

Programmering i naturfag 1

*Elevs forståelse, læringsmuligheter og
motivasjon i møte med en
programmeringsoppgave om halveringstid*

Einar Løhne Iversen



Masteroppgave i fysikkdidaktikk
Lektorprogrammet
30 studiepoeng

Fysisk institutt
Det matematisk-naturvitenskapelige universitet

UNIVERSITETET I OSLO

Januar 2022

Programmering i naturfag 1

Elevers forståelse, læringsmuligheter og motivasjon i møte med en programmeringsoppgave om halveringstid

Forfatter:

Einar Løhne Iversen

Veileder:

Ellen Karoline Henriksen

Masteroppgave

Lektorprogrammet

30 studiepoeng

Fysisk institutt

Det matematisk- naturvitenskapelige fakultet

UNIVERSITETET I OSLO

Januar 2022

© Einar Løhne Iversen

2022

Programmering i naturfag 1

Einar Løhne Iversen

<http://www.duo.uio.no/>

Trykk: Reprosentralen, Universitetet i Oslo

Sammendrag:

Programmering er nå kommet inn i den norske skolen, og til det kommer det spørsmål rundt hvordan det skal innlemmes i de ulike fagene. I naturfaget legger læreplanen opp til at programmering skal benyttes som hjelp og verktøy til modellering. Det er derfor interessant å se nærmere på hvordan bruk av programmering påvirker elevenes læring, motivasjon og opplevelse av naturfag. Problemstillingen til oppgaven er: *«Hvordan kan en programmeringsaktivitet påvirke elevenes forståelse, muligheter og motivasjon innenfor naturfag?»*

Utgangspunktet for å kunne se nærmere på dette er et undervisningsopplegg for naturfag i den videregående skole. Temaet for opplegget er knyttet opp mot radioaktivitet og har benyttet programmering for å simulere henfall og halveringstid. Opplegget er gjennomført i fire forskjellige klasser, og data er samlet fra disse gjennom tre ulike metoder. Det er gjennomført fokusgruppeintervju etter gjennomført opplegg i hver klasse, elevene har gjennomført en spørreundersøkelse i slutten av økten der de hadde opplegget, og elevenes besvarelser til opplegget er samlet inn. Det er gjort kvalitativ analyse av dette datamaterialet.

For å belyse problemstillingen min har jeg tatt utgangspunkt i tre forskjellige elementer knyttet til programmering: Hvordan påvirkes modellering av programmering, hvilke muligheter/utfordringer møter elevene gjennom programmering, og hvor relevant opplever elevene programmering. Jeg argumenterer for at programmering er et nyttig verktøy som har potensiale til å bidra positivt til elevers forståelse og motivasjon innenfor naturfag, så lenge man tar hensyn til de potensielle utfordringene som kommer på kjøpet. Resultatene mine indikerer at:

- Programmering gir elever mulighet til å effektivt jobbe med modeller, både de som de kjenner fra før og også andre mer avanserte modeller, så lenge vi klarer å gi elevene nok støtte og hjelp til å overkomme eventuelle utfordringer de møter i programmeringen.
- Elever kan oppleve programmering som nyttig og relevant, men vi må hjelpe dem til å oppnå dette. Elevene er ofte ikke klar over hva programmering er nyttig for, eller hvordan det er relevant for dem.
- Det er stor kontrast mellom de som har egen programmeringserfaring, og de som kun møter det gjennom skolen. Det blir viktig å hjelpe elevene som sliter med programmering opp på et nivå der de kan oppleve fordelene ved programmering.

Abstract:

Programming has now entered the Norwegian school system, and this raises a lot of questions about how it should be implemented in the different subjects. In science the curriculum aims for programming to be used as a tool for modelling. It is therefore interesting to look closer at how use of programming effects the pupils teaching, motivation, and experience of science. The main question for this thesis is: *“How can a programming activity affect the pupils understanding, possibilities and motivation within science?”*

The starting point to examine this closer is a teaching exercise for science in the upper secondary school. The theme for the exercise relates to radioactivity and uses programming to simulate decay and half-life. The exercise has been completed by four different classes, and data has been gathered through three different methods. Focus-group interviews have been carried out after each class had completed the exercise, the pupils have answered a questionnaire towards the end of the session they did the exercise, and the students answer-sheet have been collected. Qualitative analysis has then been carried out on this data.

To examine my main question I have looked at it from three different perspectives related to programming: How does programming effect modelling, what possibilities/challenges do the pupils encounter through programming, and how relevant do the pupils find programming. I argue that programming is a useful tool that has the potential to positively effect pupils understanding and motivation within science, as long as one takes into account the potential challenges it brings with it. My results indicate that:

- Programming gives pupils the possibility to effectively work with models, both ones they know already and other more advanced models, as long as we manage to give the pupils adequate support and help to overcome potential challenges they face during the programming.
- Pupils can experience programming as useful and relevant, but we have to help them to achieve this. Pupils often aren't aware of what programming is useful for, or how it is relevant for them.
- The difference between those who have programming experience of their own, and those who encounter it only through the school is large. It will be important to help the pupils struggling with programming reach a level where they can experience the advantages that programming can bring.

Forord:

Masteroppgaven min er slutten på en fem og et halvt år lang tid som lektorstudent ved UiO. I løpet av disse årene har jeg lært mye om fysikk, matematikk, elever og didaktikk. Denne oppgaven er på mange måter summen av det jeg har lært på universitetet og oppdaget om meg selv der. Det jeg har lært om fysikk, didaktikk og pedagogikk har vært med på å forme min oppgave. Jeg håper den kan bli satt pris på av de som leser den. Uansett så har jeg lært mye under arbeidet, både når jeg har lest, analysert og skrevet, som jeg kan få bruk for når jeg nå skal ut og jobbe som lærer.

Under arbeidet med denne oppgaven har jeg fått mye hjelp, og jeg vil særlig rette en stor takk til min veileder Ellen Karoline Henriksen, som har hjulpet meg gjennom hele prosessen. Jeg har fått råd, hjelp og god støtte fra henne, og hun har vært tålmodig og forståelsesfull. Uten hennes hjelp hadde ikke denne oppgaven blitt til.

Jeg vil også takke Cathrine W. Tellefsen, som er forfatter av det underliggende opplegget jeg baserte mitt undervisningsopplegg på. Jeg hadde også en fin samtale med henne da jeg startet på mitt masterarbeid, og jeg fikk litt teknisk bistand underveis. Jeg vil også takke alle elever som har vært med og deltatt i denne studien, og alle lærerne som villig stilte sine klasser til min disposisjon. Det har vært veldig hyggelig å komme inn i naturfagsklasserom rundt omkring, og opplevelsen har vært udelt positiv.

Til slutt vil jeg si takk til alle ved skolelaboratoriet på Fysisk institutt, der jeg har holdt til under arbeidet. Dette er alltid et hyggelig sted å komme innom, enten det er for sosialt samvær, faglig hjelp, eller bare for å slå av en prat med likesinnede. Deres innsats og glede for fysikk og undervisning har vært en inspirasjon for meg.

Innhold

1) Introduksjon	1
1.1 Bakgrunn og formål	1
1.2 Problemstilling og forskningsspørsmål	2
2) Teori og tidligere forskning	3
2.1 Modellering og radioaktivitet	3
2.2 Programmeringen og utfordringer elevene møter i den	4
2.3 Muligheter elevene får med programmering	6
2.4 Motivasjon og oppgavens verdi og utforming	6
2.5 Relevans	9
2.6 Relevans kontra interesse	10
2.7 Meningsfullhet	11
2.8 Rammeverk for relevans	12
2.9 Tidligere forskning	13
3) Metode	15
3.1 Datainnsamling	15
3.2 Utvalg	15
3.3 Spørreundersøkelse	16
3.4 Gjennomføring av spørreundersøkelse	17
3.5 Fokusgruppeintervju	18
3.6 Gjennomføring av fokusgruppeintervju	18
3.7 Elevbesvarelser	20
4) Undervisningsopplegg i naturfag med bruk av programmering	21
4.1 Undervisningsopplegg: Programmeringsoppgave om halveringstid	21
4.2 Trinket og Aunivers	23
5) Analyse	25
5.1 Transkribering og analyse av intervju	25
5.2 Spørreundersøkelse	26
5.3 Gjennomgang av oppgaveark	27
5.4 Etske hensyn	27
5.5 Troverdighet og generalisering	28
6) Resultater	31
6.1 Spørreundersøkelse	31
6.2 Oppgaveark	35
6.3 Fokusgruppeintervju: Kort oversikt og kobling mellom modell og fenomen:	38
i) Fokusgruppe 1	38

ii) Fokusgruppe 2.....	39
iii) Fokusgruppe 3.....	41
iv) Fokusgruppe 4.....	42
6.4 Motivasjon og relevans	43
6.5 Utfordringer	48
6.6 Opplegget og tidligere undervisning.....	50
7) Diskusjon	55
7.1 Modellen for halveringstid.....	55
7.2 Programmeringsutfordringer.....	57
7.3 Muligheter	59
7.4 Relevans og motivasjon	62
7.5 Tilpasninger av opplegg.....	67
7.6 Implikasjoner	69
7.7 Begrensninger ved denne oppgaven	71
8) Konklusjon/oppsummering	73
9) Kilder/referanser	75

Vedlegg A, oppgaveark:	77
Vedlegg B, spørreundersøkelse:	80
Vedlegg C, Intervjuguide:	82
Vedlegg D, Infoskriv og samtykkeskjema utdelt til klassene:	84
Vedlegg E, Infoskriv og samtykkeskjema til fokusgruppeintervjuene:	86

Liste over figurer og tabeller:

Figur 1: Rammeverk for relevans utarbeidet av Stuckey et al.	side 12
Figur 2: Bilde fra det første vinduet elevene møter i nettmodulen fra Aschehoug	side 21
Figur 3: Bilde fra Trinket-vinduet til den første programmeringsoppgaven elevene møter	side 24
Figur 4: Diagram over elevenes holdninger til programmering	side 32
Figur 5: Diagram over hvor elevene trodde programmering kunne være nyttig	side 33
Figur 6: Diagram over elevenes erfaring med programmering i naturfag	side 34
Figur 7: Sektordiagram over elevens svar på oppgave 8	side 35
Figur 8: Sektordiagram over hvilke typer feil elevene gjorde på oppgave 8	side 37
Tabell 1: Liste over koder brukt under analyse	side 25
Tabell 2: Eksempler på «riktige» og «gale» elevsvar på oppgave 8	side 36

1) Introduksjon

1.1 Bakgrunn og formål

Programmering er kommet inn i norske læreplaner med fagfornyelsen, og flere fag har læreplaner med kompetansemål som omhandler programmering. Mange instanser, slik som senter for IKT, mener kunnskapsdomenet programmering kan ha en plass i norsk skole (Sevik, 2016). Programmering skal med utrulling av de nye læreplanene sakte innføres gjennom hele skoleløpet fra barneskolen til og med videregående skole. I naturfag skal programmering inn som en naturlig del av kjerneelementet teknologi (Regjeringen, 2018) og elevene skal «vurdere og lage programmer som modellerer naturfaglige fenomener» (Utdanningsdirektoratet, 2020).

Det argumenteres nå for at programmering er en viktig del av allmenndannelsen på lik linje med matematikk, og at man med programmering kan undervise i temaer og problemstillinger der elevenes kunnskapsnivå innen matematikk tidligere har lagt begrensninger (Haraldsrud, & Tellefsen, 2018). For de elevene som nå begynner med de nye læreplanene vil dette for mange være første gang de opplever programmering. Dette kommer gjerne med noen ekstra utfordringer siden elevene ikke kjenner godt til programmering fra før. En annen utfordring elevene vil møte er å kombinere programmering, fysikk og matematikk som kan medføre stor belastning, ikke minst siden hver av de tre tingene er vanskelige nok i seg selv (Taub, Armoni, Bagno & Ben-Ari, 2015). Da naturfag har mange likhetstrekk med fysikk er det naturlig å tro at dette gjelder for naturfag også. Det er derfor av interesse å se nærmere på hvordan elevene som nettopp har begynt med programmering opplever dette, hvilke muligheter de ser i programmering, og hvilke utfordringer de treffer på i møte med programmering og dens integrering i naturfaget.

Oppgavens formål har vært å se nøyere på programmering i naturfag, og hvordan bruk av det påvirker elevenes forståelse og motivasjon. Helt konkret har oppgaven tatt for seg et opplegg fra de nye bøkene i naturfag som benytter seg av programmering, og jeg har sett på hvorledes elevene jobber med og erfarer dette opplegget og hvordan det eventuelt skiller seg fra «tradisjonell» undervisning. Det har også blitt undersøkt om elevene erfarte programmeringen som relevant og motiverende.

Opplegget med programmering er basert på et tidligere opplegg der naturfagslæreren benytter terningkast for å simulere sannsynligheten for radioaktivt henfall. Gjennom store talls lov og nok terningkast vil elevene få mulighet til å kunne se at halveringstid og sannsynlighet for henfall innen en gitt tidsperiode er koblet til hverandre.

Modellering generelt anses som et kraftfullt verktøy innen fysikk (Angell, Kind & Henriksen, 2008), men også innen andre naturvitenskaper. Siden modellen som benyttes i dette forsøket er mye utprøvd i norske klasserom er det interessant å se om programmering tilfører noe nytt til øvelsen, og om dette styrker læringsutbyttet. Det er også interessant hvordan elevene opplever denne øvelsen som en programmeringsoppgave.

1.2 Problemstilling og forskningsspørsmål

Oppgaven ønsker å se på programmering og elevenes opplevelse av bruk av programmering til å forklare et naturfenomen. Problemstillingen vil se på hvordan programmering påvirker elevenes forståelse av modellen de jobber med. Det er også ønskelig å se om de klarer å benytte seg av mulighetene programmering som verktøy gir dem. Til slutt er det også tenkt å se på hvordan elevene opplever å jobbe med programmering, og om de finner det motiverende og relevant.

Problemstillingen min blir derfor:

Hvordan kan en programmeringsaktivitet påvirke elevenes forståelse, muligheter og motivasjon innenfor naturfag?

For å svare på dette vil jeg ha følgende forskningsspørsmål.

- 1: Hvordan påvirker programmeringen elevenes forståelse rundt koblingen mellom modellen de jobber med og naturfenomenet som skal forklares?
- 2: Hvilke muligheter får elevene når programmering benyttes i naturfag, og hvordan sikrer vi at elevene får utnyttet disse mulighetene?
- 3: Hvordan påvirker programmering elevenes motivasjon, og hvordan kan vi bidra til at de opplever programmering som relevant?

2) Teori og tidligere forskning

2.1 Modellering og radioaktivitet

«Modellering er en prosess for å finne forenklet representasjon av et fenomen i virkeligheten, altså en modell» (Haraldsrud, Sveinsson & Løvold, 2020, s. 190). Modeller er et kjempeviktig verktøy i naturvitenskapen da det gir oss muligheten til å knytte matematikk opp mot naturvitenskapelige fenomen og observasjoner, og også muligheten til å forutse naturfenomener. Matematiske fenomener elevene blir presentert for i naturvitenskapelige fag kommer alle fra modeller som er laget for å forstå virkeligheten. Programmering gir elevene bedre anledning til å prøve ut modellene selv. Ved arbeid med radioaktivitet og læren om dette er det mulig å se på dette i både et makro- og et mikroperspektiv. Mange ting som angår oss direkte vil være mest synlige i makroperspektivet, men for å forstå tema fullt ut kan det være nyttig med en forståelse på mikronivå, og det er her modellering kommer inn (Millar, Klaassen & Eijkelhof, 1990). Dette kommer også frem i teksten til Wilensky, Brady og Horn hvor de benytter seg av såkalt «agentbasert modellering», hvor mange mindre «agenter» i et større system utstyres med visse egenskaper. Deretter modelleres disse agentene i det store systemet. Man kan dermed observere resultatene på makronivå, mens man jobber med agentene på mikronivå. Agentbasert modellering gir dermed mulighet for å bygge en intuitiv forståelse om individuelle agenter som handler på mikronivå, og hvordan dette påvirker mekanismer som kommer frem på makronivå (Wilensky, Brady & Horn, 2014).

Modeller er likevel ikke helt uten sine egne utfordringer. Elever har ofte misoppfatninger om hva ordet modell betyr, og de sliter ofte med å benytte dem for å analysere fenomener og løse problemer (Etkina, Warren & Gentile, 2006). Modeller har også tydelige begrensninger som gjør at de lett kan oppleves som gale eller mangelfulle. Det er også lett at misoppfatninger oppstår når elever tror at ting ser ut som modellen sin i virkeligheten, for eksempel hvordan flere mennesker tror at atomet ser ut som Bohr sin modell i virkeligheten (Haraldsrud et al., 2020). For å forhindre slike misoppfatninger kan man gjøre eleven bevisst på hva en modell skal fremstille, og hva den ikke skal fremstille, slik at elevene kan forstå både mulighetene og begrensningene iboende i modellen de ser på. Ved hjelp av programmering, som fjerner mange matematiske begrensninger når vi jobber med modeller, vil det kunne være lettere å flytte perspektivet over på analyse, forståelse og drøfting av modeller (Haraldsrud et al., 2020).

2.2 Programmeringen og utfordringer elevene møter i den

Når man programmerer er det mye som skjer på en gang. Sørby og Angell tar i deres tekst for seg at studenter ofte jobber i ulike «arbeidsmodus», og de peker på tre bestemte moduser som studentene ofte befinner seg i. Disse tre kaller de for «Fysikkmodus», «mattematikkmodus» og «programmeringsmodus». (Sørby & Angell, 2012) Vi kan for naturfag substituere «fysikkmodus» med «naturfagsmodus». I programmeringsmodusen observerte Sørby og Angell at studentene i større grad benyttet seg av en prøve-og-feile-strategi hvor de lettere hang seg opp i koden, og lette etter feil der, fremfor å tenke gjennom den bakenforliggende fysikken. I denne modusen gikk også studentene lettere i gang med å programmere, og bruke resultatet til å bekrefte om de er på rett spor eller ikke. Det kan være positivt at man i programmeringsmodus er mindre redd for å gjøre feil, og at terskelen for å komme i gang er senket. Det er likevel rom for å diskutere om dette fører til at studentene tenker mindre gjennom hva de faktisk gjør, og hva koden betyr. De risikerer også å ende opp med mer rotete og mindre oversiktlig kode, som er vanskeligere å tolke og benytte seg av senere.

En annen utfordring med programmering er også kognitiv overbelastning (cognitive load). Taub et. al. skriver i sin tekst fra 2015 at programmering kan føre til uønsket kognitiv belastning av både den iboende (intrinsic) og utenforliggende (extraneous) typen (Taub et al., 2015). Som før nevnt er programmering, matematikk og fysikk utfordrende i seg selv og kombinasjonen av dem kan bli ekstra vanskelig, og kan føre til ekstra utenforliggende belastning, når de lærende må ta hensyn til tre ulike disipliner med ulike måter å gjøre ting på samtidig. Taub et. al. trekker det derfor frem som viktig at lærerne viser frem koblingen mellom programmeringen og fysikken, og viser hvordan ulike deler av programmeringsprosessen representerer et ulikt aspekt ved det fysiske fenomenet som modelleres. For å gjøre kombinasjonen av programmering og andre fagområder noe enklere foreslår de også å gi elevene mer opplæring i fysikk der noe av denne kan foregå uten programmering som kan distrahere. Likeledes foreslår de å gi grunnleggende opplæring i programmering uten å blande inn fysikk, eller andre fagområder. Vi må likevel ikke glemme at målet er at elevene skal kunne benytte programmering i kombinasjon med andre fag som Haraldsrud og Tellefsen påpeker i sin tekst Programmering – for fysikkens skyld fra 2018.

«Naturvitenskapelig programmering handler om mer enn det tekniske i å lage et dataprogram. Det handler om å bearbeide en problemstilling så den lar seg løse ved programmering, utvikle

algoritmisk tenkning og bruke numeriske metoder for å studere fysikkens vidunderlige verden.» (Haraldsrud & Tellefsen, 2018, s. 75).

Fra ideen om å lære programmering på egenhånd før man forsøker å benytte denne i andre fagfelt kan man trekke paralleller til Vygotskij sin ide om den proksimale utviklingssonen. I denne teorien benytter vi oss av eksisterende kunnskap når vi skal tilegne oss ny kunnskap. Det som ligger innenfor «rekkevidde» for oss er det som befinner seg i vår utviklingssone. (Säljö, 2018). For å utvide vår utviklingssone må vi tilegne oss kunnskap, slik at vi kan nå nye nivåer. For å hjelpe oss å få denne kunnskapen kan «den mer kompetente» (lærer, foreldre, etc.) drive stilas-bygging (scaffolding), som hjelper den lærende til å oppnå en høyere forståelse enn de ville klart på egenhånd. I sin masteroppgave om bruk av numerisk fysikk og programmering hos førsteårsstudenter i fysikk fant Sørby at disse ofte støtte på problemer de ikke klarte å løse selv, og behovet for en lærer virket stort. Samtidig observerte han at de ofte trengte lite input for å komme til korrekt metode/løsning, og han spekulerer i om det ikke kan være andre måter å gi elevene dette på som ikke involverer direkte hjelp fra læreren, men heller gjennom oppgaven i seg selv (Sørby, 2010).

I tillegg til utfordringen med kombinasjonen av fagområder som kommer av å benytte programmering innen et annet fagfelt, så er programmering i seg selv også et utfordrende fag. Få elever opplever det som lett å lære å programmere. Det er flere teorier om hva som ligger bak dette, men Jenkins trekker i sin tekst «The difficulty of learning to program» frem to kognitive faktorer han mener spiller en viktig rolle. Den første av disse er læringsmetoder, hvor han ser på de to hovedtypene dybdelæring (deep) og overflatelæring (surface). Jenkins mener at programmering er avhengig av begge måtene å lære på da vi trenger dypere forståelse for å kunne anvende programmering til å løse nye, ukjente problemer, men også at mer overfladisk læring kan være nyttig for å få med seg alle detaljer av syntaks og lignende (Jenkins, 2002). Denne kombinasjonen er noe som de fleste elevene ikke har vært borti før, og bidrar til utfordringen ved å lære programmering.

Den andre kognitive faktoren Jenkins peker på er motivasjon, og hvordan ulike elever har ulike former for motivasjon. Han trekker frem at noen har en indre motivasjon, mens andre er motivert av mer eksterne ting. Ikke overraskende er det ofte mer utfordrende å jobbe med programmering for de med ytre faktorer som motivasjon, kontra de som har en indre motivasjon for programmering. Likevel er det viktig ikke å overdrive påvirkningen av

motivasjon, og Jenkins skriver at selv de som er ytre motiverte pleier å beholde denne motivasjonen gjennom opplæring i programmering.

2.3 Muligheter elevene får med programmering

Med programmering får elevene flere nye muligheter som de ikke hadde før. «Ved å bruke programmering gjør vi derfor tilgjengelig en rekke nye faglige problemstillinger fordi vi ikke er begrenset av hvilke ligninger som lar seg løse med penn og papir» (Haraldsrud et al., 2020, s. 14). Programmeringen lar elevene generalisere løsningsstrategier i større grad enn tidligere, og nye fenomener kan utforskes. Disse mulighetene kan benyttes for å gjøre modellering mer realistisk. I fysikken kan man for eksempel ta med luftmotstand i mange beregninger der man tidligere måtte se bort i fra denne, da elevene ikke hadde matematikkunnskaper til å ta med dette. «Programmering gjør store oppgaver mindre og umulige oppgaver mulig» (Haraldsrud et al., 2020, s. 162)

Elevene får også bedre anledning til å prøve og feile, og med programmering er det lettere å sjekke uten å gjøre en stor utregning på nytt (Haraldsrud et al., 2020). Dette gir også elevene anledning til lettere å eksperimentere, da de enkelt kan bytte ut en variabel med en annen. Det blir også muligheter for å spare tid, da utregninger går raskere for en datamaskin enn et menneske.

Elevene får også flere tilnærminger til fagstoffet. Selv om programmering fremmer modellering, og løsninger av problemer numerisk, så vil jo ikke det å løse noe analytisk være verdiløst (Haraldsrud et al., 2020). Datamaskinen og numeriske metoder vil heller være et ekstra verktøy elevene får som gir dem mulighet til å betrakte ting på flere måter.

2.4 Motivasjon og oppgavens verdi og utforming

Motivasjon er gjerne definert som en *indre tilstand som forårsaker, styrer og opprettholder atferd* (Woolfolk, 2004). Det er mye forskjellig som påvirker motivasjonen, og det kan komme fra flere kilder. En enkel måte å definere motivasjon er som før anført i indre og ytre motivasjon. «Indre motivasjon er den naturlige tendensen til å oppsøke og overvinne

utfordringer når vi forfølger personlige interesser og utvikler våre ferdigheter» (Deci & Ryan, 1985; Reeve, 1996; Ryan & Deci, 2000, sitert av Woolfolk, 2004, s. 275). Ved indre motivasjon er aktiviteten i seg selv belønning nok og vi trenger hverken annen belønning eller straff. Ytre motivasjon handler om når vi gjør noe ikke for oppgaven i seg selv, men for å oppnå en gevinst som ikke har nevneverdig med selve oppgaven å gjøre, det kan være karakter, unngå straff, eller å gjøre noen blide. Ofte kan aktiviteter vi gjør være mer kompliserte enn å kun være rent indre eller ytre motiverte, og man kan snakke om en linje som går fra fullt selvbestemte aktiviteter, til aktiviteter som er fullt ut bestemt av alle (Woolfolk, 2004).

Hvordan en oppgave er utformet kan ha mye å si for hvordan den påvirker elevers motivasjon. Vi kan se for oss tre forskjellige verdier for elevene; prestasjonsverdi, interesseverdi og nytte (Woolfolk, 2004). Prestasjonsverdi omhandler hvor viktig det er å gjøre det godt på en oppgave. Dette vil henge sammen med et individs behov og betydningen av suksess. En person som ønsker å fremstå smart og som forbinder det å fremstå smart med det å få en god karakter vil vektlegge en prøve mer og tilegne den høyere prestasjonsverdi. Interesseverdi, som relaterer seg til gleden man får gjennom selve aktiviteten, vil variere fra person til person. Noen liker å lære nye ting, mens andre blir mer trigget når de må løse en gåte. Nytteverdi henger sammen med om noe hjelper oss å nå et mål, både kort- og langsiktig. Vi kan også se for oss en fjerde komponent som vi kan knytte mot verdien til en oppgave, nemlig kostnad. Kostnad handler om de negative aspektene ved å delta i en oppgave, som prestasjonsangst eller stress, men også i forhold til innsatsen som må nedlegges og tiden som brukes som kunne vært brukt til noe annet (Eccles, Wigfield, 2002). Som man kan se vil større kostnad trekke verdien av en oppgave nedover, og det blir opp til de tre andre verdiene å sørge for en positiv netto ved å delta i en oppgave.

Et konsept som kan bidra til å øke en oppgaves verdi er autentiske oppgaver. «En autentisk oppgave har en forbindelse med de reelle problemene og situasjonene som elevene kommer til å møte utenfor skolen, både nå og i fremtida» (Woolfolk, 2004, s. 301). Om elever blir møtt med oppgaver de ikke ser nytteverdien i, eller føler at den eneste gangen de får bruk for en definisjon de lærer er under en kommende prøve, kan dette resultere i lav motivasjon for å lære. Dersom oppgavene derimot er autentiske og elevene lettere kan se nytteverdien er det mer sannsynlig at elevene opplever oppgavene som meningsfulle og interessante.

Utfordringen her er selvfølgelig at det kan være vanskeligere å lage en autentisk oppgave fremfor en generisk oppgave.

Brown og Wilson anbefaler også å bruke autentiske oppgaver, men de er mer opptatt av at elevene får en kontekst til oppgavene i det hele tatt, fremfor om denne kan relateres til elevenes liv utenom skolen. Det viktigste for dem er at en kontekst flytter oppgavene fra «vanskelig og kjedelig» til «lett og engasjerende» (Brown & Wilson, 2018). De trekker også frem at man skal passe på så eventuell kontekst ikke utelukker noen elever, og at det noen finner spennende kan gi dårlige assosiasjoner for andre. De kommer også med innspill om at man kan gi elever en del ferdig kode når man starter, slik at de starter nærmere målet og lettere kan komme seg dit.

Sørby tar også i sin tekst opp hvordan selve oppgaven kan utformes for å best hjelpe elevene. Han foreslår at oppgaven kan lages på en slik måte at elevene hjelpes eller tvinges til å sette seg inn i hva som er målet for oppgaven, og dermed trenes i å lage en plan for modelleringsøkten (Sørby, 2010). «Oppgaveteksten burde ikke være redd for å utdype begreper, presisere sammenhenger og å veilede studentene i riktig retning" (Sørby, 2010, s. 189). Oppgavens formål bør jo primært være læring og ikke testing. Oppgaven kan også gjerne være stegvis og fungere som eksempel for elevene på hvordan modellering bør foregå.

Et annet viktig konsept innenfor læring og motivasjon dreier seg om eierskap. Når elevene selv er aktive agenter for egen læring så opplever de et eierskap til kunnskapen sin, og man kan se på eierskap som et viktig element for å konstruere kunnskap (Rainer & Matthews, 2002). Milner-Bolotnin fant i sin studie fra 2001 at eierskapet studenter følte ovenfor et prosjekt bidro til økt mestringsorientering hos studentene, som igjen førte til forbedret læring (Milner-Bolotnin, 2001, sitert av Margareta Enghag, 2006) Hun definerte at en lærende opplevde høy grad av eierskap dersom han/hun:

- Fant personlig verdi i å jobbe med prosjektet, forsto hvordan kunnskapen kunne være nyttig, og kunne knytte denne nye kunnskapen sammen med tidligere kunnskap.
- Opplever å være i kontroll over læringsprosessen gjennom å ta beslutninger og være proaktiv.
- Tar ansvar for læringsprosessen og læringsprosjektet.

Vi kan se at denne måten å se eierskap på knytter seg opp mot noen andre elementer som er tatt frem tidligere, som nytteverdi, og knytte ny kunnskap opp mot gammel (utviklingssone), men det er også noen nye konsepter her som å oppleve å være i kontroll og være aktiv.

Følelsen av kontroll kan ta mange former, men primært handler det om å føle at man forstår hva man gjør, og har kontroll over dette. I sin masteroppgave fra 2020 kommer Waters frem

til at elever kan oppleve en oppgave som relevant og virkelighetsnær, men likevel ikke føle eierskap, om de ikke føler at de har kommet frem til svaret selv (Waters, 2020).

2.5 Relevans

En måte å se på verdien til en oppgave, og hvordan den påvirker elevers motivasjon, er å snakke om det noe utvidede begrepet relevans. Relevans er et begrep mange kjenner til fra hverdagen. Vi er i hverdagslivet vant til å avgjøre om noe er relevant for oss, og det er nok mange lærere som har hørt og kommer til å høre elever som lurte på om noe de lærer er relevant. For både elever og lærere er begrepet viktig, men det er ikke alltid man er enige om hva det betyr. I debatten om vitenskaps-undervisning er relevans et mye brukt begrep, men ofte på forskjellige måter, og det kan virke som en eksplisitt og akseptert definisjon av begrepet fortsatt er mangelfull (Stuckey, Hofstei, Mamlok-Naaman & Eilks, 2013). De identifiserer at det i historisk kontekst kan sies å være tre klare dimensjoner av relevans innenfor naturvitenskapelig undervisning.

1: *Relevans for å forberede elevene for potensielle karrierer innen vitenskap.* Dette var et syn som var dominerende på 50-tallet under «sputnik-æraen», og som fokuserte på å trene og utdanne flere fremtidige akademikere i samme fagfelt. Vitenskapelig undervisning skulle presentere elevene med et riktig og autentisk bilde av vitenskapen, og målgruppen var først og fremst de elevene som kunne bli fremtidige vitenskapsmennesker.

2: *Relevans for å forstå vitenskapelige fenomen og for å kunne håndtere kommende utfordringer i elevens fremtid.* Dette var en ide som ble mer dominerende mot slutten av 60-tallet og fremover. Det var mange på denne tiden som mente at ideene fra 50-tallet kun fungerte for en liten del av elevene, og at «naturvitenskapelig kunnskap skulle brukes til å besvare viktige spørsmål som folk møtte i sitt hverdagsliv, ikke bare for å gi svar omkring teoretiske vitenskaper» (Bybee and De Boer, 1994, sitert av, Stuckey et al., 2013, s. 5)

3: *Relevans for å gjøre elevene til effektive fremtidige borgere i samfunnet.* Denne tankegangen ble mer og mer gjeldende utover 80-tallet og fremover. Yeager og Hofstein skrev om fire områder hvor kvalitet i læreplanen kunne gi erfaring og fokus. Et av disse omhandlet samfunnsmessige utfordringer, og hvordan naturvitenskapelig undervisning skulle produsere informerte innbyggere, som var forberedt på å håndtere samfunnets

naturvitenskapelige utfordringer (Yeager & Hofstein, 1986). Denne siste dimensjonen kan sies å ha fått stor plass i den nye norske læreplanen, fagfornyelsen, der «demokrati og medborgerskap» har blitt et av tre tverrfaglige temaer i den nye overordnede delen av læreplanen.

2.6 Relevans kontra interesse

Et poeng å ta med seg når man snakker om relevans er at mange forbinder det med interesse. I undersøkelsen *Relevance of Science Education (ROSE)* av Schreiner og Sjøberg fra 2004, en mye sitert undersøkelse, skriver de at selv om de benytter ordet relevans så kunne de like gjerne valgt ord som «... meningsfull, motiverende, interesserende, engasjerende, viktig, etc.» (Schreiner & Sjøberg, 2004, s. 21). Videre skriver de at dette er gjort da mange elever i vestlige land i dag er mindre villige til å ta del i undervisning som de ikke finner motiverende, og at det er viktig å lære seg å kjenne perspektivet til elevene. Dette er definitivt et godt poeng, og alle har nok erfart at det går lettere å lære seg noe når man har en interesse for temaet.

Stuckey et. al. påpeker likevel at det finnes gode argumenter for å se på relevans og interesse som overlappende, men ikke like ideer. De henviser til Krapp and Prenzel, som i sin tekst skiller mellom interesse og fornøyelse, noe de mener er sammenlignbart med forskjellen på relevans og interesse. «Fornøyelse under læring kan oppstå av mange grunner, og interesse er bare en av disse.» (Krapp & Prenzel, 2011).

Noen deler av undervisning kan være relevant uten at elevene er interesserte i dem, og motsatt. Matematikk er for eksempel veldig relevant for senere undervisning i fysikk/kjemi eller yrkesliv. Likevel er det mange temaer innen matematikken som ikke faller innenfor elevenes interesseområde på tidspunktet de blir undervist og det samme kan gjelde for naturvitenskapen (Stuckey et al., 2013). På samme tid kan det være opplevelser elevene finner interessante som de fleste vil mene ikke er relevante. En sekvens der man demonstrerer et uventet fysisk fenomen eller viser frem «kule» eksperimenter vil i de fleste tilfeller oppleves som engasjerende og spennende av elevene, men med mindre man forklarer de bakenforliggende årsaker, eller kommer tilbake til eksperimentet i senere undervisning kan det jo ikke sies å være veldig relevant. En viktig utfordring rundt relevans blir derfor å

fremheve den når den ikke er umiddelbart synlig for elevene. Dette kommer til syne i flere undervisningsmodeller, for eksempel modellen ARCS utviklet av Keller, der relevansstrategien er å binde sammen de lærenes behov, interesse og motiv (Feng & Tuan, 2005).

2.7 Meningsfullhet

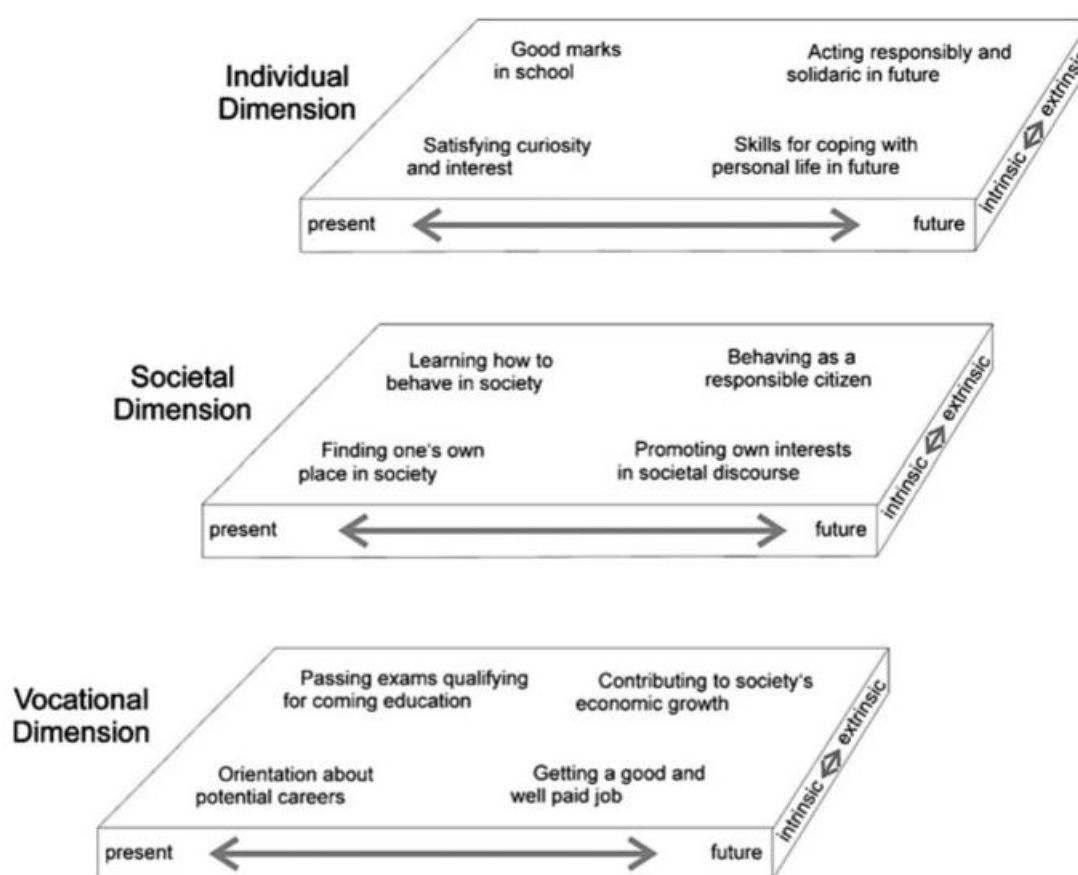
Et annet viktig ord å se på er meningsfullhet. Vi ønsker jo alle at det vi gjør har mening, og vi har nok alle erfart at det å gjøre noe som føles meningsløst ikke er like enkelt å motivere seg for. Skolen har i tidligere tider ofte ikke klart å gjøre undervisning meningsfull i den forstand at det ikke var noen forståelig kobling mellom naturvitenskapelig læring og elevenes hverdagsliv (Dewey, 1973, sitert av Stuckey et al., 2013). Bruner snakker på sin side om at kulturen gir elevene en verktøykasse, og at undervisning skal hjelpe elever til å bruke verktøyene meningsskaping og virkelighetskonstruksjon, slik at de bedre kan tilpasse seg verden de befinner seg i (Bruner, 1996). Det blir her lagt vekt på at vi mennesker ved hjelp av verktøykassen skaper vår egen verden, og vår egen oppfattelse av oss selv, sånn at vi dermed må skape vår egen meningsfullhet også.

Meningsfullhet kan dermed sees på som både noe vi kan gi elevene konkret ved å trekke koblinger mellom det elevene lærer og hverdagslivet, men også som noe vi kan hjelpe elevene til å konstruere selv, ved å gi dem verktøyene de trenger for å skape mening. For eksempel vil de fleste elever i dag gjennom media og den generelle holdningen i samfunnet være opptatt av klimakrisen, og vil nok finne mening i aktiviteter de klarer å linke opp mot denne problemstillingen. Dette speiles av Westbroek, Klaassen, Bulte and Pilot som sier at vitenskapelig undervisning blir relevant for elever, og dermed motiverende, hvis innholdet er bygget inn i en meningsfull kontekst (Westbroek, Klaassen, Bulte & Pilot, 2005, 2010, sitert av Stuckey et al., 2013). De lærende må føle et behov for å vite og ha mulighet til å delta aktivt i problemet som legges frem.

2.8 Rammeverk for relevans

Stuckey et al. har i sin tekst kommet opp med et rammeverk for å vurdere relevans. Dette rammeverket har tre forskjellige dimensjoner: den individuelle, den samfunnsmessige og den yrkesrettede.

Rammeverket har også ulike tidshorisonter slik at den tar hensyn til konsekvenser både i nær og fjern fremtid. I tillegg tar rammeverket hensyn til om relevans retter seg mot noe iboende eller ytre. I følge forfatterne får de med dette rammeverket frem en bredere forståelse av begrepet relevans, og fremhevet at det ikke bare handler om meningsfullhet eller interesse, men om flere ting som omhandler mange aspekter av elevens liv, både nå og i fremtiden.



Figur 1: Rammeverk for relevans utarbeidet av Stuckey et al.

2.9 Tidligere forskning

Det er ikke gjort så mye tidligere forskning rundt bruk av programmering i norske klasserom på videregående skole, men det finnes noen eksempler. De to jeg har tenkt å trekke frem her er begge masteroppgaver; en fra NTNU og en fra UiO. Den eldste er fra 2019 og ble gjennomført av Sigurd Torp Nordby i 2019. Han gjennomførte tre undervisningsopplegg i fysikk 1 og fysikk 2, der det ble programmert i Python. Oppleggene foregikk i vanlige norske klasser fra den samme skolen, og «erfaringer om hvordan lærere og elever opplever undervisningen, er presentert basert på data fra observasjon og spørreundersøkelser». (Sørby, 2019, side iii). Resultatene viser at flere elever finner programmeringen utfordrende, og at det er fare for kognitiv overbelastning. Det kan være utfordringer knyttet til tekniske ting som installering, lagring og lignende som gir utfordringer, eller selve programmeringen. Elevene uttrykte også at de ikke alltid forsto hva koden deres gjorde, og ikke alle likte dette da de gjerne ville ha kontroll og forstå alt. Likevel finner Nordby også mange positive sider ved bruk av programmering i fysikk. Elevene klarer de fleste utfordringer de blir servert, og får ofte til oppgavene. Programmering kan også benyttes som «krydder» til en ellers tradisjonell tilnærming til faget og være med på å bidra til variasjon. Det kan også benyttes til å avdekke misforståelser. Nordby trekker også frem at mange av utfordringene elevene møter kan lettes på om de får gode instruksjoner, og tett oppfølging underveis.

En annen oppgave om programmering er skrevet av Jonathan Brakstad Waters i 2020. Han har sett mer på hvordan programmering påvirker dybdelæring i fysikk. Gjennom sitt arbeid har han utført et opplegg for to fysikk 1 klasser, der elevene skulle simulere kast med og uten luftmotstand ved hjelp av programmering. Ved hjelp av skjerm- og lydopptak, samt fokusgruppeintervjuer har han sett nærmere på hva elevene lærer, hvilke utfordringer de møter, hvordan elevene arbeider med representasjonene i opplegget og hva aktiviteten gjør med elevenes eierskapsfølelse. Han finner at programmering gir elevene muligheter til å benytte ulike representasjoner som hjelper elevene med å se sammenheng mellom ulike størrelser i formlene de benytter. Elevene kan enkelt bytte ut variabler i sin kode og se resultatet av endringen grafisk. Han påpeker likevel at programmering er utfordrende, og at det er mange ting som kan ta fokuset bort fra fysikken om elevene ikke får nok støtte og hjelp til å unngå dette. Dette dreier seg primært om å gi elevene nok støtte og hjelp slik at syntaks og feilsøking tar minst mulig tid, og at mest tid brukes på å jobbe med de mer fagspesifikke utfordringene som omhandler de underliggende prosessene. Et tredje funn fra oppgaven er at

elevene må få hjelp til å forstå koden de arbeider med for å styrke deres kognitive eierskap til oppgaven. Elevene opplevde selve oppgaveteksten som relevant og virkelighetsnær, men at de ikke hadde for sterkt eierskap til aktiviteten likevel siden de følte at de ikke hadde kommet frem til svaret på egenhånd, men heller hadde benyttet kode skrevet av andre.

Som vi ser er det ikke gjort veldig mange undersøkelser rundt programmering i norske klasserom, og det er rom for flere undersøkelser som ser nærmere på hvordan elevene jobber med og opplever programmering som nå kommer inn i den norske skolen. Jeg ønsker å ta utgangspunkt i noe av det som allerede er gjort, og se nærmere på hvordan elevene kobler programmering til det bakenforliggende faget, samt finne ut litt mer om elevenes motivasjon og interesse knyttet til programmering. Jeg vil også gjøre dette i en naturfagsklasse, et fellesfag, som ikke bare inneholder elever som tenker seg å velge realfag videre i sitt skoleløp.

3) Metode

3.1 Datainnsamling

Undervisningsopplegget benyttet i denne oppgaven ble gjennomført i fire forskjellige klasser på tre ulike skoler i og rett utenfor Oslo. Opplegget ble gjennomført av meg, med faglærer tilstede i klasserommet. Det var jeg som var fungerende lærer under økten, men faglærer gikk rundt og hjalp elever som ba om assistanse. Opplegget var det samme i alle klassene, men undervisningsdelen i starten av økta ble justert noe basert på hva elevene opplevde som utfordrende. Det ble i de senere øktene lagt vekt på å snakke mer om fenomenet før elevene gikk i gang med oppgavesettet. Siden datainnsamlingen foregikk mens Covid-19 pandemien var i full sving, med alle de restriksjoner dette medfører gikk det en lengre tidsperiode (3 måneder) fra første til siste datainnsamling. Dette medførte også at noen av gruppene som gjennomførte opplegget hadde hatt tradisjonell undervisning i temaet fra før. Jeg valgte likevel å samle data fra disse klassene da jeg anså det som et nyttig datapunkt til sammenligning.

Datamaterialet mitt består primært av tre deler. Den største av disse er fire fokusgruppeintervjuer, en fra hver av klassene. Det ble også gjennomført en kort spørreundersøkelse mot slutten av timen som elevene svarte på, og i tillegg har jeg oppgavearkene som elevene jobbet med og skrev ned svarene sine på. På toppen av dette har jeg skrevet ned observasjonsnotater fra hver klasse jeg var inne i som kan supplere analysen og bidra til «det overordnede bildet».

3.2 Utvalg

Alle elevene brukt i denne oppgaven er fra de fire klassene hvor opplegget ble gjennomført. I den første klassen var det totalt 13 elever tilstede, 9 gutter og 4 jenter. Den andre klassen som gjennomførte opplegget besto av 19 elever den aktuelle dagen, 10 gutter og 9 jenter. Den tredje klassen hadde 20 elever med på opplegget der 10 var gutter og 10 var jenter. Den siste klassen hadde 19 elever, der 10 var gutter og 9 var jenter. Utvalget som er gjort må sies å være et bekvemmelighetsutvalg. Klassene er alle hentet via enten mitt eget eller veileders

nettverk, noe som ble gjort fordi dette var enkelt og raskt. Datainnsamlingen ble gjennomført våren 2021 da mange skoler gikk ut og inn av hjemmeundervisning, og det var en fordel å kunne skaffe klasser raskt da jeg opplevde at klasser jeg skulle ha datainnsamling i plutselig ikke kunne stille. Dette er også en av grunnene til at de fleste klassene har vært forholdsvis små, da elever har vært hjemme enten i karantene eller fordi de viste symptomer på CoVid-19.

Klassene som er brukt i denne studien var omtrent på størrelse med, eller litt over, gjennomsnittet av det man kan forvente av en norsk naturfagsklasse. Elevene var interesserte i fagstoffet og stort sett arbeidsomme, og ifølge faglærerne var det elever som oppnådde både gode, gjennomsnittlige og mindre gode resultater med under datainnsamlingene. Elevene som har deltatt i fokusgruppeintervjuer har alle deltatt frivillig. Faglærer i klassene ble bedt om å foreslå elever som kunne representere en god spredning av ferdighetsnivå og naturfagskompetanse. Inntrykket under intervjuene var at dette ble oppnådd rimelig bra, men at elevene med høyere naturfaglig kompetanse hadde lettere for å uttrykke seg under intervjuene.

Sammensetning av utvalg kan påvirke troverdigheten til en studie, og det er viktig å prøve å skape et strategisk utvalg som gjør resultatet lettere å generalisere. Jeg vil likevel hevde at mitt utvalg kan forsvares og finnes hensiktsmessig for min oppgave. Alle ferdighetsnivåer innen naturfag er representert både i klassene og også i fokusgruppeintervjuene. Oppleggene hadde også ulik grad av suksess i de forskjellige klassene og det er naturlig å tro at det dermed ble et godt grunnlag for en god diskusjon rundt programmering i naturfag der flere synspunkter var godt representert.

3.3 Spørreundersøkelse

Spørreundersøkelse er en god metode for å samle mye data på kort tid. Der det kan ta for lang tid å inkludere alle elevene i fokusgruppeintervjuer, er det fullt mulig for alle å svare på en undersøkelse. En annen fordel med en spørreundersøkelse er at det ikke er noen observatør til stede som kan ha en effekt på hva elevene svarer. Når man utformer en spørreundersøkelse er det viktig å stille klare og tydelige spørsmål, som i utgangspunktet bare spør om en ting (Kleven, 2014). Svaralternativene bør også være mest mulig faste da dette er effektivt og

tidsbesparende for de som svarer. Dette letter også bearbeiding av besvarte skjemaer. Åpne spørsmål kan benyttes hvis man ønsker veldig detaljerte eller nyanserte svar, men dette krever mye innsats fra både de som besvarer, og de som analyserer. Et mellomalternativ er å benytte en holdningsskala, for eksempel en Likert-skala. Dette er en skala der den som svarer skal ta stilling til ulike påstander, og er som oftest delt i fem (Malt & Grønmo, 2020). Om man kombinerer svarene man samler inn vil man kunne finne et samlet mål på respondentenes holdning til et fenomen.

I min spørreundersøkelse, som ligger vedlagt bak i denne oppgaven, er det benyttet noen få lukkede spørsmål, og en større del med Likert-skala. Spørsmålene her har dreid seg primært om elevenes holdninger til programmering, så det har vært ønsket å få en samlet verdi for ulike påstander. Ellers er skjemaet holdt forholdsvis kort og enkelt, slik at det skal være effektivt og ikke kreve for mye innsats av de som svarer.

3.4 Gjennomføring av spørreundersøkelse

Spørreundersøkelsen ble gjennomført i slutten av økten der elevene hadde undervisningsopplegget. Elevene fikk utdelt undersøkelsen med omtrent 5 minutter igjen av økten, og beskjed om å fylle den ut før de leverte den sammen med det andre de skulle levere inn. Da denne spørreundersøkelsen kun hadde 71 potensielle respondenter er det ikke nødvendigvis mulig å konkludere for mye kun basert på de kvantitative dataene fra den, men dataene kan sammenlignes mot svarene fra fokusgruppeintervjuene, og eventuelt styrke eller svekke disse. Dessuten ble elevene bedt om å oppgi navn på spørreundersøkelsen slik at det var mulig for meg å koble den opp mot oppgavearket som elevene jobbet med gjennom timen, og se om det var noen sammenhenger som kunne utpeke seg der. Selv om spørreundersøkelsen var utformet for å være kort og enkel var det noen elever som valgte å ikke legge for mye arbeid i svarene de ga. Spesielt i en klasse valgte noen elever å svare konsekvent det samme svaret på Likert-skalaen, også på spørsmål som burde ha motstående svar. Jeg har valgt å se bort fra disse undersøkelsene når jeg samlet har summert opp statistikken, men jeg gikk inn og undersøkte disse spørreundersøkelsene ved å sjekke dem mot oppgavearkene til disse elevene for å se om de kanskje opplevde at opplegget ikke traff dem.

3.5 Fokusgruppeintervju

Jeg har i denne studien valgt å benytte meg av fokusgruppeintervju. Dette er et intervju der en gruppe i fellesskap svarer på spørsmål fra intervjuleder eller diskuterer seg imellom temaer de får fremlagt. Om man samler en gruppe får de mulighet til å dele meninger og diskutere seg i mellom, og gir dem mulighet til å spille mellom hverandre (Morgan, 1998). Et fokusgruppeintervju tar delvis form som et strukturert intervju. Dette er en intervjuform der intervjuer har forberedt spørsmål som skal stilles deltagerne (Kleven, 2014). Spørsmålene elevene blir stilt er nøye planlagt, og det er skrevet en intervjuguide slik at spørsmålene er like mellom hver fokusgruppe. Når man utarbeider en intervjuguide gjelder et par av de samme tingene som når man utarbeider et spørreskjema. Spørsmålene skal være «... entydige, og skrevet i et enkelt og klart språk» (Kleven, 2014, s. 36). Det er også viktig at man spør om en ting av gangen

Når spørsmålet derimot er stilt vil deltagerne kunne diskutere mellom seg deres tanker rundt det de ble spurt om, eller komme med innspill til hverandres svar. Denne delen av intervjuet kan minne litt mer om et ustrukturert intervju. Dette er en intervjuform der temaet for intervjuet er klart, men samtalen blir til underveis (Kleven, 2014). Denne delen av intervjuet vil være fleksibel, og man kan følge opp interessante temaer som kommer opp. I et ustrukturert intervju vil også samtalen være mer uformell, og det er lettere at de som intervjues åpner seg opp og deler ting som kanskje ikke ville kommet frem i en formell situasjon. Denne delen av intervjuet stiller store krav til intervjueren som skal veilede, men samtidig ikke legge for sterke føringer for samtalen mellom elevene. Det blir viktig at intervjuer arbeider for å minimere den såkalte observatøreffekten. Dette er en effekt som kommer av at elevene erfarer at de blir observert, og kanskje oppfører seg litt annerledes. «Situasjonen blir litt unaturlig med en observatør til stede» (Kleven, 2014, s. 42).

3.6 Gjennomføring av fokusgruppeintervju

Det ble gjennomført fokusgruppeintervjuer med fire forskjellige grupper, der to av gruppene besto av fire elever, en besto av fem elever, og den siste besto av seks elever. Det var kjønnsbalanse under alle intervjuene bortsett fra i gruppen med fem elever der det var fire

jenter og en gutt tilstede. Jeg gjennomførte selv alle intervjuene, og faglærer var ikke tilstede. Av de 19 elevene som var tilstede snakket alle litt, og de svarte på direkte spørsmål fra meg. En gjenganger var at noen elever snakket mye, mens andre sa noe mindre. Det var ikke alltid at de faglig svake elevene sa minst under intervjuet, men totalt sett er inntrykket at de faglig sterke elevene uttrykte seg mest.

Før intervjuene ble det utarbeidet en intervjuguide som jeg benyttet under alle intervjuene. Denne ligger vedlagt bakerst i denne oppgaven. Intervjuguiden er organisert i noen overordnede temaer, med flere underspørsmål som belyser ulike sider knyttet til de ulike temaene. Noen steder er det også skrevet ned ekstra spørsmål under noen av underspørsmålene som kan benyttes som oppfølging hvis det er noen momenter rundt et spørsmål som ikke elevene kommer innom på egenhånd.

Jeg har under intervjuet forsøkt å holde egne oppfatninger og synspunkter for meg selv. Jeg har forsøkt å gi elevene mulighet til å styre samtalsretning og kun sporet samtalen inn på naturfag etter at et eventuelt sidespor har kommet til en naturlig avslutning. Jeg har også vært veldig tydelig ovenfor elevene at det ikke er noen «gale svar», og at jeg ikke har noen påvirkning for deres vurdering i faget. Dette er ting jeg har gjort for å forhåpentligvis å minke en eventuell kontrolleffekt/observatøreffekt, slik at elevene ikke skal føle at det er utsagn de ikke kan komme med foran meg. Elevene var likevel klar over at jeg var lektorstudent i fysikk, og det kan hende de antok at jeg var positiv til programmering og dens inntog i naturfag.

Jeg brukte opptaksutstyr under alle intervjuene. Dermed satt jeg ordrett igjen med informantens uttalelser etter gjennomført intervju. Jeg gikk for kun lydopptak og ikke video. Dette kan ha virket mindre inntrengende på deltagerne enn videokameraer, og ha minsket en eventuell observatøreffekt. Det var også en praktisk grunn til dette valget da det er lettere å sette opp lydopptak fremfor flere kameraer. Dette har ført til at jeg i ettertid ikke har kunnet analysere ansiktsuttrykk og kroppsspråk, men min observasjon under intervjuene var at tone og innhold i respondentens uttalelser ikke ble påvirket nevneverdig om man tok hensyn til kroppsspråk eller ansiktsuttrykk.

3.7 Elevbesvarelser

Elevene som var med fikk alle utdelt et oppgaveark ved starten av hver økt, som ligger vedlagt bakerst i denne oppgaven. Dette arket hadde et opplegg som elevene gjennomførte, til dels ved hjelp av programmering, og hvor de skrev ned svarene/resultatene sine på oppgavearket. Oppgavearket var bygget opp med økende vanskelighetsgrad på oppgavene, og det hadde en oppsummeringsdel til slutt dere elevene ble stilt en del spørsmål rundt modellen de hadde jobbet med, og om halveringstid generelt. Ved slutten av hvert gjennomførte opplegg med en klasse ble elevenes besvarelser samlet inn, for deretter å bli gjennomgått senere. Dette har gitt meg muligheten til å se direkte hva elevene kom frem til, og hva de har svart/skrevet på de forskjellige oppgavene.

4) Undervisningsopplegg i naturfag med bruk av programmering

Jeg vil i dette kapittelet gjennomgå hvordan undervisningsopplegget elevene jobbet med ble utviklet, si litt om de ulike oppgavene elevene jobbet med og de forskjellige programmeringsverktøyene som elevene benyttet seg av.

4.1 Undervisningsopplegg: Programmeringsoppgave om halveringstid

Dette undervisningsopplegget ble utarbeidet høsten og vinteren 2020/2021. Det er tett basert på, og benytter seg av nettressursen til et opplegg fra boken Naturfag SF fra Aschehoug. Det ble besluttet å benytte dette opplegget som utgangspunkt siden det baserer seg på en allerede eksisterende modellering av halveringstid, men legger til programmering, og sikter på å gi elevene mer mulighet til å utforske på egenhånd. Det ble gjennomført et møte med forfatteren av kapittelet Cathrine W. Tellefsen som også kunne berette om at hun selv hadde god erfaring med opplegget uten programmering, og at hun derfor hadde valgt å legge inn programmering i akkurat dette opplegget.



Figur 2: Bilde fra det første vinduet elevene møter i nettmodulen fra Aschehoug

Det originale opplegget hadde en tidsramme på 1 økt (2x45 minutter), og handlet om å trille terninger for å simulere radioaktivt henfall. Planen var at elevene, ved hjelp av programmering, skulle kunne trille tusenvis av terninger og produsere blant annet en graf som plottet gjenværende ikke-henfalte isotoper mot tid, samt eksperimentere med de forskjellige variablene.

Læringsmålene for timen omhandlet å gi elevene en innføring i hva halveringstid er, og hvordan den henger sammen med sannsynlighet for henfall. Rent konkret skulle elevene gjøre seg kjent med en modell som gjennom sannsynlighet og store talls lov, viste hvordan en mengde radioaktivt stoff sakte minket etter hvert som vi fikk henfall. Rent konkret ble derfor læringsmålene.

- Elevene skal lære hva henfall er.
- Elevene skal bli kjent med terning-modellen som gjennom terningkast simulerer henfall over tid.
- Elevene skal lære hvordan sannsynlighet for henfall henger sammen med, og påvirker, halveringstiden.

Om vi ser på læreplanen så er det to kompetansemål som denne timen vil gi elevene mer kunnskap/erfaring innenfor dette, henholdsvis:

- Vurdere og lage programmer som modellerer naturfaglige fenomener
- Utforske og beskrive elektromagnetisk og ioniserende stråling, og vurdere informasjon om stråling og helseeffekter av ulike strålingstyper

(Utdanningsdirektoratet, 2020)

Utformingen av opplegget har tatt hensyn til ideer fra både Sørby (2010) og Brown og Wilson (2018). Oppgavene har en kontekst, og gis trinnvis for å la elevene bygge seg mot målet steg for steg. I nettressursen elevene får tilgang til er det også en del utdypende informasjon om begreper, samt et løsningsforslag som elevene kan benytte seg av til hjelp og støtte. Elevene får også utdelt mye ferdigskrevet kode, slik at de selv kan fokusere mer på naturfenomenet de undersøker og ikke må jobbe like mye med syntaks og feilsøking. I starten av opplegget er det et par spørsmål som omhandler halveringstid og henfall som skal utfordre elevene litt. De som ikke har lest i boka kan slå opp der og dermed få en kjapp innføring i begrepet. Deretter er det flere oppgaver hvor elevene skal jobbe med et program som simulerer terningkast, og de vil etter hvert modifisere dette programmet til å trille et visst

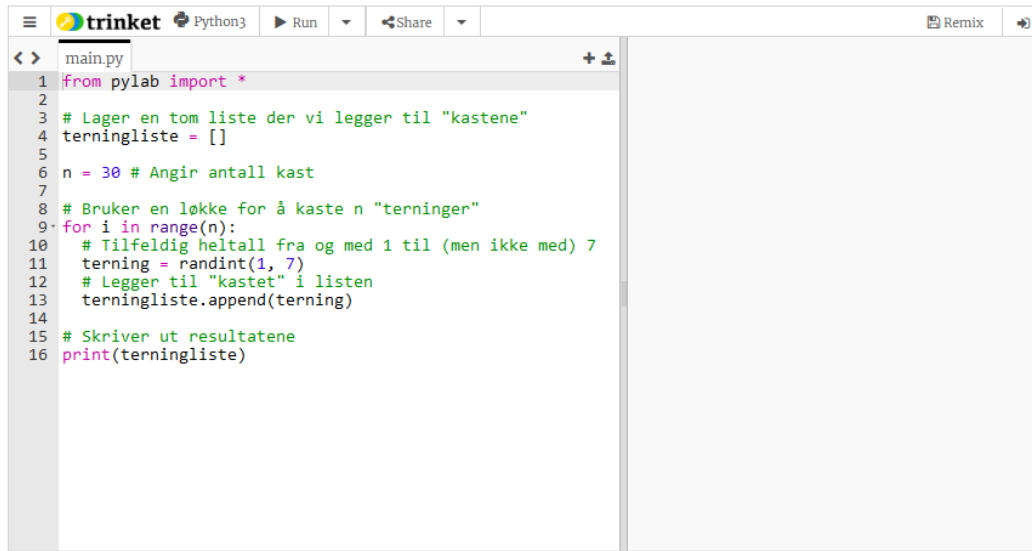
antall terninger, ta bort alle terninger med seks øyne, og så trille igjen. Terningene simulerer en viss mengde radioaktive isotoper, og å få en sekser på terningen skal simulerer at henfall har skjedd. Mengden radioaktive isotoper som er igjen vil dermed minske for hvert kast, og om man plotter antall terninger som er igjen mellom hvert kast vil man få en kurve som ligner en vanlig kurve for halveringstid. Mot slutten av opplegget er det noen oppfølgingsspørsmål der elevene skal knytte modellen og terningene de jobbet med sammen med det virkelige fenomenet halveringstid.

4.2 Trinket og Aunivers

Opplegget baserer seg rundt en nettmodul som ligger på Aunivers tilhørende Aschehoug. Denne nettmodulen inneholder instruksjoner og bilder som forklarer elevene litt om opplegget og koden. Modulen har også en del fasitsvar som elevene kan sjekke sine svar mot. Dessuten inkorporerer modulen Trinket.io. Trinket er et nettbasert programmeringsmiljø der elevene kan skrive og kjøre koden sin, uten å installere noe på sin egen maskin. Dette er veldig nyttig og er med på å fjerne utfordringer knyttet til installasjon og vedlikehold av et programmeringsspråk. Trinket kjører Glowscrip, eller VPython, som består av python og en grafikkmodul. Trinket kan også komme med mye kode ferdig utfylt fra lærer/bokforfatters side. Dette medfører at elevene kan få starthjelp slik at de ikke selv trenger kode fra «scratch», hvis de har begrenset erfaring fra før. Den ferdige koden elevene får kan være så omfattende at det for elevene kan oppleves som å jobbe med en avansert kalkulator, eller ferdigbygget maskin, der de kun skriver inn og fokuserer på variablene. I vårt tilfelle har elevene i starten kun behov for å endre på variabler, mens de etter hvert må skrive inn noen få linjer med egen kode.

I dette programmet kan du angi antall terninger som skal kastes. Ved hjelp av en løkke bruker vi funksjonen **randint(1, 7)** én gang per terning. Denne funksjonen gir oss et tilfeldig heltall fra og med 1 til (men ikke med) 7, den fungerer med andre ord som en vanlig terning. Resultatet vi får fra funksjonen, legger vi til i listen som vi har kalt **terningliste**. Til slutt skriver vi ut verdiene i listen for å se resultatene.

Nedenfor kan du se programmet slik det er presentert i læreboka.



```
1 from pylab import *
2
3 # Lager en tom liste der vi legger til "kastene"
4 terningliste = []
5
6 n = 30 # Angir antall kast
7
8 # Bruker en løkke for å kaste n "terninger"
9 for i in range(n):
10     # Tilfeldig heltall fra og med 1 til (men ikke med) 7
11     terning = randint(1, 7)
12     # Legger til "kastet" i listen
13     terningliste.append(terning)
14
15 # Skriver ut resultatene
16 print(terningliste)
```

Oppgaver Løsningsforslag

Figur 3: Bilde fra Trinket-vinduet til den første programmeringsoppgaven elevene møter

På Aunivers er Trinket innebygd i nettsiden til Aschehoug, og kommer opp med to vinduer. Et kodevindu, hvor koden man lager skrives inn, og et resultatvindu der grafikk eller annen output fra koden vil vises. Når elevene går til en ny del av nettmodulen på Aunivers kommer det opp et nytt Trinket-vindu hvor det vil være ny kode.

5) Analyse

5.1 Transkribering og analyse av intervju

Fokusgruppeintervjuene er blitt lyttet til på avspiller og transkribert. Uttalelsene derfra er tillagt en eller flere respondenter. Pseudonymer er brukt, og hvert pseudonym skal tilhøre en bestemt elev. Kjønnnet på pseudonymet og eleven er det samme. Alle uttalelser ble transkribert, også kortere uttalelser i bakgrunnen av en annen uttalelse. Det er mulig slike uttalelser i bakgrunnen er blitt tilegnet feil pseudonym da de ofte er korte og potensielt vanskelige å høre. Dette er selvfølgelig ikke optimalt, men disse uttalelsene er ikke fryktelig viktige i det store bildet, og heller ikke for resultatene. Lyder som «eh», «huh» og stamming er tatt med de steder der dette oppleves som relevant.

Etter gjennomført transkribering er intervjuene kodet og analysert ved hjelp av verktøyet NVivo 12. Det ble gjennomført en innledende koderunde, der rammeverket til Stuckey et al. er benyttet for å kode relevans, og induktiv koding er benyttet for å kode andre interessante uttalelser. Etter denne runden er forekomsten av de forskjellige kodene sjekket, og eventuelle koder med betydelig overlap er slått sammen der dette har vært hensiktsmessig. Intervjuene er deretter kodet på nytt etter de nye kodene. Kodene som har blitt benyttet til siste koderunde kan samles i tre ulike kategorier som er listet opp under.

Relevans	Opplagg og undervisning	Generelle utfordringer
Individuell relevans <ul style="list-style-type: none">Nåværende iboendeNåværende ytreFremtidig Yrkesmessig relevans Autentisk/virkelighetsnært Eierskap	Scaffolding Utfordringsnivå Modell/fenomen Tidligere undervisning	Kognitiv belastning Programmeringsteknisk Støy/distraksjon

Tabell 1: Liste over koder brukt under analyse

Kategorien «Relevans» inneholder utsagn fra elevene som kan knyttes opp mot relevansbegrepet. I forhold til rammeverket til Stuckey et al. er antall koder redusert da mange av dem ikke dukket opp i elevenes uttalelser. Dette gjaldt spesielt den samfunnsmessige delen av rammeverket, men også den yrkesmessige ble redusert betydelig da elevene stort sett kun

uttalte seg om potensielle fremtidige jobbmuligheter når de snakket om yrkesmessige ting. Koden autentisk/virkelighetsnært er også lagt inn i denne kategorien da denne kan sies å være nært beslektet relevans. Koden eierskap er også lagt inn her.

Kategorien «Opplegg og undervisning» tar for seg utsagn som omhandler det konkrete opplegget elevene jobbet med, eller tidligere undervisning de hadde hatt i programmering. Kodene Scaffolding og Utfordringsnivå ble benyttet når elevene snakket om utfordringer de støtte på, og hvor de fikk hjelp. Koden modell/fenomen ble benyttet når elevene uttalte seg om koblingen mellom modellen de jobbet med og naturfenomenet halveringstid, enten de så denne koblingen eller ikke. Koden tidligere undervisning ble benyttet når elevene uttalte seg om tidligere undervisning i programmering som de hadde hatt på VGS.

«Generelle utfordringer» er en kategori som er benyttet for å samle utsagn fra elevene som omhandler utfordringer med programmering som ikke er spesifikt knyttet til opplegget de gjennomførte. Koden kognitiv belastning ble benyttet for å samle utsagn der elevene snakker om belastningen det er å skulle lære programmering, og gjerne når de uttalte seg om mye nytt som skal læres, eller en annen måte å tenke på en de er vant til. Koden programmeringsteknisk er benyttet når elever har trukket frem mer tekniske eller mindre aspekter ved programmering som kan være utfordrende, slik som syntaksfeil eller feil kommando. Koden støy/distraksjon er benyttet når elevene har kommet med uttalelser som indikerer at programmering har tatt fokuset vekk fra naturfag, enten elevene har kommet med dette selv, eller blitt stilt direkte spørsmål om det.

5.2 Spørreundersøkelse

Det ble gjennomført en spørreundersøkelse i hver klasse som hadde opplegget. Denne ble gjennomført til slutt, etter endt opplegg. Det var noen elever som valgte å ikke levere inn spørreundersøkelse, da spesielt i den fjerde klassen som gikk litt over tiden. Totalt var det 7 elever som ikke leverte inn. I tillegg ble 6 av spørreskjemaene tatt ut av statistikken da elevene der kun hadde fylt ut det samme Likert-alternativet nedover hele skjemaet, noe som gjør at man har levert motstridene svar, og trolig ikke har tatt seg tid til å fylle ut skjemaet skikkelig. Ved en gjennomgang av oppgavearkene som tilhørte de spørreundersøkelsene som ble tatt ut er det funnet at 3 av disse ikke engang var påbegynt, mens de resterende 3 kun

hadde gjennomført en liten del av opplegget. Det er mulig opplegget ikke klarte å motivere elevene, men det må tas med at det generelle inntrykket jeg fikk i denne klassen var at noen så på økten som en mulighet til å koble litt av og ikke jobbe med skolearbeid.

De 60 gjenværende spørreundersøkelsene er blitt delt inn i to kategorier, en for de som oppga tidligere programmeringserfaring fra før VGS, og en for de som oppga ingen programmeringserfaring fra før VGS. For disse to gruppene er så Likert svarene fra spørreundersøkelsene summert opp og regnet ut. Det er sett etter trender og blitt gjort sammenligninger mellom de to gruppene. Foruten å sammenligne disse to kategoriene mot hverandre er det også sett litt på de individuelle klassene, og forskjeller mellom disse. Når dette er gjort blir dog datagrunnlaget meget lite for hver klasse, så det blir ikke trukket for mange bestemte konklusjoner ut fra dette. For å visualisere svarene er det laget diagrammer som presenterer flere Likert-verdier for de to ulike gruppene. Disse er presentert i resultatdelen.

5.3 Gjennomgang av oppgaveark

Alle oppgavene-arkene elevene leverte inn er sett gjennom. For spørsmål syv som omhandler sammenhengen mellom modellen elevene jobbet med, og naturfenomenet halveringstid er det også utarbeidet statistikk som tar for seg om elevene svarte «rett» eller «galt» på dette spørsmålet. Det er også sjekket om «gale» går på at elevene tror simuleringen kun omhandler programmering, og at hensikten med programmeringen har blitt borte for dem, eller om de har svart «galt» på andre måter. Det er også flere elever som ikke har rukket å svare på dette spørsmålet. Dette har blitt talt opp og blir tatt med som et viktig ankepunkt rundt forholdet mellom «rette» og «gale» svar.

5.4 Etiske hensyn

Jeg har under arbeidet med denne oppgaven forhold meg til viktige forskningsetiske prinsipper. Opplegget mitt, inkluderende både oppgaveark, spørreundersøkelse og intervjuguide til fokusgruppeintervju er sendt inn til NSD, Norsk senter for forskningsdata, og godkjent av disse. Alle elever har også fått et samtykkeskriv, både noen dager før

opplegget, og på nytt i starten av økten jeg hadde med dem. Dette skrivet ligger vedlagt som vedlegg bakerst i denne oppgaven. I dette skrivet kunne de finne informasjon om hva de skulle være med på, hva dataene skulle brukes til, og hvem som var kontaktpersoner. De fikk også vite om sine rettigheter, og deres mulighet til å trekke samtykke ved senere anledning. Dette ble også gjennomgått muntlig når elevene fikk utdelt skrivet på nytt i starten av økten jeg hadde med dem. Elevene ble bedt om å se gjennom skrivet, og de som ville være med ble bedt om å skrive under. Skrivene ble deretter samlet inn og etter økten sjekket jeg om det var noen som ikke hadde skrevet under. Dermed slapp elevene å si ifra i timen, eller til meg eller faglærer, om at de ikke ville være med. Det var ingen elever som lot være å gi samtykke. Det ble påpekt for elevene at det var frivillig å være med, men at det ikke innebar noen risiko å skrive under, og at det var begrenset med hvem som ville ha tilgang på deres data. Jeg mener derfor at samtykke som ble gitt var informert og frivillig fra elevenes side. Det var et eget samtykkeskriv for elevene som deltok i fokusgruppeintervju, som elevene fikk før vi begynte med disse. Elevene ble også muntlig påminnet om de viktigste tingene herfra, slik som frivillig deltakelse som en intro til hvert fokusgruppeintervju.

Når jeg samler data fra elever er det et par ting som veier tungt i et etisk perspektiv, sett fra elevenes ståsted. En av disse er anonymitet, og hvem som kan se dataene deres. Lydopptak fra elever har kun vært tilgjengelig for meg selv og veileder, og de er ikke delt med noen. De er også lagret på opptakerutstyr fra instituttet, eller servere hos UiO, og er slettet etter transkribering. Under transkribering er alle elever anonymisert, og det er kun jeg som har gjort dem som vet hvilke intervju som er fra hvilke klasser. For spørreundersøkelsene ble disse foretatt skriftlig, og det er kun jeg som har hatt disse. Disse er sett gjennom av meg, og verdier er summert opp. De er deretter blitt destruert. Oppgavearkene elevene jobbet med ble samlet inn av meg, og gjennomgått. Dette var også det eneste datamaterialet faglærerne kunne fått tilgang til, slik de ville gjort med alt annet elevarbeid fra en vanlig time, men ingen av dem valgte å gjøre dette. All annen data fra elevene var ikke tilgjengelig for lærerne, og dette ble også tydelig poengtert overfor elevene.

5.5 Troverdighet og generalisering

Dataene i dette prosjektet er primært av kvalitativ art. Det er noen spørsmål elevene får som gir kvantitative data, som tidsbruk til forskjellige oppgaver, men utenom dette omhandler de

fleste dataene elevenes tanker, følelser og opplevelser. Når vi jobber med kvalitative data blir begrepet «troverdighet» viktig, og det erstatter begrepet validitet som vi kjenner fra kvantitativ forskning (Onwuegbuzie & Johnson, 2006). For å oppnå troverdighet er det viktig at resultater er plausible og sannsynlige. Korrekt fremstilling av data, og om data tillegges gal mening under analysen er viktige punkter å se på her. Det er dessverre ikke mulig å dele lydopptakene eller spørreskjemaene som elevene fylte ut av personvern hensyn. Transkripsjonene kan likevel deles ved forespørsel, sånn at de som måtte ønske skal kunne undersøke sammenhengen mellom mitt datamateriale og resultater. Likert-verdiene fra spørreundersøkelsen er også delt i en tabell her, og kan sees av alle som leser denne oppgaven.

Jeg vil nå peke på noen ting ved opplegget som kan ha innvirkning på denne oppgavens troverdighet. Det første er selve undervisningssituasjonen elevene var i. Det var to lærere tilstede (meg selv og faglærer), og noen klasser var redusert i antall. Det var dermed flere som kunne gi elevene hjelp og støtte under arbeidet. Om det kun var en lærer tilstede er det mulig at elevene ville møtt på flere utfordringer, og at de hadde hatt et mindre positivt inntrykk etter timen. Dessuten var det en fremmed lærer i timen, som kan ha påvirket hvordan elevene jobbet, selv om jeg ikke så tegn til at dette var tilfellet. Lærerne i de ulike klassene ga tilbakemelding på at elevene hadde oppført seg normalt i alle klasserom jeg var inne i, og jeg opplevde et vidt spekter av forskjellige oppførsel, arbeidsinnsats og interesse når elevene jobbet med opplegget.

Under fokusgruppeintervjuene er det også mulig at elevene tilpasser seg situasjonen, og at dette påvirker dataene. Hvis elevene er utrygge på hverandre, eller er redd for å fremstå som «dumme», kan de tilpasse sine utsagn. Det er også mulig at elever velger å uttale seg mer positivt enn de egentlig er. Et fint motstykke til dette er at det under fokusgruppeintervjuene kom både positive og negative utsagn rundt både programmering generelt, og opplegget. Det var flere grupper der elever hadde motstående meninger og utsagn, som jo styrker troen på at det elevene sa stemmer med deres meninger. Ved å sammenligne utsagn elevene kom med opp mot deres svar i både spørreundersøkelsen og oppgavene de jobbet med er det også mulig å gjennomføre såkalt triangulering. Triangulering er en prosess for troverdighet der man søker konvergens fra flere ulike kilder til informasjon (Creswell & Miller, 2000). Jeg har under arbeidet mitt forsøkt å finne data fra både spørreundersøkelsen, oppgavearkene, og fokusgruppeintervjuene som indikerer det samme. Det er selvfølgelig en stor fallgrube her,

ved at man som forsker fort kan begynne å se sammenhenger som ikke nødvendigvis er tilstede.

Et annet moment når man analyserer data er det som kalles bias, som kan påvirke forskeren og vedkommendes tolkning av data, både bevisst og underbevisst (Creswell & Miller, 2000). Et grep som kan tas for å øke troverdigheten er at man oppgir sine antagelser til leseren. Jeg merket selv at jeg hadde tro på opplegget som elevene gjennomførte, og jeg har personlig tro på at programmering har en plass i den norske skolen. For å forsøke å motvirke dette har jeg forsøkt å være kritisk, og sett etter indikatorer fra elevene om at ting ikke fungerte, eller at det var noe de ikke likte. Jeg har også forsøkt å gi gode nøyaktige beskrivelser, slik at det skal være mulig for leseren å se hvorfor jeg kommer til mine vurderinger og selv vurdere om de er enige i de slutningene som trekkes. Der det er mulig har jeg også forsøkt å finne uttalelser fra flere ulike fokusgrupper.

Et siste punkt er om forskning er generaliserbar, og om den kan anvendes i andre klasserom rundt i landet. For funnene fra denne oppgaven og den innsikten som omhandler elevers holdninger og tanker rundt programmering vil dette være ting lærere kan ta med seg og ta hensyn til når de selv skal planlegge egen undervisning med programmering. Det må likevel huskes på at antallet elever i denne studien er begrenset, og selv om klassene var varierte, så kom de alle fra omtrent samme geografiske område. Det er også mulig at andre funn fra denne oppgaven ikke nødvendigvis lar seg generalisere, spesielt siden den ble gjennomført med elever som ikke vil ha samme bakgrunn innenfor programmering som kommende elever.

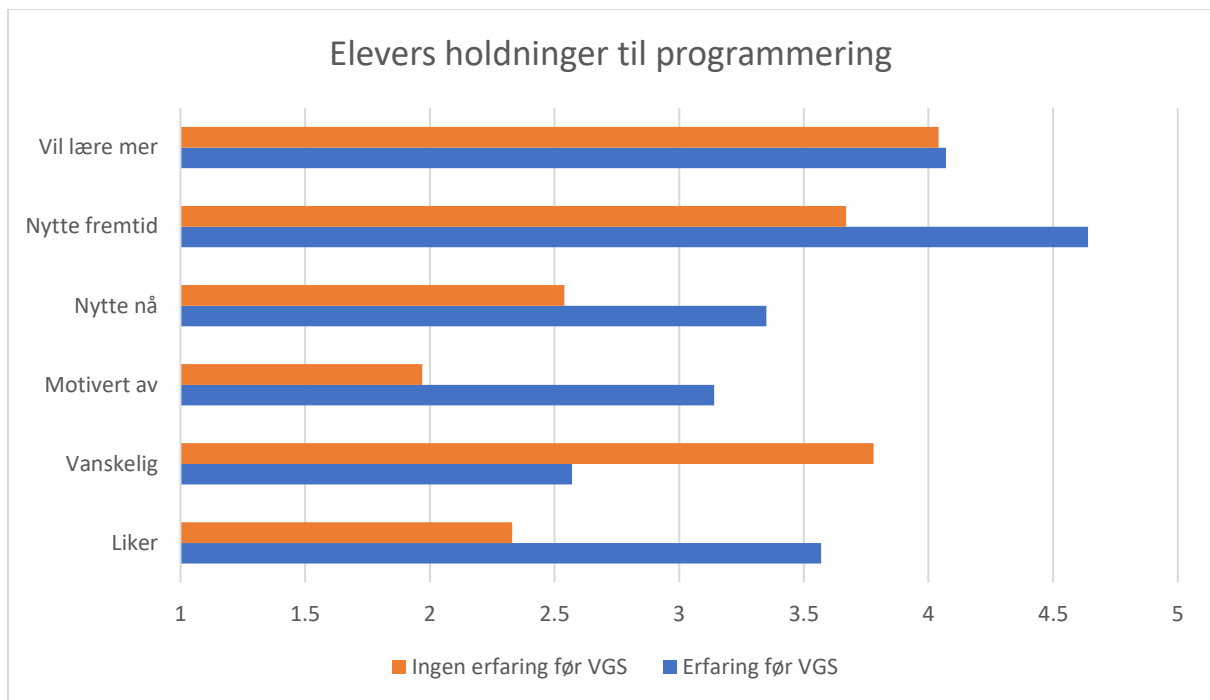
6) Resultater

Her skal jeg nå presentere resultatene fra analysen av mitt datamateriale. Jeg presenterer først resultater fra spørreundersøkelsene, og oppgavearkene elevene jobbet med. Etter dette kommer en seksjon som gir litt innsikt i hver av de fire fokusgruppene, før jeg fokuserer på ulike temaer som kom frem under disse. Det kommenteres av og til kort om resultatene her i resultatdelen, men hovedtyngden av diskusjonen kommer i sin egen del etterpå.

6.1 Spørreundersøkelse

Svarene fra spørreundersøkelsen er delt i to grupper, der en gruppe hadde erfaring med programmering fra før VGS, og en hadde det ikke. Det er flere forskjeller mellom de to gruppene, men også noen forskjeller mellom de ulike klassene. Elevene fra den første og andre klassen, hadde et generelt bedre inntrykk av programmering enn elevene fra den fjerde, og spesielt den tredje klassen. Forskjellene var på opptil 1,0 poeng på en skala fra 1 til 5.

De to gruppene, forhåndserfaring og ikke forhåndserfaring, er rimelig forskjellige i størrelse. Gruppen uten forhåndserfaring er på 46 personer, og en enkelt person har ikke påvirket gjennomsnittet veldig mye. Denne gruppen hadde et spenn av meninger, der noen tydelig satte mer pris på, og så nytten av, programmering enn andre. Likevel var det noen ganske klare trender, som kommer frem når man ser på gjennomsnittet av Likert-svarene. For de med forhåndserfaring er gruppestørrelsen mindre, og besto av totalt 14 respondenter. Dermed kan enkelsvar påvirke gjennomsnittet i større grad enn det kan for den større gruppen. Ved gjennomgang viste det seg at de 14 hadde svart ganske likt, og selv om det var enkelte punkter hvor svar varierte fra 2 (uenig) til 5 (sterkt enig), så var hovedbildet at de fleste i denne gruppen var samstemte om ganske mye.



Figur 4: Diagram med oversikt over elevenes holdninger til programmering. Verdi 1 tilsvarer alternativet «sterkt uenig» og verdi 5 tilsvarer alternativ «sterkt enig»

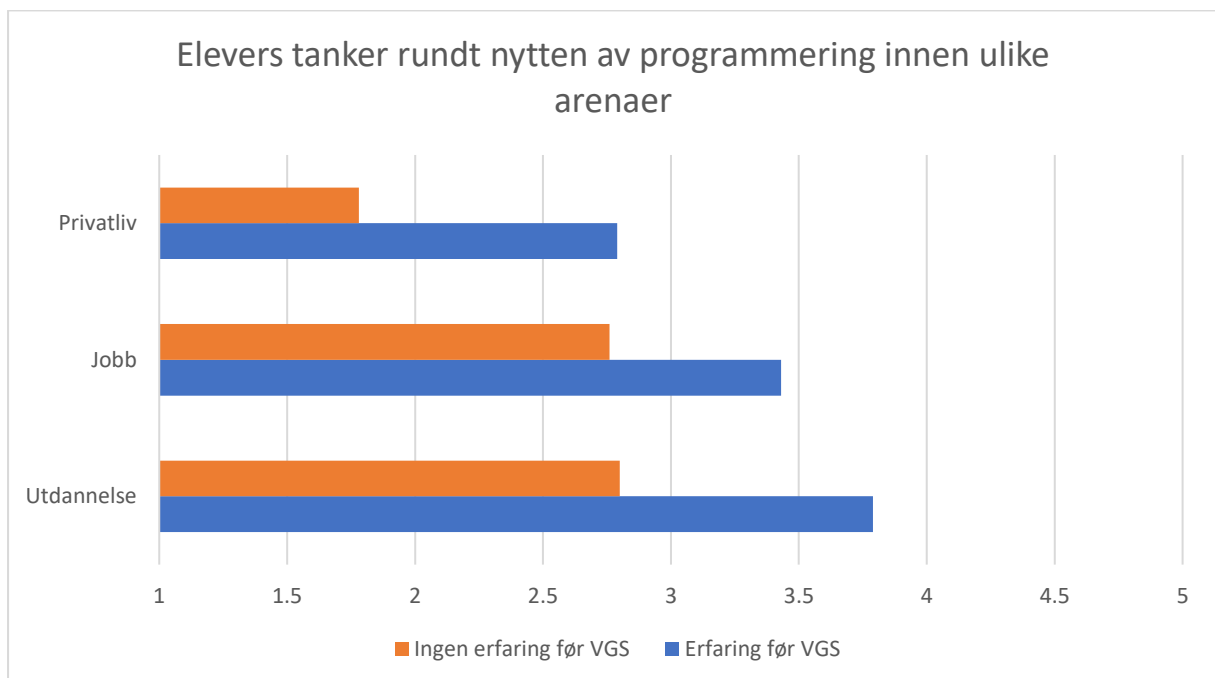
Når det kom til elevenes holdninger til programmering så kan vi se at det er både likheter og ulikheter mellom de to gruppene. Den største forskjellen (1.24 poengs differanse) finner vi på spørsmålet om elevene liker programmering, der de med tidligere erfaring svarte i snitt noe mellom «både og» (3) og «enig» (4). De uten forhåndserfaring sa seg derimot mer uenig med denne påstanden, og gjennomsnittet endte mellom «uenig» (3) og «både og» (2). Det er flere mulige grunner til denne forskjellen, men de som har hatt programmering fra før har gjerne har valgt det selv, så denne forskjellen virker ganske naturlig.

Det er også tydelig at elevene uten forhåndserfaring opplever programmering som mer vanskelig. Ved påstanden «Jeg synes programmering er vanskelig» heller svaret til elevene med erfaring i gjennomsnitt litt mot alternativet «uenig» med en verdi på 2.57, mens elevene uten forhåndserfaring tydelig er mer enige. med en verdi på 3,78. Dette virker også som et resultat man kan forvente da de med erfaring jo har holdt på lenger med programmering, og har hatt anledning til å lære mer.

Ved spørsmålet «jeg blir motivert av programmering» ser vi igjen tydelige forskjeller mellom de to gruppene, omtrent et poeng på skalaen, litt som ved påstanden «jeg liker programmering». Forskjellen fra den påstanden er at begge grupper er mer uenig i denne nye påstanden, og at ingen grupper gir et klart uttrykk for å være enig av å bli motivert av programmering, selv om det er enkeltelever, i begge grupper, som sier seg enige i påstanden.

Når det kommer til påstandene om å nyttiggjøre seg av programmering, så ser vi igjen at gruppen med forhåndsprogrammering er mer enige i påstandene. Det er likevel verdt å merke seg at begge grupper er enige i påstanden om at programmering kan bli nyttig i fremtiden. Når man endrer spørsmålet til å handle om nytte de har hatt, er elevene generelt litt mer uenige, og vi ender opp med å ha begge gruppene på hver sin side av «både og» (3) alternativet.

På spørsmålet som handler om elevene ønsker å lære mer programmering så ser vi at forskjellen mellom de to gruppene forsvinner. Begge gruppene svarer i snitt at de er enige i påstanden, med en verdi på rett rundt 4, som tilsvarer alternativet «enig». Det er vanskelig å si hvorfor de to gruppene er enige, og veldig positive, til denne påstanden, men vi kan spekulere litt. Begge gruppene mente jo at programmering kunne være nyttig i fremtiden, og da vil det være nyttig å ha lært det. Det er også mulig at de elevene som opplever det som vanskelig og ikke motiverende ønsker å lære mer programmering med den tanke om at det da blir mindre vanskelig for dem. For gruppen med erfaring, og som i snitt liker programmering virker det ganske naturlig at de vil lære mer. Dette er likevel kun spekulasjoner, men det er viktig å ta med seg at selv om elevene syns noe er vanskelig og lite motiverende nå, så kan de ha et ønske om å lære mer om det.

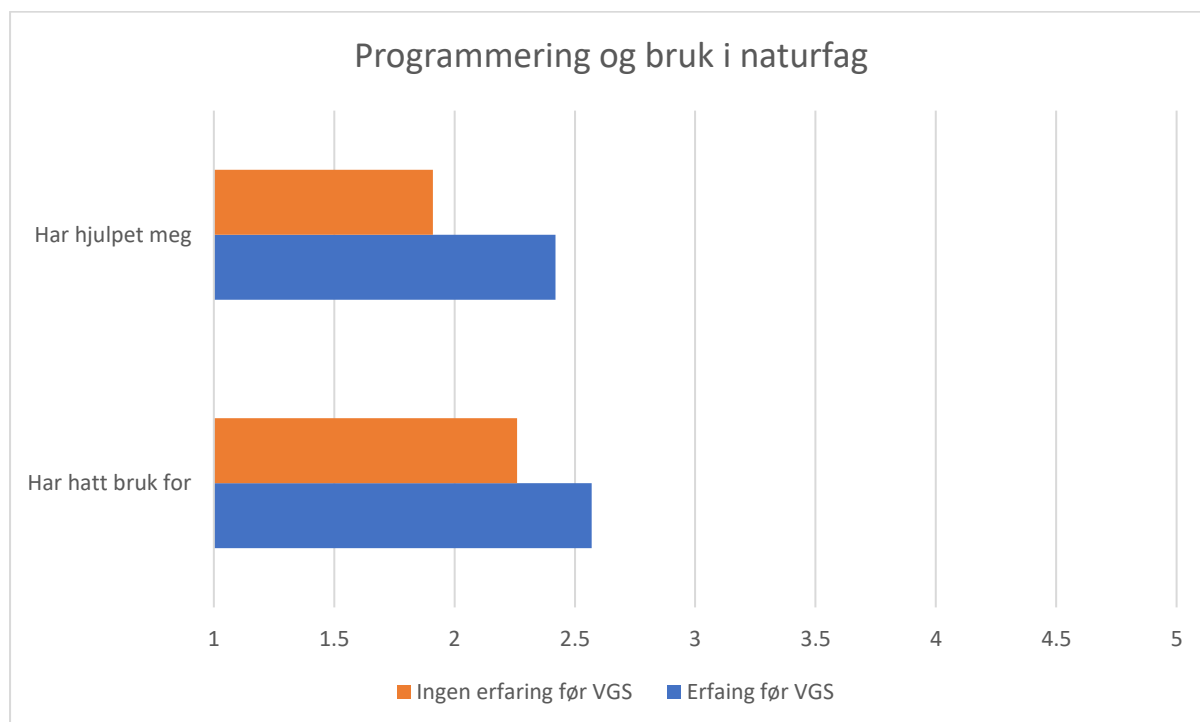


Figur 5: Diagram over hvor elevene trodde programmering kunne være nyttig

Elevene ble også spurt om på hvilke områder de tror programmering kan være nyttig for dem senere i livet, og hvor de tror de kan få bruk for det. Her var det også forskjell mellom de to

gruppene, gruppen med forhåndserfaring var konsekvent mer enige i påstander enn gruppen uten dette. Av ting det er verdt å merke seg her er at begge gruppene var minst enige i påstanden om å bruke programmering i eget privatliv, men at gruppen uten erfaring er betydelig mer uenig i denne påstanden. Vi kan se dette litt i sammenheng med et annet spørsmål fra undersøkelsen som omhandlet nettopp det å programmere på fritiden. Elevene ble nemlig spurt om de gjorde dette og der en meget liten andel av de 46 uten erfaring svarte at de hadde programmert på egen fritid (3 av 46 totalt, 6,5%), så oppga en betydelig større andel av de med erfaring at de programmerte på fritiden (5 av 14 totalt, 35,7%).

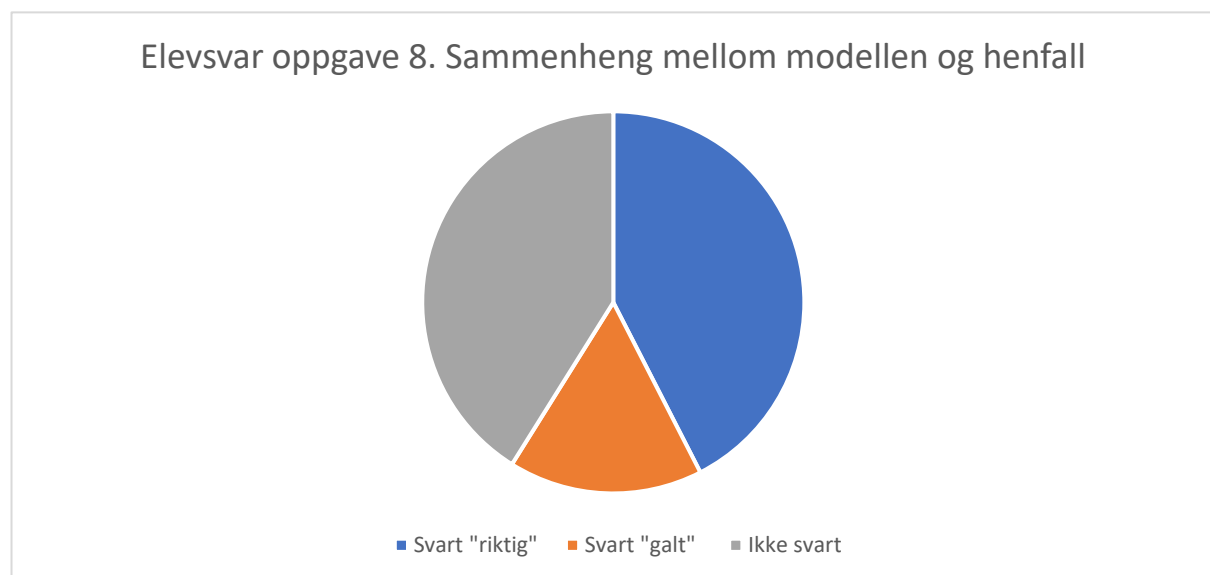
Når det kommer til mer «tvungen» programmering i fremtiden, enten gjennom utdanning eller jobb, så er det betydelig flere som tror de får bruk for dette. Blant de uten erfaring er svarene til påstandene rundt, men litt under alternativet «både og» (3), og veldig mange har svart dette alternativet. Det kan virke som at flere er litt usikre på hvordan ting blir i fremtiden, der programmering blir viktigere og viktigere innen mange, men kanskje ikke alle, arbeids- og utdanningssektorer. De som har forhåndserfaring virker tydeligere å tro at dette er noe de får behov for innen utdanning og yrke, og kanskje ser noen av disse for seg å jobbe med programmering i fremtiden allerede. Faktisk var det flere av de som svarte her som sa seg «helt enig» (5) i påstandene om behovet for programmering i fremtidig utdanning og yrke.



Figur 6: Diagram over elevenes erfaring med programmering i naturfag

Når det kommer til spørsmål om å ha benyttet, og hatt nytte av, programmering i naturfag heller svarene fra elevene stort sett mot alternativet «uenig». For både de med forhåndserfaring, men også de uten, er elevene noe mer enige i at de har hatt bruk for programmering, fremfor at det har hjulpet dem. Det var en av klassene som hadde generelt flere som svarte at de hadde hatt bruk for programmering, hvor det under fokusgruppeintervju kom frem at de hadde benyttet programmering til å modellere bakterievekst tidligere. Ellers ga de andre elevene i fokusgruppeintervjuene uttrykk for at de kun hadde benyttet programmering i matematikken, og da primært gjennom oppgaver der fokuset ligger på å lære å programmere, og ikke bruke det til å lære noe annet. Det viktig å ta med seg at disse elevene ikke har hatt programmering før de begynt på VGS, og at lærerne dermed ikke kunne forvente at de hadde kunnskaper om dette. Det kan virke som at de fleste naturfaglærerne har valgt å ikke benytte seg av programmering for disse elevene enda og heller venter med å benytte seg av dette til det kommer elever fra ungdomskolen som har hatt programmering der.

6.2 Oppgaveark



Figur 7: Sektordiagram over elevens svar på oppgave 8

Når vi ser på oppgavearkene elevene leverte inn er det et spørsmål som er meget interessant å se nærmere på, og det er hva elevene har svart på oppgave 8. Denne oppgaven spør nemlig

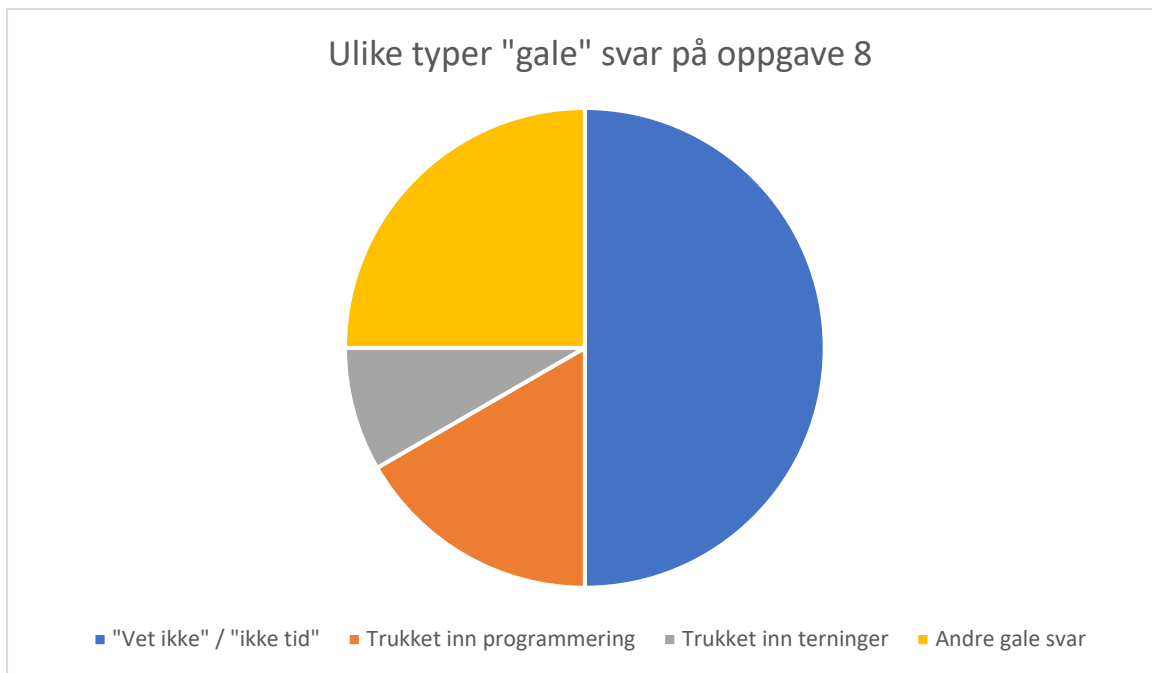
elevene om hva det å få en sekser i modellen de jobbet med representerte i virkeligheten. For å kunne svare på dette spørsmålet er det viktig at elevene klarer å koble modellen de har jobbet med opp mot fenomenet halveringstid. Et riktig eller galt svar her er ikke nødvendigvis ensbetydende med om man har forstått modellen og hvordan den fungerer, men det er en veldig god indikator på det. Når man skal se gjennom elevsvar som dette er det ikke alltid lett å si når et svar er godt nok. For å gi en indikasjon på hvordan jeg har tolket så kommer her en liten tabell med noen eksempler på hva jeg har klassifisert som «riktig» og «galt».

Riktig	Galt
At et atom henfaller	Det programmerer antall terninger som man
Et atom henfalt, og ble til et annet stoff	trenger for å få et visst antall
En sekser simulerer at stoffet mister et	At det er sannsynlig å få en sekser
radioaktivt atom	Det skulle simulere halveringstiden
	Vet ikke

Tabell 2: Eksempler på «riktige» og «gale» elevsvar på oppgave 8

Blant svarene er det noen der elevene har begynt å snakke om halveringstid, men ikke trukket inn henfall. Der elevene trakk frem at et radioaktivt atom ble «borte» eller «ikke lenger var radioaktivt», så har jeg valgt å anse dette som «rett», mens der eleven har nevnt halveringstid, men ikke utdypet noe mer, så har jeg valgt å tolke dette som «galt». Det er mulig at disse elevene her har forstått mer enn svaret deres skulle tilsi, men det kommer ikke frem gjennom oppgavearket deres.

Vi har fra de 73 elevene totalt 31 som har avgitt et «riktig» svar, 12 som har avgitt et «galt» svar, og 30 som ikke har svart. Ved ettersyn av de som har svart «galt» er det flere her som tydelig har hatt dårlig tid, og bare svart noe kjapt her. Faktisk er det 3 av besvarelsene her som er «vet ikke», og 3 andre som omhandler mangel på tid. Trolig ville noen flere av disse kunne gitt et «riktig» svar med bedre tid. Bare 3 av de 12 som svarte «feil» har begynt å dra inn programmering eller terninger i svaret sitt, så det kan virke som at det ikke er mange som har blitt forvirret av programmeringen eller hva den skulle forestille.



Figur 8: Sektordiagram over hvilke typer feil elevene gjorde på oppgave 8

En utfordring med oppgavearket er at 30 elever ikke har svart på spørsmålet. Av disse 30 er det ingen som har valgt å hoppe over spørsmålet, de har rett og slett ikke blitt ferdig. Dermed er det ikke noe som indikerer at noen av de 30 fant spørsmålet vanskelig, og derfor valgte å droppe det. Hvor mange blant de 30 som ville svart galt sammenlignet med de som har avgitt svar er vanskelig å si, men jeg har sett på et tidligere spørsmål i oppgavearket, spørsmål 5, og sett hva elevene svarte der. På spørsmål 5 hadde 63 av de 73 elevene avgitt et svar, og av disse 63 hadde 54 av dem oppgitt et «korrekt» svar. Denne statistikken kan likevel ikke tillegges for mye tyngde, da spørsmål 5 relaterer seg til resultatet elevene fikk under modelleringsoppgaven, og ikke dreide seg om selve modellen de jobbet med. Om det skulle vise seg å være litt flere av de som ikke har svart som avgir «feil» svar på spørsmål 8 enn blant de som har avgitt svar så vil det fortsatt være en overvekt av «riktige» svar på dette spørsmålet. Det er også verdt å merke seg at det fra den første klassen, som ikke hadde hatt undervisning om fenomenet halveringstid på forhånd så har ingen avgitt «feil» svar, men det er kun rett under halvparten, 6 av 13, som har besvart spørsmålet.

6.3 Fokusgruppeintervju: Kort oversikt og kobling mellom modell og fenomen:

Fokusgruppeintervjuerne er en meget viktig datakilde som forteller mer i dybden om hva elevene tenker rundt spesifikke temaer. Gjennom intervjuerne fikk elevene komme til uttrykk om sine tanker og meninger om programmering og hvordan dette brukes i skolen. De forskjellige intervjuene fortønet seg noe ulikt, og her kommer en kort oppsummering som gjennomgår de ulike intervjuene.

i) Fokusgruppe 1

Denne fokusgruppen var fra den første klassen som gjennomførte opplegget. Klassen hadde ikke hatt undervisning om halveringstid på forhånd, og gikk inn i øvelsen rett etter å ha hatt en annen økt om radioaktivitet som ikke kom inn på halveringstid. Klassen var redusert i antall på grunn av koronapandemien, og fikk dermed god mulighet til oppfølging fra både faglærer og meg selv. Av elevene i klassen var det omtrent halvparten (6 personer mot 7 personer), som rakk gjennom opplegget og svarte på spørsmålene mot slutten. Av disse seks svarte alle elevene «riktig» på spørsmålet om sammenhengen mellom terning-modellen og henfall/halveringstid. Selve klassen arbeidet bra med lite forstyrrelser under selve økten. Under intervjuet kom det frem at to av deltagerne der hadde programmeringserfaring fra før VGS, og to hadde det kun fra VGS. Det var likevel ingen av elevene i intervjuet som bedrev nevneverdig koding på fritiden, og de to med erfaring hadde hatt det som valgfag på ungdomskolen. Elevene oppga blandete følelser om hvordan programmering var blitt undervist dem på VGS, og også rundt følelse av egen kunnskap angående programmering. Elevene anså programmering som vanskelig, men gøy når man fikk det til og mestringsfølelsen dette medførte.

Når det kom til målet for økten, og om elevene trakk frem halveringstid eller programmering så var dette en av klassene som fokuserte mest på programmering, og hvor temaet halveringstid fikk litt mindre plass. Det var en av elevene som kom med temaet halveringstid etter litt samtale og tenkning, men det var fremdeles noen av elevene som fokuserte mest på programmeringsbiten.

Intervjuer: Hva tror dere var målet med økta, hva var læringsmålet?

Tine: Eh.. kanskje lære ting på litt andre måter, eller...

Intervjuer: Ja...

Tor: Bare.. forstå programmering litt bedre.. ved å eh..

Tine (bakgrunn): ja

Tor: Man skulle jo bytte ut litt sånn ting og sånn, putte inn litt sånn tall og sånn.

Tine: Finne ut hva halveringstid er

Intervjuer: Ja, for hva var det du egentlig... hva tror du var det naturfaglige læringsmålet i timen?

Per: Det var vel egentlig hvordan man kunne bruke koding til å gjøre naturfaget enklere

Tine: Finne ut hvordan halveringstiden påvirkes og sannsynligheter og ulikheter og sånn, eh.. og atomkjerner eller.. ja sånn, halveringstid for atomer da... atomkjerner og sånn.

Selv om det var blandet hvorledes elevene så på halveringstid eller programmering som det viktigste målet, så oppga de at programmeringen hjalp deres forståelse av fenomenet halveringstid. Elevene trakk frem overblikket de fikk og muligheten til å se endringer og effekten av dem som viktige momenter for dette.

Kari: Det ga liksom et finere overblikk da, så det var enklere å se det istedenfor å enten lese seg opp eller regnet ut selv

Tine: Ja det ble på en måte regnet ut for oss

Kari: Du måtte endre tallene da, så da så du liksom endringene veldig oversiktlig

Tine (bakgrunnen): Hvordan ting påvirker...

Elevene uttrykte også under intervjuet at de hadde satt pris på denne måten å gjøre programmering på, der det var tettere knyttet opp til et tema, og hvor de fikk benyttet programmering til å gjøre noe som ville vært vanskelig eller tatt lengre tid å gjøre manuelt. Dette sto i kontrast til måten de oppga at de hadde brukt programmering tidligere der de ofte hadde følt at de kunne gjort ting lettere uten bruk av programmering. Dette siste er et poeng som ble gjentatt i andre fokusgrupper fra andre skoler senere.

ii) Fokusgruppe 2

Fokusgruppe 2 var fra den andre klassen som gjennomførte opplegget. De hadde hatt om halveringstid på forhånd, og kjente til fenomenet ganske godt. Klassen var tilnærmet fulltallig, men spredt ut over et større areal på grunn av koronapandemien, og oppfølging fra både faglærer og meg selv ble dermed noe vanskeligere sammenlignet med normalsituasjon. Av elevene i klassen var det omtrent halvparten (9 personer mot 10 personer), som rakk gjennom opplegget og svarte på spørsmålene mot slutten. Blant disse 9 svarte 6 av elevene «riktig» på spørsmålet om sammenhengen mellom terning-modellen og henfall/halveringstid, og 3 av dem svarte «feil». Arbeidsinnsatsen i klassen var variert og det var en del elever som lett ble distraheret eller holdt på med ting utenfor opplegget. Det kom under intervjuet frem at en av deltagerne der hadde mye programmeringserfaring fra før VGS, og fire andre hadde det kun fra VGS. Eleven med erfaring var den eneste som drev med koding på fritiden. Elevene oppga blandete, men generelt negative, følelser om hvordan programmering var blitt undervist dem på VGS. Det ble trukket tydelig frem av noen av elevene at de opplevde programmering som «lite relevant» og at de ofte ikke så nytten med programmering.

På spørsmål om temaet for opplegget ble halveringstid raskt trukket frem av elevene etter litt snakk om programmering og nytten av denne.

Intervjuer: Ja, hva tror du egentlig var meningen med opplegget? Hva var det du skulle lære om?

Nora: Kanskje litt det jeg sa i sted da, å se litt hvor nyttig programmering kan være og hva man kan bruke det til

Intervjuer: Ja, hva var selve temaet?

Jane: Eh.. halveringstid (sagt tydelig og kontant).

Elevene uttrykte også at selv om de kjente temaet fra før, så ga programmeringen og modellen med terningene dem en ny betraktning av halveringstid.

Jane: Nei... jeg lærte bare å se det på en annen måte med liksom terningene.. eller bare en annen måte å visualisere det på. Så jeg tror at hvis jeg ikke hadde visst noe om det tror jeg det hadde vært lett å lære det liksom gjennom de terningene.

Utover i intervjuet kom elevene mer inn på deres tanker rundt programmering, og det kom tydelig frem at det var ulike perspektiver på programmering blant elevene. Der en elev hadde mye erfaring og så for seg å benytte seg av programmering i fremtiden, så var det flere andre

som var usikre på om de ville benytte seg særlig av programmering senere i livet. Disse andre så noe potensiale i programmering, men følte at det burde integreres bedre, og læres skikkelig, hvis de skulle få nytte av det.

Anne: men det er ikke sikkert at programmering er for alle, for det er jo komplisert

Jane: Jeg lurer litt på hva vi skal ta bort for å legge inn programmering, fordi hvis vi skal legge.. jeg syns ikke de burde kutte ut alle de fellesfagene for eksempel, som de tenker på, og så legge heller inn programmering. Da føler jeg ikke det er verdt det på en måte.

Nora: Men å bruke det i fag, sånn som vi gjorde i naturfag da, eller skal begynne med kan jo være lurt, men da er det viktig at vi lærer hvorfor vi gjør det for ofte føler jeg at vi ikke vet hvorfor.. jeg gjør dette her, og det er veldig umotiverende for da føler jeg ikke at jeg trenger dette

Jane: og så føler jeg vi må lære det ordentlig, for det er sånn som når vi hadde programmering i matte hadde vi det liksom en time, skjønte ikke hvorfor vi gjorde det og så kom vi liksom aldri tilbake til det igjen. Så jeg føler det var bare helt bortkastet

iii) Fokusgruppe 3

Fokusgruppe nummer 3 var fra den tredje klassen som gjennomførte opplegget. Klassen hadde hatt litt om halveringstid på forhånd, men det var en stund siden, og ikke alle husket så godt fra temaet. Klassen var fulltallig, og var i sitt vanlige klasserom, og fikk oppfølging fra både faglærer og meg selv. Av elevene i klassen var det halvparten (10 personer mot 10 personer), som rakk gjennom opplegget og svarte på spørsmålene mot slutten. Blant disse 10 svarte 7 av elevene «riktig» på spørsmålet om sammenhengen mellom terning-modellen og henfall/halveringstid, og 3 av dem svarte «feil». Arbeidsinnsatsen i klassen var, som i klasse nummer 2, variert og det var flere elever som lett ble distraheret eller holdt på med ting utenfor opplegget. Det kom under intervjuet frem at to av deltagerne der hadde litt programmeringserfaring fra før VGS, men dette var begrenset til et år med valgfag eller litt arbeid med det på egen fritid. De fire andre elevene hadde ingen erfaring før VGS. Elevene hadde hatt en ok erfaring med undervisning i programmering på skolen, men antok ikke at de ville benytte seg nevneverdig av programmering i fremtidig arbeid.

Elevene i intervjuet uttrykte at de så koblingen mellom modellen og terningene, selv om ikke alle så ut til å være like sikre. Ved sjekk av oppgavearket etterpå var det bare to av elevene i

fokusgruppen som hadde svart «riktig» på spørsmålet som linket modellen til henfall. Av de fire andre hadde en svart galt, og tre av dem ikke svart. Arket til den som hadde svart galt bar tydelig preg av å ha hastet seg gjennom de siste spørsmålene, så det kan hende ting hadde utfoldet seg annerledes om de hadde hatt mer tid til å jobbe på.

Det var flere elever som følte de kunne se nytten av programmering, men var usikker på om det kom til å bli relevant for dem. De ga uttrykk for at de så potensialet i programmering, men at de ikke hadde kunnskapen til å utnytte dette potensialet enda.

Alice: ja, jeg ville ikke brukt det, men jeg skjønner at det er en lettere måte å bruke det på kanskje, hvis man forstår det...

...

Jens: Det som er nå er.. bruker vi jo på en måte.. undervisning til å lære programmering, og ikke programmering som et hjelpemiddel i undervisning

iv) Fokusgruppe 4

Den fjerde og siste fokusgruppen var fra en annen klasse fra samme skole som den første fokusgruppen. Klassen hadde vært borti temaet på forhånd, men ikke alle husket temaet like godt. Klassen var noe redusert i størrelse grunnet strenge sykdoms- og karanteneregler, og var i sitt vanlige klasserom. De fikk oppfølging av meg selv samt to kontaktlærere som var tilstede (der en av disse var faglærer i naturfag). Av elevene i klassen var det omtrent to tredjedeler (13 personer mot 6 personer), som rakk gjennom opplegget og svarte på spørsmålene mot slutten. Blant disse 13 svarte 7 av elevene «riktig» på spørsmålet om sammenhengen mellom terning-modellen og henfall/halveringstid, og 6 av dem svarte «feil». Arbeidsinnsatsen i klassen var jevnt over god, og elevene jobbet godt sammen. To av deltagerne under fokusgruppeintervju hadde programmeringserfaring fra før VGS, da som valgfag på ungdomskolen, en av disse to hadde mye erfaring og programmerte selv på fritiden. De to andre elevene hadde lite erfaring før VGS. Selv om de hadde vært borti konseptet blokkbasert programmering, så kjente de ikke igjen dette som programmering før en av de to andre deltagerne nevnte det som sin egen erfaring. Elevene hadde hatt en ganske bra erfaring med undervisning i programmering på skolen, selv om de følte de hadde hatt lite av det, og ønsket seg mer. De hadde ikke benyttet programmering i fag utenom

matematikken, men hadde likt oppgavene derfra. På spørsmål om målet med økta trakk elevene primært frem programmering og ikke fenomenet halveringstid.

Intervjuer: Men hvis vi ser på økta, hva tror du ville vært målet at du skulle lært etterpå?

Maren: Kanskje å bruke programmering i naturfag istedenfor bare matte som vi er vant til

Ida: Til noe praktisk

Sven: Det er for å vise at man kan bruke koding til forskjellige ting

På videre oppfølgingsspørsmål kom det frem at elevene i fokusgruppen kjente til fenomenet ganske godt, og dermed ikke anså å lære om dette som det viktigste målet for timen, men at de heller fokuserte på det nye som var å benytte programmering til å simulere noe. Dette stemmer godt overens med svarene fra oppgavearkene der tre av de intervjuede hadde svart riktig på koblingen mellom modellen de jobbet med og radioaktivt henfall, mens en ikke hadde rukket å svare.

Når det gjaldt nytten og mulighetene programmering medbrakte så kunne elevene se den, men de følte at de ikke hadde lært nok programmering enda til å kunne benytte seg av disse enda.

Ida: Personlig så føler jeg at hovedproblemet er at vi ikke kan nok om programmering, sånn hvis hadde startet med det tidligere så..

Intervjuer: Så kunne du gjort «nyttige» ting nå?

Ida: Ja, for eksempel, for nå har vi hatt det litt under et år, og vi har jo ikke hatt det hver eneste time. så det blir litt sånn..

Maren: Sånn overfladisk

6.4 Motivasjon og relevans

Under fokusgruppeintervjuerne var det flere elever som kom med utsagn som omhandlet motivasjon, eller relevans. På direkte spørsmål om hvordan elevenes motivasjon ble påvirket av programmering var det 4 som svarte at det bidro positivt til deres motivasjon, 5 svarte det bidro negativt, og 8 personer svarte noe mellom de to ytterpunktene. De som svarte noe

mellem positivt og negativt trakk gjerne frem at det kom an på oppgaven, og om de fikk til det de skulle gjøre. Faktisk kom alle fokusgruppene inn på viktigheten av å «få til» kodingen, og hvordan det å skjønne eller mestre noe bidrar til motivasjon. Elevene var her stort sett enige på tvers av alle fokusgruppene, selv om de hadde ulike erfaringer med å få til og mestere kodingen selv.

Fokusgruppeintervju 1:

Tine: det går liksom litt bedre etter tid når man forstår det litt mer.. eh.. men i starten var det litt umotiverende å måtte sitte ned og skrive egne koder og.. prøve å finne ut hva det er på en måte

Fokusgruppeintervju 2:

Jane: det er veldig demotiverende å gjøre ting man ikke forstår syns jeg.

Fokusgruppeintervju 3:

Elsa: ja, jeg er egentlig enig. Altså det kommer veldig an på om man får det til eller ikke hvor motiverende det er, og på en måte hvor gøy det blir å gjøre det.

Fokusgruppeintervju 4:

Maren: Jeg liker bare programmering hvis jeg får det til, da er det veldig gøy, så det er litt både og.

To elever trakk også frem gleden og motivasjonen ved å få en utfordring de kunne bryne seg på. Disse elevene var relativt sterke innen programmering, og bedrev det aktivt på egen fritid.

Fokusgruppeintervju 4:

Intervjuer: Hvordan påvirker programmering motivasjonen din for et fag?

Johan: Det spørs hvilke fag, i realfag er det mye mer interessant.. og få en utfordring, og finne ut hvordan du skriver kode til å løse et problem

Fokusgruppeintervju 3:

Frank: At det var spennende å prøve å lage.. eller bygge videre på den ene koden, og prøve å gjøre at den kunne repetere seg selv

Intervjuer: Ja, kom du i mål med det?

Frank: Nei, men det var fortsatt gøy å prøve

Flere elever kom også inn på om de opplevde programmering som relevant eller motiverende for dem. Elevene trakk frem, eller kom med indikasjoner på flere former for relevans, eller mangel på dette. Det mest vanlige elevene snakket om var deres egen interesse for, og fornøyelse ved, programmering. Dette ble kategorisert som «individuell relevans, nåværende, iboende». Det er markert 26 tilfeller av denne koden og så godt som alle elevene kom med et eller flere utsagn relatert til dette. Det var god spredning i meningene til elevene på dette området, og det var tydelig at det var mange forskjellige nivåer av interesse for programmering hos elevene.

Fokusgruppeintervju 2:

Ole: Jeg syns jo det her var kjempegøy.. fordi jeg syns programmering er gøy til å begynne med. Så det at jeg også hadde noe erfaring i programmering eh.. ja, jeg syns det her var.. mye morsommere enn å sitte og trille terninger for hånd hvert fall

...

Anne: Men det er ikke sikkert at programmering er for alle, for det er jo komplisert.

Fokusgruppeintervju 1:

Kari: Nei, altså jeg syntes det var veldig gøy. Hvert fall sånn som vi jobbet nå med det da.

Fokusgruppeintervju 3:

Elsa: ja, jeg er egentlig enig. Altså det kommer veldig an på om man får det til eller ikke hvor motiverende det er, og på en måte hvor gøy det blir å gjøre det

Dina: For å være helt ærlig syns jeg egentlig ikke programmering er så kjempegøy selv om jeg får det til heller, jeg bare syns ikke det er så interessant

En ting som gikk igjen flere ganger under intervjuene var om elevene opplevde at de kunne nyttiggjøre seg programmering, både nå og i fremtiden. For nytte ved bruk av koding på nåværende tidspunkt er det blitt kodet som «individuell relevans, nåværende, ytre», og det er 18 tilfeller av denne koden. Her sa noen elever tydelig ifra om at de ikke opplevde programmering som nyttig for dem selv på nåværende tidspunkt, mens andre så muligens noen fordeler, men også potensielle utfordringer.

Fokusgruppe 2:

Jane: Jeg føler det er litt.. når jeg jobber med programmering føler jeg liksom at det er sånn forbigående sånn som når vi hadde det i matte så tenkte jeg liksom at det har ikke så mye å.. jeg gadd ikke å sette meg inn i det for jeg visste at vi ikke skulle bruke det videre.

...

Jane: jeg har ikke opplevde det som nyttig, men jeg kan se for meg at det kan være nyttig i flere situasjoner, men jeg har ikke vært i den situasjonen selv hvor jeg har tenkt sånn, nå burde jeg kunne programmert.

Fokusgruppe 3:

Hans: Jeg ser ikke helt behovet med programmering, for min del da.

Fokusgruppe 4:

Maren: Det gir mening, og jeg kan se for meg at det blir brukt til andre ting, men ikke når vi gjør det. Vi finner ikke ut noe som har betydning.

For fremtidig nytte og bruk er dette kodet som «individuell relevans, fremtidig», og vi har 5 tilfeller av denne koden. Vi har i tillegg 4 tilfeller av koden «Yrkesmessig» som omhandler hvorvidt elevene ser for seg programmering som karrierevei, eller om de tror det vil mye inn i fremtidige yrker. Det er her verdt å få med seg at flere elever viser usikkerhet rundt hvordan programmering kan bli mer relevant i flere yrker i fremtiden, og at inntrykket er at de ofte bare ser for seg typiske programmeringsyrker som utvikler, eller web-designer når de tenker programmering som karrierevei.

Fokusgruppe 1:

Per: antageligvis ikke.. eller hvis du skal jobbe med mye.. sikkert sånn web og sånn, så kommer sikkert det til å bli relevant.

Fokusgruppe 2:

Jane: Jeg føler hvert fall det er bra at det tilbys som et valgfag, for det er jo ikke alle yrker det blir relevant å ha programmering. Det er jo akkurat som at man kan jo velge å ta kjemi.. fordi det kommer jo ikke til å være relevant i alle fag.

Flere elever mente også at de burde lært programmering tidligere i sin undervisning, og at dette ville medført at de så på programmering mer som en sentral del av deres undervisning. De tok opp at de frem til VGS har lært seg å jobbe uten å bruke programmering, og nå skal de

plutselig lære å jobbe med bruk av programmering, og at det dermed oppleves å komme litt på toppen av alt det andre de har lært fremfor å være sidestilt med dette.

Fokusgruppe 2:

Anne: I så fall må vi på en måte lære mer om programmering da, hvis det skal brukes, fordi nå er det nytt.. da blir det liksom nå har vi lært oss det uten, og så skal vi lære oss enda en ting. Det hadde vært enklere hvis det på en måte var litt sånn grunnleggende..

Nora: Og at man begynte med det tidligere.. så det ble litt mer vanlig

Jane: ja

Intervjuer: Når.. hvor tidlig tror du er tidsnok?

Nora: Jeg tenker egentlig på barneskolen jeg.. fordi at hvis det blir liksom vanlig er det enklere å forstå det.. og så er vel da man begynner å lære best er det, når man er litt ung?

Fokusgruppe 3:

Hans: Jeg føler kunne vært mer nyttig hvis vi hadde hatt mer kunnskap, nå er det litt sånn.. nå starter vel folk tidligere med det, mens vi starter jo alt nå på et år

...

Hans: Hvis du hadde spurt de som er ti år yngre enn oss nå om ti år så tror jeg svarene hadde blitt ganske annerledes

Elevene sa ikke så mye direkte om deres opplevelse av eierskap til det de jobbet med, men de kom med noen utsagn rundt forståelse og om de følte de hadde kontroll over det de lærte.

Noen elever var opptatt av å skjønne i grove trekk hva koden gjorde, mens andre var litt mer opptatt av å ha orden på de mindre detaljene. Det var ingen som mente at man behøvde å forstå absolutt alle linjene for å forstå hva et program gjorde, men det var noen som trakk frem dette som viktig om man skulle gjenskape koden selv senere.

Fokusgruppe 2:

Jane: Jeg forsto veldig godt hva programmet gjorde selv om jeg ikke visste hva alle linjene betydde.. jeg skjønte jo liksom.. hva svaret betydde, og hvorfor det var sånn, jeg følte ikke det var relevant å vite hva liksom.. print x.. bla, bla, bla, hashtag betyr på en måte

Fokusgruppe 3:

Frank: Når du endret på den ja, og så føler jeg at hvis man får lest en kode og så ser hva den gir i output så forstår du også litt mer av koden, og da lærer du også av det.

Fokusgruppe 4:

Per: Jeg synes det er bra å få kode, og lese koden og forstå koden istedenfor å skrive koden, for jeg vil si forståelse av kode er like viktig som å skrive koden

...

Sven: Jeg føler at hvis du skal forstå hva koden gjør så må du jo kunne så si alle.. eller ha en forståelse av hva hver linje gjør liksom

Maren: Eller hvert fall hvis du skal skrive den selv da, men hvis du skal lese den selv så burde du kanskje forstå de viktigste

Per: Ikke nødvendigvis alle linjene, men selve funksjonene

6.5 Utfordringer

Elevene trakk frem flere utfordringer de opplevde når det kom til programmering. Det de trakk frem flest ganger var at det var mye nytt å lære seg, og holde oversikt over. Dette ble uttalt fra både elever uten programmeringserfaring før VGS, og av elever som hadde gjort dette før VGS. Elevene trakk også frem at man ofte må tenke annerledes når man programmerer, og at programmering kan oppleves som et nytt element som øker vanskelighetsgraden til en oppgave fremfor å gjøre den lettere. Dette er alt kodet under «kognitiv belastning», som er en kode som forekommer 17 ganger.

Fokusgruppeintervju 1:

Per: komplisert

...

Tor: Mye å huske

Fokusgruppeintervju 2:

Jane: Jeg vet ikke helt om det er verdt å liksom lære å gjøre sånne ting med programmering, fordi du må jo ofre mye tid du kunne brukt på en del andre ting for å lære deg programmeringen ordentlig.

Fokusgruppeintervju 3:

Elsa: Jeg tror man må skjønne programmering for at det skal være lettere, fordi hvis man ikke gjør det så blir det bare enda en ting til som kan gjøre det vanskeligere. Så for de som skjønner på en måte tema og som kan programmering så tror jeg det vil være lettere, men hvis ikke så tror jeg ikke

...

Alice: Så er det en helt ny måte å gjøre ting på så det blir jo på en måte ganske mye hvor man må tenke annerledes.

Flere elever kom også innom forskjellige programmeringstekniske utfordringer som kan dukke opp når man programmerer. Både når det gjaldt å kjenne til metoder og kommandoer, men også når det kom til mer enkle ting som skrivefeil og feilsøking kom elevene inn på at utfordringene man kan møte ved programmering er mange.

Fokusgruppeintervju 1:

Tine: Jeg synes den siste.. var litt vanskelig å forstå, men jeg skjønte de andre, men det var litt mye metoder der som vi ikke har lært enda.

Fokusgruppeintervju 2:

Ole: fordi å skrive kode tar lang tid, skriver du en ting feil så kan du ødelegge alt

Jane: Ja i sted skrev jeg noe med stor p og så funket det ikke

Anne: Jeg gjorde det samme

Fokusgruppeintervju 3:

Jens: Og så er det jo veldig kjipt hvis man printer noe feil, eller skriver noe feil, for da blir jo nesten hele greia feil.. man skal være veldig fokusert

Under intervjuene ble elevene også direkte spurt om de opplevde at programmeringen distraherete dem i undervisningen av selve naturfenomenet, eller skapte støy i undervisningen på noen måte. Det var ikke så mange som sa noe konkret her, selv om noen kom inn på at det

selvfølgelig gir mer utfordringer og vanskeligheter som kan distrahere. Fokusgruppe 3 var den eneste klassen som rett ut sa at det tar tid, og kan ta fokus, bort fra naturfag når man benytter programmering.

Fokusgruppe 3:

Jens: Det blir mindre naturfag og mer programmering

...

Dina: Jeg ser på programmering mer som matte enn naturfag egentlig

6.6 Opplegget og tidligere undervisning

Elevene ble også spurt om opplegget de gjennomførte, og tidligere undervisning i programmering. Elevene uttrykte jevnt over at de likte opplegget de jobbet med, selv om det var noen som følte det kunne vært forklart litt mer om fenomenet halveringstid før de begynte å jobbe med oppgaver om det. Det var elever som sa de måtte søke opp begrepet da de traff på det i oppgaveteksten. Noen oppga derimot at de kjente begrepet fra før, som kom av at de hadde gjennomgått temaet tidligere.

Fokusgruppe 1:

Kari: Det vi jobbet med? Jeg synes det var veldig bra jeg sånn...

Per: jeg synes også det var greit.

Tor (bakgrunn): jeg synes også det var bra.

Kari: veldig mye sånn oppfølgingsspørsmål, så man liksom.. skjønte mye mer på en måte da.

...

Tor: Jeg ble veldig forvirret når jeg leste oppgaven.

Tine: (ler litt), det var nytt.

Tor: Jeg måtte google, og så fant jeg ut hva det var.

Fokusgruppe 2:

Jane: Hvis jeg ikke hadde visst så godt hva halveringstid (er) fra før av, tror jeg ikke jeg hadde forstått oppgaven så godt

Anne: Nei en måtte vite hva det var fra før av, men det ga meg litt mer sånn eh.. sånn forståelse på hva egentlig halveringstid er når man får det mer sånn fysisk

Elevene ga ellers uttrykk for at de hadde fått hjelp og støtte når de behøvde det. De satte også pris på nettmodulen som de kunne utforske om de sto fast eller trengte mer informasjon.

Totalt sett virket det som elevene hadde et godt inntrykk av oppgavene og modulen, selv om de fleste syntes oppgave 6, som ba dem lage egen algoritme, var litt for utfordrende i forhold til deres nivå.

Fokusgruppe 1:

Tine: Jeg likte på en måte.. hvordan det var lagd, at man leste på arket og så skulle man gå over på nettsiden og så forklarte det litt mer der også og sånn.

Fokusgruppe 2:

Intervjuer: Hva syns dere om selve nettmodulen?

Nora: Den var veldig fin egentlig.. eller sånn.. opplegget at man kunne finne løsningsforslag og sånn, hvis det er det du mener

Jane: ja jeg likte også den, det var veldig enkelt å liksom.. skjønn hva alt betydde

Fokusgruppe 3:

Dina: Jeg følte i hvert fall at alt sto veldig sånn punktvis, da er det lett å følge med

Et flertall av elevene ga også uttrykk for at de syntes utfordringen de fikk var «passe» vanskelige, og ga dem en god utfordring, selv om det var noen som syns det både var for vanskelig og for lett. De som syns det var ganske lett fikk likevel en utfordring i oppgave 6, så det var ingen som uttalte at hele oppgavearket var enkelt å løse.

Fokusgruppe 1:

Per: Utfordringen var ganske.. følte jeg var litt sånn gradvis.. hvor jo flere oppgaver du gjorde så ble det litt mer komplisert.

Fokusgruppe 4:

Sven: de oppgavene jeg har gjort inntil nå har vært ganske passe, men nå er det oppgave seks som..

Johan: Jeg vil si det er den beste, fordi den.. du skriver jo ikke så mye kode på de andre, det er mer tolkning og forståelse av selve temaet

Et moment elevene tydelig uttrykte at de satte pris på var å jobbe i nettmodulen med Trinket, og få utdelt mye av koden. Det var bred konsensus om at et fåtall eller ingen hadde klart å lage programmene helt på egenhånd med sin nåværende kunnskap, og at det for dem var nødvendig med starthjelp. Med såpass få linjer og kommandoer som elevene skulle skrive inn selv ble det også relativt få problemer med syntaks eller andre typiske programmeringsutfordringer underveis, selv om det fortsatt forekom et par ganger. Noen elever trakk også frem muligheten til å endre på variabler og å raskt kunne teste hva som skjer når du gjorde endringer i programmet. En siste ting noen elever kom med var den økte presisjonen som kom med programmering, og hvor gode datamaskiner er til å regne.

Fokusgruppeintervju 1:

Intervjuer: Hva synes dere om å få utdelt kode fremfor å skulle skrive alle linjer selv?

Per: Var veldig deilig

Tine: mye deiligere, jeg hadde ikke forstått det hvis jeg måtte skrevet alt selv

Tor: jeg hadde brukt ganske lang tid og hadde brukt mer tid på å skrive ned koden enn å løse oppgaven

...

Tine: det er veldig mye sannsynlighet og at man skal lage litt sånn løkker og at programmet skal kjøre seg selv om igjen og om igjen og finne ut av ting.

Tor: Altså mye mer sånn tall og.. mer sånn nøyaktige beregninger og litt sånne ting

Fokusgruppeintervju 3:

Intervjuer: Hva synes dere om selve nettmodulen, hvordan var det å få masse linjer med kode?

Dina: Det var bra synes jeg

...

Jens: Og du får testet litt hva de forskjellige tingene gjør i tillegg, som når du fikk testet ut hva $n =$ når du endret på den

Frank: Når du endret på den ja, og så føler jeg at hvis man får lest en kode og så ser hva den gir i output så forstår du også litt mer av koden, og da lærer du også av det.

Fokusgruppe 4:

Maren: Kan jo bli mer nøyaktig kanskje, jeg vet ikke..

...

Johan: Jeg synes det er bra å få kode, og lese koden og forstå koden istedenfor å skrive koden, for jeg vil si forståelse av kode er like viktig som å skrive koden

Når elevene ble spurt om tidligere undervisning de hadde hatt innen programmering var kommentarene blandet, men ofte litt negative. Det var de som mente det gikk for fort, og de som mente det gikk for sakte. Noen elever mente også at lærerne ikke hadde hatt nok kompetanse, og at det hadde gjort ting vanskeligere.

Fokusgruppeintervju 1:

Tine: Men.. ja.. eller det som er, i alle fall i matte, er at man går igjennom temaer så veldig fort, så man gikk egentlig veldig raskt gjennom disse programmeringstingene så det satt liksom ikke ordentlig.

Fokusgruppeintervju 2:

Jane: Det var vanskelig å følge med, at hvis de skal undervise i programmering burde de ha flere lærere i timen som har god kompetanse i det

Nora: Ja, for det virket ikke som læreren kunne så mye da

Jane: Vi hadde bare en lærer hos oss som kunne, «lærernavn» kunne jo ingen ting. Vi hadde besøk av en lærer, men det var litt vanskelig for han sto liksom og prata og..

Nora: Og så fikk vi heller ikke vite hvordan vi kan bruke.. få bruk for det, så da virket det litt meningsløst

Fokusgruppe 4:

Ida: Det er gøy å få det til synes jeg hvert fall, og jeg synes.. undervisningen har vært grei, men det er lett å.. lett å falle av også, fordi det er veldig mye

Maren: Spesielt når vi hadde over teams, da var det veldig vanskelig å følge med

Sven: Og så har vi hatt ganske lite av det og, føler jeg

Per: Jeg føler det burde vært mer aggressivt, mer.. dypt inn i læreplanen da

Elevene kom også inn på at noen oppgaver de hadde jobbet med i tidligere undervisning ofte hadde vært ganske simple, uten å være spesielt knyttet opp mot naturfag eller andre ting fra virkeligheten. Det var en klasse hvor elevene trakk frem at de hadde fått opplevd programmering benyttet til mer avanserte oppgaver som var mer knyttet til temaer de hadde om, og at det hadde føltes mer relevant og nyttig.

Fokusgruppe 4:

Ida: Vi har jo bare hatt sånn båt-tall, hus-tall, lag en kode som skriver ut..

Intervjuer: lag en kode som skriver ut noe?

Ida: Ja, det er jo bare det vi har hatt

Fokusgruppe 3:

Alice: De har vært litt relevant da for vi har hatt litt om sånn bakterier og.. hadde vi ikke det i en oppgave med sånn veksten av bakterier og sånn.. men ellers så er det litt sånn..

Jens: Men jeg føler de er jo relevante liksom i den forstand det er jo å trekke til det temaet vi har da på skolen, og det skal liksom løse oppgaver som vi har i boka, bare gjennom programmering da

Frank: Og i starten var det litt mer basic bare sånn for å.. legge til tall og sånn, og det var jo ikke så relevant, men det er jo selvfølgelig relevant etter hvert

7) Diskusjon

I diskusjonen vil jeg se på og belyse flere interessante elementer fra opplegget og dataene som er samlet inn. Jeg vil ta for meg hvordan elevene opplever modellen for halveringstid som de jobbet med, og om de forsto denne. Jeg vil også se nærmere på om programmeringen de benyttet seg av kan ha distraheret elevene fra det som var fokuset for timen, nemlig halveringstid. Utfordringer elever støter på under programmering, og hvordan opplegget de hadde endret forekomsten av disse utfordringene vil også blitt tatt nærmere i øyensyn. Muligheter elevene får gjennom programmering vil også bli diskutert. Mot slutten vil jeg se nærmere på om elevene opplever programmering som relevant, og komme inn på ting vi kan gjøre for å øke denne. Det vil også bli foreslått noen forbedringer til undervisningsopplegget jeg gjennomførte, basert på erfaringene som ble gjort da disse ble gjennomført.

7.1 Modellen for halveringstid

Elevene jobbet i opplegget med en modell som skulle simulere et fenomen fra virkeligheten, halveringstid. Det er derfor viktig for læringsutbyttet at elevene ser denne koblingen, og forstår hva modellen de jobbet med faktisk skulle simulere. Modellen vi hadde benyttet seg også av såkalt agent-basert modellering, der elevene fikk modellert hver radioaktiv isotop på mikronivå, for så å kunne observere effekten fra disse på makronivå. Det er derfor to potensielle steder der sammenhengen mellom modellen og halveringstid blir utydelig. Den første dreier seg om sammenhengen mellom mikro- og makronivå. I vårt tilfelle hva endringer i sannsynlighet på mikronivå førte til i endringer av halveringstid på makronivå. Denne sammenhengen ble elevene spurt om i oppgaveteksten, og vi vil snart se litt nærmere på dette. Det andre stedet der sammenhengen kan bli utydelig relaterer seg til om programmeringen elevene bedriver distraherer dem fra hva de faktisk jobber med, og om dette ekstra laget mellom dem og fenomenet gjør det vanskeligere å se sammenhengen.

Ved gjennomgang av oppgavearkene til elevene var det mange (31) som hadde sett sammenhengen, men ikke alle. Blant de som har svart «galt» (12) på denne sammenhengen er det flere forskjellige svar, og det er ikke noe spesielt mønster som utpeker seg. Det må også sies at et spørsmål ikke nødvendigvis beviser at man forstår sammenhengen, selv om det gir en indikator på dette. Et viktig ankepunkt ved disse dataene er at mange (30) elever ikke rakk

å svare på dette spørsmålet. Det kan dermed bety at om flere hadde rukket å svare ville vi fått flere «gale» svar, men dette er heller ikke sikkert. Ved gjennomgang av andre, tidligere spørsmål, der flere har rukket å svare finner vi at prosentandelen «rette» svar er tilnærmet lik det vi fant for det senere spørsmålet som gikk på sammenheng mellom modellen og naturfenomenet. Det er også viktig å få med seg at en av klassene hadde jobbet med akkurat den samme modellen før, men uten bruk av programmering. I denne klassen var det litt flere som rakk å skrive noe om sammenheng mellom modell og naturfenomen, men her var det faktisk flere som svarte «feil» enn i andre klasser. Dette er en interessant observasjon, som kan peke mot at tidligere arbeid med en modell ikke nødvendigvis betyr at elevene forstår og husker modellen senere. Likevel er det viktig å ikke trekke for mange konklusjoner fra denne observasjonen siden det statistiske sammenligningsgrunnlaget er meget lite, og resultatet kan være tilfeldig.

Når det gjelder spørsmålet om programmeringen var med på å distrahere elevenes fokus bort fra det bakenforliggende temaet halveringstid, og skape «støy», så har vi både ting som indikerer, og argumenterer mot dette. På direkte spørsmål om programmeringen var distraherende svarte de fleste elever at det ikke påvirket dem så veldig mye, men vi har noen unntak. Spesielt i klasse nummer 3 var det et par elever som kom inn på at programmering kan ta bort fokuset fra naturfag, og en elev sa rett ut at det ble «mindre naturfag og mer programmering». Å måtte bytte mellom ulike arbeidsmoduser er en utfordring programmering tar med seg når man bringer det inn i andre fag. Under arbeidet med opplegget observerte jeg flere elever som tydelig var i en programmeringsmodus, som Sørby og Angell navngir det i sin tekst, der de benyttet seg av mer prøv-og-feil teknikk og eksperimenterte med koden sin. Dette er ikke udelt negativt, og kan føre til mer utforskning fra elevenes side, så lenge man ikke lar det distrahere elevene fra det endelige læringsmålet. Når vi ser på oversikten for hvilke typer «gale» svar elevene har gitt på sammenhengen mellom modell og fenomen, er det kun to av disse som kan knyttes opp mot programmering. Vi kan derfor ikke fra spørreundersøkelsen se at programmeringen har forstyrret elevene. Vi må likevel huske at den er ufullstendig, og at om vi hadde svar fra alle kunne bildet sett annerledes ut. I tillegg er antallet deltagere i vår undersøkelse lavt, så det er viktig å ikke tillegge svarene herfra for mye vekt.

Når elevene ble spurt mer indirekte om programmering distraherer dem, kom det også frem interessante momenter. På spørsmål om målet for økten elevene nettopp hadde gjennomgått valgte flere elever å fokusere på, og trekke frem, programmering. Noen elever oppga at dette

var fordi de kjente til halveringstid fra før, så de valgte å fokusere på det som hadde vært nytt i økten, nemlig programmering. Likevel valgte også klassen som ikke kjente til fenomenet fra før å trekke frem programmering når de snakket om målet for økta. Selv etter at noen i denne fokusgruppa hadde kommet inn på halveringstid, var det elever her som fremdeles fokuserte på programmering og dens nytte i naturfaget. I en av klassene var det en elev som tydelig trakk frem at temaet for opplegget var halveringstid, men før det var det i denne fokusgruppa også noen som trakk frem programmering som et viktig mål for økten. Basert på dette kan det virke som at programmering har hatt en noe mer «støyende» effekt enn det elevene selv tror. Det er tydelig at for flere av dem er programmeringen det første som kommer frem når de skal huske momenter fra, og gjengi, timen. Det er vanskelig å si hva som er grunnen til dette, og jeg skal være forsiktig med hva jeg her antar, men et mulig scenario er at elevene i økten jobbet med naturfag og programmering på en måte som er helt ny for dem. Det er dermed lettere å fokusere på programmeringen som oppleves som ny og spennende, mens forståelsen av en modell er noe de vil ha sett før i naturfag, så det er ikke like spennende. En ting som kan støtte litt opp under dette er at fokusgruppen fra klassen som hadde benyttet seg av programmering i naturfag før, også var den som trakk frem at programmering kunne være distraherende for den bakenforliggende naturfagskunnskapen.

Jeg vil derfor si at det ikke er helt enkelt å svare på om programmering distraherer, og tar fokuset bort fra det som skal undervises ved hjelp av programmering, men jeg vil si at dataene mine viser at det er en mulighet. For å kunne svare bedre på spørsmålet rundt dette vil det være nødvendig med mer forskning, og da gjerne med elever som i fremtiden, takket være de nye læreplanene, vil kjenne bedre til programmering. Fremtidige elever vil også ha mer erfaring med å benytte programmering innenfor andre fag, så det vil være en mindre nyhetsfaktor over å introdusere elevene til et slikt opplegg som gjennomført under arbeidet med denne oppgaven.

7.2 Programmeringsutfordringer

En annen ting som vil ta bort fokus fra det man jobber med er å bli distrahert av programmeringsutfordringer. Om man opplever at man hele tiden må rette feil, eller bruke mye av tiden på å jobbe med syntaks og andre programmeringstekniske utfordringer, vil det være større sannsynlighet for at man ikke tenker på modellen man faktisk programmerer.

Det er mange potensielle utfordringer som dukker opp når vi skal programmere, og elevene i fokusgruppene valgte å trekke frem noen av dem. Flere valgte å peke på at det er mye å lære seg, og at det tar mye tid å oppnå forståelse for programmering. Dette stemmer godt overens med annen forskning fra blant annet Taub et. al. og Sørby som peker på den kognitive belastningen som en stor utfordring med programmering. Et annet moment som Jenkins trekker frem som viktig innen programmering er at man ofte må kombinere overfladisk og dypere læring. Dette trekkes frem av et par elever som nevner at programmering er en annen måte å tenke på, og at det ikke bare er mange metoder å passe på, men også når og hvor man bruker disse.

En siste stor utfordring med programmering er syntaks og feilsøking. Dette er en del av programmeringen som ofte kan oppleves som vanskelig og frustrerende for de elevene som sliter med dem. Elevene i fokusgruppene kom innom at den minste feil, som for eksempel stor bokstav kontra liten bokstav i et variabelnavn, ville gi dem syntaksfeil og at det å lete etter disse var vanskelig. Her er det veldig forskjellig mellom de ulike programmeringsspråkene hvor mye hjelp du får til å finne feil. Noen språk er ganske flinke til å markere hvor de mener en feil ligger, og hva slags type det er, mens andre språk gir deg mindre hjelp. Om man benytter et språk som gir gode feilmeldinger og gir tydelige instruksjoner til elevene kan feil av denne typen reduseres noe, men det vil alltid snike seg inn mindre skrivefeil eller annet når elevene jobber, så det er viktig å gi elevene trening i å tolke og forstå feilmeldinger de får slik at de kan håndtere dem på egenhånd.

På spørreskjemaet som elevene fylte ut var det også mange som oppga at de anså programmering som vanskelig. Dette gjaldt spesielt gruppen som besto av elever som ikke hadde egen erfaring med programmering før de begynte i den videregående skolen. Det virker rimelig å anta at de som har mer erfaring etter hvert vil begynne å forstå mer, og oppfatte programmering som mindre vanskelig. Det kan også bety at flere av de som fant programmering vanskelig kanskje vil oppleve det som litt mindre vanskelig i fremtiden når de får tilegnet seg mer erfaring. Et ankepunkt mot denne ideen er at de elevene som har programmert før de begynte på VGS, gjerne er mer motiverte for å programmere, og dermed også mer villige til å legge ned innsats for å lære programmeringen. Det må også huskes på at gruppen med elever som hadde programmert før VGS er liten, så man kan ikke trekke for bastante slutninger ut fra data fra denne gruppen.

Et gledelig moment å ta med seg var at når elevene trakk frem utfordringer med programmering, så snakket de stort sett om opplevelser fra tidligere programmering de hadde hatt, og ikke så mye om utfordringer de støtte på under arbeidet med opplegget. Noen nevnte at de hadde fått feil i syntaks, og at de ikke hadde kommet gjennom alt de arbeidet med, men så godt som alle uttrykte at de likte opplegget, og at de hadde funnet det fint å jobbe med. Det kom også frem at de hadde satt pris på den trinnvise oppbygningen som Sørby trekker frem som et tiltak for å lette prosessen for elevene. Ferdig utdelt kode, som Brown og Wilson trekker frem ble også verdsatt av elevene, og de fleste ga uttrykk for at de ikke hadde nivået på nåværende tidspunkt til å programmere det de jobbet med fra bunnen av selv.

Nettmodulen, hvor elevene kunne finne mer om oppgaven og fasit, ble også trukket frem av elevene som noe de satte pris på og benyttet seg av. Sett under ett kan det virke som at bruken av nettmodul, og mye ferdig kode kan hjelpe elevene til å komme i gang, og sørger for god stilas-bygging for de elevene som har noe mer utfordringer med programmering. Vi kan dermed si at oppgaven og modulen er med på å utvide elevenes utviklingssone, slik at den treffer flere elever. Dette må likevel ikke bli en sovepute for all programmering fremover, da det kan føre til at elevene opplever programmering som en «kode-maskin» der de bare benytter seg av det de får, eller alltid skal fullføre et påbegynt program. Noen elever oppga i fokusintervjuet at de så på programmering som en avansert kalkulator som de kunne benytte til å regne ut mer avanserte ting. Dette er for så vidt en nyttig anvendelse av programmering, men det bør ikke være hele målet for programmering. Målet er jo at elevene skal kunne programmere på egenhånd der de selv identifiserer metoder, variabler og selve modellen de jobber med og vi burde stadig utfordre elevene til å komme et skritt nærmere dette hver gang de programmerer.

7.3 Muligheter

Programmering medfører også flere muligheter som kan komme elevenes læring til gode. Som nevnt av Haraldsrud et al. får elevene bedre anledning til å undersøke og teste modellen de jobber med selv. Jeg så under arbeidet med opplegget at flere elever benyttet seg av en prøve-og-feile modus som beskrevet av Angell og Sørby, og at de hadde lettere for å teste ut variabler og annet i koden sin. Denne utforskningen er nyttig for at elevene skal kunne lære mer om og forstå modellen de arbeider med, og må sies å være en stor fordel elevene får ved

å benytte programmering. Det var ikke alle elevene som var linke flinke til å bedrive denne utforskningen, og flere av de svakere elevene var gjennomgående mer redde for å «ødelegge noe» hvis de endret på ting. Det vil derfor være viktig at læreren bidrar til å gjøre det trygt for elevene å prøve seg litt på egenhånd, og hjelpe dem med å rette eventuelle feil som oppstår, slik at de er mer villige til å utforske mer i fremtiden.

Noen elever oppga at det å programmere ga dem et nytt perspektiv på fenomenet som ble undersøkt. For de elevene som oppga dette dreide det seg da om at selve modellen de jobbet med var ny for dem, og ga dem ny innsikt knyttet til halveringstid. Man kan derfor si at dette perspektivet for akkurat denne oppgaven kunne oppnås ved å benytte denne modellen uten programmering også. Fordelene med programmering er likevel at vi kan gjøre ting raskere og enklere, og som Haraldsrud et al. trekker frem, så kan vi fjerne matematiske begrensninger som hindrer elevene fra å benytte og studere andre avanserte modeller. I mekanikk i fysikken får elevene plutselig mulighet til å ta med luftmotstand når de jobber, og innen naturfag kan vi benytte oss av større datasett eller simuleringer som tar hensyn til flere variabler.

Vi kommer her også innom hva som er tanken bak å lære programmering. Handler det om å lære programmering som en ferdighet i seg selv, eller er programmering mer som et verktøy som vi benytter til å lære andre ting? Om vi ser på læreplanen i naturfag er det et tre kompetansemål jeg vil trekke frem som kan si oss noe om hvordan programmering er tenkt brukt innenfor naturfaget. De tre kompetansemålene er:

- Vurdere og lage programmer som modellerer naturfaglige fenomener
- Utforske en selvvalgt naturfaglig problemstilling, presentere funn og argumentere for valg av metoder
- Drøfte hvordan utvikling av naturvitenskapelige hypoteser, modeller og teorier bidrar til at vi kan forstå og forklare verden

(Utdanningsdirektoratet, 2020)

Om vi ser på det første av disse kompetansemålene, som også er det eneste som har ordet programmering skrevet direkte i seg, så forteller dette kompetansemålet oss at programmering er tenkt benyttet til arbeid med modeller, og at elevene ikke bare skal lage programmene, men også kunne vurdere dem. De to andre kompetansemålene sier ikke noe om programmering direkte, og det er mulig å oppnå disse kompetansene uten bruk av programmering, men de peker begge mot at naturfaget handler om å forstå og forklare verden rundt oss, gjerne gjennom modeller. Det kan virke som at læreplanen legger opp til at elevene

i naturfag skal jobbe mye med modeller, og at programmering skal kunne hjelpe til i dette arbeidet ved å la dem jobbe med modeller de tidligere ikke kunne benytte seg av.

For at elevene skal kunne få full utnyttelse av mulighetene som programmering gir vil det derfor være viktig å gi dem mer opplæring i arbeid med modeller. Som Etkina et. al. skriver så sliter en del elever ofte med misoppfatninger av hva modeller er, og de er ikke alltid så flinke til å analysere dem utenom å benytte dem slik som de får dem servert. Når elevene med programmering får muligheten til å se på mer avanserte modeller må vi, som Haraldsrud et al. skriver, flytte perspektivet over på analyse, forståelse og drøfting av modeller. Under fokusgruppeintervjuene snakket elevene ikke mye om arbeidet med modellen annet enn når de snakket om selve programmeringen og hvordan den knyttet seg til modellen. De kom ikke med innspill til analyse av modellen utover hva den skulle simulere eller så for seg måter å forbedre den på. Det virker for meg som at det her er en jobb å gjøre for å ruste elevene bedre til å jobbe med modeller, da det med innføring av programmering er noe som de høyst sannsynlig vil treffe mer på i fremtidig undervisning.

En ting som elevene derimot trakk frem og kommenterte var at det å jobbe med programmering kan være tidsbesparende og gjøre ting lettere. Elevene sa seg jevnt over enige i at det var lettere å trille mange terninger på en datamaskin, fremfor å gjøre det manuelt, og at dette sparte dem for en del monotont arbeid. Mange uttrykte at de satte pris på dette aspektet ved programmering, men at de ikke var sikre på om de med deres nåværende ferdigheter ville spart tid om de måtte skrive all programkode selv. Noen elever trakk også frem at en datamaskin regner mer presist og nøyaktig enn de ville gjort selv, og at den ikke gjør regnefeil. De mente dermed at man med programmering kunne få mer nøyaktige svar, og finne ting med større presisjon når man benyttet programmering.

Et utsagn som flere elever kom med ved flere anledninger var at de ikke selv hadde lært så mye programmering enda. Elevene mente at deres egne ferdigheter ikke var gode nok, og noen trakk også frem at de opplevde undervisningen de hadde fått som mangelfull. Dette medførte at noen elever hadde et mindre positivt syn på programmering, og måten det ble benyttet i skolen på. De fleste elevene mente likevel at det kunne ha sin plass i undervisningen, men at det måtte undervises tidligere. Dette kan virke som en rimelig antakelse og synet støttes opp av at spørreundersøkelsen viste at de elevene som følte de hadde mest nytte av programmering på nåværende tidspunkt var de elevene som hadde egen erfaring med programmering før de møtte på det i VGS. Programmering er jo nå en del av

læreplanen gjennom hele skoleløpet, og om ikke mange år vil elevene som begynner i den videregående skolen ha god kjennskap til programmering.

Noen elever kom også innom at programmering kan gi muligheter for mer variasjon i undervisning, og de ønsket å se at man benyttet seg av mer prosjekter der man lot elevene jobbe med noe de var interessert i, men med programmering som grunnstein. De så gjerne for seg at man trakk andre fag mer inn i programmeringen, slik at det ikke bare ble et matematikkfag, slik som elevene opplevde faget som nå.

Det er ganske tydelig at det ligger flere muligheter innenfor programmering, og at elevene er positive til en del aspekter rundt programmering. Likevel var det ikke alle elevene som var udelte positive til programmering, og da særlig til hvordan de selv hadde hatt det til nå. Det er mange potensielle grunner til dette, og et viktig moment ligger nok i det jeg kaller for relevans.

7.4 Relevans og motivasjon

Ordet relevant var noe som forekom mange ganger under fokusgruppeintervjuene. Det var også flere spørsmål relatert til det i spørreundersøkelsen. Flere elever hadde tydelige meninger om relevans, og om programmering var relevant for dem. Det var likevel litt ulike måter elevene benyttet ordet på. Noen benyttet det som en erstatning for ord som interesserende, engasjerende eller motiverende, slik som Schreiner og Sjøberg gjorde i sin undersøkelse *Relevance of Science Education*. For disse elevene handlet programmeringens relevans, eller i alle fall et aspekt ved dens relevans, om egen interesse for programmering. Her var det mange blandete svar fra elevene i fokusgruppene, og det var både de som sa de likte programmering og bedrev det på fritiden, og de som sa de mislikte det.

I spørreundersøkelsen som ble gjennomført var det to spørsmål som kan sies å måle denne formen for relevans. De to spørsmålene var «jeg liker programmering» og «jeg blir motivert av programmering». For begge disse spørsmålene viser svarene tydelig at de med mer erfaring innenfor programmering var mer motiverte for, og likte programmering bedre enn de med mindre erfaring. Dette virker sannsynlig og logisk, og man kan håpe at elevene i fremtiden, som alle vil ha erfaring med programmering når de begynner på VGS, vil oppleve det som mer gøy og motiverende. Vi må likevel huske på at de elevene med tidligere erfaring

med programmering alle har valgt dette selv, og dermed ikke nødvendigvis representerer alle fremtidige elever som vil ha blitt «tvunget» til å drive med programmering på barne- og ungdomskolen.

Det er også verdt å merke seg at begge gruppene oppga at de likte programmering mer enn de ble motivert av den. Der den erfarne gruppen ga ganske positiv respons på spørsmålet om de liker programmering, så var de mindre positive når spørsmålet om motivasjon kom opp.

Denne nedgangen gjentok seg også hos gruppen uten forhåndserfaring, som gikk fra å være litt uenig med påstanden «jeg liker programmering» til å være tydelig uenige med påstanden «programmering motiverer meg». Krapp and Prenzel tar i sin tekst opp at det er forskjell på fornøyelse og interesse, der de nevner at fornøyelse kan oppstå fra flere faktorer og at interesse bare er en av disse. Om vi overfører denne tankegangen til våre funn, kan vi si at det er forskjell på fornøyelse og motivasjon, og at motivasjon kan komme av flere ting, der fornøyelse bare er en av disse. En mulig forklaring på dette kan være at selv om flere elever uttrykte at de likte programmering, så var det mange som også uttrykte at de fant programmering vanskelig. Eleven Jane uttrykte dette veldig fint med utsagnet «det er veldig demotiverende å gjøre ting man ikke forstår syns jeg». Eccles og Wigfield tar opp at kostnad er en viktig faktor når man ser på en oppgave elever skal utføre, og for mange elever er nok kostnaden knyttet til programmering fremdeles ganske stor.

Denne kostnaden som elevene opplevde vil naturligvis kunne senkes ved at elevene får mer opplæring i programmering. Fremtidige elever vil naturligvis oppleve denne kostnaden som mindre siden de vil ha mer erfaring med programmering. Det er likevel sann at denne kostanden aldri kan bli helt borte, og programmering er kanskje en ferdighet hvor dette gjelder mer enn andre steder. Selv erfarne programmerere, som daglig jobber innenfor dette feltet, må bedrive feilsøking og rette syntaksfeil i egen kode. Så det er ikke realistisk at denne kostnaden blir helt null, men forhåpentligvis kan den i fremtiden ligne mer på kostnaden elevene forbinder med andre skolefag.

Om man skal overkomme denne kostnaden må man ha andre faktorer som veier opp for den. Woolfolk drar frem de tre faktorene prestasjonsverdi, interesseverdi og nytte, når hun snakker om en oppgave og verdien den har for elevene. Interesseverdi er jo allerede nevnt, og for noen elever er denne ganske høy allerede, mens den for andre er mindre. Det er ikke så lett å legge en generell strategi for å påvirke interessen til elevene. Ulike elever har ulike interesser, og selv om vi for eksempel kan forsøke å koble elevenes andre interesser til programmering,

er dette ikke noe som er gjort i en håndvending. Med mer programmering i skolen vil det dog være flere elever som får prøve programmering. Disse vil bli bedre kjent med det, og kanskje vil det være noen som finner sin interesse for programmering vekket som ikke ville oppdaget dette på egenhånd. Prestasjonsverdi er heller ikke noe vi enkelt kan øke hos elevene, da elevene som begynner på VGS gjerne har et etablert bilde av seg selv og hva slags elev de er, og dermed også hvordan de burde prestere.

Det er når vi ser på nytteverdien at vi lettest kan gjøre grep for å øke verdien til programmering blant elevene. Fra både intervjuene, og spørreundersøkelsen virker nemlig nytteverdien elevene har opplevd i programmering tidligere ikke å være velig høy, og det var ikke mange som mente det hadde hjulpet dem i naturfag før. Dette er noe vi kan forsøke å forbedre ved å gjøre dem mer bevisste på hva programmering kan brukes til, og hvordan det kan komme til nytte for dem. Det var flere elever under intervjuene som oppga at de ikke erfarte at programmering ble benyttet til noe nyttig. Det er her viktig å trekke frem at elevene hadde begynt med undervisning i programmering det samme skoleåret som de gjennomførte dette opplegget, og at de derfor gjerne hadde hatt kun grunnleggende undervisning i programmering dette første året hvor det ble innlemmet i læreplanen. Mange erfarte at oppgavene de jobbet med ikke var nyttige utenfor klasserommet, og flere elever følte at det de jobbet med og fant ut, var ting de lettere kunne gjort uten bruk av programmering. Dette er som tidligere nevnt en unik situasjon, som vil endre seg når elever i fremtiden begynner på VGS med eksisterende erfaring innen programmering, men det er verdt å merke seg denne holdningen. Det å holde programmeringsoppgaver enkle når man lærer om det for første gang er også noe som generelt er anbefalt, for eksempel av Taub et. al., men det er likevel viktig at man ikke blir redd for å vise elevene hva de kan utrette med programmering i fremtiden, og gi dem en liten smakebit på hvor nyttig programmering kan være som verktøy.

Når elevene ble spurt om de opplevde opplegget de hadde jobbet med som nyttig var responsen fremdeles blandet, men noe bedre. Det var flere som ga uttrykk for at de foretrakk å ikke trille terninger, og at mulige tidsbesparelser som følge av programmering kunne oppleves som nyttige. Det var likevel flere som her påpekte at de selv ikke hadde for mye igjen av sitt eget skoleløp, og at de heller ikke så for seg å benytte programmering i videre utdanning. Flere elever følte dermed at programmering kunne være nyttig, men ikke for dem, og at elever i fremtiden som lærte om programmering fra tidligere alder ville kunne få mer nytte av det. Om vi ønsker å bedre på denne situasjonen for elevene nå under overgangsperioden må vi gjøre grep som hjelper elevene. Et mulig slikt grep kan være å

senke innsatsen som kreves ved å gi elevene mer ferdig kode. Det vil kunne gjøre at elevene opplever at de tjener på å benytte seg av programmering her og nå. Et ankepunkt er at elevene ikke vil føle denne nytten like mye da den kommer fra at de får noe, og ikke fra noe de kan på egenhånd. Ferdig kode kan også gjøre det lettere for lærere i andre fag å benytte programmering, da det minker kostnaden og tidsbruken, som for noen lærere kan virke vanskelig å rettferdiggjøre. På spørsmål elevene fikk kom det nemlig frem at 3 av de 4 klassene kun hadde benyttet programmering i matematikk, og den siste klassen hadde bare så vidt benyttet det innenfor naturfag.

Det er også viktig å huske på at det er forskjeller mellom de ulike elevene, og at spørreundersøkelsen viste at de med mer erfaring generelt var mer positive til programmering og så nytten i den. Dette er viktig å huske på når man skal legge opp undervisning som inneholder programmering. For elevene kan det oppstå positive og negative sirkler for de som henholdsvis opplever at de får til programmering, og de som sliter mer. Det blir viktig å ta hensyn til de som har mindre erfaring, og sørge for at de opplever programmeringen som nyttig og relevant, slik at de kan bli motivert til å lære seg mer programmering. Samtidig må man ikke glemme at det også vil være elever med mer erfaring enn gjennomsnittet tilstede, spesielt nå i overgangsperioden der «grunnkunnskapen» i elevmassen er lav, og at disse også må få oppleve at programmering blir brukt til noe som er relevant for dem og deres kunnskapsnivå.

Når programmeringens nytte og relevans i fremtiden kommer i fokus er det flere ting å se på. Det er viktig å huske det Stuckey et. al. skriver om at mange ting ikke faller innenfor elevers interesseområde når de blir undervist det, selv om det blir relevant for dem senere. Ofte kan vi som lærere bli nødt til å hjelpe elevene med å løfte blikket og se langt nok fremover slik at de ser relevansen. Det er noen datapunkter fra spørreundersøkelsen som er veldig interessante i forbindelse med dette. Elevene oppga nemlig jevnt over at de trodde programmering kunne bli nyttig i fremtiden. Om de derimot ble spurt om de trodde de ville få bruk for programmering innen bestemte arenaer fremover så bildet plutselig annerledes ut. Her var det flere som var usikre på, eller til og med litt uenige i, at de kom til å få bruk for programmering i senere utdanning, jobb eller privatliv. Denne forskjellen mellom om elevene antok at programmering kunne være nyttig i fremtiden, og om de trodde de ville få bruk for det innen ulike arenaer, var noe som gjaldt for elever både med og uten forhåndserfaring innen programmering. Om vi ser litt nærmere på intervjuene som ble gjennomført, så er det flere elever som sier de ikke vet hva det kan brukes til i fremtiden, eller

at de ikke ser for seg å jobbe med programmering selv. Et utsagn som kan peke på noe av forklaringen kommer fra eleven Anne i fokusgruppe 2.

Anne: Jeg tror bare det kommer til å bli mye mer av det etter hvert da, sånn i fremtiden og sånn.. at det er lurt å ha.. kunne det

Nora (bakgrunnen): ja

Intervjuer: Lurt å ha bare fordi at det blir mer av det eller fordi det har noen andre fordeler?

Anne: Det har jo fordeler, men jeg tror jo at det kommer til å bli.. at ting blir mer sånn.. eh.. teknologisk og sånne ting da

Dette utsagnet, i sammenheng med andre utsagn der elevene ofte kun så for seg at utviklere og webdesignere benytter programmering i sin hverdag, kan være med på å forklare at elevene tror programmering blir nyttig, men de vet ikke helt hvordan enda. Elevene ser vår stadig mer teknologiske hverdag og antar at programmering må passe inn i den på et vis, men de er usikre på akkurat hvordan dette vil se ut.

Dette betyr jo at mange elever har potensiale til å oppleve programmering som veldig relevant for dem, men vi må klare å vise dem hvordan. Om man skal kunne utnytte elevenes tro på at programmering kan bli nyttig i fremtiden, og la dem føle og bygge denne relevansen, så må vi klare å vise dem de ulike måtene programmering kan bli viktige for dem i fremtiden. Elevene må kunne se og forstå hvilke yrker som benytter seg av programmering, eller hvilke yrker hvor dette kan bli relevant etter hvert som ferdigheter innen programmering blir mer utbredt i befolkningen. Elevene burde også bli vist og forklart hvordan programmering kan hjelpe deres egen undervisning, og gi dem et ferdighetssett som vil kunne bidra positivt til deres utdanning.

Det er også et poeng at elevene skal kunne føle et eierskap til det de holder på med, da dette ifølge Milner-Bolotnin fører til økt læring. Et av de viktige elementene hun trakk frem for eierskap handlet om å finne verdi i prosjektet, oppleve nytte, og kunne knytte sammen gammel og ny kunnskap. Disse trekkene har vi vært innom tidligere og erfaringen er at flere elever opplevde deler av dette, men at det er ting som kan forbedres. For eksempel var det ikke alle elever som kunne se nytten i programmeringen de gjorde. Et annet viktig moment for eierskap handlet om kontroll, og å være aktiv i læringsprosessen. Her ville det i første omgang kanskje være nærliggende å tro at flere elever som jobbet med opplegget ikke nødvendigvis opplevde kontroll da de fikk utdelt mye kode. Flere elever ga likevel uttrykk

for at de opplevde kontroll hvis de forsto i grove trekk hva koden gjorde. På spørsmål om man behøver å forstå hver linje i et program kom de fleste frem til at det ikke nødvendigvis var nødvendig, så lenge man forsto hva koden skulle gjøre, og omtrentlig hvordan den gjorde det. Det er vanskelig å spekulere rundt om dette vil stemme for alle elever, men det er grunn til å tro at det er litt lettere å akseptere at man ikke forstår alt som står i kodelinjene, når man forstår modellen. Alt datamaskinen gjorde var tross alt å trille mange terninger, og det er noe elevene kunne gjort selv om de ville.

Ellers var elevene litt usikre på om de opplevde oppgaven som realistisk eller ikke. Som Dewey trekker frem har skolen tidligere slitt med å skape en forståelig kobling mellom naturvitenskapelig læring og elevenes hverdagsliv. Opplegget til elevene hadde et tema som er virkelighetsnært og som med jevne mellomrom kommer opp i media, men det er likevel ikke noe elevene behøver å tenke over til daglig. Dermed er det ikke helt lett å si om konteksten til oppgaven gjorde den mer meningsfull. Elevene satte pris på at de regnet ut noe som de opplevde som noe mer meningsfullt enn å regne ut sin egen alder, eller andre typiske begynneroppgaver innenfor programmering, men det var noen ga uttrykk for at det fortsatt føltes som en skoleoppgave. Noen trakk også frem at det med datamaskin er noe ekstra mellom deg og det som skjer, og at man ikke får noen fysisk interaksjon når man programmerer noe.

7.5 Tilpasninger av opplegg

Etter å ha gjennomført undervisningsopplegget i fire omganger, for fire ulike klasser, er det noen erfaringer jeg har gjort meg. Basert på disse ville jeg gjort noen endringer på opplegget, og gjennomføringen av dette, hvis jeg skulle gjort dette på nytt. Jeg velger her å kommentere rundt disse endringene, slik at andre som ønsker å benytte dette opplegget kan nyte godt av mine erfaringer. Før jeg tar for meg disse justeringene vil jeg likevel påpeke at elevene stort sett uttrykte at de likte opplegget. Det var flere elever som kommenterte at de satte pris på denne måten å trekke programmering inn i andre fag. De likte også at de benyttet seg av programmeringen til noe som de opplevde som mer praktisk sammenlignet med programmeringsoppgavene de var vant til.

Det første jeg ville gjort er å sette av mer tid til opplegget. Det var en del elever som ikke kom gjennom alt, og de som gjorde det rakk heller ikke å diskutere modellen de hadde jobbet med eller reflektere rundt hva de fant. Jeg ville trolig latt opplegget gå over to økter av 90 minutter, totalt fire skoletimer, fremfor en økt som nå. Den ekstra tiden ville blitt benyttet til å gi elevene mer tid til å komme gjennom alle oppgavene, men også til å gi elevene litt grunnleggende undervisning i halveringstid før man begynte med selve opplegget.

Den første klassen der opplegget ble gjennomført hadde aldri hørt om fenomenet, og selv om de ble oppfordret til å lese litt om halveringstid før de gikk i gang, var det ikke alle som gjorde dette. Dermed var det mange som opplevde at de måtte frem med google, eller spørre læreren, rett etter at de startet med oppgavene. Om det hadde vært en kortere økt der fenomenet ble forklart for elevene så tror jeg dette hadde vært nyttig. Elevene kunne så fått resten av timen til å komme i gang med oppgavene som hørte til den første programmeringsbiten, og de som kom langt nok kunne muligens begynt på oppgave 6, og forsøkt å lage sin egen algoritme.

Når det gjelder oppgave 6 så er det også en viktig observasjon at elevene i intervjuene oppga at de ikke hadde skrevet ned en algoritme før, og at de heller ikke hadde brukt dette under tidligere programmeringsundervisning. Det kan derfor være nyttig å undersøke om elevene kjenner til dette fra før, og eventuelt gi elevene opplæring i hva dette er.

Den siste endringen av opplegget ville vært å kjøre en større avslutning mot slutten av den andre økten, etter at elevene fullfører de siste programmeringsoppgavene. Her ville jeg lagt inn tid til å diskutere modellen man jobbet med, og bedt elevene analysere og forklare modellen. En skikkelig gjennomgang bidrar til å sikre at elevene forstår hva de har jobbet med, og gir dem samtidig mulighet til å øve seg på å diskutere og analysere modeller.

For å gi mer mulighet for differensiering vil det også være mulig å la noen elever skrive mer av koden selv, og spesielt i den andre runden med koding legger nettmodulen opp til nettopp dette. Man kan da utfordre de elevene som trenger det til å lage noe fra bunnen, mens de som trenger mer støtte finner det i den ferdige koden i nettmodulen. Håpet vil jo også være at etter hvert som flere elever går inn i VGS med mer programmeringserfaring, så vil flesteparten av elevene kunne skrive mye kode selv.

7.6 Implikasjoner

Denne oppgaven har sett på programmering i naturfag, og tatt for seg elevenes erfaringer med dette. Gjennom observasjoner, fokusgruppeintervju, spørreundersøkelse og elevenes oppgaver har jeg fått et innblikk i hvordan elever som møter programmering nå opplever dette. Det er mange forskjellige ting jeg har sett, noen tydeligere enn andre, og noe av dette mener jeg har implikasjoner for hvordan programmering bør undervises og benyttes i skolen fremover.

Den kanskje tydeligste observasjonen er at elever tror programmering kan bli nyttig i fremtiden, men at de ikke er helt sikre på hvordan. Det er tydelig at elevene har behov for å se situasjoner, yrker og utdanninger hvor programmering er nyttig og anvendbart. Dette kan komme i form av at lærerne setter av tid til å informere elevene bedre om dette, enten i dedikerte økter eller som en del av undervisningen i programmering. Vi burde også bli flinkere til å vise elevene nytten de får av programmering på nåværende tidspunkt, og benytte oppgaver der elevene kan forstå at programmering sparer dem for tid, eller gjør noe lettere for dem.

Relatert til nytte er det også tydelig at flere elever opplever programmering som vanskelig, og noe som krever en investering fra deres side å lære seg. For elevene som er mot slutten av sitt skoleløp var det det flere som var usikre på om denne investeringen ville lønne seg. Elever som begynner på skolen nå vil gjøre denne investeringen tidligere, og dermed vil de kunne nyttiggjøre seg av den gjennom mer av sitt skoleløp. Det betyr derimot ikke at kommende elever vil oppleve programmering som relevant på egenhånd. Elevene må bli vist hvordan programmeringen de lærer er relevant for dem, enten når de lærer det, eller hvordan det vil bli relevant senere i skoleløpet. For de elevene som er lenger ut i skoleløpet kan det være nødvendig at vi arbeider med å senke kostnaden knyttet til bruk av programmering.

Opplegget som ble gjennomført viser også at det er fullt mulig å senke kostnaden ved programmering mye, selv for elever uten masse erfaring innenfor programmering. Selv om flere elever ikke rakk gjennom opplegget, og det fremdeles var noen utfordringer elevene møtte, så bidro opplegget til å fjerne eller redusere flere vanlige programmeringsutfordringer. En trinnvis, forsiktig oppbygning ga elevene tid til å jobbe seg inn i oppgaven, og det var veldig få ganger at elever spurte om hjelp fordi de ikke forsto hva de skulle gjøre. Ferdiglaget kode ga også elevene et utgangspunkt å jobbe med, noe som jeg anser som viktig for elever

med mindre erfaring fra programmering og modellering. En fasit ga også elevene mulighet til å finne noe hjelp på egenhånd, og sørget for at lærerne som var i timen ikke ble overbelastet. Selv om målet er at elever skal kunne benytte seg av programmering og lage modeller mer på egenhånd i fremtiden så viser opplegget at det går an senke kostnaden såpass mye at programmering kan trekkes inn i mange fag allerede i dag. Dette er kanskje en av de viktigste implikasjonene, da dette vil være veldig verdifullt for elevene siden de da ser og opplever prosessen med modellering og programmering og får trent på denne. Jo flere ganger elevene gjennomgår prosessen, jo mer vil de forstå og huske, og over tid vil man kunne trappe ned på hjelpen som gis i oppgaven. Ferdiglaget kode kan da anses litt som støttehjul som elevene med trening kan klare seg uten.

Et moment der det er litt vanskeligere å si noe om implikasjonene av i denne studien omhandler støy og distraksjon som kommer med programmering. Elevene som jobbet med opplegget klarte stort sett å relatere modellen de jobbet med til naturfenomenet, og de virket heller ikke å ha blitt for mye distrauert av programmeringen. Noen tegn var det likevel på at programmeringen fanget mye av oppmerksomheten til elevene, og programmering vil jo ta med seg flere elementer inn i naturfagstimene, som elevene så må klare å holde styr på. Å håndtere dette potensialet for støy og distraksjon blir viktig fremover, og når programmering etter hvert vil være en helt normal del av elevenes opplæring vil det være lettere å finne ut hvor stort dette problemet eventuelt er.

Når det kommer til holdninger så ser vi også tydelig at det er viktig for elevene å forstå og få til koding for at det skal oppleves som individuelt relevant for dem. Dette ser vi komme til uttrykk både gjennom flere elevuttalelser om at de ikke er gøy når man ikke får til ting, men også gjennom utsagn fra både intervju og spørreundersøkelsen om at elevene gjerne vil lære mer programmering. Å gi elever i fremtiden god opplæring i programmering blir derfor viktig, og dette ønsket om å lære er viktig å ta med seg. Elevene virker generelt å ha flere positive tanker rundt programmering, selv om det er kommet til orde flere utfordringer, og det blir viktig at vi skaper en undervisning som kan bygge opp under elevenes entusiasme og ønske om å lære. Det er også viktig å være klar over at vi kan se en tydelig trend der elever som har forhåndserfaring med programmering er mer positive til omtrent «alt» som har med programmering å gjøre, og de uten denne erfaringen er mer negative. Det blir viktig å forsøke å minske dette gapet i fremtiden, og unngå at disse to gruppene går inn i hver sin utviklingsspiral, en positiv og en negativ.

7.7 Begrensninger ved denne oppgaven

Det er flere begrensninger ved denne oppgaven som jeg her vil ta opp. Jeg anser ingen av disse som kritiske, men de er viktige å ta med seg når man ser på oppgaven. Den første av disse begrensningene er mengden datamateriale. Det er gjort fire fokusgruppeintervjuer i fire ulike klasser, det er gjennomført spørreundersøkelse, og elevenes oppgaveark er samlet inn, men det er likevel kun materiale fra 73 elever totalt. Koronapandemien har medført at flere klasser har vært reduserte i antall, noe som har ført til at deltagerantallet er blitt lavere enn om klassene hadde vært fulltallige.

Dette medfører at antallet som har svart på for eksempel spørreundersøkelsen er noe lavere enn man kunne ønsket seg, men det er likevel mange nok som har svart slik at trender sannsynligvis har kommet frem. Det er også mulig å spørre seg om den delen av klassen som har deltatt har vært representativ eller ikke. I utgangspunktet ville det være naturlig å anta, da oppdelingen av klassene som har vært med har vært tilfeldig. Det som derimot ikke nødvendigvis er tilfeldig er de elevene som ikke var tilstede. Under datainnsamlingen var koronapandemien på et punkt hvor elever selv meldte seg syke, og gikk i karantene om de hadde luftveisymptomer. Flere lærere opplevde høyt fravær denne aktuelle våren, og flere lærere ga uttrykk for at det gjerne var de mindre motiverte elevene som hadde høyere fravær.

Utvalget til fokusgruppene er også verdt å si litt om. Det var spredning blant elevene som deltok både når det kommer til programmeringskunnskaper, og innstilling til programmering, men det er ikke sikkert at representasjonen ble proporsjonal. Om man ser på spørreundersøkelsen og sammenligner de med fokusgruppene kan det virke som at antallet elever som var positive til programmering har vært noe overrepresentert. Trolig har de fleste elev-typene likevel blitt representert i diskusjonen, i alle fall litt. Blant elevene i fokusgruppene som var mindre positive til programmering fant vi noen av dem som uttrykte seg mest, sånn at disse elevenes meninger til tross mindre representasjon har kommet tydelig frem.

Det er også viktig å huske på at opplegget elevene gjennomførte var holdt på et relativt enkelt nivå, og at de fikk mye kode utdelt. En del elever kjente også til halveringstid på forhånd, og noen få kjente til og med til modellen de skulle jobbe med. Det er derfor mulig at elevene kan ha funnet opplegget enklere enn de vanligvis ville gjort, og at flere elever kan ha hatt en mer positiv opplevelse enn de normalt ville hatt. Et positivt punkt er at elevene som ikke kjente til

halveringstid også responderte positivt på opplegget. Deres resultater er sammenlignbare med de andre elevene, så det kan være at det ikke har hatt for mye påvirkning at noen kjente til fenomenet. Flere elever som kjente til fenomenet sa også at de kun visste litt om det, og at de ikke hadde gått spesielt i dybden når det hadde kommet opp i undervisning.

Når jeg har analysert mine data har jeg vært opptatt av å belyse mine forskningsspørsmål, som jo selvfølgelig er en begrensning. Jeg har også jobbet ut i fra visse teoretiske perspektiver, som relevans, som nok har hatt lettere for å komme til syne en andre perspektiver. Om jeg hadde benyttet andre analytiske rammer er det sannsynlig at jeg ville fått andre typer resultater, som kunne trukket frem andre aspekter ved programmering i naturfag.

Til slutt er det også verdt å merke seg at spørreundersøkelsen som elevene gjennomførte var i slutten av hver økt, og at elevene skulle ha friminutt eller avslutte dagen etter økten. For å hindre at elevene svarte for raskt ble det satt av 5-6 minutter i slutten av hver økt, og elevene fikk ikke gå før økten var ferdig slik at det ikke skulle være noe ekstra incentiv til å fullføre spørreundersøkelsen raskt. Likevel kan noen ha gått raskt gjennom hele eller deler av undersøkelsen, for å unngå å bruke egen fritid på denne.

8) Konklusjon/oppsummering

Nå som programmering innføres i den norske skolen har mitt utgangspunkt for denne oppgaven vært problemstillingen: *Hvordan kan en programmeringsaktivitet påvirke elevenes forståelse, muligheter og motivasjon innenfor naturfag*. For å undersøke dette nærmere har jeg sett på tre forskningsspørsmål.

- 1: Hvordan påvirker programmeringen elevenes forståelse rundt koblingen mellom modellen de jobber med og naturfenomenet som skal forklares?
- 2: Hvilke muligheter får elevene når programmering benyttes i naturfag, og hvordan sikrer vi at elevene får utnyttet disse mulighetene?
- 3: Hvordan påvirker programmering elevenes motivasjon, og hvordan kan vi bidra til at de opplever programmering som relevant?

Hva angår de to første spørsmålene, så fant jeg at elevene tilsynelatende ikke ble for mye påvirket av programmeringen når det kom til å forstå modellen de jobbet med.

Programmering kan føre med seg noen ekstra elementer som kan trekke fokus vekk fra naturfag, men dette virket ikke å ha mye effekt under mitt opplegg. Programmering gir derimot elevene anledning til å benytte mer avanserte modeller, og jeg har også opplevd at elevene gjennom programmering kan spare tid. Dette gir elevene mer anledning til utforskning av modellen, både ved at de får mer tid, og ved at de får et kraftig verktøy der det er lettere å bytte variabler og eksperimentere.

For å sikre at elevene får utnyttet disse mulighetene er det noen viktige elementer som må være på plass. For det første burde elevene få mer opplæring i bruk av modeller som er litt mer avansert enn bare å forstå hva modellen forestiller. Om elevene får mer trening i å analysere og diskutere modeller, vil det være lettere for dem å foreslå og teste endringer i modellene de jobbet med. Elevene må også få god hjelp og støtte til programmeringen.

Programmering er vanskelig, og innføringen av programmering i naturfaget bringer med seg et ekstra element som elevene må håndtere. Det ønskelige er at alle elever får nok opplæring slik at mange av utfordringene med programmering blir noe de håndterer stort sett på egenhånd, men som jeg observerte under mitt opplegg så kan vi også gi elevene ekstra hjelp gjennom oppgavene de jobber med.

Mitt siste forskningsspørsmål dreide seg om elevenes opplevelse av relevans knyttet til programmering. Siden programmering nettopp er innført er det mye elevene ikke har rukket å lære enda, og de har primært opplevd programmering i matematikktimen. Dette bidro til at følelsen av relevans knyttet til programmering for mange av elevene hang sammen med om de opplevde at de fikk til programmeringen eller ikke. Elevene hadde stort sett ikke benyttet programmering til annet enn nettopp oppgaver hvis mål var å lære programmering. Gjennom opplegget elevene hadde fikk de erfare programmering benyttet til å lære om noe mer virkelighetsnært, og at programmering gjorde deler av arbeidet lettere, spesielt siden de fikk mye hjelp til programmeringen i nettmodulen de jobbet i. Om vi ønsker at flere elever opplever programmering som relevant blir det viktig å gjennomføre opplegg der elevene føler de nyttiggjør seg av programmering, og hvor vanskelighetsgraden er tilpasset elevenes nivå. Det er også viktig å gjøre elevene mer bevisst på hva programmering kan brukes til i fremtiden, og hvordan det å kunne programmerer vil gagne dem. Til sist blir det også viktig å gi elevene god undervisning i programmering tidlig nok, slik at de kan benytte denne kunnskapen som et verktøy gjennom store deler av deres skoleløp.

Jeg håper det gjennom denne oppgaven er kommet frem at programmering er et flott verktøy å trekke inn i naturfagundervisningen. Det gir elevene store muligheter som de ikke har nå både hva gjelder hvilke fenomener som det undervises og hvilke modeller som benyttes i undervisningen. Om man klarer å håndtere de ekstra utfordringene som kommer med programmering, vil elever i fremtiden gå ut av skolen med et kraftig verktøy som de kan få nytte av senere i livet.

9) Kilder/referanser

- Angell, C., Kind, P. M. & Henriksen, E. K. (2008). Implementation of empirical-mathematical modelling in upper secondary physics: Teachers' interpretations and considerations. *Nordic Studies in Science Education*, 4(2), 113-122. Hentet fra <https://doi.org/10.5617/nordina.284>
- Brown, N. C. C. & Wilson G. (2018). Ten quick tips for teaching programming. *PLoS Comput Biol*, 14(4). Hentet fra <https://doi.org/10.1371/journal.pcbi.1006023>
- Bruner, J. S. (1996). *The culture of education*. Cambridge, Massachusetts: Harvard University Press.
- Creswell, J. D. & Miller, D. L. (2000). Determining validity in qualitative inquiry. *Theory Into Practice*, 39(3), 124-130. Hentet fra https://doi.org/10.1207/s15430421tip3903_2
- Eccles, J. & Wigfield, A. (2002). Motivational Beliefs, Values and Goals. *Annual Review of Psychology*, 53(1), 109-132. DOI:10.1146/annurev.psych.53.100901.135153
- Enghag, M. (2006). *Two dimensions of student ownership of learning during small-group work with miniprojects and context rich problems in physics* (Doktoravhandling). Mälardalen University, Västerås
- Etkina, E., Warren, A. & Gentile, M. (2006). The Role of Models in Physics Instruction. *The Physics Teacher*, 44, 34-39. Hentet fra <https://doi.org/10.1119/1.2150757>
- Feng, SL. & Tuan, HL. (2005). Using ARCS Model to Promote 11th Graders' Motivation and Achievement in Learning about Acids and Bases. *International Journal of Science and Mathematics Education*, 3, 463-484. Hentet fra <https://doi.org/10.1007/s10763-004-6828-7>
- Haraldsrud, A. D. & Tellefsen, C. W. (2018). Programmering – for fysikkens skyld. *Fra Fysikkens Verden*, 3/2018, 70-75. Hentet fra <https://www.mn.uio.no/ccse/om/aktuelt/i-media/programmering-for-fysikkens-skyld.pdf>
- Haraldsrud, A. D., Sveinsson, H. A. & Løvold, H. H. (2020). *Programmering i skolen*. Oslo: Universitetsforlaget.
- Jenkins, T. (2002). ON THE DIFFICULTY OF LEARNING TO PROGRAM. Hentet fra <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.596.9994&rep=rep1&type=pdf>
- Kleven, T. A. (2014). Kap. 2: Data og datainnsamlingsmetoder. I T. A. Kleven (Red.), *Innføring i pedagogisk forskningsmetode: En hjelp til kritisk tolkning og vurdering* (s. 27-47). Bergen: Fagbokforlaget
- Krapp, A. & Prenzel, M. (2011). Research on Interest in Science: Theories, methods, and findings. *International Journal of Science Education*, 33(1), 27-50. DOI: 10.1080/09500693.2010.518645
- Malt, U. & Grønmo, S. (2020, 26. november). Likert-skala. Hentet fra <https://snl.no/Likert-skala>
- Millar, R., Klaassen K. & Eijkelhof, H.M.C. (1990). Teaching about Radioactivity and Ionising Radiation: An Alternative Approach. *Physics Education*, 25(6), 337-342. DOI:10.1088/0031-9120/25/6/310
- Morgan, D. L. (1998). *The Focus Group Guidebook*. Thousand Oaks, California: SAGE publications

- Nordby, S. T. (2019). *Programmering og algoritmisk tenkning i fysikkundervisning* (Masteroppgave, Norges teknisk-naturvitenskapelige universitet). Hentet fra <https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2610766>
- Onwuegbuzie, A. J. & Johnson, R. B. (2006). The Validity Issue in Mixed Research. *RESEARCH IN THE SCHOOLS*, 13(1), 48-63.
- Rainer, J. D. & Matthews, M. W. (2002). Ownership of Learning in Teacher Education. *Action in Teacher Education*, 24(1), 22-30. Hentet fra DOI: 10.1080/01626620.2002.10463264
- Regjeringen. (2018, 26. juni). Forbyter innholdet i skolen. Hentet fra <https://www.regjeringen.no/no/dokumentarkiv/regjeringen-solberg/aktuelt-regjeringen-solberg/kd/pressemeldinger/2018/forbyter-innholdet-i-skolen/id2606028/?expand=factbox2606066>
- Sevik, K. (2016). *Programmering i skolen*. Senter for IKT i utdanningen. Hentet fra https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf
- Sjøberg, S. & Schreiner, C. (2004). *ROSE: The Relevance of Science Education*. Hentet fra <https://www.duo.uio.no/bitstream/handle/10852/32303/1/AD0404.pdf>
- Stuckey, M., Hofstein, A., Mamlok-Naaman, R. & Eilks, I. (2013). The meaning of 'relevance' in science education and its implications for the science curriculum. *Studies in Science Education*, 49(1), 1-34. Hentet fra DOI: 10.1080/03057267.2013.802463
- Säljö, R. (2016). *Læring – en introduksjon til perspektiver og metaforer*. Oslo: Cappelen Damm akademisk.
- Sørby, S. A. (2010). Beregningsorientert fysikk i bachelorkurs ved Universitetet i Oslo: En didaktisk studie av førsteårsstudenters møte med numerisk matematikk og programmering med anvendelser i mekanikk (Masteroppgave, Universitetet i Oslo). Hentet fra <https://www.duo.uio.no/bitstream/handle/10852/11290/Masteroppgaven.pdf?sequence=1>
- Sørby, S. A. & Angell, C. (2012). Undergraduate students' challenges with computational modelling in physics. *Nordic Studies in Science Education*, 8(3), 283-296. Hentet fra <https://doi.org/10.5617/nordina.534>
- Taub, R., Armoni, M., Bagno, E. & Ben-Ari, M. (2015). The effect of computer science on physics learning in a computational science environment. *Computers & Education*, 87(2015), 10-23. Hentet fra <https://doi.org/10.1016/j.compedu.2015.03.013>
- Utdanningsdirektoratet. (2020). *Læreplan i naturfag* (NAT01-04). Hentet fra <https://www.udir.no/lk20/nat01-04/kompetansemaal-og-vurdering/kv77>
- Waters, J. B. (2020). *Programmering og dybdelæring i fysikk: En kvalitativ studie av elevers arbeid med programmering i fysikk 1* (Masteroppgave). Universitetet i Oslo, Oslo.
- Wilensky, U., Brady, C. & Horn, M. (2014). Fostering Computational Literacy in Science Classrooms. *Communications of the ACM*, 57, 24-28. DOI:10.1145/2633031
- Woolfolk, A. (2004). *Pedagogisk psykologi*. Bergen: Fagbokforlaget.
- Yager, R. E. & Hofstein, A. (1986). Features of a quality curriculum for school science. *Journal of Curriculum Studies*, 18(2), 133-146. DOI: 10.1080/0022027860180203

Vedlegg A, oppgaveark:

Programmeringsoppgave om halveringstid

Navn:

Dere skal, parvis eller alene, jobbe med dette arket og fylle ut oppgavene her. Arket følger og benytter seg av en oppgavemodul som ligger på

<https://aunivers.lokus.no/fagpakker/realfag/naturfag-sf/innhold/programmering/halveringstid?nof=1>

I slutten av timen skal dere levere inn dette arket, med navn, til læreren

Dere kan også se på eksempel 8 i boka på side 243, der dere kan se mer informasjon om halveringstid og programmering.

Denne oppgaven omhandler halveringstid og benytter programmering for å illustrere og vise hvordan dette fenomenet fungerer. Før du går i gang med nettressursen kan det være lurt å tenke over følgende spørsmål:

1: Hva er det som halveres etter en halveringstid?

Svar:

2: Hvis du begynner med 1000 atomer av et radioaktivt stoff, og det går 3 halveringstider, hvor mange atomer av det radioaktive stoffet sitter du igjen med (som ikke har henfalt)?

Svar:

Når du har svart på dette kan du begynne med nettressursen. Se på de to første sidene der, test ut det første programmet du får (i trinket-vinduet), og gjør de tre oppgavene under:

3: Prøvekjør programmet, og noter deg det som skjer. Prøv så programmet med ulike verdier for **n**. Hva skjer med «resultatet/outputen» når du endrer **n**?

Svar:

4: Legg til **print(terningsliste.count(6))** nederst i programmet. Test programmet med denne kodebiten og forsøk å forklare hva denne kodebiten gjør.

Svar:

5: Sett **n = 1000** og kjør programmet. Noter antall seksere. Gjenta programmet, men reduser verdien til **n** med antall seksere du fikk når du kjørte programmet. Gjenta prosessen og reduser **n** til den er blitt mindre enn 500. Hvor mange gjentakelser var nødvendige? Hva er det du har funnet nå?

Svar:

Nå skal vi se på kode som både kaster terning og plotter resultatet. Ønsker at programmet skal gjøre kast etter kast, og redusere verdien til **n** etter antall seksere man får (som i forrige oppgave, men nå uten at du må endre tallet selv). Deretter er ønsket at koden skal plotte hvor mange terninger som var igjen ettersom man kastet flere og flere ganger.

Før du begynner å kode denne oppgaven vil jeg at du skal lage såkalt *pseudokode* eller *algoritme*. Dette er rett og slett en slags beskrivelse som forteller hva du vil at koden skal gjøre, for å få det resultatet vi ønsker. For den første koden du brukte under denne øvelsen ville algoritmen sett ut som noe slikt:

Algoritme

- 1: Lag en tom liste som kan lagre verdien av alle terningkastene (terningliste)
- 2: Angi hvor mange kast (n)
- 3: For hvert kast
 - Generer et tilfeldig tall fra 1 til 6 (terning)
 - Legg dette tallet til listen (terningliste)
- 4: Skriv ut terninglisten

6: Prøv nå å lage en algoritme/pseudokode som gjør kast etter kast, reduserer verdien til **n** etter antall seksere man får og til slutt plotter det (som i oppgave 5, men med et plott til slutt). Skriv ned forslaget ditt i boksen under.

Algoritme

På neste side (den tredje) av modulen kan du finne et forslag til algoritme, og kode som utfører denne. Du kan enten bruke denne videre eller forsøke å erstatte den med din egen.

7: Kjør programmet flere ganger for å finne «halveringstiden» til terningene oppgitt i antall kast. Sammenlign med hvor mange kast du brukte i oppgave 5. Prøv å doble, halvere og kjøre med flere forskjellige verdier for antall terninger, har dette noe å si for halveringstiden?

Svar:

På siste side av oppgavemodulen som dere nå har jobbet med skal du se litt mer på et par kjente stoffer og deres halveringstid. Her finner du kode som plottes hvor mye radioaktivt stoff (karbon-14) vi har igjen ettersom tiden går. Kjør programmet først med den ferdig utfylte informasjonen, og forsøk deretter å lage graf for noen andre stoffer ved å bytte ut informasjonen i koden med informasjonen om enten cesium-137, uran-235 eller radon-222 som du finner lenger ned i nettmodulen.

Til slutt kommer et par utforskningsspørsmål om det du har holdt på med

8: I forbindelse med radioaktivitet/halveringstid og henfall, hva betydde det i den første og andre kodeoppgaven at du fikk en sekser på terningen? Hva skulle dette simulere/modellere fra virkeligheten?

Svar:

9: Måtte vi i vår modell alltid fjerne sekserne eller kunne vi for eksempel valgt å fjerne toere eller firere?

Svar:

10: Hvorfor ble ikke halveringstiden nevneverdig påvirket når du endret antall terninger (med mindre du satte veldig lav n), og hva betydde det for antall terninger igjen etter en halveringstid når du gikk fra å starte med f. eks. 1000 til 2000 terninger?

Svar:

Vedlegg B, spørreundersøkelse:

Spørreskjema for prosjekt «Programmering i naturfag»

Navn:

Full ut dette spørreskjemaet som best du klarer, og svar så ærlig som mulig.

Hvor enig/uenig er du i følgende:

	Sterkt uenig	uenig	Både og	Enig	Sterkt enig
Jeg liker programmering	1	2	3	4	5
Jeg syns programmering er vanskelig	1	2	3	4	5
Jeg blir motivert av programmering	1	2	3	4	5
Jeg har hatt bruk for programmering i naturfag	1	2	3	4	5
Jeg føler programmering har hjulpet meg i naturfag	1	2	3	4	5
Jeg opplever programmering som nyttig nå	1	2	3	4	5
Jeg tror programmering kan bli nyttig i fremtiden	1	2	3	4	5
Jeg vil lære mer programmering	1	2	3	4	5
Jeg opplever at jeg lett mestrer programmering	1	2	3	4	5
Jeg får mer lyst til å jobbe med en oppgave om den kan løses med programmering	1	2	3	4	5
Jeg tror jeg kommer til å få bruk for programmering senere i min utdanning	1	2	3	4	5
Jeg tror jeg kommer til å få bruk for programmering senere i mitt yrkesliv	1	2	3	4	5
Jeg tror jeg kommer til å få bruk for programmering senere i mitt privatliv	1	2	3	4	5
Jeg har som mål å bli bedre innen programmering	1	2	3	4	5

Jeg synes programmering er utfordrende

1

2

3

4

5

I denne andre delen skal du skrive ned hvor mye tid du brukte på ulike deler av opplegget, Oppgi svar i antall minutter.

Hvor lang tid (i minutter) brukte du/dere omtrent på de to første oppgavene (før dere begynte med nettressursen)?

Svar:

Hvor lang tid (i minutter) brukte du/dere omtrent på oppgave 3 – 5 (som benyttet seg av den første siden i nettressursen)?

Svar:

Hvor lang tid (i minutter) brukte du/dere omtrent på å lage/utforme algoritmen i oppgave 6?

Svar:

Hvor lang tid brukte du/ dere på å lage, eller forsøke å lage, egen kode til oppgave 7? Om dere benyttet koden i nettressursen med en gang så svar 0.

Svar:

Hvor lang tid (i minutter) brukte du/dere omtrent på oppgave 8 – 10 (utforskningspørsmål)?

Svar:

I denne tredje delen skal du sette ring rundt det alternativet som passer best

Hadde du programmeringserfaring før du begynte på VGS?

- Ingen
- Litt
- noe
- mye

Har du hørt om begrepet algoritmisk tenking?

- Ja
- Nei
- Usikker

Har du siden du begynte på VGS programmert på fritiden som ikke var lekse/skole?

- Ingen
- Litt
- Noe
- mye

Identifiserer du seg som:

- Jente
- Gutt
- Annet/ønsker ikke å oppgi

Vedlegg C, Intervjuguide:

Intervjuguide

Intro

Takk for at dere stiller opp til dette intervjuet. Jeg har tenkt å benytte informasjon herfra til å skrive masteroppgaven min, som skal handle om programmering i skolen. Dere kommer ikke til å kunne gjenkjennes i oppgaven jeg skriver, der står det ingenting om hvem dere er, hvilken skole dere går på, eller hvor dere bor. Alt dere sier her har ingen betydning for hvilken karakter dere får i faget. Læreren deres har ikke tilgang til opptaket av intervjuet. For å huske hva dere sier, tar jeg lydopptak av intervjuet. Planen er å skrive ned hva dere sier så fort som mulig, og anonymisere det slik at dere ikke kan kjennes igjen. Lydopptakene slettes ved prosjektets slutt, i juni 2022. Til da er det kun jeg, veilederne mine og et fåtall andre ansatte på universitetet som vil ha tilgang. Dere er her på frivillig basis. Hvis dere finner ut nå – eller senere i intervjuet – at dere ikke vil være med likevel, kan dere si ifra om det, og gå ut av rommet.

Bakgrunn

- Hvor mye erfaring har dere med programmering
 - o Har noen drevet med blokkbasert programmering før?
 - o Har dere erfart dette gjennom skolen, fritidsaktivitet eller egeninteresse?
- Hvordan syns dere undervisning innen programmering har vært så langt
 - o Har det vært lett/vanskelig?
 - o Hva er gøy og mindre gøy?
 - o Har oppgavene dere har fått vært spennende/kjedelige?
 - o Har dere fått anvendt programmering innenfor andre fag
- Hva synes du om naturfag
 - o Trekk frem noe du synes er lett/vanskelig med naturfag?
 - o Hva er gøy og hva er mindre gøy?

Programmeringsopplegg i naturfag

- La oss se litt mer på læringsutbyttet fra opplegget dere hadde i naturfagsøkten nå. Hva husker dere best fra opplegget?
 - o Hva tror dere var målet med denne timen?
 - Hva skulle dere lære denne økta?
 - o Hva er noe dere forstår bedre nå enn før økta?
 - Er det noe som fremdeles er uklart etter opplegget?
 - o Hva visste dere fra før om halveringstid?
 - Lærte dere noe nytt om halveringstid?
- Hvor lett var det å forstå fenomenet halveringstid, og hvor krevende var selve programmeringen?
 - o Føler dere at dere forstår fenomenet halveringstid bedre nå?
 - Kunne du gjort en presentasjon om halveringstid nå
 - o Hvor mye tid brukte dere på hver komponent?
 - Var det programmering eller det rundt som tok mest tid?
 - o Hjalp programmeringen til med forståelsen av fenomenet?
 - o Ble det mye «støy» fra programmeringsøvelsene, og har dere i så fall ideer om hvordan det kunne vært gjort annerledes?
- Hvordan opplevde dere å jobbe med oppgavearket og opplegget
 - o Var det nok informasjon? For mye?
 - o Var det passe utfordrende eller ble det kjedelig?

- Fant dere hjelp om dere sto fast, og fra hvor eller hvem kom denne hjelpen?
- Rakk dere gjennom alt?
- Hvordan opplevde dere å jobbe med nettmодulen?
 - Hva syntes dere om å få utdelt ganske mye av koden dere skulle lage?
 - Hvordan var dette i forhold til å skrive alle linjer selv?
 - Er det viktig å forstå alle linjene i et program før man bruker det?
 - Hvor godt forsto dere koden dere brukte?
 - Prøvde noen å lage egen kode for program nummer 2?
 - Var det noen utfordringer med bruk av nettmодulen som dere støtte på
- Hva synes dere om å benytte programmering innen naturfag i motsetning til å ikke bruke dette verktøyet?
 - Hva ble lettere?
 - Hva ble vanskeligere?
 - Hva synes dere om å trille terning på datamaskin fremfor å gjøre det manuelt?

Programmering generelt

- Hva tenker dere om programmering som verktøy i naturfag eller andre realfag?
 - Hva var lett/vanskelig?
 - Hva var gøy/kjedelig?
- Hvordan påvirker programmering din motivasjon i fag?
 - Syns du det er gøy/motiverende å bruke programmering?
 - Får du mer lyst til å jobbe med en oppgave om den inneholder programmering?
- Hva er din oppfattelse av en «typisk programmeringsoppgave»?
 - Føler du programmeringsoppgaver er «praktiske/realistiske»?
 - Benytter dere programmering i andre fag?
 - Blir det dere jobber med forbedret av å bruke programmering?
- Jeg vil finne ut hvordan programmering kan brukes for å skape mest mulig læring og motivasjon for naturfag, har dere tanker om hvordan dette kan gjøres?
 - På hvilken måte mener du at programmering kan best innlemmes i naturfag eller andre fag?

Vedlegg D, Infoskriv og samtykkeskjema utdelt til klassene:

Invitasjon til å delta i forskningsprosjektet **Programmering i naturfag**

Dette er et spørsmål til deg om å delta i et forskningsprosjekt som ønsker å undersøke elevers oppfatning av, samt holdninger og motivasjon i forhold til, programmering innenfor naturfag.

Formål

Det har kommet nye læreplaner i naturfag der programmering er blitt inkludert som en viktig del. I forbindelse med dette gjennomfører vi et forskningsprosjekt som undersøker hvordan elever jobber med programmeringsoppgaver i naturfag, hvilke utfordringer de støter på, og hvordan programmering påvirker motivasjonen deres for naturfag.

Hva vil det innebære å delta i undersøkelsen

Vi ønsker å observere klassen mens dere arbeider med et opplegg som involverer programmering i naturfag, og vi ønsker tilgang på dine skriftlige svar på et oppgaveark knyttet til opplegget. I tillegg ønsker vi å gjennomføre en kort spørreundersøkelse på papir i forbindelse med gjennomføring av opplegget. Ditt arbeid i naturfagtimen (både skriftlig og muntlig) vil selvsagt bli vurdert av læreren din på vanlig måte uansett om du samtykker eller ikke til å delta i forskningsprosjektet. Hvis du gir forskningsprosjektet tilgang til dine skriftlige svar på oppgavearket vil dette ikke påvirke lærerens vurdering eller karaktersetting av din innsats i naturfag. Forskningsprosjektet vil bruke opplysningene bare til forskning på elevers forståelse, motivasjon og læring knyttet til programmering i naturfag.

Hva skjer med informasjonen

Vi vil behandle alle personopplysninger konfidensielt, og kun bruke innsamlet data til forskningsformål. Vi vil ikke bidra til vurdering av deg, og det vi finner ut vil ikke påvirke din vurdering i naturfag. Det er kun forskningsprosjektet som vil ha tilgang til personopplysninger, og det vil ikke være mulig å gjenkjenne deg ut fra rapporter/oppgaver som blir laget ved hjelp av data samlet inn gjennom dette prosjektet. Innen juni 2022 vil alt materiale der du kan identifiseres være slettet eller anonymisert. Anonymiserte data vil kunne lagres også etter juni 2022 for oppfølgingsstudier. Som forskere forholder vi oss til etiske regler om hvordan personopplysninger kan brukes. Vi behandler opplysninger om deg basert på ditt samtykke. På oppdrag fra Universitetet i Oslo har NSD – Norsk senter for forskningsdata AS vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

Frivillig deltagelse

Det er frivillig å delta i undersøkelsen. Klassen kommer til å gjennomføre opplegget uansett, men data fra deg samles kun inn om du gir samtykke. Hvis du *ikke* samtykker vil det kun være læreren din som har tilgang til dine svar fra programmeringsopplegget. Om du velger å delta kan du når som helst trekke ditt samtykke tilbake uten å oppgi grunn. Hvis du gjør dette vil alle personopplysninger om deg slettes. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg.

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke.

Dine rettigheter

Om du deltar i studien, og fremdeles kan identifiseres i datamaterialet, har du rett til:

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

Kontaktinformasjon prosjektledere

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

Einar Løhne Iversen: einarliv@student.uv.uio.no

Ellen K. Henriksen: e.k.henriksen@fys.uio.no



UiO

Postadresse: Fysisk institutt, Postboks 1048 Blindern, 0316 OSLO
<https://www.mn.uio.no/fysikk/forskning/grupper/skolelab/>

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

NSD – Norsk senter for forskningsdata AS på epost (personverntjenester@nsd.no) eller på telefon: 55 58 21 17.

Vedlegg E, Infoskriv og samtykkeskjema til fokusgruppeintervjuene:

Invitasjon til å delta i fokusgruppeintervju til forskningsprosjektet **Programmering i naturfag**

Dette er et spørsmål til deg om å delta i et fokusgruppeintervju til forskningsprosjektet **Programmering i naturfag** som ønsker å undersøke elevers oppfatning av, samt holdninger og motivasjon i forhold til, programmering innenfor naturfag.

Formål

Det har kommet nye læreplaner i naturfag der programmering er blitt inkludert som en viktig del. I forbindelse med dette gjennomfører vi et forskningsprosjekt som undersøker hvordan elever jobber med programmeringsoppgaver i naturfag, hvilke utfordringer de støter på, og hvordan programmering påvirker motivasjonen deres for naturfag.

Hva vil det innebære å delta i undersøkelsen

Vi ønsker å invitere noen elever til å delta i mindre gruppediskusjoner etter gjennomføringen av et programmeringsopplegg for å høre hvordan dere opplevde å jobbe med dette. Som en liten påskjønnelse vil det bli lett bevertning under gruppeintervjuet. Vi ønsker å gjøre lydopptak fra disse diskusjonene. Hvis du gir forskningsprosjektet tillatelse til dette vil det kun være prosjektet som har tilgang til dataene. Læreren din vil ikke ha anledning til å høre lydopptaket eller få vite hva som ble sagt under fokusgruppeintervjuet. Forskningsprosjektet vil bruke opplysningene bare til forskning på elevers forståelse, motivasjon og læring knyttet til programmering i naturfag.

Hva skjer med informasjonen

Vi vil behandle alle personopplysninger konfidensielt, og kun bruke innsamlet data til forskningsformål. Vi vil ikke bidra til vurdering av deg, og det vi finner ut vil ikke påvirke din vurdering i naturfag. Det er kun forskningsprosjektet som vil ha tilgang til personopplysninger, og det vil ikke være mulig å gjenkjenne deg ut fra rapporter/oppgaver som blir laget ved hjelp av data samlet inn gjennom dette prosjektet. Innen juni 2022 vil alt materiale der du kan identifiseres være slettet eller anonymisert. Anonymiserte data vil kunne lagres også etter juni 2022 for oppfølgingsstudier. Som forskere forholder vi oss til etiske regler om hvordan personopplysninger kan brukes. Vi behandler opplysninger om deg basert på ditt samtykke. På oppdrag fra Universitetet i Oslo har NSD – Norsk senter for forskningsdata AS vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

Frivillig deltagelse

Det er frivillig å delta i undersøkelsen. Om du velger å delta kan du når som helst trekke ditt samtykke tilbake uten å oppgi grunn. Hvis du gjør dette vil alle personopplysninger om deg slettes. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg.

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke.

Dine rettigheter

Om du deltar i studien, og fremdeles kan identifiseres i datamaterialet, har du rett til:

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

Kontaktinformasjon prosjektledere

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

Einar Løhne Iversen: einarliv@student.uv.uio.no, Ellen K. Henriksen:
e.k.henriksen@fys.uio.no

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

NSD – Norsk senter for forskningsdata AS på epost (personverntjenester@nsd.no) eller på telefon: 55 58 21 17.



UiO

Postadresse: Fysisk institutt, Postboks 1048 Blindern, 0316
OSLO <https://www.mn.uio.no/fysikk/forskning/grupper/skolelab/>