurte vi om hjelp for når vi hadde laget 3 firkanter så ble den så lang nedover og da skjønte vi etter hvert at vi kunne sette inn en sånn gjenta blokk. Vi hadde prøvd det, men da funka det ikke, for da lagde den bare tre stykker, men vi satte den gjenta blokken på feil sted. Så når vi satt den inn på nytt et annet sted så kunne vi få lagd 5 med en gang.

Rebekka: Så bra. Lagte dere en plan da før dere startet?

Anna: Litt, siden vi visste litt hvordan vi skulle lage firkanten.

Amanda: Ja, det gjorde vi.

Anna: Så da bladde vi tilbake til der det stor hvordan vi skulle lage firkant. Så gjorde vi det, så tok vi av pennen så gikk vi litt slik at det skulle bli et mellomrom.

Amanda: Vi hadde sånn 50 skritt til firkanten. Og 80 skritt imellom.

Anna: Ja, så vi hadde 50 skritt på sidene i rutene og 80 skritt mellom.

Amanda: Vi prøvde først 10, så 20, så gikk vi opp til 90. Det ble altfor høyt. Så prøvde vi 100 og det ble altfor høyt igjen. Så prøvde vi 80 da og det var passe.

Rebekka: Så bra. Hvis der skulle løst denne oppgaven på ny, hvordan ville dere løst den da?

Amanda: Samme.

Anna: Cirka samme.

Rebekka: Ja, dere hadde gjort det på samme måte.?

Begge: Ja.

Rebekka: Er det noen andre oppgaver dere tenker dere kunne løst på en annen måte?

Anna: Kanskje huset.

Rebekka: Ja, oppgave 4. Hvordan tenker dere at dere kunne løst den på en annen måte?

Anna: Kanskje starta med trekanten eller noe sånn.

Amanda: Eller bare lagd et rektangel på en måte. Så prøvd å lage små vinduer og dør inni den.

Rebekka: Ja, så hvis dere skulle gjort det på ny så ville dere kanskje prøvd å lage et litt vanskeligere hus på en måte?

Begge: Ja.

Rebekka: Så sa du at dere kanskje skulle ha starta med trekanten først og at det kanskje hadde blitt enklere?

Anna: Ja, det hadde nok vært lettere.

Rebekka: Så bra, da har dere reflektert litt over det. Og så tenkte jeg litt sånn på oppgave 1 her da sa dere at det var et kvadrat. Hvordan kan dere være helt sikre på at det er et kvadrat?

Anna: Em ...

Rebekka: Dere snakket litt om det i stad.

Anna: Ja, kvadrater har ofte 90 grader i vinkelen og så ...

Amanda: Gjentas alle linjene 4 ganger

Anna: Og det er ikke noe sånn som sier eller viser på en måte at den ene skal være lenge, eller den andre skal være kortere. Så man kan på en måte, for eksempel

når vi slet med det rektangelet da slet vi med å få laget to av de andre sidene like. Så så vi på firkanten ...

Amanda: På om det kanskje kunne vært noen endringer der.

Rebekka: Ja, så bra. Så dere brukte egentlig oppgave 1 når dere skulle løse oppgave 2 og, når dere skulle lage rektangelet?

Begge: Ja.

Rebekka: Hvordan fungerte det da?

Amanda: Det fungerte bra.

Rebekka: Ja, dere fikk det til?

Amanda: Ja.

Rebekka: Så bra.

Rebekka: Da lurer jeg på, i dette oppgavesettet er det noen av oppgavene som ligner på en annen oppgave som dere har løst før?

Anna: Sånn i det samme settet eller i en annen oppgave som vi har hatt på skolen før?

Rebekka: Nå tenkte jeg først og fremst på en annen type oppgave på skolen.

Amanda: Ja, vi har jo for prøvd å lage sånne spill, programmerere spill. Så det synes jeg er ganske likt som dette.

Anna: Ja, det er litt likt.

Rebekka: Hva var det dere gjorde da som var likt?

Amanda: Du skulle programmere en figur til å gjøre forskjellige ting.

Rebekka: Ja okei. Så dere har jobba med noe lignende før.

Anna: Du måtte tenke litt likt på en måte.

Rebekka: Kunne dere løse den på samme måte som dere løste denne oppgaven i dag, eller måtte dere endre litt på stegene?

Amanda: Kanskje endre litt på stegene.

Anna: Ja, endre litt.

Rebekka: Du sa dere kunne tenke litt likt. Hva tenker du på da, hvordan kunne du tenke likt?

Anna: Når du skulle programmere den figuren til å gjøre sånn forskjellige ting, så må du på en måte tenke litt likt siden han skal på en måte, han skal ikke tegne noe, men han skal gå gjennom en løype og da må du programmere den til å gjøre det litt det samme siden den skal jobbe gå i forskjellige linjer og sånt, men han skal ha ikke med seg noen penn til å fargelegge.

Rebekka: Så i tillegg til at dere programmerte litt annerledes, så var det pennen som var annerledes, som dere måtte tenke litt over.

Rebekka: Dere har jo sagt at dere har brukt både oppgave 1 og oppgave 2, når dere løste de andre oppgavene, ikke sant. Så dere har jo brukt de tidlige oppgavene til løse oppgaver senere i oppgavesettet.

Anna: Ja.

Rebekka: Da tenkte jeg på oppgave 3. Så spurte vi dere om det er noen av de blokkene her som er unødvendig å ha med.

Anna: Hm ...

Amanda: Det er ikke så mange.

Anna: Det var trekanten vi skulle lage da?

Rebekka: Ja.

Anna: Når du skal lage den så kunne du tatt ut det firetallet for eksempel eller noe sånt.

Rebekka: Ja, du kunne tatt bort firetallet.

Amanda: Ja, så bytter du heller med 3.

Rebekka: Ja, hvorfor det?

Anna: Fordi en trekant har bare tre hjørner.

Rebekka: Ja, 3 sider ikke sant, og 3 hjørner.

Amanda: Du kan også bytte litt på retning sånn at hvert hjørne er jo egentlig 60 grader i en trekant. Likesidet trekant hvert fall.

Rebekka: Ja, så du mener at han burde snudd 60 grader. Er det du tenker på?

Amanda: Ja, på en måte.

Rebekka: Ja okei. Er det noen andre ting i blokkene som kanskje ikke hadde trengt å være der? Hvis dere ikke ser noe så det er helt greit.

Anna: Nei, jeg tror ikke det er noe mer egentlig.

Rebekka: Så dere tenker at alt det burde vært med?

Begge: Ja.

Rebekka: Ja, supert. Er det noen forskjell på å jobbe med slike oppgaver som dere har jobbet med i dag og oppgaver dere pleier å jobbe med i matten.

Amanda: Ja, det er jo det.

Anna: Ofte så er det mer sånn at det ikke er digitalt, men mer sånn på ark og i bok og sånn. Men her var det mer digitalt.

Rebekka: Ja, så dere fikk jobbet digitalt, og det var annerledes?

Begge: Ja.

Rebekka: Syns dere det var greit eller hvordan syns dere det var?

Amanda: Ja, det er bedre.

Rebekka: Ja, det var bedre?

Anna: Ja.

Rebekka: Er det andre ting som er forskjellig? Hvordan oppgavene blir stilt kanskje?

Anna: Ja, det er vel ikke helt samme oppgaver. Det er forskjellige typer ofte i mattetimen som vi har til vanlig så er det ofte mer regnestykker. Men her er det litt mer stykker som man må heller tenke litt mer på hvordan du skal løse det. Og når du setter inn ting på den oppgaven så betyr det noe for hva du gjør videre. Men hvis du regner sammen et stykke så pleier ikke det å bety noe for hva som skjer videre.

(Intervjuet ble avbrutt)

Rebekka: Ja, kan du si det en gang ting?

Anna: Når man har en vanlig mattetime så pleier ikke det, eller når du gjør noe der så må du tenke litt på hva du gjør videre. Mens når du løser et regnestykke i matten så betyr det ikke så veldig mye hvilket regnestykke du gjør etterpå. Så du må tenke mer fremover.

Rebekka: Ja, du må tenke mer framover ja.

Anna: Når du setter inn en av de boksene så må du også tenke på hva som skal komme etterpå eller om du burde sette den inn senere.

Rebekka: Så bra, det var jo kjempefint tusen takk. Er det noe annet dere har lyst til å legge til eller fortelle eller hvis dere synes det er noe vi kunne gjort bedre i oppgavene?

Anna: Nei, jeg kommer ikke på noe egentlig.

Rebekka: Det er ikke noe dere har lyst til å fortelle om opplevelsen eller?

Anna: Ikke noen annet enn at det var gøy.

Rebekka: Så bra, supert! Da tenker jeg at dere kan få lov til å gå tilbake til at klasserommet. Tusen takk for hjelpen.

## 8.6 Transkripsjon case 2

### 8.6.1 Oppgaveløsning

#### **Opptak 2.1**

*Leser opp oppgave 1.*

Beate: Skal vi liksom kode den figuren da?

Bendik: Ja.

\*Legger inn koden for kvadrat og tester den. \*

Beate: Hva skal vi gjøre nå?

Bendik: Sånn.

Beate: Okei, var det det vi skulle gjøre? Fordi her står det klarer dere å se hvilken figur vi har laget bare ved å se på koden.

Bendik: Ja den går bare i ring.

Beate: Ja, da ha vi gjort det da. Hvordan ser dere ut ifra koden at det er akkurat denne figuren. Men jeg tror vi er ferdig med den oppgaven nå.

Bendik: Ja det var ikke vanskelig.

*Leser oppgave 2.*

Beate: Da må vi kanskje ha en lengre.

\*Endrer til «pek i retning 45». \*

Beate: Nei, det var riktig sånn den var i stad.

Beate: Men kanskje det er den.

Bendik: Ja, der er denne.

\*Endrer til «snu 180 grader» inni løkka. \*

Bendik: Nei.



Beate: Nei det var ikke riktig.

Bendik: Hvis vi prøver 100.

Beate: Hæ?

\*Endre til «snu 90 grader» og så til 10 grader. \*

Beate: Oi. Nei, men vent, se nå. Vi må bare prøve. Hva var det vi tok i sta?

Bendik: 10.

Beate: Nei, var det ikke 90 i sta? Prøv halvparten av 90, 45.

Bendik: Nei, vent nå kan vi få en heksagon. Jeg tror jeg vet hva det er da.

\*Endrer til «snu 60 grader». \*

Bendik: Se, nå begynner den å gå nedover sånn.

Beate: Ja, men vi skal ikke ha heksagon, vi skal ha rektangel.

\*Endrer til «snu 70 grader». \*

Beate: Der to og to sider er like. Prøv 25.

\*Prøver 80 grader. \*

Beate: Ja, fortsett.

\*Prøver 25 grader. \*

Beate: Nei, greit.

\*Endrer til «snu 90 grader». \*

Bendik: Å, det er former ikke figurer. På oppgave 1 skulle vi se hvilken form, jeg trodde det var selve han figuren.

Beate: Ja, men nå skal vi prøve å lage rektangel.

Bendik: Å, rektangel.

Beate: Ja, så den er bare lengre enn.

\*Endrer til «gå 110 steg». \*

Bendik: Hm, litt større.

\*Endrer grader til 95. \*

\*Endrer til 200 steg. \*

Bendik: Nei, nå blir den bare større.

\*Endrer til 90 grader. \*

\*Fjerner hele løkken. \*

Beate: Jeg tror vi skal ha de, for vi skal ha samme form som sist.

Bendik: Ja, men vi kan jo endre på det og da blir det jo lettere egentlig.

Beate: Nei, for vi må ha samme form.

Bendik: Okei.

\*Legger løkken inn igjen. \*

Bendik: Jeg tror vi skal legge til flere.

Beate: Nei vi skal ikke det. For vi skal bruke koden fra oppgave 1.

Bendik: Det må være mindre enn 90.

Beate: Ja jeg tror og det.

Bendik: Nei, det må være mer enn 90.

Beate: Mer enn 90?

Bendik: Ja, for se her, hvis den er mindre enn 90 så går den sånn.

\*Prøver med 100 og 110 grader. \*

Bendik: Se her, det ligner mer på et rektangel.

\*Prøver 120 grader. \*

Bendik: Sånn.  
Beate: Det er ikke et rektangel, det er en trekant.  
Bendik: Trekanter er bedre enn rektangler. De er sterkere.  
Beate: Men hva var det du gjorde nå. Kanskje vi prøver 130 da.  
Bendik: 130, da får vi bare en sånn en. Vent det blir stjerne. Vi kan lage stjerne.

\*Endrer til 140 og 150 grader. \*

Bendik: Se stjerne.  
Beate: Men hvis vi prøver 25 grader.  
Bendik: Det har vi sjekket fra før av. Da går den bare sånn.  
Beate: Det er så rart, for det gir ikke mening.

\*Endrer til 90 grader. \*

Beate: Hvis den skal være lengre.. 135, det er sikkert 135.

\*Endrer til 135 grader og tester. \*

Bendik: Nei, det er det ikke.  
Beate: Jeg gir opp snart.  
Bendik: Jeg tipper hvis vi gjør sånn, og så endrer vi den (**løkken**) til 2.  
Beate: Og den (**snu**) til 90.  
Bendik: Og da går den sånn.  
Beate: Ja.  
Bendik: Og vi vil jo egentlig ikke ha den (**en strek**) så da gjør vi sånn.

\*Endrer løkken til 1. \*

Beate: Ta 4 (**løkken**).  
Bendik: 4, da er den på samme.  
Beate: Var den på 4 i sta? Åja.  
Bendik: Hvis vi tar 5.  
Beate: Nei.  
Beate: Men du tok den på 2 ikke sant. Og så var det 90 og 135.

\*Endrer til «snu 135 grader». \*

Beate: Hæ?

\*Endrer til «snu 100 grader», så 90 grader. \*

\*Endrer løkken til 1 gang. \*

Beate: Dette gir ikke mening.

Bendik: Det er denne (**gå til x: y**)

Beate: Okei prøv da.

\*Endrer til «gå til x:90 og y: 10». \*

Bendik: Nei, det er hvor den starter.

Beate: Men du vet det du gjorde i sta. Da var det 2 der (**løkken**). Hva må vi gjøre nå?

Bendik: Aner ikke.

\*Endrer forskjellige ting. \*

Beate: Men et rektangel er jo to og to like sider.

Bendik: Ja, det er to lange.

Beate: Vent kanskje vi må ta 100 her (**steg**) og 50 her (**snu**), for da blir det på en måte halvparten.

\*Endrer og tester koden. \*

Beate: Nei det funka ikke.

Bendik: Det der er hvor mye den snur.

\*Endrer til «gjenta 4 ganger, gå 100 steg og snu 50 grader». \*

Beate: Nei hva skjedde nå.

Beate: Hvis vi bare får den lengre. Begge de lengre. For da blir det et rektangel.

Beate: Jeg dør snart.

Bendik: Det er ikke mulig å lage et rektangel uten å legge på mer.

*Leser oppgaven på ny.*

Bendik: Se her står det bruk koden fra oppgave 1 og endre den koden. Det betyr at vi må endre den. Det er det jeg tror hvert fall.

Beate: Ja, det er sikkert det.

Bendik: Det gjør det jo sykt mye lettere.

Beate: Men det står bruk koden fra oppgave 1, så da må vi bruke den.

Bendik: Ja, så vi kan ikke ta bort denne, men vi kan legge til flere.

Beate: Jeg tror de heller mener tallene.

Bendik: Ja, men det er ikke mulig da.

Bendik: Men vi kan bare følge det bilde her (**oppgave 3**).

Beate: Nei. det er til trekant tror jeg.

Bendik: Åja, det er da vi skal lage trekant. Men den har vi jo gjort allerede egentlig.

Beate: Ja, jeg vet det vi gjorde det i sta.

Bendik: Så oppgave 2 er vanskeligere enn oppgave 3.

Beate: Den er sikkert ikke det, det er bare vi som ikke helt tenker.

Bendik: Jeg klarte å løse oppgave 3 uten å se på bilde, men de løste nok bare den ved å se på den.

Beate: Men vi må bare få den lengre, men jeg vet ikke hvordan vi skal få den lengre.

Bendik: Er det lov å spørre om hjelp?

Beate: Vet ikke.

*Spør oss om det er lov å spørre om hjelp, får ja til svar og får hjelp.*

Beate: Vi skjønner ingenting av hvordan vi skal..

Bendik: Det er ikke mulig egentlig.

Rebekka: Hva da?

Beate: Å lage et rektangel.

Beate: Vi skjønner ikke om vi skal legge til.

## **Opptak 2.2**

Rebekka: Da skal jeg stille dere et spørsmål. Hva er det som er forskjellen på et kvadrat og et rektangel?

Beate: I rektangel er det to og to sider som er like. Det der er et rektangel.

Rebekka: Så rektangel ser ut som en iPad.

Rebekka: Men kan du da gjenta noe fire ganger, hvis to og to skal være like.

Beate: Nei ...

Bendik: Jeg vet det. Det er derfor denne (**løkken**) må være to. Og så må vi bare ha en til av de her.

Beate: Kanskje vi må lage en til sånn. Hvis vi prøver det.

\*Legger til en lik løkke. \*

(Sprite tegner et kvadrat).

Bendik: Nei.

Beate: Nei, det funket ikke.

\*Fjerner løkke nr. 2. \*

Beate: Men nå er den sånn. Vi må få den til å gå her og sånn.

\*Drar Sprite på skjermen for å vise – viser et kvadrat. \*

Bendik: Nei vi må få den til å gå her.

\*Drar Sprite og viser et rektangel. \*

Beate: For nå er den halvparten av en firkant. Nei, det her gir ingen mening.

Beate: Eller jo vent. For hvor mange grader er et rektangel?

Bendik: 90 grader i alle hjørnene.

\*Endrer på antall steg. \*

Beate: Jeg tror den er sånn 120, for se da blir den lengre. Og hvis vi tar den (**snu**) til for eksempel 80.

Beate: Det funket ikke.

Bendik: Da får vi parallelogram.

*Tenker uten å snakke.*

Beate: For se et rektangel er jo sånn. Og si at den er 4 cm og den er 10 cm.

\*Endrer på «gå til x: 10 og y: 4». \*

Bendik: x er opp og y er bortover sånn. Altså y-aksen.

Beate: Da må vi kanskje ha 2 av de (**gå til x: y:**).

Bendik: Nei. Eller jo det må vi.

\*Legger til en til «gå til x: y:». \*

Bendik: Nei, det her er jo bare hvor den starter. Tror jeg.

Bendik: Vi sitter fast på en av de letteste oppgavene.

Beate: Jeg vet det.

Bendik: Skal vi bare hoppe over den?

Beate: Nei, vi må gjøre det.

Beate: Vi skal gjøre ferdig rektangel først.

\*Fjerner den andre «gå til x: y:». \*

Bendik: Vi er de eneste som sitter fast på denne her.

Beate: Jaja.

Beate: Den der, vi må bare forlenge den.

\*Endrer størrelsen på katten. \*

Bendik: Se nå er den mye mindre.

Beate: Ja, da kan vi faktisk se hva vi holder på med.

Beate: Åh, det her gir ingen mening.

Bendik: Nei, jeg vet det.

Bendik: Hvorfor klarer vi det ikke?!

Beate: Må ikke den være lengre?  
Bendik: Jo, selvfølgelig må den være lengre eller den må være høyere.  
Bendik: Prøv å gjøre den høyere istedenfor lengre.  
Beate: Vi må gjøre den dobbelt så stor, så vi må gjøre dobbelt igjen.  
Bendik: Vi må ha en til av de der.  
Bendik: Den er ganske unødvendig.

\*Drar bort løkken. \*

Bendik: Vi trenger ikke den egentlig.

\*Flytter på det som står over løkken. \*

Bendik: Men jeg beholder den der. Men da trenger vi egentlig bare denne.

\*Bruker ikke løkke. Legger til «gå 100 steg, snu 90, gå 150, snu, 90, gå 100». \*

Beate: Jeg tror ikke det er riktig.

Bendik: Jo, se.

\*Legger til «snu 90 grader» og «gå 150 steg». \*

Bendik: Sånn.

Beate: Ja, nå fikk vi rektangel. Endelig.

Bendik: Halleluja.

Bendik: Jeg skal bare spørre om det er riktig.

Beate: Jeg tror ikke det er riktig.

Bendik: Telles dette som riktig? Det er den eneste måten som jeg kunne tenke meg til, fordi hvis du bruker «repeat» så vil den jo gjenta en spesifikk ting og hvis ikke måtte vi hatt to av de, men det hadde blitt mer komplisert enn å bare gjøre det.

Rebekka: Ja. Hvorfor tenker du at det er riktig?

Bendik: Fordi det viser jo det. Og fordi du ser jo her, for først er det 100 steg ...

(Begynner å snakke om å endre katten)



Beate: Men skal vi fortsette på oppgaven?  
Rebekka: Kan dere ikke snakke litt om hvorfor dette blir riktig?  
Bendik: Jeg bare visste at det blir riktig.  
Rebekka: Du bare viser det?  
Bendik: Nei, jeg visste det. Men det forklarer seg selv egentlig. Hvis ikke du skjønner det så..  
Beate: Det er jo fordi den skal gå, og så skal den snu seg ... (Blir avbrutt av Bendik).  
Rebekka: Du begynte på noe, hva var det du sa?  
Beate: Den går steg så snur den seg og så går den 150 eller noe.  
Bendik: Nei, først er det 100, 150, 100, 150 og så snur den 90 grader fire ganger. Ja, kan vi gå over til oppgave 3. Eller vi har allerede gjort den, vi gjorde den med uhell.  
Rebekka: Ja okei, men da går dere videre til der dere har komt.  
Bendik: Til huset.

*Begynner rett på oppgaven uten å lese oppgaveteksten.*

Bendik: Dette bli ikke så vanskelig, vi gjør bare sånn og sånn.

\*Fjerner blokkene de har brukt til rektangel og legger til blokkene til kvadrat. \*

Beate: Og så må vi bare lage en trekant oppå.  
Bendik: Ja, det er ikke så vanskelig.  
Bendik: Da legger vi til 60 grader.  
Beate: Ja, for vi har jo allerede en firkant, og så må vi bare ha en trekant oppå. Og da blir det et hus.

*Tegner på arket.*

\*Endrer løkken til kvadratet til å gjentas 6 ganger. \*

Bendik: Og da er den der.  
Beate: Og da må vi ta å lage en trekant.  
Bendik: Og da er det denne (**snu mot høyre**) og denne (**steg**).

Beate: Må vi ikke ha 60 grader eller noe.  
Bendik: Jo, 60 grader. Og da må vi ha 100 her (**steg**). Nei vi må ha 90.  
Beate: Og så skal vi ha den andre siden.

\*Legger til «snu 60 grader mot venstre» og «gå 90 steg». \*

Beate: Nei, for vi må få den til å gå ned.  
Bendik: Vi må ta to av den der (**snu**).

\*Legger til en til «snu blokk» slik at det er to blokker med snu mot venstre etter hverandre, begge på 60 grader. Altså 120 grader til sammen. \*  
(Elevene tester koden, og det er et hus, men taket treffer ikke det ene hjørnet til kvadratet).

Beate: Åh, det er irriterende.  
Bendik: Men vi fikk jo huset da.  
Beate: Ja, men det er ikke riktig, for den er jo helt på skeiva.

\*Endrer fra 90 til 95 steg. \*

Beate: Åh!  
*Prøver seg frem.*

Beate: Åh, herligheten da.  
Beate: Kanskje vi bare må flytte på den ene.  
Bendik: Nei, vi må ha flere grader.

\*Endrer litt og litt på gradene. \*

Bendik: Der! Nå fikk vi det til.  
Beate: Halleluja.  
Bendik: Endelig!  
Beate: Okei, mer her står det ...

*Leser oppgaveteksten.*

*Tegner huset på arket.*

*Leser oppgave 5.*

(Bevegelsesblokkene forsvant).

Beate: Okei, så vi skal lage sikksakk.  
Beate: Da må vi ta sånn 1 grad og så 1 og så 1.  
Bendik: 1 grad?  
Beate: Ikke 1 grad, men liksom.  
Bendik: 70 grader mener du.  
Beate: Ja, 70 grader.

\*Legger til bevegelses blokkene steg og snu. \*

(Koden viser «gå 100 steg, snu 15 grader mot høyre, gå 100 steg, snu 15 grader mot venstre»  
to ganger uten løkke).

Beate: Men nå tar du forskjellig turn.  
Bendik: Ja, selvfølgelig.

(Sprite beveger seg, men de har ikke tatt på penn så den tegner ikke).

\*Endrer størrelse på katten. \*

### **Opptak 2.3**

Beate: Hvis vi flytter den (**første snu blokken**) til for eksempel 50.  
Bendik: Ja, men først må vi få katten inn i bilde. Og da må vi bruke denne (**gå til x:y**).

\*Legger inn «gå til x: 294 og y: -211». \*

(Sprite forsvinner ut at bilde).

Beate: Hva skjer, hvorfor går den sånn?  
Beate: Nei, jeg gir opp.  
Beate: Okei, 50.  
Bendik: Så må du gjøre det på alle.

\*Endrer alle snu blokkene til 50 grader. \*

Beate: Det skjer jo ingenting.

Bendik: Åja, han må jo tegne. Han må jo ha tegnegreia si.

Beate: Åja.

*Penn blokkene har forsvunnet og får hjelp til å legge dem inn igjen.*

\*Legger til slett alt og penn på, og tester koden. \*

Bendik: Oi hva! Hva gjør han?

Beate: Oi, nå klarte vi noe.

Bendik: Å, den må bort.

Bendik: Sånn.

Beate: Nå har vi sikksakk. Er ikke det sikksakk?

Bendik: Jo.

Beate: Det er ikke ordentlig sikksakk.

Bendik: Jo, det er sikksakk, bare at vi ikke ser den helt.

Beate: Ja, men det er ikke helt ordentlig sikksakk.

(Flytter på katten – slik at nesten hele streken viser på skjermen).

Beate: Sånn det er sikksakk. Vi ser det er sikksakk.

Bendik: Ja, det er sikksakk nå.

Beate: Men er vi ferdige nå?

Bendik: Ja.

Beate: Eller skal vi lage den? Eller vi kan lage et av mønstrene.

Beate: Hvilken vil du lage?

Beate: Vi kan prøve å lage pyramide.

Bendik: Pyramide?

Beate: Den her. Eller den, den er bare 3.

Bendik: Nei. Først vil jeg endre hvordan han ser ut.

*Bendik endrer på hvordan Sprite ser ut og Beate blir oppgitt over Bendik.*

Beate: Okei, nå skal vi lage en ny en. Hvilken vil du lage? Vil du lage rutenett?  
Bendik: Em, jeg vil egentlig bare ...  
Beate: Eller den?  
Bendik: Nei, den er mye vanskeligere.  
Beate: Da tar vi rutenett.  
Bendik: Rutenett, okei.  
Beate: Den er sikkert enkel.

\*Legger til «gå 100 steg, snu 90 grader mot høyre» to ganger og «gå 100 steg». \*

Beate: Men hvordan skal vi lage rutenett?  
Bendik: Sånn.

*Elevene har verken tatt på penn eller slettet den forrige figuren.*

Beate: Hva skjedde nå?  
Beate: Dette er ikke riktig «Bendik»

*\*Endrer koden. \**

Bendik: Okei, nå skal du få lov til å kode.

*\*Sletter alt. \**

\*Legger inn «gå 50 steg og snu 15 grader mot høyre» – 4 ganger etter hverandre. \*

Beate: Jeg vet ikke. Hvor mange må vi ha, hvis vi skal lage den der, det rutenettet.  
Bendik: Aner ikke.  
Beate: Vi må jo på en måte lage fire firkanter.  
Bendik: Her starter vi. Og vanligvis, når vi lagte firkanten så gikk vi bare sånn. Men nå må vi gå sånn, sånn, sånn, sånn.  
Beate: Eller så kan vi gjøre sånn her, at vi starter her.  
Bendik: Ja, det er jo det vi gjør.  
Beate: Og så gjør vi sånn, sånn, sånn, og så sånn, sånn, sånn.

Beate: Hvis vi starter i det ene hjørnet, det har ikke så mye å si, men hvis vi starter der.

Bendik: Jo, det har ganske mye å si.

Beate: Okei, vi starrer der og så opp, til siden, ned, til siden, og så fortsetter vi til siden, opp der, bam bam bam. Sånn.

\*Bendik endrer på Sprite igjen. \*

Beate: Det her er feil.

\*Sletter hele koden. \*

Beate: Okei, hvordan var det vi lagde firkanten igjen?

\*Legger inn koden for kvadrat, uten å bruke løkke. \*

Beate: Nå har vi en firkant. Så må vi lage fire av de.

Bendik: Fire av de. Det kan du få lov til å gjøre.

Bendik: Vent vi må egentlig bare kopiere den samme der, fire ganger.

Beate: Jeg vet.

Beate: Men jeg vet ikke hvor jeg starta. Jo, det gjør jeg.

(Har lagt inn koden for kvadrat to ganger, uten løkke).

\*Tester koden. \*

Beate: Oi, oi, nå.

Bendik: Du bare kopierte hele greia.

Beate: Nå begynner det å komme. Jeg er så flink.

Bendik: Du er bedre enn meg i hvert fall tipper jeg. Jeg hadde brukt lengre tid.

Beate: Men jeg vet ikke hvor mye ...

Bendik: Vent.

Beate: Jeg vet ikke hvor jeg starta på den igjen.

\*Bendik drar bort koden for kvadrat nr. 2 og legger til en løkke på koden for kvadrat nr. 1. \*

Beate: Ja, det er sant vi kan bare ta «repeat» fire ganger.

Bendik: Fire ganger sånn.

(Sprite tegner to kvadrater ved siden av hverandre).

Beate: Men ja så gjør vi det bare igjen. Legg til en til.

\*Legger til en løkke nedenfor den første, med gjenta fire ganger. \*

(Sprite tegner fremdeles bare 2 kvadrater).

Beate: Nei!

Bendik: What?

Bendik: Okei, hvis vi gjør sånn. Tar den bort (**løkke nr. 2**) og så tar vi 8 (**på den første løkken**)

Beate: Ja!

(Sprite tegner fremdeles bare 2 kvadrater).

Beate: Hæ?!

Bendik: Hæ?

Beate: Det her gir ingen mening.

\*Endrer løkken til gjenta tre ganger. \*

(Sprite tegner ikke lengre to kvadrater, men noe annet).

Bendik: Nei hvordan gikk det fra å være to til det?

Beate: Nei, hva har du gjort nå?

(Sprite tegner to kvadrater igjen uten at elevene har gjort endringer i koden).

Bendik: Å, jeg tror jeg vet hvordan vi gjør det.

Bendik: Vi hadde riktig med fire (**i løkken**).

Beate: Ja, og så må vi bare trykke og se om vi får det.

Bendik: Nei, jeg tror vi må ha åtte.

\*Tester koden mange ganger. Noen ganger blir det to kvadrater, andre ikke. \*

Beate: Den går bare rundt i ring.

Bendik: Nei, se den beveger seg.

Bendik: Okei, jeg tror vi bare har ødelagt den her.

Bendik: Okei, jeg gjør bare sånn.

\*Fjerner løkken og legger den inn i en gjenta for alltid løkke. \*

Bendik: Hæ?

Bendik: Vent, hvis vi gjør sånn.

\*Legger inn en løkke gjenta 10 ganger, inni løkken gjenta for alltid. \*

Beate: Nei.

Bendik: Jeg må ta bort den «forever» greia.

Beate: Åja, er det «forever» der.

\*Fjerner «forever» løkken. \*

Bendik: Hjernen min koker.

Beate: Jeg skjønner ingenting.

\*Endrer på verdien til løkken. \*

Bendik: Det er samme uansett.

Beate: Bare gjør det jeg gjorde å nytt.

Beate: Eller ta 16.

Beate: Nei, ikke sant det funket ikke.

Bendik: Hvis vi bare tar 200 da.

Beate: 200?!

Bendik: Det endret ingenting.



Beate: Nei, jeg gir opp snart.  
Bendik: Jeg tror kanskje det er den «erase all» greia.  
Beate: Nei.  
Bendik: Se!

(Sprite tegner nå to kvadrater oppå hverandre og en på siden).

Beate: Hva skjedde nå?  
Bendik: Vi prøver 300 (**i løkken**)

(Sprite tegner nå tre kvadrater i bunn og en oppå den midterste).

Bendik: Vi kan lage en mer avansert en.

\*Endrer løkken til «gjenta 400 ganger». \*

Beate: Jeg har ikke peiling på hva vi holder på med, vi bare lager masse former.  
Beate: Nei, «Bendik» nå.  
Bendik: Nei, nå må vi ha en sånn «erase all».

(Sprite tegner nå to kvadrater).

Bendik: Så 300 er egentlig jackpoten.

\*Endrer løkken til 300 og tester. \*.

Bendik: Nopp.  
Beate: Hæ, dette gir ingen mening.  
Bendik: Endre til 400 da.  
Beate: Nei. Men vi fikk det jo til i sta. I sta så fikk vi jo sånn her.  
Bendik: Åja, det du gjorde i sta var riktig.  
Beate: Se vi må bare gjøre sånn og legge til en kloss der.  
Bendik: Vent vi hadde glemt de her i «repeat».

\*Flytter på to blokker som står utenfor løkken, «gå 100 steg og snu 90 grader», slik at de er inni løkken. \*

(Sprite tegner et rutenett).

Bendik: Se sånn.

Beate: Å, du skremte meg.

Bendik: Endelig!

Beate: 1000 repeat?

Bendik: Ja, men vi trenger egentlig bare gjøre sånn.

\*Endrer løkken fra «gjenta 1000 ganger» til «gjenta 4 ganger». \*

Bendik: Det var fordi disse er du var utenfor her.

Beate: Åja!

Bendik: Det var derfor det bare ble to. Så jeg flytta de inn.

Beate: Da gir det mening.

Beate: Okei, hvilken skal vi gjøre nå. Nå har vi gjort den.

Bendik: Ja, da prøver vi på det andre kvadratet der.

Beate: Hvilken, den?

Bendik: Nei, kvadrat 1.

Beate: Da må vi gjøre det vi gjorde i sta.

Bendik: Ja. Sletter hele greia.

\*Sletter hele koden\*

Beate: Så da må vi starte med å lage en firkant, må vi ikke?

Beate: Men ta på den der repeat, sånn at de blåe kommer i repeat.

\*Legger inn kode for kvadrat\*

Beate: Prøv å ta repeat 10 ganger.

(Sprite tegner bare et kvadrat)

Beate: Prøv 11, siden den har 11 kanter.

Bendik: 11 kommer ikke til å endre noe, for det er bare det samme her.

Bendik: Vi må bare gjøre det er om og om igjen.

Beate: Så vi må bare gjøre akkurat det samme som vi har gjort alle andre gangene egentlig.

\*Endrer til gjenta 4.\*

Beate: Må det ikke være 3, for det er bare 3 sånne. Se vi må ha 3. Det står repeat 4 fordi der måtte vi ha 4 ruter. Men her er det 3 ruter.

Bendik: Ja jeg vet det. Men vi må ha en som står omvendt retning. Og så må vi bare ta den på repeat.

(Inni løkken står det gå 100, snu høyre 90 grader, snu venstre 90 grader, gå 100. Gjentas 4 ganger)

Beate: Nei.

Bendik: Vent, vi må bare ha en ny repeat.

\*Legger inn løkke som gjentas 10 ganger med de to siste blokkene fra over inni. \*

Beate: Hva skjedde nå?

Bendik: Vi kan lage 3D objekt.

Beate: Nei, det trenger vi ikke.

Bendik: Vi har jo allerede basiclly et 3D-objekt her.

Beate: Hæ?

Bendik: Ser du ikke.

Beate: 3D objekt? Jeg vet ikke hva du prater om engang.

Bendik: Her er 3D. Her har vi en kube på en måte bare at vi mangler noe her.

\*Endrer løkke nr. 2 til å gjenta 4 ganger. \*

(Får 2 kvadrater som møtes i hjørnene)

Bendik: Ja se!

## **Opptak 2.4**

- Rebekka: Prøver dere å bli ferdig med den?  
Bendik: Ja. Kan vi prøve å bli ferdig med alle?  
Rebekka: Ja, hvis dere rekker det.  
Bendik: Ja!  
Beate: Ja, okei.

### *Opptak avbrutt*

- Bendik: Jeg tipper at vi gjør den bare flere ganger.  
Beate: Ja, for nå har vi den formen her.  
Bendik: Ja, men fortsatt så må vi få flyttet den bort hit.  
Beate: Men skal vi ikke bare ta at den må roteres 90 grader en gang til?  
Beate: Vent jeg skal bare sjekke en ting.

*\*Endrer første snu til 180\**

- Bendik: Da går den bare rett opp. 180 grader er bare rett.  
Beate: Men da kan vi gjøre sånn «move».  
Bendik: Ja, men vil vi ikke at den skal være vannrett?  
Bendik: Men hvis vi gjør sånn.

*\*Flytter det som står i løkke nr. 2 inn i løkke nr. 1.\**

- Beate: Nei det er vertfall ikke riktig.  
Bendik: Nei, jeg lagde en stor firkant.

*\*Endrer koden tilbake slik den var\**

*Spør om hvor gruppe 3 er. Får til svar at de er ferdige*

- Bendik: Vi er egentlig også ferdige, men vi vil bare gjøre mer. Eller jeg vil gjøre mer.  
Rebekka: Hvis dere blir ferdige med den så kan dere gå.

Bendik: Men jeg vil gjøre alle oppgavene.  
Beate: Nei, trenger vi ikke.  
Rebekka: Hvis du gjør den ferdig nå, så kan du gå inn på Chromebooken din og prøve litt på den.  
Bendik: Ja. Er det mulig å gjøre det på noe som ikke er skole ment på en måte? Kan jeg bruke det på pc-en min hjemme? Kan jeg gå inn på Scratch på pc-en min hjemme?  
Rebekka: Det kan du, det er helt gratis.  
Bendik: Yes. Men da kommer jeg til å trenge dette arket med meg hjem da.  
Beate: Nei, men det skal de ha.  
Rebekka: Ja, vi trenger det arket.

*Elevene begynner å arbeide med oppgaven igjen*

Beate: Okei, men dette her er jo riktig da. Se nå hvis vi får den til å...  
Bendik: Du kan ikke snu den. Da må vi ha katten bort der.  
Beate: Vent hvis vi flytter på katten, slik at den er der. Og da snur den seg ikke..  
Beate: Vi har det på en måte nå, vi må bare opp sånn så siden går ned. Men den er jo feil vei. Den står jo sånn nå.  
Bendik: Ja, men samme det. Jeg tror jeg vet nå.  
Bendik: Så vi må bare legge til en til av disse **(løkke)**.  
Beate: Oi, nå fikk vi en 3D-figur. Hva skjedde nå?

\*Flytter på katten. \*

\*Legger inn gå 100 og snu 90 i løkke nr. 3.\*

(Får 2 kvadrater som møtes i hjørnene.)

Bendik: Sånn.  
Beate: Kan vi ikke bare si at det er greit det?  
Bendik: Nei.  
Beate: Jo, vi trenger ikke flere nå.  
Bendik: Jeg tenkte liksom at vi hadde den der.  
Beate: Åja, det har vi ikke.  
Bendik: Jeg har mest lyst å bare lage en stjerne egentlig.

\*Legger til ny løkke lik de andre. \*

(Får to kvadrater på siden av hverandre og en oppå.)

Beate: Hvorfor gjorde vi ikke bare det i sta?

Beate: Vent, legg til en til sånn. Så kan vi lage ...

Beate: Går det an å ta det i slow motion, så vi kan se?

\*Endrer på katten igjen. \*

Beate: Men vi lager jo ikke den formen nå, nå lager vi jo et rutenett.

Bendik: Mhm.

Beate: Ja, vi skulle jo lage den pyramide 1.

Beate: Vi må slette en av dem.

Beate: Det irriterer meg at vi ikke kan ta i slowmo. Siden da kan vi ikke se hvem som gjør hva.

\*Endrer katten igjen. \*

*Elev 2 virker oppgitt.*

Beate: Det er ikke det vi skal gjøre. Vi må gjøre det vi skal gjøre.

Beate: Okei, hvis vi tar den her da (**løkken**). Repeat 4 ganger og «move» 100 og turn 90.

Beate: Hva skjedde der?

\*Endrer koden\*

Bendik: Vi må ha annen hver snu og gå på en måte.

(Får rutenett)

Bendik: Hva hvis vi prøver å lage 3x3.

Beate: 3x3? Ja vi kan sikkert prøve det.

Bendik: Først av alt trenger den å være større. Den må ta 200 skritt.

\*Endrer alle til 200 skritt. \*

(Får ikke lengre rutenett.)

Bendik: Den katten klarer ikke gjøre noe. Hva skjer, den var jo perfekt i sta.

Beate: Ja, men hva gjorde du for noe?

Beate: Du tok 200.

Bendik: Ja, 200 skritt.

Beate: 200 skritt er jo veldig mye.

Bendik: Okei, 150 da.

Beate: Eller 300, fordi der er liksom 3x3.

\*Endrer skritt til 150.\*

(Blir mellomrom mellom to av kvadratene.)

Bendik: Hva?

Beate: Vi kan bare si vi er ferdige hvis det er et problem.

Bendik: Nei, vi skal ikke det.

Beate: Vi har jo gjort alle oppgavene.

\*Endrer på koden. \*

(Får to kvadrater ved siden av hverandre.)

Beate: Men da kan vi gjøre pyramiden når vi har den sånn.

Bendik: Ja!

Bendik: Da må vi bare ikke ødelegge dette.

Bendik: Da tar vi 90 grader og ...

Beate: 150 skritt.

Bendik: Nei.

Beate: Jo, for det var det du gjorde på alle de andre.

Bendik: Ja, men da går den helt bort dit.

Beate: Ja, ta halvparten.

Bendik: Ja! Hva er halvparten av 150.

Beate: 75.

Bendik: Plis ikke si vi ødela alt nå.

Beate: Nei, nå funket det!  
Bendik: Ja!  
Beate: Og så må vi opp til 90. Opp 90 grader. Snu og så 90 grader.  
Beate: Ja det er riktig. Og så må du ta 150 skritt.  
Bendik: Sånn, så må vi bare sjekke at den fortsatt fungerer. Nei.  
Beate: Kanskje vi bare må ta 75 skritt.  
Bendik: Ja 75.

\*Tester koden\*

Beate: Hæ? Ja, men se den kommer inni. Den skal jo utpå.  
Bendik: Vent, jeg tror jeg vet hva som er problemet. Vi har to av de her (**løkkene**) det er helt unødvendig.  
Beate: Hvorfor det?

\*Sletter to løkker og prøver. \*

Beate: Hva gjør du nå?  
Bendik: Aner ikke.  
Beate: Nei, det er irriterende.  
Bendik: Ja jeg vet det.  
Beate: Ja nå er den der. Og så opp halvparten av 75. Vi må opp halvparten av 75.  
Bendik: Så gjør jeg bare sånn.

\*Legger til gå 75 skritt. \*

Beate: Hva skjedde nå?  
Bendik: Appen er ødelagt!  
Beate: Den er ikke ødelagt, vi bare får ikke til det vi vil.  
Rebekka: Hæ?  
Bendik: Vi har gjort alt riktig her, men så ender det opp med at den krysser linja. Den skal jo ikke gjøre det.  
Rebekka: Nei, hvordan kan du gjøre slik at den ikke gjør det da?  
Beate: Kode den riktig.



Rebekka: Mhm.  
Bendik: Sånn nå er den tilbake sånn. Og nå skal vi egentlig bare..  
Beate: Nei, nå er den feil igjen.  
Bendik: Men den skal jo ikke gjøre det. Og så blir den bare verre og verre hver gang vi trykker.  
Rebekka: Men hva er det som gjør at den blir sånn da?  
Beate: Er det fordi vi har tatt den der?  
Rebekka: Men hvis du tar bort den nå.

### **Opptak 2.5**

Rebekka: Sier du noe om hvordan firkantene skal stå i forhold til hverandre?  
Rebekka: Hvis du tar bort den (**løkke nr. 2**), så ender den opp der. Hvordan kan du få den til å gå i den firkanten du vil?  
Beate: Det var jo noe i sta, det sto sånn y og x tingen. Der den y og x.

\*Legger den til, men fjerner den igjen. \*

\*Legger på lyd. \*

Beate: Bendik, nå tror jeg det holder.  
Rebekka: Føler dere dere ferdig?  
Beate: Ja.  
Bendik: Nei.  
Beate: Plis, vi er ferdig.

\*Fortsetter med lyd. \*

Rebekka: Jeg tror vi skrur av nå.

### 8.6.2 Intervju

Rebekka: Hvordan syns dere det var å jobbe med slike oppgaver som dette?  
Bendik: Jeg syns det var veldig gøy.

Beate: Jeg synes det var gøy og så er det liksom på en måte noe nytt. Vi har liksom aldri gjort det på skolen. Vi har bare holdt på med Lego og sånne ting.

Bendik: Jeg synes det var litt utfordrende med den siste formen vi prøvde på.

Rebekka: Ja, den oppgave 5. Den var vanskelig?

Bendik: Ja, litt. Form nummer 3.

Rebekka: Det er den pyramiden er det ikke det?

Bendik: Ja.

Rebekka: Ja, pyramide kvadrat 1, den synes dere var vanskelig.

Bendik: Ja, men pyramide kvadrat 2 den er nok vanskeligere. Men den på slutten er nok den vanskeligste.

Rebekka: Ja, dere tror den med de 5 kvadratene er vanskeligst

Bendik: Ja, for da må du kode at pennen går opp, for å få mellomrom på en måte.

Rebekka: Tror dere dere hadde klart å løse den oppgaven hvis dere hadde prøvd?

Beate: Ja, jeg tror kanskje det.

Bendik: Når jeg tenker på det så er kanskje den lettere enn de andre

Rebekka: Ja, kanskje. Hvorfor tenker dere at den kunne vært lettere enn de andre?

Beate: Fordi der må man på måte ikke stable ting. På for eksempel den pyramide 1 der måtte vi ha sånn to kuber sånn at det var en på midten av begge, så da er det enklere når alle er på rad.

Rebekka: Ja, når alle er på rad. Og så er det jo like også, ikke sant?

Beate: Ja.

Rebekka: Så dere synes det var kjekt altså?

Beate: Ja.

Bendik: Ja der var veldig gøy.

Rebekka: Så bra. Men synes dere det var noen oppgaver som var lettere enn andre?

Beate: Ja.

Rebekka: Hvilke oppgaver var det som var greie å løse da?

Beate: Trekanten var veldig enkel.

Bendik: Ja den var veldig lett. Vi klarte å gjøre den med et uhell.

Bendik: Men denne, nummer 1 var nok lettest, å bare lage en firkant.

Beate: Og rutenettet det var også ganske enkelt.

Bendik: Ja det var også ganske lett.

Bendik: Em, nummer 2 var vanskeligere.

Beate: Og huset, det var gøy, men det var også enkelt, men vi fikk sånn at kanten var kanskje  
sånn en millimeter unna for å få det riktig så det ble litt sånn ja.

Rebekka: Ja, jeg skjønner. Så den var akkurat litt feil på en måte.

Beate: Ja.

Bendik: Ja, men vi klarte å få den riktig til slutt da.

Rebekka: Ja, så bra.

Rebekka: Dere snakket jo om at dere synes det var litt annerledes i forhold til andre oppgaver dere har jobbet med. På hvilken måte da?

Beate: Det var liksom sånn at vi skulle lage former, ikke liksom sånn som på Lego så skal vi liksom lage en ting som skal kjøre. Men på Scratch ...

Bendik: At man ikke får instruksjoner i selve appen liksom. Fordi det er noe av det jeg hater. For eksempel tutorials i spill, jeg hopper bare over de, fordi jeg vil utforske selv. Så det er en ting som er veldig bra med Scratch, som jeg likte, at vi fikk det i ark og at vi kunne bestemme..

Beate: Selv hvilke vi ville gjøre.

Bendik: Ja, og det her er jo egentlig ikke tutorial, det her er bare oppgaver som på en måte sånn quest i spill for eksempel. De er liksom gøy. Og jeg synes det var helt fantastisk gøy jeg.

Rebekka: Ja, så bra da. Kunne dere tenke dere å jobbe med flere slike oppgaver?

Bendik: Ja. Det hadde vært veldig gøy.

Beate: Ja, jeg synes det var veldig gøy.

Rebekka: Så dere sier liksom at forskjellen mellom de oppgavene dere har jobbet med tidligere og de dere jobbet med nå, var at dere fikk prøve mer selv.

Beate: Ja, at vi fikk liksom utforske litt hva vi ville gjøre selv. At vi ikke måtte gjøre blablabla, at vi kunne velge litt selv hva vi skulle gjøre.

Bendik: Ja. Jeg har jo kodet på mange andre apper, men der har det jo vanligvis vært sånn at de viser det. Det er en egen nettside som heter kode.org og der går du gjennom en tutorial hver gang du skal kode noe. Og så kan du ikke kode på frihånd i det hele tatt.. Du må kode etter en spesifikk ting. Og når du har gjort ferdig tutorialen på en måte så får du den på nytt neste gang du skal lage det på nytt.

Rebekka: Ja, så du får ikke lov å prøve selv?

Bendik: Nei. Og så er det veldig få ting som faktisk inneholder koding. Og en ting er at noen kode apper så koder du hvordan du skal gå og så når koden er ferdig og

du trykker på nytt så går den ikke tilbake, da fortsetter den bare å gå. Men på kode.org så er det ikke sånn, der er det sånn Minecraft koding på en måte. Og der hopper man helt tilbake til start etter at kodelinjen er ferdig. Det er litt irriterende.

Rebekka: Ja skjønner. Kan jeg spørre dere om hvorfor dere har lyst å gjøre flere slike typer oppgaver?

Beate: Det er liksom gøy å gjøre, for det er noe annerledes enn vanlig skole. Vi sitter jo bak en pult nesten hele tiden og skal skrive matte eller lese eller noe, så det var liksom gøy å gjøre noe annet mer kreativt enn liksom vanlig skolearbeid.

Rebekka: Så bra da. Det liker vi å høre.

Rebekka: Så var det en plass der dere stoppet litt opp i oppgaven? I hvilken som helst oppgave.

Bendik: Rektangelet.

Beate: Den satt vi fast på.

Bendik: Ja den satt vi fast på ganske lenge. Fordi vi tenkte at vi måtte kun endre på de tallene her og ikke selve kodingen. Men så skjønte jeg da etter en god stund at ...

Beate: Vi måtte bytte på nesten alle.

Bendik: Ja, at må endre på ganske mye egentlig, for at det skal bli lettere.

Rebekka: Ja, men hva gjorde dere da når dere stoppet opp?

Bendik: Det jeg egentlig gjorde var at jeg bare tok bort disse to og disse tre. Så brukte jeg egentlig bare gå og snu. Jeg tok gå 150 steg og så snu 90 grader og så gjorde jeg det om og om igjen, 4 ganger.

Beate: Jeg tenkte at vi bare måtte ha lengre, på den firkanten vi hadde laget, at den måtte gå lengre sånn at den ble lengre på to av sidene. Slik at det ble til et rektangel. Det var det jeg tenkte.

Rebekka: Ja, så bra.

Rebekka: Så i den første oppgaven her så står det «Her er en kode, klarer dere å se hvilken figur vi har laget bare ved å se på koden». Hva gjorde dere da?

Bendik: Em, vi gjorde feil. Så vi koda det. Vi trodde at det skulle endre formen på selve figuren som beveger seg.

Beate: Den katten.

Bendik: Ja. Så etter hvert så skjønte vi, når vi var på oppgave 3 så skjønte vi at det da var hvilken form den lagte ikke hvilken figur så det var litt vanskeligere det.

Rebekka: Ja, okei, men kan dere se nå hvilken figur den prøver å lage?

Beate: Ja, firkant.

Rebekka: Ja, hvordan kan dere se at det er den figuren?

Bendik: Du trenger egentlig bare disse tre for å se det.

Rebekka: Ja du kan se det på de tre som står under gjenta 4 ganger.

Bendik: Ja. Nei de to som er under gjenta 4 ganger.

Rebekka: Ja, men hvordan kan dere se det?

Beate: Den skal gå 100 steg, så skal den snu 90 grader. Da blir det en firkant.

Bendik: Det er et rektangel.

Beate: Ja, men det skulle være en firkant.

Rebekka: Ja, kan jeg spørre dere om hva som er forskjellen på en firkant og et rektangel?

Beate: Et rektangel har to og to like sider. Mens en firkant har 4 like sider.

Bendik: Firkanter har like sider.

Bente: Men hva tenker dere det er da?

Bendik: Firkant.

Beate: Firkant.

Bente: Ja, hvis dere skal være mer spesifikke?

Bendik: Et kvadrat.

Bente: Ja, fint.

Rebekka: Da lurte jeg litt på en ting, denne oppgaven her. Hvis dere skulle løst en av disse figurene på ny, hvordan hadde dere løst den da?

Bendik: Rutenett var egentlig ganske lett egentlig. Der brukte vi samme metode som på den første, bare at vi gjentok den, og den tok en annen retning.

Beate: Ja, at den snudde sånn annen hver.

Bendik: Ja, den snudde annen hver gang og det var hvordan vi lagte rutenettet. På slutten hadde vi tenkt å lage en 3x3 rutenett, men det fikk vi ikke til.

Rebekka: Jeg så at dere holdt lenge på med den, det er da pyramide kvadrat 1. Hvordan hadde dere løst den på ny, dersom dere skulle løst den nå?

Beate: Vi starta på en veldig avansert måte, vi starta med liksom firkant, firkant, firkant, ja det ble helt sånn feil. Så vi måtte slette noen ting og så ble det helt feil. Så vi starta på en veldig vanskelig måte for å finne ut hvordan vi skulle gjøre det.

Rebekka: Så hvis dere skulle gjort det på ny, så ville dere prøvd å gjøre det litt lettere fra starten. Er det det du prøver å si?

Beate: Ja.

Rebekka: Okei, hvordan kunne dere ha gjort det lettere da, enn å starte på den avanserte måten?

Beate: Vi kunne kanskje ha starta med bare den, for vi starta med den, og da ble det helt feil. Så hvis vi hadde starta der så hadde det kanskje blitt litt enklere.

Rebekka: Så istedenfor å prøve å lage to kvadrater på en gang, så prøver dere å lage et og et kvadrat.

Beate: Vi prøver å starte med den istedenfor for vi starta med den, så vi skulle starta med den istedenfor.

Rebekka: Ja, så dere ville startet på toppen av pyramiden?

Beate: Ja.

Rebekka: Supert.

Rebekka: Er det noe dere har lært i matematikken fra før av som dere kunne bruke når dere løste disse oppgavene her?

Beate: Ja. Geometri. Sånn former og sånn.

Rebekka: Ja, kan dere være litt mer spesifikke. Hva var det dere akkurat brukte?

Beate: Rektangler og kvadrat og sånne ting.

Rebekka: Hvordan kom dere frem til hvilke grader dere skulle bruke da?

Beate: Matte.

Rebekka: Prøvde dere dere frem eller skjønnte dere det?

Beate: Vi skjønnte det. Jeg går på turn så da må vi vite litt om grader, så der hadde jeg en fordel.

Rebekka: Ja så bra, så du kunne bruke noe du hadde brukt i turnen da egentlig til å løse disse oppgavene?

Beate: Ja.

Rebekka: Er det noen andre ting dere kommer på, som dere kanskje kan ha brukt? Utenom geometrien, eller noe annet i geometrien.

Beate: Jeg kommer ikke på noe akkurat nå.

Bendik: Nei, ikke jeg heller.

Rebekka: Nei.

Rebekka: Er det noen deler i disse oppgavene som ligner på noe som dere har løst før?

Beate: Nei.

Bendik: Nei.

Beate: Vi har aldri gjort sånn type koding før.

Bendik: Eller jo det har vi jo. Men da koda vi figurer til å gå forskjellige plasser og ikke til å lage figurer.

Rebekka: Ja, okei. Så i oppgavene så har dere ikke løst noen lignende oppgaver fra før i matten?

Beate: Eller vi har hatt om kvadrat og sånne ting. Men jeg føler aldri vi har liksom gjort det før. Ikke som jeg kan huske.

Bente: Har dere brukt tid på å tegne ulike former?

Beate: Ja.

Bente: Ja, hvordan er det dere tegner det da i boka?

Beate: Vi har rutenett. Så da bruker vi rutene og linjal og blyant.

Bente: Ja, måler dere grader da?

Beate: Ja, så skriver vi på 90 grader og sånn.

Rebekka: Ja, så dere har kanskje brukt det litt, men dere klarte ikke helt å se sammenhengen.

Beate: Ja.

Bente: Ja, for jeg tenker da har dere jo egentlig tegnet figurene før, men nå tegner dere ved hjelp av programmet. Men dere bruker jo egentlig mye av det samme gjerne.

Rebekka: Var det noe dere brukte tidligere i oppgaven som dere kunne bruke senere i oppgavene i dette heftet?

Beate: Den der. Vi skulle bruke den samme koden.

Bendik: Og i den neste så er det endra på koden her.

Rebekka: Det er da på oppgave 3.

Bendik: Ja der er det endret på gradene.

Rebekka: Så her skulle dere prøve å lage en trekant.

Bendik: Ja. Vent, jeg lagde trekant uten den.

Bente: Ja, det var også målet med oppgaven. Vi har lagt inn litt informasjon der som ikke trengte å være med.

Rebekka: Og nå så du den, veldig bra.

Rebekka: Men de kodene dere brukte litt tidlig i oppgaveheftet, fikk dere bruk for dem i noen av de andre oppgavene dere skulle løse i dette heftet?

Beate: Ja, på den siste når vi skulle lage rutenett og sånne ting.

Rebekka: Ja, da brukte dere det?

Beate: Ja.

Rebekka: Ja, veldig bra. Hvordan så huset deres ut da?

Bendik: Firkant med trekant på toppen.

Rebekka: Ja, har dere laget en firkant og en trekant før i denne oppgaven?

Beate: Ja.

Rebekka: Brukte dere det?

Beate: Ja.

Bendik: Det tok god tid før vi klarte å få trekanten til å faktisk bli festet på taket, istedenfor å være sånn.

Rebekka: Nå viser du at den ble litt skeiv.

Beate: Ja, den ble litt skeiv.

Bendik: Ja, vi brukte god tid på det.

Bente: Det var jo det vanskeligste med den oppgaven, det å få den til å stå oppå.

Rebekka: Er det noen forskjell med hvordan dere jobber med disse oppgavene og hvordan dere har jobbet med oppgaver før. Nå tenker jeg ikke på typer oppgaver, men når dere jobber med oppgavene.

Beate: Jeg skjønnte ikke.

Rebekka: Nei, jeg kan stille spørsmålet på ny på en annen måte. Det vi lurte på, når dere jobber i matten til vanlig, så jobber dere gjerne alene, eller jobber i grupper, kanskje dere prater sammen. Så når dere jobbet med oppgaver i dag, jobbet dere da på en annen måte enn dere har gjort tidligere?

Beate: Ja, i vanlige mattetimer så sitter vi alene og jobber, men nå samarbeider vi. Så det er liksom flere hjerner som kan tenke sammen.

Rebekka: Det var fint det du sa at det var flere hjerner som kunne jobbe sammen. Hvordan syns dere det var da?

Beate: Greit.

Bendik: Ja.

Rebekka: Det var greit ja. Følte dere det ble lettere å løse oppgavene, eller ble det sånn cirka likt?

Beate: Det var egentlig greit.

Bendik: Teknisk sett så er det jo likt fordi man kan ha sånn 100 folk, men som ikke klarer å løse en oppgave selv om en person som har IQ-en bare dobbelt av en,



de kan klare å løse den. Fordi uansett hvor mange du har så er det fortsatt like vanskelig å løse oppgaven. Fordi IQ-en bli jo ikke kombinert.

Rebekka: Det er sant. Men var det greit for dere å prate sammen eller synes dere det var uvandt?

Beate: Det var greit. For i for eksempel kunst og håndverk og sånne ting så pleier vi å samarbeide.

Rebekka: Så dere har gjort det litt før i forskjellige fag?

Beate: Ja.

Rebekka: Hvordan var det å gjøre det i matten da?

Beate: Greit.

Rebekka: Ja, dere sa jo dere kunne tenke dere å jobbe med flere slike oppgaver. Kunne dere tenke dere å jobbe mer sammen?

Beate: Ja.

Bente: Hva er fordelen med å jobbe sammen?

Beate: Flere hjerner som kan tenke sammen, så blir det på en måte mer logisk. For eksempel si at jeg hadde tatt helt feil på noe, så hadde «Elev 1» tatt riktig på noe. Så kan han forklare det til meg så blir det logisk i min hjerne også. Så da er det enklere å jobbe sammen.

Bente: Ja, så dere kan hjelpe hverandre til å komme frem til en bedre forståelse sammen?

Bendik: Ja.

Beate: Ja.

Bente: For det kan jo være du kan noe som «Elev 1» ikke kan, og at «Elev 1» kan noe som du ikke kan. Så kan dere hjelpe hverandre sånn.

Rebekka: Er det noe dere har lyst å dele eller fortelle? Om hvilken opplevelse dere hadde eller om det er noe annet, eller om det er noe som kunne vært bedre med oppgaver for eksempel.

Beate: Vet ikke.

Bendik: Vet ikke. Kodinga feilet på en måte, fordi vi hadde kodet riktig, men så bare ... Vi hadde koda to firkanter, så istedenfor å gjøre det, den fungerte jo første gangen, men neste gang så bare ... Vi hadde ikke endra noe på koden.

Rebekka: Men hvorfor tror du det skjedde?

Beate: For vi hadde koda feil.

Rebekka: På hvilken måte da tror dere? Hvis den startet på en plass og en gang lagte den riktig figur og en annen gang gjorde den ikke det. Hva tror dere kan ha skjedd da?

Bendik: Den bytter også plass på hvor den blir laget.

Rebekka: Ja, der sier du noe lurt. Den bytter plass på hvor den blir laget. Hvor er Sprite oppi alt dette her? Hvor er det han står?

Beate: Han står på kanten. Eller sånn på siste stedet den tegnet, tror jeg, eller jeg vet ikke.

Bendik: Så det er egentlig ganske viktig at du får den på samme plass som den startet.

Rebekka: Ja, tror du det kan ha noe å si for hvordan den figuren ble? I hvilken vinkel Sprite sto eller om han sto på hodet eller.

Bendik: Han sto liksom sånn på midten av arket, eller selve kode greien. Han er i midten så lager han de to firkantene så er han fremdeles på midten. Men neste gang vi gjør det, og vi har ikke endre noe på kodingen, så bare går den helt vilt.

Rebekka: Ja, det er nok noe i kodingen der som sikkert har gått litt feil, i forhold til hvilken vinkel Sprite skal snu og litt sånne ting. Så det kan være det som har skjedd.

Bente: Så kan det være at dere har manglet den som heter gå til x: 0 y: 0, hadde dere med den? Og pek i retning 90?

Beate: Jeg tror vi har hatt med dem. Hvis jeg husker riktig.

Bendik: Mm, nei.

Bente: Ja for hvis de ikke er mer så stopper Sprite der han sluttet i forrige figur. Der står han, og så lager han neste figur fra det stedet der.

Bendik: Ja, for vi glemte den.

Bente: Ja, så da kan det være at han står og ser den veien istedenfor til siden. At han ser oppover istedenfor til høyre da.

Rebekka: Og når du da har klikket på den mange ganger så kanskje den ender opp på riktig plass igjen og så lager den riktig figur. Så det tror vi er derfor det har skjedd.

Rebekka: Har dere noe mer dere har lyst å legge til?

Beate: Nei.

Bendik: Nei.

Rebekka: Da sier vi tusen takk for hjelpen.