

Lasse Husby & Sindre Lines Arntzen

Fremme programmering i barneskolen

Utviklingsrettet master om bruk av
programmering i naturfag, med fokus på lærere
uten programmeringskompetanse

Masteroppgave i MGLU5209

Veileder: Elisabeth Inge Romijn

Medveileder: Ellen Marie Andersson

Mai 2024



NTNU

Kunnskap for en bedre verden

Lasse Husby & Sindre Lines Arntzen

Fremme programmering i barneskolen

Utviklingsrettet master om bruk av programmering i naturfag, med fokus på lærere uten programmeringskompetanse

Masteroppgave i MGLU5209
Veileder: Elisabeth Inge Romijn
Medveileder: Ellen Marie Andersson
Mai 2024

Norges teknisk-naturvitenskapelige universitet
Fakultet for samfunns- og utdanningsvitenskap
Institutt for lærerutdanning



Kunnskap for en bedre verden

Sammendrag

Dette er en utviklingsrettet masteroppgave, hvor vi har laget en lærerveiledning med undervisningsopplegg forankret i teori. Vi utarbeidet en lærerveiledning og elevoppgaver forbeholdt hovedmålet om å *bidra til at lærere uten programmeringskompetanse kan knytte sammen undervisning i fagtemaet vei, fart og tid med programmering*. For å underbygge målet for oppgaven har vi valgt ut to delmål som passer inn under målet.

Delmål 1: *Utvikle et oppgavehefte og en lærerveiledning som egner seg for lærere som mangler kompetanse innen programmering.*

Delmål 2: *Utvikle et undervisningsopplegg, lærerveiledning og oppgavehefte som knytter fagtemaet vei, fart, tid med programmeringsundervisning.*

Vi har tatt utgangspunkt i et undervisningsopplegg fra Super:bit, og modifisert dette ved hjelp av teorier og rammeverk fra blant annet Benton et al. (2016) og Bjursten et al. (2022). Deretter testet vi det ut, og modifiserte dette etter tilbakemeldinger fra informant gjennom et semi-strukturert intervju, og observasjoner av lærer og elever. Datainnsamlingen ble gjort i en 6. klasse på en skole i Trondheim, med en lærer uten programmeringskompetanse. Det ble gjort 4 gjennomføringer med 4 forskjellige klasser på 6. trinn.

Studien viser tegn til at opplegget og tilhørende lærerveiledning og oppgaveheftet kan være nyttig, og at både lærer og elev opplever et positivt utbytte av økten. Likevel viser funnene fra utprøvingen at ressursen er vel omfattende, da aspekter ble bortprioritert og at det dermed er mangel på innsikt i hvordan de kan fungere. Begrensninger i læreres forberedelsestid kan være utslagsgivende når det kommer til forming av læringsressurs, og her kan det vise seg at videre arbeid kunne vært og laget en film basert på læringsressursen, da dette kan være tidsbesparende.

Studien har fokusert på bruk av stillasbyggende elevtilpasninger som skulle fungere som støtte for både lærere og elever uten programmeringskompetanse. De ble i stor grad bortprioritert i utprøvingen, og vi står dermed med spørsmål om teorien som forteller at disse skal fungere er troverdig. Funnene viser også at implementasjon av fagtema i programmeringsundervisning er utfordrende fordi en må ta hensyn til to sett med forkunnskaper. Observasjonene som ble gjort i klasserommet viser mangel på forståelse mellom variablene vei, fart og tid og hvordan de skulle anvende programmeringen til å gjennomføre utregninger av verdiene. Vår studie viser og at lærere oppfatter programmering som mer meningsfullt dersom de kan anvende dette til å undervise fag. Det krever tid å gjennomføre og planlegge en helhetlig undervisningsøkt hvor det er utvikling i begge fagemnene.

Abstract

This is an educational design research master's thesis, where we have created a teacher's guide with a teaching session, rooted in theory. We prepared a teacher's guide and student assignments to reach the assessment's main goal *of helping teachers without programming skills to link teaching in the subject topic of stretch, velocity and time with programming*. To support the goal of the thesis, we have selected two sub-goals.

Sub-goal 1: *Develop an assignment booklet and a teacher's guide that is suitable for teachers who lack competence in programming.*

Sub-goal 2: *Develop a teaching plan, teacher guide and assignment booklet that links the subject topic of stretch, velocity, time with programming education.*

We have taken inspiration for our teaching plan from Super:bit, and modified this resource with the help of programming theories and frameworks from, among others, Benton et al. (2016) and Bjursten et al. (2022). We then tested our resource and modified this based on feedback from the informant through a semi-structured interview, and observations of the teacher and students. The data collection was done in a 6th grade class at a school in Trondheim, with a teacher without programming skills. There were 4 executions with 4 different classes in 6th grade.

The study shows signs that the teaching plan and the associated teacher guide and the assignment booklet can be useful, and that both teacher and pupil experience a positive outcome from the session. Nevertheless, the findings from the executions show that the teacher's guide is quite extensive, as aspects were not prioritized and that there is thus a lack of insight into how they could work. Limitations in teachers' preparation time can be decisive when it comes to shaping the teacher's guide, and here it may turn out that further work could be the making of a film based on the learning resource, as this can be timesaving.

The study focused on scaffolding student adaptations that should function as support for teachers and students without programming competence. They were largely deprioritized in the executions, and we are thus left with questions about whether the theory that says that these should work is credible. The findings also show that implementing subject topics in programming teaching is challenging, because one must consider two sets of prior knowledge. The observations made in the classroom show a lack of understanding between the variables stretch, velocity and time and how to use the programming to carry out calculations of the values. Our study also shows that teachers perceive programming as more meaningful if they can apply it to teach subjects. It takes time to carry out and plan for a comprehensive teaching session where there is development in both subjects.

Forord

Denne masteroppgaven er avslutningen på våre 5 år på lektorutdanningen ved Kalvskinnet NTNU. Det har vært 5 lange, men fine og lærerike år. Vi har lært mye gjennom master i naturfag, om biologi, kjemi, fysikk og ikke minst programmering. Vi håper oppgaven vår blir lest og kan bane vei for videre forskning innen programmeringsdidaktikk, og bruk av programmering for lærere uten programmeringskompetanse. Vi har hvert fall lært en hel del om programmeringsdidaktikk gjennom utallige teorier, noe som vi kommer til å ta med oss videre nå som vi snart er ferdigutdannede lektorer.

I dette arbeidet er det flere som har bidratt underveis. Vi vil gjerne takke veilederne våre: Elisabeth Inge Romijn og Ellen Marie Andersson, som har satt av mye tid på å lese gjennom oppgaven vår. De har kommet med konstruktive tilbakemeldinger, kommet med innspill og lagt opp til mange fine diskusjoner. Både sene kvelder og tidlige morgninger har blitt satt av til oss, og vi har følt en takhøyde for å spørre om det meste knyttet til masteroppgaven. Vi har følt oss sett, hørt og godt ivaretatt gjennom hele semesteret.

Videre vil vi takke skolen og læreren som tok seg tid til å gjennomføre opplegget vi hadde laget. Det var en positiv og nysgjerrig lærer og elevgruppe som møtte oss, og gjorde denne gjennomføringen mulig. Læreren som møtte oss tok utfordringen på strak arm uten programmeringskompetanse og kastet seg ut i ukjent farvann, for at oppgaven vår skulle komme i mål.

Vi vil også sende en takk til Arne og Bernt som har vært veiledere på seminarene gjennom semesteret. Vi har fått gode innspill og ikke minst gode samtaler om master og litt utenom. Takk også til våre medstudenter for innspill og beroligende ord gjennom flere år.

Takk til naturfagseksjonen på Kalvskinnet for utlån av utstyr, masterplass og grupperom. Også må vi sende en takk til Sit Kafe Kalvskinnet for en god start på dagen, og litervis med kaffe og monster.

Innholdsfortegnelse

1 Innledning	1
1.1 Aktualisering	1
1.2 Programmering i skolen	1
1.3 Tidligere forskning.....	2
1.4 Hensikt.....	3
1.5 Kort om metode.....	4
1.6 Oppgavens struktur.....	4
2 Teori	5
2.1 Hva er programmering?	5
2.2 Programmering i skolen	6
2.3 Programmeringsdidaktikk.....	9
2.3.1 Verktøy.....	9
2.3.2 Samarbeid i programmering	10
2.3.3 Utforskende undervisning i naturfag.....	11
2.4 Vei, fart, tid-didaktikk.....	13
3 Metode.....	14
3.1 Pedagogisk designforskning	14
3.2 Vår studie	15
3.2.1 Første fase: forberedelse av designeksperimentet.....	18
3.2.2 Andre fase: Gjennomgang av designeksperimentet	18
3.2.2.1 Utvalg	18
3.2.2.2 Syklus 1	19
3.2.2.3 Mellom syklusene	19
3.2.2.4 Syklus 2	19
3.2.2.4.1 Datainnsamling	20
3.2.2.4.2 Observasjon.....	20
3.2.2.4.3 Semi-strukturert intervju	21
3.2.2.4.4 Rammer for utprøvingen.....	21
3.2.2.4.4.1 Lærerens erfaringer/bakgrunn	21
3.2.2.4.4.2 Elevenes erfaringer med programmering	22
3.2.3 Tredje fase: Retrospektive analyser	22
3.3 Behandling av data.....	23
3.3.1 Forskningsetikk.....	23

3.3.2 Meldepliktig til Sikt	23
3.4 Metodens pålitelighet og validitet	24
3.4.1 Pålitelighet	24
3.4.2 Validitet	24
4 Resultat og diskusjon	26
4.1 Produktet	26
4.2 Delmål 1: Utvikle et oppgavehefte og en lærerveiledning som egner seg for lærere som mangler kompetanse innen programmering	27
4.2.1. Forberedelse: Lærerveiledningen	27
4.2.2 Forberedelse: Elevtilpasninger	31
4.2.3 Utprøving: Brukervennlighet	33
4.2.4 Utprøving: Lærerveiledningens design	34
4.2.5 Utprøving: Motivasjon vs. læringsutbytte	35
4.2.6 Utprøving: Elevtilpasninger	37
4.2.7 Generell retrospektiv analyse av delmål 1	39
4.3 Delmål 2: Utvikle et undervisningsopplegg, lærerveiledning og oppgavehefte som knytter fagtemaet vei, fart, tid med programmeringsundervisning	41
4.3.1 Forberedelse: Naturfag i programmering	41
4.3.2 Forberedelse: Oppgaveheftet	42
4.3.3 Utprøving: Bruk av oppgaveheftet	43
4.3.4 Utprøving: Elevenes opplevde læringsutbytte	45
4.4 Målet: Bidra til at lærere uten programmeringskompetanse kan knytte sammen undervisning i fagtemaet vei, fart og tid med programmering.	46
5 Avslutning og veien videre	48
Litteratur	50
Vedlegg	55
Vedlegg 1: Lærerveiledningen	56
Vedlegg 2: Oppgaveheftet	56
Vedlegg 3: Intervjuguide	57
Vedlegg 4: Observasjonsplan 4 gjennomføringer	60
Vedlegg 5: Prosessdokument	62
Vedlegg 6: Samtykkeskjema – Elever	63
Vedlegg 7: Samtykkeskjema - Lærer	66

Tabeller og Figurer

Tabell 1: Tabell som trekker frem de karakteriske trekkene for pedagogisk designforskning. Begrepene og forklaringene er hentet fra Van den Akker et al. (2006).	15
Tabell 2: Viser strukturen og innholdet i studien. Fasene og syklusene er vist til venstre, og de gjeldene målene, metodene og resultatene presenteres til høyre. .	17
Figur 1: a) Skjermdump fra et undervisningsopplegg fra superbit (u.å.c), b) og c) skjermdump av eksempler på våre tilpasninger. (Oppgave 2 i oppgaveheftet (Vedlegg 2))	29
Figur 2: Skjermdumper fra lærerveiledningen (Vedlegg 1), bildene viser eksempler på skripter som besvarer oppgave 2 i oppgaveheftet (Vedlegg 2).	31
Figur 3: Skjermdump av elevtilpasningene for oppgave 2 oppgaveheftet (Vedlegg 2).....	33
Figur 4: Skjermdump av kloss fra https://makecode.microbit.org/	42

1 Innledning

1.1 Aktualisering

I en tid hvor datavitenskap og teknologisk innovasjon blir stadig viktigere får ferdigheter som programmering en økende betydning både i samfunnet og i skolen. Gjennom skolens samfunnsmandat skal elevene tilegne seg ferdigheter, kunnskap og kompetanse og åpne døren mot verden og fremtiden, og forberede elevene på dagens liv og fremtidens samfunn (Meld. St. 21 (2016-2017); Kunnskapsdepartementet, 2017). Elevene skal forberede seg for fremtiden, og ifølge UiB (2023) er kompetanser som kunstig intelligens, stordata og koding ansett som viktig i fremtiden.

Samtidig som elevene utvikler en kompetanse innenfor digitale ferdigheter kan det være en positiv bieffekt som følger med programmering. Elevene kan utvikle sine evner som problemløsere gjennom at programmering tilbyr muligheten til å løse problemer på en kreativ måte, det gir skaperglede, nysgjerrighet og muligheten til teste ut ideer for å løse oppgaver (Statped, 2021).

1.2 Programmering i skolen

En utfordring med programmering i skolen er at det ofte er lite kjent blant lærere og elever. I dagens skole er det få lærere som har lært programmering i sin utdanning, det betyr at programmeringskompetansen de sitter på må ha kommet av egeninteresse eller kursing. Dermed er det mange lærere som underviser i dette temaet uten å ha grunnlag for å møte kompetansemålene på en hensiktsmessig måte.

I kompetansemålene etter 7. trinn er det tre mål hvor programmering er relevant. Ett av disse er: "elevene skal utforske, lage og programmere teknologiske systemer som virker sammen" (Kunnskapsdepartementet, 2019.a). Dette forklarer lite om hva elevene skal lære, hvordan de skal lære, ei heller hvilke programmeringsverktøy elevene skal ha kjennskap til. Dermed må lærerne ta valg og sette seg inn i hvordan en kan oppnå måloppnåelse.

Vinnervik (2022) viser at dette er utfordrende for lærerne uten programmeringskompetanse. Dette gjelder også de andre kompetansemålene, da de også er vage og krever tolkning for å kunne håndteres effektivt.

Selv om det er flere kompetansemål for programmering i matematikkfaget (Kunnskapsdepartementet, 2019.b) enn i naturfag (Kunnskapsdepartementet, 2019.a), er

også de ukonkrete og krever tolkning fra lærere for å bestemme hva som skal læres og hvordan øktene skal gjennomføres.

1.3 Tidligere forskning

Vinnervik (2022) gjennomførte intervju med lærere som underviste i 7. til 9. trinn i Sverige. Der kom det frem at lærere med lite eller ingen erfaring med programmering eller teknologi syntes at læreplanen ikke var dekkende nok. Sverige har i motsetning til Norge, ingen krav om at programmering skal implementeres i naturfag, men heller matematikk og teknikkfag, som er ulikt fra Norge. Det kommer heller ikke der tydelig frem hva programmering innebærer når det gjelder øvelser og konsepter, og hvordan en skal tilegne seg programmeringskunnskap (Vinnervik, 2022). Programmering har vært en del av matematikkopplæringen i Sverige siden 2017 (Humble, 2022), Norge er i startfasen når det gjelder programmering i ulike fag, da dette ble et krav i LK20. Derfor finnes det lite forskning i Norge på akkurat dette.

Selv om naturfag ikke har ansvar for programmeringsundervisning i Sverige, har de i likhet Norge valgt å innlemme programmering i allerede eksisterende fag og inkludere det i læreplanen for matematikk. I andre land som for eksempel England og Danmark blir programmering introdusert som et eget teknologifag (Humble, 2022; Bråting & Kilhamn, 2021). Norges tilnærming bringer med seg visse utfordringer, men åpner samtidig opp for unike muligheter. Denne tilnærmingen gjør det naturlig å anvende programmering og teknologi som verktøy for å utforske faget på en ny måte (Humble, 2022). Likevel krever dette at lærerne, som tidligere har undervist i naturfag i tråd med LK06, må tilegne seg ny kompetanse innen programmering og de didaktiske prinsippene som følger med det. Dette innebærer en betydelig investering av lærernes tid.

Det finnes noen ressurser som kan hjelpe uerfarne lærere. Kunnskapsdepartementet forstod at om programmering skulle inn i skolen, måtte de innføre noen tiltak. Tiltaket de satt i gang i 2018 var å sette landets vitensentre i oppdrag om å lage et opplegg som skulle forsterke elevenes og lærernes kompetanse innen programmering. Dette ble formet til Super:bit (Super:bit, u.å.a).

Super:Bit er et eksempel på en nettbasert veiledning utviklet for lærere og elever (Super:bit, u.å.a). Det gir en grundig gjennomgang av hvordan utstyret virker sammen, og

tilbyr mange spennende oppgaver elevene kan gjennomføre. Der vår studie skiller seg ut fra eksisterende ressurser er at vi har valgt å fokusere på anvendelse av programmeringsverktøy for å lære om de naturfaglige fenomenene vei, fart og tid, og sammenhengen mellom dem. Det finnes flere studier som har brukt simulering for å implementere fagtemaer i programmeringsundervisning, og da med for eksempel *Scratch* som verktøy. Vi bruker derimot Micro:Bit, og fokuserer på et fysisk programmeringsverktøy som er observerbart i et fysisk rom.

1.4 Hensikt

Selv om det finnes ressurser for å hjelpe lærere uten kompetanse innenfor programmering er det foreløpig ikke gjort forskning på hvordan disse faktisk fungerer og hvordan det kan tas i bruk. Vår studie skal bidra med en lærerveiledning (Vedlegg 1) og et tilhørende oppgavehefte (Vedlegg 2) som lærere uten programmeringskompetanse kan ta i bruk. Vi ser også på hvordan et undervisningsopplegg som knytter programmering med fag blir brukt av en erfaren lærer uten kompetanse innenfor tema som skal undervises.

Målet vårt for oppgaven er:

Bidra til at lærere uten programmeringskompetanse kan knytte sammen undervisning i fagtemaet vei, fart og tid med programmering.

Hensikten med vår oppgave er altså å lage et undervisningsopplegg i naturfag med programmering, for læreren som ikke har tilegnet seg denne kompetansen. Vi har utviklet to delmål for å svare til det overordnede målet. Det første delmålet vårt er å hjelpe en lærer med liten eller ingen programmeringskompetanse å ta i bruk et undervisningsopplegg ved hjelp av en lærerveiledning og et tilhørende oppgavehefte. Det andre delmålet tar for seg hvordan undervisningen evner å integrere vei, fart, tid med programmeringsundervisningen.

1. **Delmål 1:** *Utvikle et oppgavehefte og en lærerveiledning som egner seg for lærere som mangler kompetanse innen programmering.*
2. **Delmål 2:** *Utvikle et undervisningsopplegg, lærerveiledning og oppgavehefte som knytter fagtemaet vei, fart, tid med programmeringsundervisning.*

Før dette prosjektet startet delte vi en del forutsetninger med vår informant og sannsynligvis mange av lærerne vi lager produktet for. Vi var begge interesserte i programmering og syntes undervisning i dette virket spennende, men vi hadde ikke gjort

det med skoleelever tidligere. Vi har gått vår skolegang under læreplanene for LK97 og LK06 og har, i likhet med lærerne i skolen i dag, aldri møtt programmering i grunnskolen. I LK06 kom digitale ferdigheter (Kunnskapsdepartementet, 2013), som betyr at vi har møtt bruk av teknologi i skoleløpet, men aldri programmering. Vi ble møtt med programmering ved universitetet, noe som vi erfarte som krevende, men samtidig spennende og nyttig. Dette skapte en interesse, som gjorde at vi falt på dette valget av masterretning. Vi valgte en utviklingsrettet masteroppgave da vi mener dette er mest relevant for oss i fremtiden, og noe vi kan ha stor nytte av senere.

1.5 Kort om metode

Vi har gjennomført en pedagogisk designforskning. Hver fase og hver syklus blir diskutert i denne teksten og formålet er å konstruere et omfattende bilde av utviklingen til produktet. Datamaterialet vi har samlet inn kommer fra samtaler med veilederne våre, observasjoner gjort under utprøvingen, gjennomført av vår informant og påfølgende intervju av denne læreren. Basert på dette vil vi til slutt diskutere potensielle justeringer og forbedringer som kan implementeres i vår læringsressurs. Det er viktig å understreke at det ikke er realistisk med en ferdigstilling av vår læringsressurs innenfor rammene av denne masteroppgaven, gitt tidsbegrensninger og oppgavens kompleksitet. Derfor er produktet (Vedlegg 1 & 2) det samme som vi sendte til informanten for utprøving.

1.6 Oppgavens struktur

Oppgavebesvarelsen er bygd opp ved at vi først går gjennom litteraturen vi har brukt gjennom studien (Kapittel 2). Deretter presenterer vi metoden (Kapittel 3) og hvordan vi har lagt opp forskningen. I resultat og diskusjon (kapittel 4) presenterer vi kort innholdet i produktet. Deretter analyserer vi de viktigste funnene fra fase 2 for å avslutningsvis reflektere over hvilke endringer som kunne blitt gjort ved videre forskning.

2 Teori

I dette teorikapitlet utforsker vi programmeringens essens, dens rolle i skoleundervisningen i dag, og de kravene som blir stilt lærere. Vi dykker inn i programmeringsdidaktikk, med fokus på anvendelse av verktøy og parprogrammering som en pedagogisk strategi. Videre ser vi på hvordan utforskende læring implementeres i programmeringsundervisning for å fremme en dypere forståelse og ferdighetsutvikling blant elevene. Se rammeverket til Benton et al. (2016). Aspekter fra strategiene til Bjursten et al. (2022) blir så presentert, disse kan være nyttige for å innlemme programmering i naturfagsundervisningen og mens en beholder det utforskende aspektet ved faget og holdninger til programmering.

2.1 Hva er programmering?

Herunder har vi en kort gjennomgang av begrep som er relevante for fagfeltet denne oppgaven faller under. Dette gjør vi for å plassere oppgaven innenfor én forståelse av begrepene, og dermed sørge for en fellesforståelse av hvordan vi bruker begrepene videre. Vi trekker også frem begrep vi ikke tar med videre i oppgaven.

Forskjellen mellom programmering og koding kan være vanskelig å definere, men i Gjøvik & Torkildsen (2019) blir det gjort distinksjoner mellom begrepene for å gi et bilde av omfanget. Koding defineres som en enkelaktivitet hvor man reformulerer logikk i et spesielt programmeringsspråk, mens programmering tar for seg helheten ved aktiviteten. Programmering blir definert som helheten av handlingsmønstrene når man løser et problem med hjelp av et program (Gjøvik & Torkildsen, 2019). For vår del betyr det at vi må se på forberedelsen hos elevene, uttestingen, datainnsamlingen og om de kan definere problemet som løst eller ikke.

Basert på teorien som er tilgjengelig for programmeringsundervisning ser man at begrepet *algoritmisk tenkning* har en sentral posisjon (f.eks. Haraldsrud et al., 2020; Sunde & Øen, 2022; Pörn et al., 2021). Begrepet tar for seg evnen til å løse matematiske problemer ved hjelp av organisasjon og analyse av informasjon. Det er fremgangsmåten som er interessant her, og ved abstraksjon og modeller skal man kunne løse et problem i den fysiske verden (Sunde & Øen, 2022). Siden vi skriver en oppgave om naturfaglige temaer, tar vi for oss fysiske fenomener som elevene skal manipulere ved hjelp av programmering. Dermed kan begrepet være like relevant for oss som de nevnte artiklene og masteroppgavene. Likevel velger vi å ikke forholde oss til dette, da det ikke hjelper oss å svare på våre forskningsmål.

Tekstprogrammering kan anses som den beste måten å lære programmering på (Haraldsrud et al., 2021), men vi har likevel bestemt å forholde oss til blokkprogramering, da det er et pedagogisk verktøy som egner seg bedre for begynneropplæringen innen temaet. Man slipper å forholde seg til syntaks og skrivemåten (Haraldsrud et al., 2021), og kan fokusere på gjenkjennelse av koden fremfor å huske riktig skrivemåte (Bau et al., 2017).

2.2 Programmering i skolen

I dette delkapittelet ser vi på programmering i kompetansemålene for naturfag og litteratur som forteller om læreres forhold til disse og hvordan mange lærere ikke føler seg klare til å undervise i eller med programmering.

I lærerplanen for naturfag fremheves tre kompetansemål etter 7. trinn som omtaler programmering formulert som følger:

- "Designe og lage et produkt basert på brukerbehov"
 - "Reflektere over hvordan teknologi kan løse utfordringer, skape muligheter og føre til nye dilemmaer"
 - "Utforske, lage og programmere teknologiske systemer som består av deler som virker sammen"
- (Kunnskapsdepartementet, 2019.a, s. 8).

Disse sier lite om hva elevene skal lære i programmering, hvordan elevene skal lære, og hvilke verktøy eller programvarer læreren skal ta i bruk for at elevene skal oppnå ønsket kompetanse. Disse manglende spesifikasjonene setter store krav til læreres profesjonelle kunnskap for å lage programmeringsopplegg opp mot lærerplan (Vinnervik, 2023). Dermed

er det skuffende å lese hvordan lærere faktisk opplever programmeringsundervisning. I Pörn et al. (2021) blir det forsket på lærernes holdninger til programmering etter det ble innlemmet i læreplanen. I artikkelen vises det at flere lærere ikke føler seg klar for å undervise i tema da dette er nytt for dem, de har ikke gjennomgått nok kursing og noen vet heller ikke hva programmering er (Pörn et al., 2021). Denne mangelen på kompetanse blant lærerne vil direkte påvirke elevenes utvikling av programmeringskompetanse (Ball et al., 2008).

En annen utfordring for fagfeltet og som til slutt kan svekke undervisningen er en foreløpig mangel på relevant forskning på programmering i Norge. Det finnes mye internasjonal forskning som kan brukes, men med utenlandsk forskning kommer utenlandske kontekster. Eksempelvis har mange land programmering- og andre IKT-relevante læreplanmål under et eget teknikkfag (Carroll & Ward, 2016; Humble, 2022). Det kan gi teorier som ikke bør gjelde den norske skolen. Carroll og Ward (2016) har for eksempel presentert noen regler for bruk av teknikk i naturfaget, der de blant annet sier at naturfagtimene ikke skal brukes til å lære nye programvarer, men heller bruke allerede lærte verktøy. Dette er i strid med den norske læreplanen. I Norge deler naturfaget ansvaret om å utvikle en forståelse av programmering med andre fag (Kunnskapsdepartementet, 2019), og man har dermed ikke mulighet til å kun bruke ferdig forståtte programvarer. Derfor mangler det ressurser som baserer seg på norsk/nordisk forskning, men som heller kommer av generelle, internasjonale idéer og teorier.

Vi har etter beste evne forsøkt å finne litteratur som forteller oss hvordan programmering blir undervist i Norge etter kunnskapsløftet 2020 tredde i kraft. Siden Norge er i oppstartfasen ved å implementere programmering i kompetansemålene er det lite forskning på dette området i dag. Det vi har funnet er forskning på hvordan nabolandene våre Sverige og Finland har implementert programmering i skolen, og hvordan lærerne forholder seg til det. Finland og Sverige introduserte programmering i læreplanverket for første gang i henholdsvis 2016 og 2017 og har, i likhet med Norge, integrert tema i allerede eksisterende fag (FNBE, u.å; Humble, 2022.). Derfor velger vi å bruke litteratur fra disse landene for å diskutere resultatene våre opp mot forskning.

Kravet fra kunnskapsdepartementet (2019.a) om at kjerneelementene som gjelder for fagene skal gjennomsyre undervisningen bygger opp viktigheten av programmering i naturfaget. Kjerneelementet "Teknologi" forteller leseren om at elever blant annet skal kunne bruke og lage teknologi, programmering og modeller til å løse problemer (Kunnskapsdepartementet, 2019.a). I læreplanen (Kunnskapsdepartementet, 2019.a) står

det samtidig at kjerneelementene som gjelder for faget skal kunne kombineres, og man kan dermed kombinere programmeringsundervisning med andre læreplanmål. I denne studien legger vi stor vekt på å oppfylle dette kravet ved å tilrettelegge for kombinasjonen av programmeringsundervisning med det naturfaglige temaet vei, fart og tid.

Det eksisterer et ønske fra lærere om å få elevene til å lære mer enn bare koding når man gjennomfører programmeringsundervisning (Pörn et al., 2021). Ser man på kjerneelementene og kompetansemålene i læreplanen eksisterer det samme ønske fra politikerne (Kunnskapsdepartementet, 2019.a). Her skal programmering inngå blant annet i fag som naturfag, matematikk og kunst og håndverk. Ved å integrere programmeringsundervisning i allerede eksisterende fag, skal man kunne sørge for at programmering blir brukt som et verktøy til å lære andre tema, samtidig som man for eksempel utforsker hvordan en Bit:bot kan kjøre en gitt strekning. Forskning sier at ved integrering av programmering i allerede eksisterende fag utvikler man ikke bare programmeringskompetansen, men også andre fagrelevante tema (Popat & Starkey, 2019).

Når man til slutt er i klasserommet er det viktig å være klar over sin egen rolle som lærer. Det kan være et vanlig standpunkt for lærere å ta, at noen elever har evner som tillater å lære og andre ikke, når det kommer til programmering (Brown & Wilson, 2018). Brown & Wilson (2018) skriver at det er mulig for lærere å tro at noen elever ikke kan lære seg dette, mens andre kan. Dermed er det mulig at noen elever ikke får tilstrekkelig veiledning, ettersom de er klassifisert som uhjelpelige. Dette er selvfølgelig en myte (Brown & Wilson, 2018). Derimot har lærere en viktig rolle når det kommer til å kartlegge og adressere misoppfatninger og andre utfordringer elevene har (Qian & Lehman, 2017). Lærere med liten erfaring og kunnskaper om programmering, kan ofte bidra til å skape misoppfatninger, og lærer ofte elevene regler i programmering fremfor årsaker til utfall, som kan være dårlig for elevenes forståelse av programmering (Qian & Lehman, 2017). For å unngå å skape misoppfatninger er det viktig at læreren viser gode programmeringseksempler og viser at de fungerer, da mindre kompetente elever ofte støtter seg til denne type modellering (Fleury & Bentley, 1991).

Roalsvik (2023) sin forskning på emnet viser at elevgruppen i hennes studie er motivert for programmering. Hun opplever at elevene har en utforskningstrang, samt et ønske om å gjøre aktiviteter utenom lærebøkene. På den andre siden eksisterer det også studier der elevgruppen er mindre engasjert og opplever opplegget som kjedelig (Haraldsrud et al., 2020). I likhet med andre fagemner er det dermed nødvendig å sørge for at undervisningen er nyttig, engasjerende og inkluderende for alle. Dette kan ta tid, spesielt i tilfeller der

læreren ikke har tilstrekkelig med kompetanse (Haraldsrud et al., 2020; Humble, 2022). Ifølge Vinnerviks studie (2022), er det mange lærere som underviser i programmering uten den riktige kompetansen. For disse lærerne kan utviklingen av et undervisningsopplegg bli svært krevende spesielt med en vag læreplan, som krever at læreren velger innhold, metode og læremål selv (Haraldsrud et al., 2020). Det blir derfor er det viktig å sette seg inn i hvilke didaktiske muligheter man har innenfor fagtemaet.

2.3 Programmeringsdidaktikk

Dette delkapittelet skal ta for seg aspekter fra litteratur som forteller om hensiktsmessig programmeringsundervisning. Det kan leses som utgangspunktet for produktet vi har designet og grunnlaget for videre diskusjon.

Forsker ved NMBU i programmeringsdidaktikk Morten Munthe, forteller i en video for fagforeningen Tekna at programmering egner seg best for dybdelæring i faget, og ikke ved innlæring av nytt stoff (Tekna, 2021, S. Tidskode 13:09). Dybdelæring forklares følgende av utdanningsdirektoratet:

“Vi definerer dybdelæring som det å gradvis utvikle kunnskap og varig forståelse av begreper, metoder og sammenhenger i fag og mellom fagområder. Det innebærer at vi reflekterer over egen læring og bruker det vi har lært på ulike måter i kjente og ukjente situasjoner, alene eller sammen med andre” (Utdanningsdirektoratet, 2019).

Dette er i tråd med det Pörn et al. (2021) fant, nemlig at for å kunne oppnå utvikling innenfor fagrelevante tema så er man nødt til å integrere dette eksplisitt i programmeringsundervisningen. Det holder ikke å forvente at det kommer av seg selv fordi man jobber med det aktuelle tema. Derfor må undervisningsopplegget designes slik at faginnholdet som ikke er programmeringsrelatert kommer tydelig frem. Oppnår man å bruke programmering sammen med andre fagtemaer kan man se en utvikling hvor elevene husker bedre, de forstår og evner å anvende innholdet på en bedre måte (Pörn et al., 2021).

2.3.1 Verktøy

Skal man oppnå de didaktiske aspektene for programmeringsundervisning er man samtidig avhengig av å gjøre hensiktsmessige vurderinger av valg av verktøy. Det finnes mange

programmeringsverktøy som lærere kan ta i bruk og et av de mest populære verktøyene i barneskolen i dag er Micro:bit. Satsningen rundt Super:bit hadde som mål å få utstyret ut i alle skolene i Norge (Sunde & Øen, 2022). Man vil derfor finne både Micro:bit og Bit:bot i de fleste skolene i landet.

Micro:bit har mange av de samme funksjonene som en datamaskin og er på størrelse med en fyrstikkeske. Den har knapper, lyssensorer, mikrofon og er en programmeringsplattform som er tilgjengelig gjennom nettleser, nettbrett og mobil (Statped, 2023; The Micro:bit Educational Foundation, u.å). Makecode er programmet til micro:bit og kan programmeres til å kjøre servoer, bit:bots, andre roboter og ledpærer. Dette programmeringsverktøyet adskiller seg ut fra sine forgjengere ved å tilrettelegge fra rent skjermbasert programmering til fysisk programmering. Micro:bit muliggjør utforskning på et bredere spekter av problemløsningsoppgaver som er lette å gjennomføre og etterprøve. Elevene får håndfaste resultater som lett kan verifiseres gjennom fysiske resultater gjennom programmeringen.

2.3.2 Samarbeid i programmering

Parprogrammering er en øvelse der to elever programmerer over én pc, på det samme designet og/eller kode (Nagappan et al., 2003). Undersøkelsen til Nagappan et al., (2003) av et programmeringskurs med universitetsstudenter viste at studenter som programmerte i par ga pedagogiske fordeler, som bedre karakterer, mer tilfredshet med oppgaver og større selvtillit mot nye oppgaver. Forskningen viste i tillegg at elever som har programmert i par har produsert koder med høyere kvalitet og større gjennomførelse enn de som jobbet alene, og at de gjorde det bedre individuelt i programmering enn de som programmerte individuelt hele veien (Nagappan et al., 2003).

Studien viste også at studenter som jobbet i par brukte hverandre mer enn lærere om de lurte på noe, og de jobbet mer selvstendig. Dette gjorde at lærerne hadde bedre tid til å gå og hjelpe studenter som satt fast. Studenter som programmerte i par forklarte senere fordeler ved å programmere i par som at de ikke trengte å vente på hjelp, hvor de heller fikk umiddelbare svar fra læringspartner (Nagappan et al., 2003). Som støttes av annen forskning (Newhall et al. 2014; Brown & Wilson, 2018; Haraldsrud et al., 2020).

Gjennom undersøkelser viser det seg at elever jobber best i par, da dette utviklet programmeringsspillet og språklige ferdigheter (Bjursten et al., 2022; Haraldsrud et al., 2020). Dette er altså et didaktisk knep som ofte trekkes frem for å få elevene til å utvikle hverandres kompetanse er parprogrammering (f.eks. Nagappan et al., 2003; Brown &

Wilson, 2018; Haraldsrud et al., 2020). Med forbeholdt at elevene klarer å samarbeide i par, hvor det i noen tilfeller kan være hensiktsmessig å dele opp enkelte par til individuelle grupper.

Det kan oppstå enkelte utfordringer ved bruk av parprogrammering. I noen grupper der elevene var delt i tilfeldige programmeringspar, var enten ferdighetsnivået veldig forskjellig eller det var en dårlig personlig match. Dette førte til at gruppene ble oppløst. En annen utfordring med parprogrammering er at en part kan være dominerende og tar for seg hele arbeidsmengden, dette setter et krav til lærer om å være observant på gruppekjemien og hvordan de forskjellige parene arbeider Nagappan et al., (2003). Det vil derfor være hensiktsmessig om en lærer har god kjennskap til klasserommet og hvilke grupper som kan arbeide godt sammen. Et annet valg en lærer må gjøre når det kommer til programmering er hvilke verktøy en vil ta i bruk, noe som ofte påvirkes av flere faktorer som skolens ressurser, hvordan læreren eller ledelsen har tolket lærerplanen og læreren teknologiske bakgrunn (Makki et al., 2018).

2.3.3 Utforskende undervisning i naturfag

Det er ingen entydig definisjon på utforskende undervisning (Staberg et al., 2020). En forklaring på dette er fra Hazelkorn et al. (2015), som forklarer at utforskende undervisning er en prosess der elevene undrer seg, gjennomfører undersøkelser for å bygge nye forståelser, meninger og kunnskap. Elevene skal lære å kommunisere det de har lært med andre medstudenter, og anvende dette i nye situasjoner.

Staberg et al (2020) forteller at lærere ofte vurderer utforskende arbeid etter engasjement blant elevene, og diskusjoner som hva elevene gjør og ikke gjøre. Og læreren tolker ofte engasjementet blant elevene til en vellykket økt, uten at elevene nødvendigvis sitter igjen med stort læringsutbytte, hvor det ville vært hensiktsmessig for læringsutbytte om elevene fikk muligheten til å reflektere over aktiviteten (Staberg et al., 2020). Derfor har vi sett etter teori som beskriver metoder å gjennomføre utforskende undervisning.

Vi har valgt å forholde oss til én måte å se på utforskende undervisning i programmering da denne passer vårt produkt og opplegg best. Vi har valgt å ta i bruk rammeverket til Benton et al., (2016) som omhandler pedagogiske strategier ved et undervisningsopplegg i programmering. Sammen med et utvalg lærere lagde Benton et al., (2016) et rammeverk kalt 5e som skulle ta for seg pedagogiske strategier for et undervisningsopplegg. Dette rammeverket var bestående av 5 konstruksjoner: Explore (utforske), explain (forklare),

envisage (se for seg), exchange (utdype seg i andres strategier), BridgE (linker mellom programmering og her matematiske idéer). Som supplement til dette rammeverket har vi også tatt med de pedagogiske strategiene for programmering av Bjursten et al., (2022). Deriblant kontekstualisering, som omhandler det samme som bridgE, hvor elevene skal se sammenhenger mellom fag, programmering, og det virkelige liv.

I tillegg til de didaktiske idéene som direkte er knyttet til utforskende undervisning, er det andre teorier som skal hjelpe elevene til å utvikle sin programmeringskompetanse. Blant disse er det Bjursten et al., (2022) kaller stillasbyggende oppgaver. Disse har til hensikt å støtte opp under elevenes forståelse, da programmeringsspråket kan virke vanskelig uten støtte. Gomes & Mendes (2007) forklarer blant annet at det kan være fruktbart for elevene med utfylling av uferdige programmer for å skape en sammenheng mellom dataprogramløsning og generell problemløsning. Samtidig skal elevene utsettes for en hensiktsmessig progresjon i programmeringsarbeidet (Bjursten et al., 2022). Dette betyr at elevene skal møte oppgaver som gir en hensiktsmessig utfordring, og som støtter deres utvikling.

Det er flere ulike teorier om programmeringsdidaktikk, blant annet algoritmisk tenkning oversatt fra computational thinking (ct), som er et begrep som går igjen. I oversettelsen til norsk er det allerede problemer, da dette ikke må forveksles med algorithmic thinking, som er et matematisk begrep. Algoritmisk tenkning tar for seg å vurdere hvilke steg som skal til for å løse et problem, og bruke sin teknologiske kompetanse til å løse problemet ved hjelp av programmering eller en datamaskin (Kunnskapsdepartementet, 2019b). Vi har valgt bort computational thinking, da vi tolker det dit hen at dette omhandler læringsutbytte blant elever, som ikke er relevant for vår oppgave. Vi valgte også bort mye omtalte PRIMM, som er en forskningsbasert tilnærming til å planlegge opplegg i programmering, og først og fremst rettet mot matematikkundervisning gjennom modellering og utforskende læring (Wordpress. 2018, August). Vi ser rammeverket til Benton et al., (2016) som mer spesifikt utviklet programmeringsundervisning, og kan knyttes opp mot naturfag. 5e rammeverket til Benton et al., (2016) må ikke forveksles med 5e modellen til Bybee et al., (2006), som rammeverket bygger på. Vi referer til Benton et al., (2016) når vi nevner rammeverk heretter, slik at det ikke skal forveksles med Bybee et al., (2006).

Ifølge Carroll og Ward (2016) gjelder det anbefalinger for implementering av teknologi og IKT-verktøy i naturfaget. De mener blant annet at bruken av det ikke må gå på bekostning av naturfagets egenart, og hva det vil si å ha god naturfaglig undervisning. I denne sammenhengen handler det om å beholde det utforskende aspektet til emnet. Dårlig

undervisning kan blant annet kjennes igjen ved at elevene blir passive observatører (Carroll & Ward, 2016), ved bruk av 5e-rammeverket i Benton et al. (2016) oppnår man at elevene holder seg involverte og delaktige.

2.4 Vei, fart, tid-didaktikk

For å kunne presentere et opplegg som integrerer det naturfaglige temaet vei, fart, tid må vi legge frem det teoretiske grunnlaget for valgene vi gjorde i utviklingen av opplegget. Det er flere av teoriene som allerede er nevnt som også vil være gjeldende for denne delen av oppgaven, men herunder vil vi fokusere på didaktiske idéer som kun passer for delmål 2.

Det er kjent at elever kommer til klasserommet med mange hverdagsforestillinger knyttet til forholdet mellom vei, fart og tid. Disse idéene er det viktig for elevene å få utforske, da det kan bli utfordrende å forstå modeller og andre hjelpemidler, om de ikke fungerer innenfor deres forståelse av tema (Marshall & Carrejo, 2008). Å forstå hva variablene betyr og hvilke konsekvenser endring av de har er viktig for å senere kunne anvende kunnskapen opp mot standard modeller, som for eksempel vei, fart, tid-trekanten (Hestenes, 1992). Marshall og Carrejo (2008) trekker frem denne trekanten og forklarer manipulasjon av ligningen som den største utfordringen. Elever forstår ofte konseptet den tilbyr, men uten forståelse for hvordan variablene fungerer på hverandre, viser det seg at elevene ofte sliter med å bruke denne korrekt (Marshall & Carrejo, 2008).

3 Metode

I dette kapitlet skal vi beskrive metodene som ble benyttet i denne utviklingsrettede masteroppgaven i naturfag. Først i kapitlet skal vi presentere pedagogisk designforskning som danner grunnlaget for oppgaven. Senere vil vi beskrive detaljene i datainnsamlingsprosessen, valg av deltakere, forskningskontekst og de forskjellige metoder som ble brukt til innsamlingen av datamateriale. Vi har valgt å representere informanten vår ved det fiktive navnet "Kim", og omtales som "hen". Vi henviser til Kim når vi skriver "informanten".

Deretter skal vi omtale dataanalyseprosessen, herunder hvordan vi strukturerte datainnsamlingen av et semistrukturert intervju og observasjonene vi gjorde. Vi skal se dette i lys av de etiske retningslinjene til SIKT og generell etikk ved bruken av denne type datainnsamling. Avslutningsvis skal vi diskutere metodens pålitelighet og validitet, samt oppsummere implikasjoner med metoden.

*Dette kapitlet kan minne om eksamensbesvarelsen vår i emnet MGLU5208. Vi fikk mulighet til å levere tenkt metodekapittel før mastersemesteret begynte, og med det vil det være tenkelig at det blir noen utslag på selvplagiering. Det er gjort flere endringer, men mye av innholdet har forblitt likt.

3.1 Pedagogisk designforskning

Pedagogisk designforskning er en metode som i nyere tid har blitt en anerkjent forskningsmetode. Fra begrepets første benevnelse i 1992, har det stadig utviklet seg til å bli en mer og mer akseptert måte å utvikle undervisningsopplegg på (Bjørndal, 2013). Metoden kan beskrives som en prosessorientert måte å drive forskning, hvor det generelle målet vil være å legge til rette for konkrete tiltak som kan gi bedre undervisning og dermed sørge for bedre læring (Bjørndal, 2013; Van den Akker et al., 2006).

Det kan være flere måter å drive designforskning på, men det er presentert noen kriterier som man bør være klar over for å ta i bruk metoden. Van den Akker et al. (2006) bruker et utvalg av tidligere forskning på temaet til å gi fem karakteristiske trekk de mener designforskning bør være bygd opp av.

Tabell 1: Tabell som trekker frem de karakteristiske trekkene for pedagogisk designforskning. Begrepene og forklaringene er hentet fra Van den Akker et al. (2006).

1	<i>Intervensjon</i>	Forskningen skal gjøre inngrep i den virkelige verden
2	<i>Syklisk</i>	Forskningen skal brukes <i>syklisk</i> , i design, evaluering og revisjon
3	<i>Proessorientert</i>	<i>Proessorientert</i> tilnærming til forskningen, som vil si at det skal være fokus på forståelse og utvikling av inngrep
4	<i>Nytteorientert</i>	Måloppnåelsen til designet skal være målt i dens evne til å være praktisk for brukeren i en faktisk kontekst
5	<i>Teoriorientert</i>	Designet skal være basert på teori og felttesting skal/kan videreutvikle teorien

I Tabell 1 er de fem karakteristiske trekkene presentert, de består av (1) *Intervensjon*, hvor forskningen skal gjøre inngrep i den virkelige verden. (2) Forskningen skal brukes *syklisk*, i design, evaluering og revisjon. (3) *Proessorientert* tilnærming til forskningen, som vil si at det skal være fokus på forståelse og utvikling av inngrep. (4) *Nytteorientert*, måloppnåelsen til designet skal være målt i dens evne til å være praktisk for brukeren i en faktisk kontekst. (5) *Teoriorientert*, designet skal være basert på teori og felttesting skal/kan videreutvikle teorien (Bjørndal, 2013; Van den Akker et al., 2006).

3.2 Vår studie

Vi ønsket at opplegget introduserte en ny måte å implementere Micro:bit i ordinær undervisning, og gjennom observasjoner og intervju skulle vi se hvor godt dette fungerer. Hensikten av pedagogisk designforskning er ikke bare å utvikle et undervisningsopplegg, men å prøve nye måter å takle en utfordring og dermed utvikle teorier for læringsprosessen

(Øgreid, 2021). Gjennom syklusene, som blir presentert senere, ønsket vi å utvikle en forståelse av vår intervensjon i klasseromsundervisningen og ved å beskrive prosessen gjennom masteroppgaven, gi et teorigrunnlag for bruk av vårt undervisningsopplegg. Tabell 1 og litteraturen til Van den Akker et al. (2006) og Bjørndal (2013) har visse krav til hva kreves av en studie å være pedagogisk designforskning. Vår studie er en intervensjon (1) og gjør et inngrep i den virkelige verden, fordi vi lager et produkt som skal hjelpe yrkesaktive lærere og dette blir testet i et eksisterende klassemiljø. Oppgaven er syklisk (2), dette kommer vi tilbake til fra kapittel 3.2.2. Vi har vært prosessorientert (3) gjennom studien, dette vises blant annet i en stadig utvikling av produktene. Fra å lage et opplegg kun basert på egne tanker, til å gjøre dypdykk i eksisterende litteratur og revidere deretter. Studien er nytteorientert (4) gjennom vårt mål om å lage et produkt som potensielt kan bedre læreres opplevelse av programmeringsundervisning. Til slutt, som nevnt, er oppgaven teoriorientert (5) fordi vi har basert lærerveiledningen (Vedlegg 1) og oppgaveheftet (Vedlegg 2) på eksisterende teori.

Et pedagogisk designeksperiment består av tre faser, forberedelse, gjennomføring og retrospektiv analyse og deres tilhørende sykluser (Bjørndal, 2013; Cobb et al., 2003). Da vår studie har et begrenset omfang, deler vi ikke karakteristikken som man kan tenke en doktoravhandling ville hatt, å ha flere sykluser i hver av disse fasene. Syklusene i vår studie kommer bare i spill i gjennomføringsfasen.

Vi har laget en tabell (Tabell 2) som viser hvordan forberedelser og syklisk arbeid vil fungere/fungerer som prosessen for arbeidet i dette prosjektet.

Inspirert av masteroppgaven til Brattetaule (2022) og hvordan hun bruker tabellen presentert i Prediger & Zwetschler (2013).

Tabell 2: Viser strukturen og innholdet i studien. Fasene og syklusene er vist til venstre, og de gjeldene målene, metodene og resultatene presenteres til høyre.

FASE 1 Forberedelsesperiode Forskere (vi) og veiledere.	Mål	Utvikle en forståelse for Micro:bit, bit:bot, makecode, teori om programmeringsdidaktikk og utvikle et undervisningsopplegg og tilhørende lærerveiledning og oppgavehefte.
	Metode	Arbeid med teori angående tema, både programmering og didaktikk. Bruke blant annet Benton et al. (2021) rammeverk for å sørge for nytteverdi.
	Resultat	Første utkast til undervisningsopplegg.
FASE 2 Første syklus Forskere (vi) og veiledere.	Mål	Videreutvikle lærerveiledning og oppgaveheftet og gjøre det klart til utsendelse til informant.
	Metode	Bruk av veiledere. Gjennom en åpen dialog sørget vi for å få innspill på endringer som måtte gjøres og hva som kunne fungere godt.
	Resultat	Andre utkast av undervisningsopplegg.
FASE 2 Andre syklus Forskere (vi) og informant	Mål	Få resultat og grunnlag for diskusjon
	Metode	Læreren/informanten fikk utlevert andre utkast av undervisningsopplegget og skulle gjennomføre dette. Gjennom observasjon og et semi-strukturert intervju får vi datagrunnlaget som brukes i den retrospektive analysen.
	Resultat	Datagrunnlag for videre diskusjon
FASE 3 Retrospektiv analyse Forskere (vi)	Mål	Gi innsikt i hvilke endringer som bør gjøres i vår ressurs
	Metode	Gjennom analyse og presentasjon av våre viktigste funn, forsøker vi å gi innblikk i hva som fungerer godt og hva som bør endres for å oppnå målene vi har satt for oppgaven.
	Resultat	Konkluderende refleksjon av studiens betydning.

3.2.1 Første fase: forberedelse av designeksperimentet

Utgangspunktet for første fase av pedagogisk designforskning var å utvikle et undervisningsopplegg som kunne testes. Valgene vi gjorde ble forankret i teori og i samråd med veiledere. I denne delen bruker forskeren spesifikk teori for emnet til grunn for å gjøre valg, og forskeren velger selv hvordan disse kan anvendes (Bjørndal, 2013).

Vårt opplegg er utviklet basert på kompetansemålene for det gjeldene alderstrinnet, programmeringsdidaktikk, generell undervisningsteori og teori knyttet til undervisning om vei, fart, tid. Samtidig har vi sett på allerede eksisterende undervisningsopplegg knyttet til bruken av Micro: bit. Alt dette ble brukt sammen med våre antakelser av hvordan programmeringsundervisning kan gjennomføres.

3.2.2 Andre fase: Gjennomgang av designeksperimentet

Denne fasen handler om å skaffe seg resultater som kan gi grunnlag for utvikling innenfor målene vi hadde satt. Gjennom utprøving av lærerveiledningen og oppgaveheftet (Vedlegg 1 & 2), skulle vi kontrollere hva som fungerte godt, og hva som hadde mangler. Vi presenterer først utvalget av informanter som har bidratt i denne studien, før vi går gjennom hvordan vi utførte forskningen.

3.2.2.1 Utvalg

Formålet for oppgaven styrer utvelgelsen av informanter, og det som vektlegges i utvelgelsen er det som best kan besvare våre forskningsmål (Johannessen et al., 2020). Vi skal også gjøre to forskjellige utvalg, hvor det stilles forskjellige krav og kriterier.

Utvalg av informanter ved syklus 1: her har vi bare noen enkle kriterier om hvilke egenskaper informantene vi søker bør inneha. Til syklus 1, ved første gjennomgang av undervisningsopplegget burde vi ha en informant med grunnleggende kjennskap til programmering, microbit og didaktikk. Vi tenkte her å ta i bruk veilederne våre, som er lærere på NTNU. Samtidig ønsket vi en faglig forståelse, slik at vi fikk en produktiv gjennomgang av undervisningsopplegget med hensyn til eventuelle faglige mangler.

Ved siste gjennomgang trengte det mer spesifikk lærerekspertise, hvor informanten skal vurdere egen økt og reflektere rundt opplegget. Siden dette krevde mer fra informanten, måtte vi ha større kriterier til nettopp dette ved 2. syklus. Vi kom dermed i kontakt med en

erfaren lærer, med gode relasjoner til klassen, ingen kompetanse i programmering, som var villig til å teste ut læringsressursen vi hadde laget.

3.2.2.2 Syklus 1

Første test av undervisningsopplegget, lærerveiledningen (Vedlegg 1) og oppgaveheftet (Vedlegg 2) kommer i form av kritisk gjennomgang fra fagfolk. I vårt tilfelle var det veilederne våre. For å unngå at informanten i *Syklus 2* gjennomfører et opplegg med faglige mangler, gjennomførte vi i *Syklus 1* en kontrollsjekk. Ved bruk av en åpen dialog ønsket vi å få tilbakemeldinger som ga oss konkrete ting som måtte endres og idéer til hvordan vi kunne utvikle undervisningsopplegget ytterligere.

3.2.2.3 Mellom syklusene

Etter tilbakemelding fra veiledere på produktet ble det gjort flere modifiseringer. Vi hadde også en uformell samtale med informanten for å få og gi noe praktisk informasjon. Deretter leverte vi lærerveiledningen og oppgaveheftet til informanten åtte dager før gjennomføringen.

3.2.2.4 Syklus 2

I denne syklusen testet vi ut det reviderte undervisningsopplegg, dette overlot vi til informanten vi hadde plukket ut til å gjennomføre opplegget. For at forskningen skulle bli valid måtte vi fjerne forstyrrende elementer, som at vi selv skulle undervise en klasse uten relasjoner til den. I gjennomføringen var planen at vi skulle være ikke-deltakende observatører, som observerte lærerens valg og bortvalg og elevenes opplevelse av undervisningsøkten. Dette kommer vi tilbake til i delkapittelet hvor vi diskuterer forskningens validitet (kapittel 3.4.2). Observasjonene som ble gjort ble tatt med inn intervjuet vi gjorde i etterkant av utprøvingen. Intervjuet ble brukt til å få svar på informantens opplevelser av opplegget, årsaker til valg og bortvalg, oppfatninger knyttet til læringsutbytte hos elevene, mm..

3.2.2.4.1 Datainnsamling

Datainnsamlingen ble gjort gjennom observasjoner og supplert med et semi-strukturert intervju med informant i etterkant av undervisningsopplegget. Intervjuet ble transkribert og de mest sentrale funnene kommer med i resultat og diskusjonskapittelet. For å kunne observere på en mest mulig effektiv måte valgte vi å forholde oss til observasjonstypen Bjørndal (2011) beskriver som lav grad av deltakelse, men høy grad av åpenhet. Alle i rommet skal være klar over vår rolle, og hvorfor vi er der. Dette ga de involverte muligheten til å trekke seg fra aktiviteten, samtidig som vi får freden som kreves for å notere det vi finner av interessant data. Selv om vi hadde planlagt en observasjon med lav deltakelse, var dette umulig å gjennomføre, da læreren manglet tilstrekkelig kunnskap om læringsressursen til at oppgaven ble gjennomførbare.

3.2.2.4.2 Observasjon

Vi har valgt å gjennomføre observasjoner av én lærer og fire elevgrupper som utfører et undervisningsopplegg vi har laget. Vi har en deduktiv inngang på observasjon hvor vi setter observasjonene inn i en kontekst med tidligere lest teori og forskning. Vi har en strukturert observasjon, hvor vi velger ut noe å fokusere observasjonene våre på. Det vi skal fokusere observasjonene våre på er læreren og hvordan hen gjør valg i undervisningen med tanke på undervisningsopplegget. Både valg av bruk og bortvalg skal observeres, og tas opp i intervju for å skape en klarhet til lærerens gjennomgang. Vi skal dermed kanalisere grunner til bortvalg, som tid eller forberedelser, at opplegget ikke er tilpasset elevgruppen, og eventuelle andre årsaker.

Bjørndal (2017) forklarte at når observasjoner persiperes vil denne informasjonen gå tapt i brøkdelen av et sekund, derfor noterte vi underveis i undervisningen, med fokus på hvordan lærer tok i bruk undervisningsopplegget, og om læreren viker fra det tenkte opplegget. Hovedvekten av observasjonene skal være mot læreren og hvordan hen tar i bruk undervisningsopplegget. Vi observerte samtidig elevene får å få et innblikk i deres opplevelse av undervisningen. Med det var ønske å få et inntrykk av hvordan de løste oppgavene og i hvilken grad de var motiverte til å gjøre de.

Vi gjennomførte det Christoffersen & Johannesen (2012) forklarer som en strukturert observasjon, ved å lage et skjema (Vedlegg 4) hvor vi noterte relevante observasjoner, med en klar plan på hva vi er interesserte i å observere, noe som læreren også skal være klar over. Skjemaet skulle gjøre det enkle for oss å observere, og at vi skulle få med oss det

som var relevant for oppgaven. Vi delte observasjonsskjemaet i to, en del for observasjon av elever, og en for observasjon av lærer. Og videre underkategorier som "tilpasninger" "didaktiske grep" "bortvalg".

3.2.2.4.3 Semi-strukturert intervju

Intervju brukes ofte for å få øye på detaljer som observatøren selv kan overse (Bjørndal, 2011). Intervju vil være essensielt for utviklingen i og etter *Syklus 2*, da vi var avhengig av innblikk i informantens opplevelse av opplegget. Vi valgte å ta i bruk et semistrukturert intervju, der vi hadde noen forberedte spørsmål gjennom en intervjuguide (Vedlegg 3) og oppfølgingsspørsmål ble valgt ut ifra det deltakerne forteller. Dette skaper en mer naturlig dynamikk, da det føles mer som en samtale og ikke et avhør (Bjørndal, 2011), samtidig som vi får mulighet til å få svar på spørsmål vi ikke visste vi hadde før intervjuet begynte. Vi endte opp med et 45 minutter langt intervju.

3.2.2.4.4 Rammer for utprøvingen

Rammene for utprøvingen skal ta for seg lærerens og elevenes erfaringer med programmering. Da dette spiller en viktig rolle for hvordan utprøvingen utspilte seg, og at dette påvirket datainnsamlingen.

3.2.2.4.4.1 Lærerens erfaringer/bakgrunn

Kim forklarte innledningsvis i intervjuet at selv hadde hen lite kompetanse innenfor programmering, men at det har blitt introdusert programmering på fellestid i tilknytning til den nye lærerplanen. Kim følte dette var noe som ble tvunget på og husket heller ikke hvilket programmeringsverktøy som ble tatt i bruk i leksjonene gjennom fellestiden.

Kim forklarte i intervjuet at det har vært et krav i forbindelse med ny læreplan med leksjoner i programmering, holdt av andre lærere og ikke eksterne. Informanten virket ikke å ha noen positiv opplevelse med implementering av dette, da hen forklarer at de ble "tvingt" til å gå gjennom leksjoner i programmering av ledelsen ved skolen. Teorien til Misfeldt et al., (2019) bygger på dette utsagnet da det ble forklart at det var mange lærere som ikke klarte å se en sammenheng mellom programmering og ulike fag. Dette sammenfaller med det Kim fortalte at lærere ble introdusert til programmering, men så ikke sammenhengen med det og fag, og dermed opplevde dette noe meningsløst. Dette var også

noe vi hadde antatt gjennom praksis og tidligere forskning, noe vi har valgt å bygge delmål 2 rundt.

3.2.2.4.4.2 Elevenes erfaringer med programmering

Elevene vi observerte gikk i en 6. klasse på en skole i Trondheim. Klassen var bestående av rundt 60 elever fordelt på 4 klasser denne dagen. De fleste elevene hadde ikke programmert med micro:bit, men vært gjennom programmering i matematikk og var kjente med programmeringsverktøyet trinket, som er tekstprogrammering.

Kim forteller videre at elevene har gjort enkelte programmeringsoppgaver i matte gjennom Salaby og Campus, som er blokkprogrammering. Hen forteller at elevene syntes dette var gøy, og at dette var noe de trivdes å jobbe med. Elevene hadde dermed erfaringer fra blokkprogrammering, men ikke fysisk programmering som micro:bit, dette var nytt for elevene. Dette var noe vi fikk kartlagt da vi hadde et møte med informanten i forkant av opplegget, og dermed endret vi lærerveiledning til en grundigere introduksjon av de grunnleggende elementene ved oppkobling av micro:bit.

Noen elever hadde også erfaringer fra programmering utenfor skolen, gjennom egne interesser og/eller science camp, som er en sommerskole som blant annet introduserer programmering og koding. Flere av elevene hadde da mer kompetanse enn læreren i programmering, noe Vinnervik (2023) fortalte var ganske vanlig, og som stiller andre krav til lærerens profesjonelle tilstrekkelighet. Det stilles et større krav til å skripte kodene mer i detalj og kjennskap til koderelaterte problemer som kunne oppstå. Her manglet informanten kompetanse gjennom dårlige forberedelser grunnet tid, og vi som i utgangspunktet skulle være passive observatører måtte gripe inn mer delaktig i undervisningen for å hjelpe elevene videre.

3.2.3 Tredje fase: Retrospektive analyser

Den retrospektive analysen skal danne grunnlaget for endringer basert på funnene etter syklus 2 og skal invitere til videre forskning. Denne fasen baserer seg på teorigrunnlaget, observasjonene og intervjuet, og skal forklare hvilke endringer som kreves for at produktet skal utvikles og hvordan målene skal oppnås. I vår studie gjennomfører vi ikke endringene, og *vedleggene* (Vedlegg 1 & 2) er dermed de samme vi sendte informanten. Likevel diskuterer vi hvilke endringer vi ville gjort om omfanget til oppgaven hadde tillatt det. I denne fasen er det bevisene som bestemmer valgene, og det er viktig for oppgavens troverdighet at bevisføringen samsvarer med endringene vi ser for oss (Cobb et al., 2003).

Analysen baserer seg på de funnene vi anså som mest interessant. Om det er fordi de samsvarte med observasjonene våre eller om de var opplevelser som informanten hadde som vi ikke delte. Vi veide funnene opp mot teoriene vi har brukt og så om det var noe vi måtte endre basert på dette. Utdragene fra intervjuet som blir brukt i kapittel 4 er det vi har ansett som mest påfallende og det er derfor de vi har diskutert.

3.3 Behandling av data

Datamaterialet vi samler inn er egne notater knyttet til observasjoner, lydopptak av oss og informant som gjennomfører et intervju og transkripsjon av dette intervjuet. Det er ingen personopplysninger i noen av disse dataene, men likevel er det lovfestede og etiske regler vi må forholde oss til når vi samler inn og behandler datamateriale knyttet til forskning (NESH, 2021; Sikt, 2023.a).

3.3.1 Forskningsetikk

Som en forskningsetisk hovedregel skal det gis informasjon og innhentes samtykke fra alle som deltar i forskning (NESH, 2021; Sikt, 2023.a). Dette gjelder også om det ikke innhentes personopplysninger, står det i NESH (2021) hvor det er vårt ansvar når vi skal innhente datamateriale, å informere alle som deltar i undersøkelsen.

Når det kommer til personvern er det viktig at også alle involverte får informasjonen som trengs slik at de kan skape seg en forståelse av forskningsfeltet, deres følger av å delta i undersøkelsen eller forskningen, og formålet med oppgaven (Sikt, 2023.a). Vi informerte elevene som deltok i utprøvingen og deres foresatte i klassen om opplegget og hva deres rolle ville være, og gjorde det klart at det var frivillig å delta (Vedlegg 6). Elever som skal delta i en undersøkelse har krav til å vite hva som skal undersøkes og hvordan dette skal gjøres, enten direkte til dem eller informasjon til foreldre (Sikt, 2023.a).

3.3.2 Meldepliktig til Sikt

Da anonyme observasjoner hvor ingen personopplysning ikke er meldepliktig hadde vi i utgangspunktet ingen årsak til å søke om tillatelse til å gjennomføre forskningen for andre enn, elevene, foresatte, informantene og ledelsen på skolen (Sikt, 2023.a). Barn er likevel en sårbar gruppe som krever ekstra støtte når det kommer til personvern (Sikt, 2023.a). Derfor tenkte vi at vi heller søker en gang for mye, enn en gang for lite. Sikt (2023.b)

ønsker åpenhet fremfor alt når det kommer til hensyn av personvern, derfor la vi frem planen for forskningen i så god detalj som vi evnet i søkeprosessen. Og fikk godkjenning til å gjennomføre alt vi hadde planlagt. Vi fikk også tillatelse fra samtlige av de involverte i prosjektet. Vedlagt har vi samtykkeskjema til både informant og elevene (Vedlegg 6 & 7).

Av de samme hensyn som ovenfor er oppbevaring av datamateriale noe som krever omtanke. Ettersom vi har det tilgjengelig og det gir oss en trygghet jf. lovverket, brukte vi et kryptert nettskjema for å oppbevare intervjuet, og informerte om at alt av innsamlet data skal slettes etter sensur er mottatt. På denne måten har vi tillatelse til oppbevaring av datamateriale gjennom hele forskningsprosessen (NTNU, 2023).

3.4 Metodens pålitelighet og validitet

I denne delen skal vi belyse metodens pålitelighet og validitet, som står sentralt når en skal drøfte over metodens troverdighet.

3.4.1 Pålitelighet

Pålitelighet eller reliabilitet som det ofte oversettes til, beskriver hvor sannsynlig det er at en får samme svar eller funn om noen andre gjennomfører det samme (Kvale & Brinkmann, 2015). Den ultimate testen på reliabilitet ble sett på som "test-retest" hvor en skal gjenta en studie på et annet tidspunkt å se om resultatet endrer seg (Postholm & Jacobsen, 2018).

Vi gjennomførte utprøvingene våre 4 ganger, med 4 observasjoner, som støtter opp under oppgavens reliabilitet, gjennom re-testing. Vi noterte observasjoner i et observasjonsskjema som var forhåndsskrevet i kategorier. Dette styrker også oppgavens pålitelighet, selv om det som ble observert varierer noe fra klasse til klasse.

3.4.2 Validitet

Metodens validitet eller gyldighet deles inn to underkategorier, indre og ytre gyldighet, indre gyldighet omhandler om resultatene er gyldige for de vi har studert, eller om datainnsamlingen har målt det vi sier eller tror at vi måler (Postholm & Jacobsen, 2018). Ytre gyldighet omtales ofte som overførbarhet og betegner i hvor stor grad resultatene representere denne spesielle konteksten, eller mer generalisert. I hvor stor grad er

resultatene fra denne ene skolens overførbarhet til andre skoler (Postholm & Jacobsen, 2018).

Vi har fortolket observasjonene og diskuterte dette med hverandre, men også i et seminar i etterkant av observasjonene. Senere brukte vi observasjonene våre i et semi-strukturert intervju med informanten, dette styrker den indre gyldigheten til metoden. Men vi klarte ikke holde oss like utenfor bilde som vi hadde planlagt på forhånd, og var mer deltakende observatører, som påvirker resultatene i stor del. Her kan man ikke si med sikkerhet om informanten hadde kommet i mål de første gjennomføringene, og vi fikk ikke observert her om det ble gjort klare bortvalg fra læringsressursen.

Det er store forskjeller fra skole til skole, og lærer til lærer, så det kan være utfordrende å beskrive resultatet som noe som er spesielt ved denne skolen, eller trinnet i motsetning til andre kontekster. Dette er en svakhet ved denne metodens ytre gyldighet, overførbarheten da vi ikke har testet det i andre kontekster.

4 Resultat og diskusjon

I dette kapitlet skal vi diskutere målene som gjelder for denne oppgaven opp mot ressursen og utprøvingen. Overordnet handler denne oppgaven om å

Bidra til at lærere uten programmeringskompetanse kan knytte sammen undervisning i fagtemaet vei, fart og tid med programmering.

Vi har delt kapitlet i de to delmålene som skal hjelpe oss med å svare til det overordnede målet.

Delmål 1: *Utvikle et oppgavehefte og en lærerveiledning som egner seg for lærere som mangler kompetanse innen programmering.*

Delmål 2: *Utvikle et undervisningsopplegg, lærerveiledning og oppgavehefte som knytter fagtemaet vei, fart, tid med programmeringsundervisning*

Disse er igjen oppdelt basert på de ulike fasene i pedagogisk designforskning, med resultater, forankret i teori og eventuelle endringer.

Først, med henvisning til *Vedlegg 1 og 2*, gir vi et innblikk i hvordan produktet faktisk er, før vi forklarer hvordan vi i forberedelses fasen gjorde valg av innhold basert på teorien vi har brukt. Deretter viser vi til resultater fra utprøvingen gjennom våre observasjoner og fra intervjuet vi gjennomførte etter utprøvingen. Hvert delkapittel avsluttes med en retrospektiv analyse av aspektet, hvor vi diskuterer hva vi skulle gjort annerledes, hva som fungerte. Hva kan generaliseres og eventuelt hvorfor det ikke kan generaliseres, hvordan vi ser for oss veien videre for en forsker som vil fortsette med utprøving av dette produktet kunne vært. Avslutningsvis diskuterer vi det overordnede målet vårt i en generell retrospektiv analyse av produktet.

4.1 Produktet

Når vi utformet lærerveiledningen hadde vi teorien til Carroll & Ward (2016) frisk i minne, og valgte et undervisningsopplegg hvor elevene skulle være delaktige og involverte. Dette for å unngå at elevene blir passive observatører, som Carroll & Ward (2016) begrunnet som tegn på dårlig undervisning.

Produktet (Vedlegg 1 & 2) har vi valgt beholde i sin ordinære form, fremfor å gjøre endringene vi senere diskuterer som hensiktsmessige. Dermed er, som nevnt, vedleggene like de informanten fikk utdelt før gjennomføringen.

Lærerveiledningen (Vedlegg 1), skulle forklare alt læreren måtte vite og alt som skulle gjennomføres. Den inneholder informasjon om verktøyet, og hvordan det skal anvendes, noe teoretisk informasjon, forklaring av innholdets relevans opp mot kompetansemålene og gjennomgang av tenkt undervisning. I tillegg forklares elevoppgavene og tilknyttede tilpasninger. Tilpasningene er laget i makecode, og gjør det nødvendig for å ha veiledningen digitalt.

Opgaveheftet (Vedlegg 2) er dokumentet elevene skulle forholde seg til under gjennomføringen. Denne er man også avhengig av å ha digitalt, da tilpasningene har en sentral rolle. Utover det forklarer heftet gangen i timen i elevperspektiv. Alle oppgavene de skal forholde seg til er beskrevet og knyttet opp mot de relevante tilpasningene vi tilbyr. Det er laget rom for elevene til å notere svarene de får samtidig som de også oppfordres til å diskutere med partner.

4.2 Delmål 1: Utvikle et oppgavehefte og en lærerveiledning som egner seg for lærere som mangler kompetanse innen programmering

Under dette målet skal vi utvikle en læringsveiledning og et undervisningsopplegg for lærere med lav programmeringskompetanse. Vi skal presentere hvordan vi gjennom teori har utviklet en læringsveiledning forbeholdt lærere med manglende programmeringskompetanse.

4.2.1. Forberedelse: Lærerveiledningen

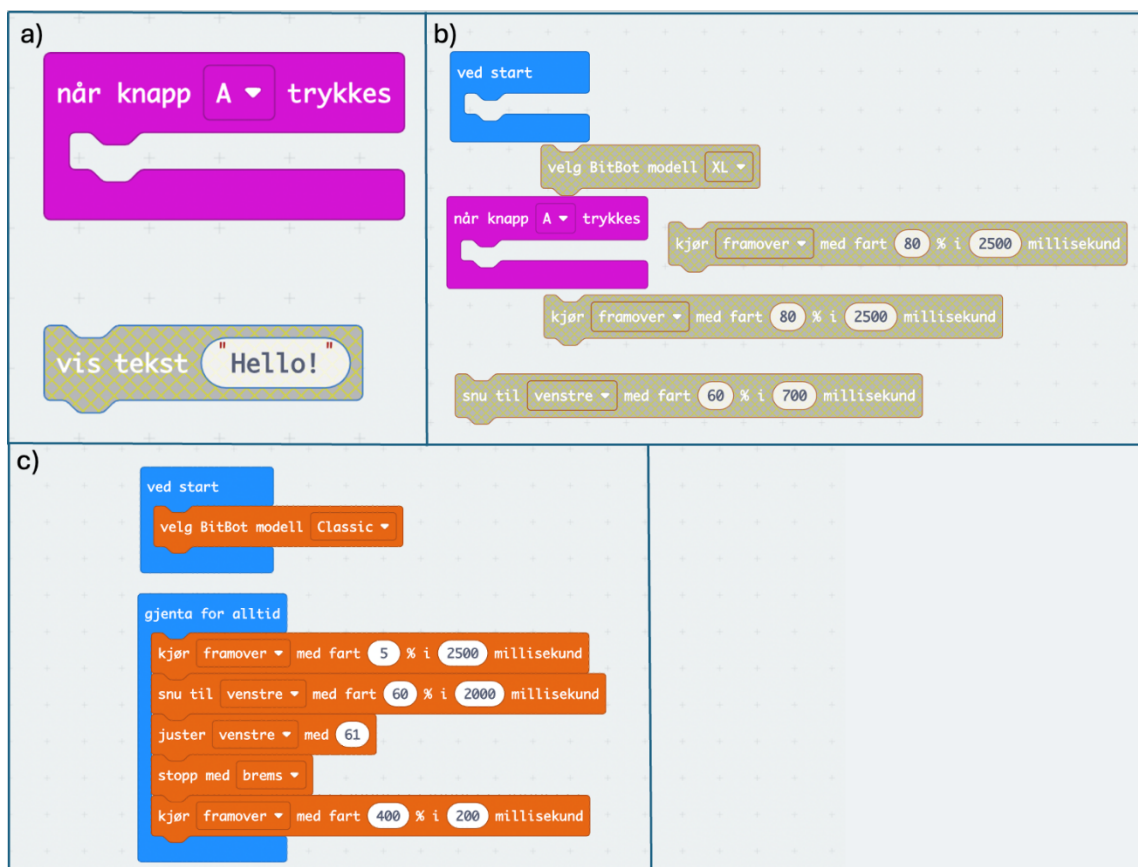
Valg av programmeringsverktøy er et viktig første steg i å lage et undervisningsopplegg for læreren med begrenset programmeringskompetanse (Makki et al., 2018). Til å begynne med så vi i all hovedsak på koblede aktiviteter innen programmering. Vi konkluderte med at elevene hadde kompetanse forbi utkoblede aktiviteter som nevnes som introduksjon til programmering (Bell & Vahrenhold, 2018), siden elevene gikk i 6. klasse, og hadde programmert i matematikk tidligere. Vi så nærmere på koblede aktiviteter og simuleringsverktøyet *Scratch*, som nesten er uavhengig av skolens ressurser og mer tilgjengelig for lærere. Scratch manglet aspektet ved programmering av fysiske

gjenstander, som vi selv mener har en positiv effekt på den utforskende læringen til elevene og skaper engasjement, og valget falt dermed på micro:bit.

Ved bruk av *makecode*, som vår studie bygger seg på, er man avhengig av at skolen er utstyrt med et rikelig *micro:bit*-lager. Spesielt for vårt opplegg er at man også trenger like mange *bit:bots*, en *micro:bit* og en *Bit:bot* hver gruppe. *Grunnet superbites nasjonale satsning er det ofte rikelig med micro:bit og bit:bot på skolene i Norge (Sunde & Øen, 2022).* Vi valgte å fokusere på *makecode* da dette tilbyr en mer aktiv læring for elevene. Elevene møter programmeringsverktøy som *scratch* blant annet i matematikk, men Haraldsrud et al (2020) mener at verktøyet som egner seg best for å oppnå kompetansemålene i naturfag er *micro:bit*.

Vi fant et undervisningsopplegg utviklet av *Super:bit*, og gjorde tilpasninger til opplegget for at det skulle passe alle elever. Vi la til stillasbyggende oppgaver som *Bjursten et al., (2022)* nevner som en pedagogisk strategi i programmering, og vi forenklet elevoppgavene som en del av progresjonen av programmeringen, og gjorde oppgavene vanskeligere med å øke mengden koder elevene trengte.

En av måtene vi skiller oss fra ressursene *Super:bit* allerede har gjort tilgjengelig er at vi har fokusert på koblingen mellom programmering og fag. Som det skrives i *Benton et al. (2016)*, er ett av punktene for gunstig programmeringsundervisning å lage en bro mellom fagene, og med det se nytteverdien av programmeringen. En annen endring vi gjorde fra *Super:bit* var å legge til tilpasningen elevene kunne ta i bruk dersom de stod fast eller oppgavene ble for utfordrende. I figur 1 kan man se hvordan tilpasningene ser ut. I figur 1a ser man en skjermdump fra *superbit (u.å.c)*, dette er et eksempel på *parsons puzzle* som vi har valgt å kalle *puslespill*. Vi har også med denne type tilpasning i alle våre oppgaver (figur 1b). Der vi skiller oss ut med tanke på tilpasningene er blant annet *debugging* (figur 1c). Her har vi, istedenfor å gi elevene hjelp, gitt de en utfordring. De må her finne ut hvilke klosser som faktisk skal være med, og hvilke som må fjernes fra skriptet. De er også nødt til å plote inn verdier som gir mening for oppgaven, da de som er skrevet inn i klossene nå vil gi feilkoder.



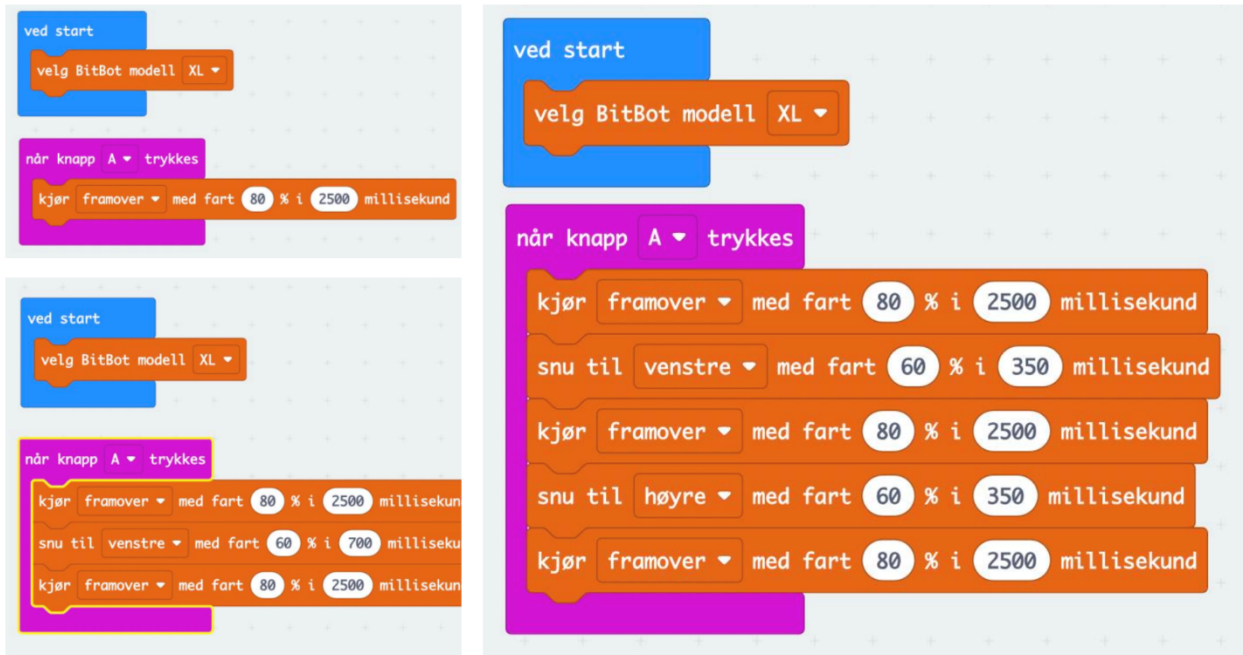
Figur 1: a) Skjermdump fra et undervisningsopplegg fra superbit (u.å.c), b) og c) skjermdump av eksempler på våre tilpasninger. (Oppgave 2 i oppgaveheftet (Vedlegg 2))

Basert på kompetansemålene for naturfag etter 7. trinn (Kunnskapsdepartementet, 2020) og litteratur på programmeringsundervisning designet vi et undervisningsopplegg som skulle sørge for måloppnåelse både innenfor programmering og naturfag. I tillegg ønsket vi at opplegget skulle inneholde kjerneelementet for "naturvitenskapelige praksiser og tenkemåter", som forteller oss at elevene skal oppleve naturfaget som praktisk og utforskende (Kunnskapsdepartementet, 2019.a). I tillegg stiller lærerplanen krav til at opplegg i programmering skal inneholde elementer fra andre fag (kunnskapsdepartementet, 2020).

Gjennom bruk av rammeverket presentert av Benton et al. (2016) hvor elevene skulle utforske, forklare, se for seg, utdype seg i andres strategier og å lage koblinger mellom programmering og naturfaglige temaer (Benton et al., 2016), sørget vi for å opprettholde naturfagets egenart som et involverende, delaktig og utforskende fag (Kunnskapsdepartementet, 2019.a; Carroll & Ward, 2016). I lærerveiledningen kommer

dette frem i at hver oppgave i elevheftet skulle besvares i par og diskuteres i fellesskap i etterkant. Oppgavene elevene skulle svare på, omhandlet vei, fart og tid, og ikke programmering. Derimot skulle programmeringen oppleves som et verktøy, slik valgte vi å bygge programmering og naturfag sammen. Teorien om parprogrammering forteller oss om nytten av fellesskapet i utprøvningsfasen (Nagappan et al., 2003). Dermed bygger opplegget i stor grad på samarbeid mellom elevene. Samtidig som dette i seg selv gir bedre kompetanseutvikling hos elevene, er dette en strategi for lærere med liten eller ingen programmeringskompetanse. Ved å oppfordre til samarbeid og idéutveksling vil byrden på veiledning fra lærere minke, og man får bedre tid til hver gruppe (Haraldsrud, 2020; Brown & Wilson, 2018). Dette underbygger Newhall et al. (2014), da de mente at elever som jobbet i par, hadde bedre forutsetninger til å skape programmeringsforståelse, sammenlignet med elever som jobbet alene. Dette har vi tatt i betraktning når vi har satt opp oppgavene slik at elevene skal kunne jobbe i par. Dette er et av mange valg en lærer må gjøre i forkant av en undervisningstime, og går lettere med godt kjennskap til elevgruppen.

For å kunne bidra med et undervisningsopplegg som hvilken som helst lærer kunne ha nytte av måtte vi sørge for at ingenting var for dårlig beskrevet. Om det var noe som virket uklart måtte vi påse at det ble forenklet. Vi oppdaget tidlig at det kan bety at lærerveiledningen blir et omfattende dokument med lange forklaringer og mange bilder og eksempler. Eksempelvis ønsket vi å gi læreren gode nok forutsetninger med tanke på forståelse av programmet. Det innebærer skjermdumper av eksempelkoder til hver oppgave (Figur 2), en omfattende forklaring av verktøyet og forklaringer til de ferdiglagde tilpasningene.



Figur 2: Skjermdumper fra lærerveiledningen (Vedlegg 1), bildene viser eksempler på skripter som besvarer oppgave 2 i oppgaveheftet (Vedlegg 2).

For å nå frem til læreren som ikke er kjent med makecode la vi inn en veiledning til programmet. Med bilder og forklaringer ønsket vi å geleide læreren frem til en forståelse som kunne videreføres til elevene.

Vår egen testing av verktøyet viste tegn til at vi selv mangler programmeringskompetanse, og vi brukte dermed lang tid på å finne ut hvordan maskin- og mykvaren fungerte sammen. Det å være uerfarne programmere selv, er ikke utelukkende noe negativt for denne forskningen. Grunnet vår manglende kompetanse, kan det være enklere å legge opp til et undervisningsopplegg som simplifiserer programmeringsundervisningen til en lærer uten programmeringskompetanse. Dette ble dermed gunstig for vår forskning, siden vi jobbet grundig med ressursen og kunne assosiere oss med en uerfaren programmeringslærer. Oppgavene vi lagde var basert på egen utprøving, og egen oppfattelse av i hvilken grad elevene, og ikke minst læreren, kunne gjennomføre dette.

4.2.2 Forberedelse: Elevtilpasninger

I lærerveiledningen ønsket vi, i tillegg til å gi kjennskap til verktøyet, å gi læreren et godt utgangspunkt for å ha en innholdsrik økt med godt læringsutbytte. For å gjøre det presenterte vi elementer som kommer fra teori om programmeringsdidaktikk. Qian og Lehman (2017) skriver at det er viktig for læreren å kartlegge elevenes misoppfatninger, og

deretter jobbe strategisk for å bli kvitt disse, men når de samtidig skriver at lærere med manglende kompetanse kan bidra til å skape misoppfatninger, må man legge til rette for å unngå dette. For å støtte læreren gjennom undervisningen, og for å unngå overføring av misoppfatninger, er det lagt inn flere tilpasninger til hver av elevoppgavene. Disse fungerer som en støttestruktur for både lærer og elev og er alltid tilgjengelig hvis en av partene sliter med å komme seg videre. Utviklingen av disse var basert på eksempelkode vi selv laget og la ved hver oppgave i lærerveiledningen (Vedlegg 1), og tilbyr varierende vanskelighetsgrad for å få et ønsket resultat. I utviklingsfasen la vi stor vekt på disse tilpasningene. Vi plasserte de strategisk tidlig i lærerveiledningen og forklarte de grundig for å underbygge hvor viktig vi syntes de var for gjennomføringen. De fikk også en tydelig plassering i oppgaveheftet (Vedlegg 2).

Utfordringen med denne støttestrukturen er mangelen på frihet. Å veilede elevene til et resultat på denne måten strider mot 5e rammeverket til Benton et al. (2016). Mangelen på frihet gjør det mindre naturlig å *utforske* programvaren og å *forklare* egne idéer. Dermed, selv om vi var fornøyde med innholdet og mulighetene med tilpasningene, oppfordret vi læreren til å presentere disse som alternativer til gjennomføring. Slik blir utforskningen det første elevene forsøker. De andre aspektene av rammeverket ble tatt hånd om med blant annet oppsummerende samtaler. Ved å dele strategier ville elevene få innsikt i andres idéer, noe som støttes av samarbeidsteori fra blant annet Sentance et al. (2019) og Nagappan (2003). Vi ønsket å skape et klasserom hvor parene sammen skapte hypoteser og utforsket disse, før de delte funnene sine med resten av gruppen. Det siste punktet i rammeverket, *BridgE*, er også å finne i lærerveiledning. Den kommer vi tilbake til i *mål 2*.

Vi valgte tilpasninger til hver enkelt oppgavene for en forenkling både for læreren og for elevene. Dette valgte vi på grunnlag av at hverken lærer eller elever hadde kjennskap til micro:bit, og dermed kunne bruke dette som stillasbygging til oppgavene de skulle igjennom. Tilpasningene var "ferdig kode", "halvferdig kode", "forutsi kode" og "debugging". Vi la ved disse 4 tilpasningene til hver av oppgavene elevene fikk utdelt (Figur 3). Planen var at elevene skulle løse programmeringen ved å ta i bruk en eller flere av disse tilpasningene om de sto fast.

Hvis dere trenger hjelp til oppgave 2:

Forutsi kode

Her får dere en kode som vil kjøre en strekning, snu, og kjøre tilbake.

Hvor langt tror dere den kjører før den snur?

Hvordan kan dere få den til å kjøre dobbelt så langt?

https://makecode.microbit.org/_7tRDxqbfICLx

Ferdig program

Klossene er klare, men bilen mangler tall som forteller hvor fort og langt den skal kjøre - Det må dere skrive inn. Mål opp hvor langt dere vil at bilen skal kjøre, og så finne ut hvilke tall dere må sette i koden.

https://makecode.microbit.org/_8qR5kh3sPbap

Puslespill

Her får dere klossene dere trenger for å gjøre oppgaven, men dere må plassere dem riktig.

Obsobs! Her kan det finnes koder som ikke skal tas i bruk i programmeringen.

https://makecode.microbit.org/_VKf14D16297s

Halvferdig program

Her har dere allerede starten på koden. Dere må fullføre.

https://makecode.microbit.org/_1VyEoeHhiC2X

Debugging

Her er det mye feil. Ta bort det som ikke trenger å være i koden og endre på tallene slik at det fungerer som dere vil.

https://makecode.microbit.org/_MHgEcvhDWUIR

Figur 3: Skjermdump av elevtilpasningene for oppgave 2 oppgaveheftet (Vedlegg 2).

4.2.3 Utprøving: Brukervennlighet

I denne og de påfølgende delene forklarer vi resultatene fra utprøving og intervjuet gjort i etterkant. Herunder presenterer vi funn og diskuterer dette sammen, så dette blir en sammenslåing av fase 2 og 3.

Vi stilte spørsmål om informantens tanker om opplegget og om han var fornøyd med innholdet i lærerveiledningen. Lærerveiledningen skulle være detaljert, og det skulle ikke være noe som skulle være uklart for brukeren av den. Dermed ble det laget et omfattende dokument som skulle kunne veilede læreren gjennom hele gjennomføringen. Dette sendte vi informanten en uke i forveien, og i intervjuet i etterkant forklarte hen:

For skulle jeg gjort det fra punkt til prikke, så tror jeg det kunne ha stresset meg veldig. Det tror jeg ikke jeg helt hadde fått til. Fordi jeg ikke kunne det godt nok. Opplegget, det var bra det.

Vår informant var fornøyd med innholdet i veiledningen, men samtidig forklarte at hen måtte gjøre det på sin egen måte, etter egne begrensninger. Som vi tolker dette utdraget av intervjuet, er det en blanding av mangel på kompetanse i programmering, programmeringsdidaktikk og kunnskap om lærerveiledningen. Kim innrømte å tildele for liten tid til forberedelse. Basert på observasjonene og de uformelle samtalene før og mellom gjennomføringene kom det frem at hen gjorde bortvalg fra opplegget basert på usikkerhet knyttet til veiledningen. Vi kan ikke forvente at en informant lærer seg alt vi har tildelt hen, men en lærer med manglende kompetanse på fagfeltet gjør nødvendigvis ikke hensiktsmessige bortvalg. Dermed må man i en eventuell ny gjennomføring påse at informanten tar seg god tid til å sette seg inn i det vi har planlagt. Vi ble aldri kontaktet i informantens forberedelsestid, og dermed skulle man tro hen hadde forstått innholdet godt nok. Likevel var det aspekter ved veiledningen vi anså som essensielle som ble for dårlig gjennomgått. Et eksempel på dette var mangelfull gjennomgang av tilpasningene. Dette kommer vi tilbake til.

4.2.4 Utprøving: Lærerveiledningens design

Som nevnt ønsket vi å lage en lærerveiledning for lærere med manglende kompetanse innenfor programmering. Dette innebar å presentere programvaren og verktøyet så godt som mulig. På bakgrunn av detaljerte forklaringer av nettsiden makecode.org i veiledningen klarte informanten å formidle denne til elevene på en god måte. Derimot var det mer utfordrende å ta i bruk verktøyet. Observasjonene vi gjorde i klasserommet forteller oss at dokumentet læreren mottok fra oss var mangelfull på dette punktet. Hen slet med å sette micro:biten riktig vei, og trengte muntlig veiledning for å finne frem til på-knappen på bit:boten.

Basert på disse funnene ser vi at vi manglet bilder på dette punktet i veiledningen. Å få verktøyet presentert bare med ord viste seg å ikke være tilstrekkelig. Derfor ville vi, om vi skulle gjort dette igjen, lagt ved bilder der vi forteller hvordan læreren bruker maskinvaren. I tillegg burde vi tilstrebe å gi brukeren av lærerveiledningen en bit:bot og micro:bit samtidig som hen får dokumentet. Vi kunne gjort forberedelsestiden vesentlig enklere om vi hadde gjort disse endringene tidligere. Dette er noe som eventuelt kunne vært forenklet ved en representasjon av programmering gjennom video, med tale og bilder. Som Mayers multimodale teori omhandler, gir ord og bilde bedre læring enn bilde alene (Digital learning institute, u.å.). En kan også stoppe og spole tilbake i filmen dersom noe er uklart for elevene (Swendsen, 2014), og elevene kunne lært dette i ønsket tempo.

4.2.5 Utprøving: Motivasjon vs. læringsutbytte

I lærerveiledningen vektla vi oppsummerende diskusjoner tungt, noe informantene valgte å ikke gjennomføre. Vi lagde et opplegg hvor elevene fikk mulighet til å dele strategier i oppsummerende diskusjoner etter hver oppgave. Informanten begrunnet bortvalget av oppsummerende diskusjoner på dette viset:

Og, på en måte, den responsen som vi så hos elevgruppen, tyder på at det her gikk veldig bra. For alle gruppene, så å si, var jo veldig motivert og påkoblet. De gjorde det de skulle, og de ønsket nesten ikke å avslutte økta.

Basert på dette utsagnet kan man tenke seg at Kim gjorde bortvalg for å unngå endringer i klasseromsmiljøet og elevenes iver og respons til oppgavene. Når elevene ble så motiverte av å få store frihetsgrader, virket det utfordrende å stramme dette inn igjen. Vi hadde planlagt oppsummerende samtaler om programmeringsstrategier etter hver oppgave. Disse ble nedprioritert på bakgrunn av et ønske om å opprettholde elevenes motivasjon. Her ble motivasjon prioritert over læringsutbytte som vi har tolket det.

Siden vi har sammenslått fase 2 og 3 går vi nå over på en diskusjon av bortvalget informantene gjør over. Vi argumenterer ved hjelp av Staberg et al., (2020) hvorfor vi mener bortvalget av de reflekterende samtalene på slutten av hver gjennomføring var negativt, og at økten ikke nødvendigvis er vellykket fordi elevene er engasjerte. Vi skal nå se nærmere på mulige endring basert på det vi har nevnt tidligere.

At lærere i utforskende undervisning bedømmer hvordan økten gikk etter nivået av engasjement blant elevene er noe som ofte blir gjort ifølge Staberg et al., (2020). Fokuset

på aktiviteten ligger som nevnt tidligere på å gjøre utforskningen heller enn å reflektere over læringsutbytte, og konkludere med at økten var vellykket.

Samtidig som friheten elevene fikk stred imot det planlagte opplegget, ser vi en positiv effekt av det også. Riktignok vet vi ikke hvordan det ville gått om Kim hadde brutt opp øktene med diskusjonsbolker, men det er ingen tvil om at elevene forble motiverte. Dette kommer tydelig frem både i intervjuet og i våre egne observasjoner. Flere av elevene slet med å finne seg i at timen var omme. Elevene ble oppfattet som svært motiverte gjennom hele dagen, og Kim uttrykte underveis i økten at det var flere elever som til daglig sliter med å holde følge, og å gjøre det de skal, bidro i langt større grad enn vanlig.

Programmeringsparene førte til en inkludering av alle elevene i klasserommet, da det ikke ble observert noen som falt utenfor. Elevene hadde utforskertrang, og uttalte en glede av å jobbe med programmering av en fysisk gjenstand, istedenfor simulering på en datamaskin som de har gjort tidligere. Både sterke og svake elever var aktive gjennom hele timen de hadde til disposisjon, som gjerne kunne vært lengre da vi måtte avslutte økten imot elevenes vilje.

For forskningens skyld, om vi skulle gjort dette igjen, er det et tydelig behov for å stramme inn frihetsgradene. Ikke bare til elevene, men også til læreren. Slik som gjennomføringen ble, var det noen av aspektene som vi på forhånd vektla tungt som ikke ble testet ut godt nok. Flere av teoriene vi presenterte tidligere i denne oppgave, og som vi bygde undervisningsopplegget på, forteller om et behov for deling av strategier (Benton et al., 2016; Sentance et al., 2019; Nagappan, 2003). Dette behovet ble ikke tilfredsstilt på tvers av læringspartnere. Parprogrammeringen ble opprettholdt, men de reflekterende samtalene må tydeliggjøres i lærerveiledningen, da dette, ifølge Staberg et al., (2020) også bør vektlegges. For å sørge for at diskusjonene og strategidelingen blir gjennomført, ser vi et behov for å tydeliggjøre disse i lærerveiledningen. Per nå står de som en alminnelig del av det planlagte opplegget, og det er dermed lett å velge bort eller glemme. Legger vi derimot ved en begrunnelse av viktigheten gjennom teori, kan det tenkes at også læreren ser dette som essensielt.

Samtidig som denne type bortvalg kan forklares ved manglende forståelse av viktigheten av aspektene, kan det også forklares ved manglende forberedelse. Opplegget er laget for lærere med manglende programmeringskompetanse og det er derfor en forutsetning at lærer har jobbet seg gjennom lærerveiledning før undervisning. Dermed anbefaler vi å tilpasse veiledningen til å inneholde et estimat for hvor lang tid læreren burde sette av til å forberede seg, samt ta i bruk micro:bit og bit:bot under forberedelsene.

Det kan også diskuteres at lærerveiledningen er for omfattende. Det er kjent at lærere har begrenset med tid, og at drøyt 20 sider med veiledning dermed kan sies å være overdrevent. Det er tenkelig at en lærer som i fremtiden leter etter gode undervisningsopplegg for programmering som integrerer fag overser vårt opplegg, og heller ser videoene til smart:bit. Likevel lager vi et opplegg for læreren som mangler kompetanse innenfor programmering. Dermed mener vi at alle aspektene i lærerveiledningen bør forbli.

Det må også nevnes at tidsbegrensninger farget gjennomføringene, da det bare var 60 minutter til rådighet for hver av gruppene. Som timene gikk var det i grunn naturlig at det ble gjort bortvalg. Ingen av gruppene ble ferdige med samtlige oppgaver, og noen kom ikke lengre enn halvveis. Dermed kan man se hvorfor læreren ønsket å gi elevene mulighet til å utforske opplegget mest mulig uavbrutt. Selv om vi som skapere av lærerveiledningen og elevoppgavene helst skulle sett det annerledes. Man kan dermed se at opplegget i seg selv kan modifiseres slik læreren selv ønsker. Hvis man ignorerer forskningen tilknyttet gjennomføringen, ville dette vært et realistisk scenario. En lærer kan sjelden endre timeplanen basert på et undervisningsopplegg de har funnet, dog kan de endre opplegget til å passe timeplanen. Omfanget til økten ble større enn vi selv tenkte, og gjennomføringen trengte mer tid. Ser man på tiden som var til rådighet opp mot øktens omfang er det klart at det må bli gjort bortvalg. Dessverre krever ofte bortvalg kompetanse fra læreren om de skal være hensiktsmessige. Derfor anbefaler vi, basert på at opplegget er designet for en lærer uten programmeringskompetanse, å faktisk gi undervisningsopplegget mer tid. Slik at læreren ikke trenger å ta vurderinger om hva som må kuttes, da dette kan svekke utbytte av økten. Vi drøftet dette med informanten, og hen var enig i at opplegget godt kunne fungert gjennom en hel dag, eller som en del av et lengre prosjekt.

4.2.6 Utprøving: Elevtilpasninger

Under gjennomføringen ble elevtilpasningen "ferdig kode" brukt for å introdusere programmeringen som en modellering slik at elevene lettere kunne gjenskape det de nettopp hadde sett. Dette var et valg informanten gjorde, og annerledes fra det vi hadde anbefalt, hvor informanten selv skulle skape en kode med innspill fra elevene.

I intervjuet spurte vi informanten om hvordan hen følte tilpasningene kom frem, og hvordan det ble tatt i bruk.

Det var jo litt sånn på den første oppgaven, at de gikk på ferdig utfylt program, den brukte jeg jo nesten på hver gjennomgang, tror jeg. Det var litt med hensikt til at de

skulle komme godt i gang, og så bare skjønne litt hvordan det her fungerte. Når de hadde en "ferdig kode" med 4-5, er det blokker det heter? Så skjønte de det med en gang, og da kan de begynne å bytte ut litt, og så sette de det inn, det er ikke sikkert jeg har trengt å gjøre det, men det var en sikker måte å få med alle på hvertfall.

Hen forklarer at tilpasningen "ferdig kode" fungerte godt som introduksjon, da elevene selv skulle sette inn verdier. Dette gjorde at alle elevene klarte å være med fra start, og sette i gang på egenhånd. I teorien til Skaalvik & Skaalvik (2015) skal introduksjonen være på et forståelig nivå så elevene kunne bygge opp mestringsforventningene til oppgaven, og senere utvikle faglig høyere selvvurdering blant elevene. Dette gjorde informanten da hen introduserte micro:bot, oppkobling og oppgavene på en grundig måte slik at eleven visste hva de skulle gjøre og fikk modellert at programmeringen fungerte dersom kodene var korrekte. Dette mener vi fungerer bedre enn anbefalingen vi hadde, og bør få plass i lærerveiledningen. Denne tilnærmingen ble brukt for at også svakere elever skulle få en forståelse av hva som kreves for en fullstendig kode. Dette skulle også forhindre at elever skulle falle av underveis i undervisningen.

Gjennom observasjonen og intervju erfarte vi at det var svært få, om noen elever i det hele tatt, som brukte noen av tilpasningen utenom "ferdig kode" som de fikk se i introduksjonen. Grunner til dette kan ha vært en dårlig gjennomgang av hvordan vi så for oss at elevene skulle ta i bruk tilpasningene, eller om oppgavene var såpass overkommelige at de ikke trengte tilpasningene for å komme i mål.

Læreren satt en gruppe, som i forkant av undervisningen var utnevnt som eksperter på programmering, både av seg selv og lærer, til å ta i bruk "debugging". Dette var noe elevene ikke fikk til, og dermed kan det også virke som den tilpasningen virket mot sin hensikt sett i retrospektiv. Vi erfarte gjennom våre observasjoner at "debugging" som var det mest omfattende stillasbyggende tilpasningen, ikke egnet seg som stillasbygging for denne klassen, selv om det var motiverende for elevene med en utfordring. Hensikten med tilpasningene var å bygge opp kompetanse og forståelse til oppgavene, og her var nok "ferdig kode", "puslespill" og "halvferdig kode" mer egnet som tilpasninger, mens "debugging" mer som en utfordrende oppgave til elevene.

Vi spurte informanten om det var tenkelig at hen kunne ta i bruk tilpasningene ved en senere anledning til mer krevende oppgaver.

Tilpasningene ja - Så det kan jo være at jeg prøver litt etter at dere drar også. Vi har jo bilene her, jeg må bare finne dem, de bit:botene. Så går det an at vi jobber litt videre med det. Og da kan det jo være at jeg prøver litt på den tilpasningen.

Informanten la også til at hen så verdi i tilpasningen som hen gjerne kunne ta i bruk ved en senere anledning. Kim erkjenner at hen støttet seg vel mye på oss, da hen gjennomgikk opplegget til hen følte seg trygg. Men at hen nå følte seg trygg til å gjennomføre et opplegg, hvor hen tenkte ta i bruk tilpasningene til å jobbe videre med programmering.

Retrospektivt kan man diskutere bortvalget læreren gjør. Forskningsmessig mangler vi innsikt i hvordan elevene opplever tilpasningene. Samtidig ser vi at det i disse gjennomføringene ikke var nødvendig med mer fokus på dem. Som informanten sier gjentatte ganger, var hen strålende fornøyd med opplegget og elevenes tilstedeværelse i økten. Tilpasningene var rett og slett ikke nødvendige for elevenes utbytte. Selv om vi skulle ønsket at de ble testet, var det utforskningen av programvaren som ga elevene motivasjon og læringsutbytte gjennom programmering.

4.2.7 Generell retrospektiv analyse av delmål 1

I dette delkapittelet skal vi oppsummere hvordan vi ville bearbeidet ressursen vår ut ifra observasjoner og intervju. Og kort om hvordan vi kunne lagt til rette for å oppnå delmål 1: *Utvikle et oppgavehefte og en lærerveiledning som egnert seg for lærere som mangler kompetanse innen programmering.*

Basert på funn og erfaringer gjennom observasjon kan en se at lærere trenger mer tid på forberedelse før et undervisningsopplegg i programmering. De bør sette seg inn i programmeringsverktøyet, og misoppfatninger knyttet til dette. Elevene kan skrive inn koder som ikke passer inn i make:code, for eksempel ved at de skriver inn en verdi over 100%, bil make:code automatisk sette farten til 0. Noen av bilene vil kjøre litt mot en av sidene, og dermed bør en ny kloss ved navn "juster venstre/høyre" introduseres til elevene i forkant. En god forberedelse kartlegger slike tilfeller av misoppfatninger og brukerfeil, og kan motvirke dette. Lærere i dag har som regel en ganske full timeplan, og da å sette av flere timer på å gå gjennom en lang lærerveiledning og tilknyttede oppgaver kan virke omfattende. Vi ser her at vi med fordel kunne gjort lærerveiledningen mer kortfattet og

visuell, med flere bilder eller introdusert opplegget gjennom en video, som forklarte tilkobling, tilpasninger og hvordan vi så for oss gjennomføringen.

Tidspresset ved en slik gjennomføring vil alltid være til stede. Selv om det er realistisk, mener vi at det gir en forskning som er mangelfull. Mange aspekter ved lærerveiledningen gikk bort fra gjennomføringen da tidsbegrensningen for økten ble for stor, og vi har dermed manglende innsikt i hvordan disse kunne fungert. Det kan tenkes at gjennomføringen vi fikk ble så bra som det er mulig å få av opplegget vi laget, men det vi fikk var ikke nødvendigvis likt det vi hadde tenkt. Dermed kan man påstå at vi fikk testet et tenkt undervisningsopplegg, som er noe ulikt vårt design.

Samtidig som man ser flere mangler med forskningen og opplegget for øvrig, er det tegn til at opplegget fungerer. Vi ble tildelt en relativt tilfeldig gruppe elever. Det vi hadde kontroll over var kompetansen læreren hadde. Det vi kan se er at en elevgruppe som har berørt programmering tidligere, men aldri vært innom micro:bit eller bit:bot, kan bruke dette opplegget til å utforske dette verktøyet. Dette baserer vi på funnene fra intervjuet og observasjonene, som viser en elevgruppe som behersket nivået på oppgavene godt, og evnet å programmere verktøyet til å svare til disse oppgavene. Elevene viste stor utforskertrang, og evnet å manipulere kjøremønsteret til bilen relativt enkelt. Om de fikk læringsutbytte innenfor naturfagtemaet vei, fart, tid, kommer vi til i kapittel 4.3.

En annen tanke vi har gjort oss i ettertid er i hvor stor grad et undervisningsopplegg som vårt kan lages av en lærer. Det vi mener med det er at vi selv og antakelig mange lærere med programmeringskompetanse kan produsere elevtilpasninger som ligner våre i undervisningsøyeblikket. En lærer med manglende kompetanse på tema derimot, vil ikke ha forutsetningene for å koble opp PC-en sin og impulsivt gi elevene en "puslespill", da dette krever innsikt i hvilke klosser som er hensiktsmessige å ta med. Dermed anbefaler vi at de som lager undervisningsopplegg som ligner vår gjør denne jobben for læreren, enten om det er gjennom lærebøker eller til nye masteroppgaver, og gir elevene de stillasbyggende tilpasningene som vi mener er viktige.

I henhold til kjerneelementene i lærerplan skal ikke programmering undervises alene, og kan dermed knyttes til kompetansemål i flere fag (Kunnskapsdepartementet, 2019.a). Vi skal videre diskutere hvordan man kan koble programmering opp mot fag, og hvordan dette kan gjøres hensiktsmessig.

4.3 Delmål 2: Utvikle et undervisningsopplegg, lærerveiledning og oppgavehefte som knytter fagtemaet vei, fart, tid med programmeringsundervisning

Delmål 2 kommer fra et ønske om å oppnå dybdelæring hos elevene. Selv om vi lager et opplegg som skal være overkommelig for læreren å gjennomføre, og det er i stor grad fokusert på i forskningen, var det alltid en tanke om at undervisningen skulle være nyttig for elevene også. For selv om læreren utvikler sin programmeringskompetanse gjennom dette prosjektet, skal elevene også utvikle sin kompetanse, både innenfor programmering, men også innenfor det valgte naturfaglige temaet.

Det kan argumenteres for at det er unødvendig å integrere et naturfaglig tema i en lærers første undervisning i programmering, men dette er et bevisst valg vi har gjort. Det å oppleve programmering som et verktøy fremfor noe separat som må læres gjør det mulig for både lærere og elever å se på programmering som noe mye mer nyttig for deres hverdag. En lærer som ikke ser behovet for temaet, ser heller ingen grunn til å implementere dette i deres arsenal av undervisningsopplegg. Informanten vår spøkte for eksempel med inntoget av "smartboard" i skolen, og at noen av hans kolleger slet med å se hvorfor de måtte begynne å bruke disse, da krittavlene tilbød det de trengte. Programmering kan sees i samme lys. Vet man ikke hva dette kan tilby, vet man heller ikke hvorfor man bør bruke det i undervisningen.

4.3.1 Forberedelse: Naturfag i programmering

Vi brukte teori fra (Tekna, 2021) og Pörn et al. (2021) for å legge til rette for at delmål 2 skulle oppnås. De skriver om integrering av fagstoff i programmering, og hvilke muligheter og utfordringer man må tenke på. Det viktigste poenget for vår studie er at man må arbeide med det integrerte tema eksplisitt for å utvikle kompetansen (Pörn et al., 2021). Det er altså ikke tilstrekkelig å la elevene programmere en robot til å kjøre lengder ut ifra valg av fart og tid. De må derimot forstå at det er det de gjør og for eksempel forklare for hverandre hvilke endringer som skal til for å endre de andre variablene. Det er samtidig et poeng som sier at integrering av fagtema fungerer best om det allerede er kjent for elevene (Tekna, 2021). Man skal altså ikke innlede et nytt fagstoff gjennom programmering, men heller bruke programmering til å videreutvikle forståelsen av et tema elevene allerede sitter på. Tatt i betraktning nevnt teori og hvilket verktøy vi bestemte oss for å bruke i programmeringsundervisningen var det ganske enkelt å bestemme seg for hvilket naturfaglig tema vi ønsket å integrere i opplegget.



Figur 4: Skjermdump av kloss fra <https://makecode.microbit.org/>

Klossen (Figur 4) og kompetansemålet i naturfag etter 7. trinn som sier at elevene skal kunne "stille spørsmål og lage hypoteser om naturfaglige fenomener, identifisere variabler og samle data for å finne svar" (Kunnskapsdepartementet, 2019.a), ble utgangspunktet for valg av tema. Kodeklossene i *makecode* (Figur 4) som bestemmer kjøremønsteret til bit:boten krever en utforskning av forholdet mellom fart og tid for å oppnå en gitt strekning. Å fokusere på forståelse og utregning av vei, fart, tid, ble for oss naturlig. Vei, fart, tid kan diskuteres som et tema som ikke har en plass i kompetansemålene, men som vi ser det er det nevnte målet perfekt for utforskning av fenomenet. Identifisering av variabler og manipulasjon av disse for å få et utbytte kommer tydelig frem gjennom både oppgaveheftet og gjennomføringen. Samtidig samler elevene data ut fra observasjonene de gjør og baserer endringene på disse.

Basert på inn-puten man må tilføre klossen (Figur 4) for å bestemme kjøremønsteret til bit:boten kan man lage en rekke oppgaver som krever utregning. Farten, som blir bestemt ut ifra tilført motoreffekt i prosent vil alltid være ukjent. Tiden, som blir bestemt i programmet, vil alltid være kjent. Mens strekningen må måles etter testing. Dermed kan man, bare med denne klossen, for eksempel regne ut hvilken fart man får med en gitt prosent motoreffekt. Endrer man deretter tiden, vil strekningen være en funksjon av de kjente variablene. Slik så vi for oss at elevene skulle utforske fenomenet. Dermed designet vi et oppgavehefte som elevene skulle bruke gjennom gjennomføringen.

I tillegg til å gi elevene noe håndfast som ville utfordre deres forståelse av vei, fart og tid, ville dette fungere som en støtte til læreren.

4.3.2 Forberedelse: Oppgaveheftet

Oppgaveheftet (Vedlegg 2) ble designet basert på egne erfaringer av en sjetteklasses forståelse av og evne til å forstå vei, fart og tid, samt teorien som er presentert tidligere i oppgaven. Den ble utviklet uten kjennskap til hvilke forkunnskaper elevene i informantens klasse hadde. Oppgavene bar et preg av en tanke om at elevene helst ville kjøre med maksimal motoreffekt så ofte som de kunne. Dermed innleder vi oppgavene med å regne

fart ut ifra maks effekt og tre selvvalgte strekninger. For å tvinge elevene til å variere farten, skulle de videre regne hvor tregt bilen kunne nå de samme strekningene.

Vi lagde totalt tre oppgaver som krevde tre skriptdesign. Altså de måtte programmere tre forskjellige sammensetninger av klosser, hvor hver deloppgave i tillegg krevde variasjon i input data. Vi så for oss at oppgavene skulle være gjennomførbare, og at det dermed ble mange muligheter til fellesdiskusjoner etter hver oppgave, hvor elevene fikk dele strategier og svar. Samarbeidsteoriene til blant annet Bjursten et al. (2022) og Benton et al. (2017) som vi tidligere har presentert fungerer også her.

Vi så for oss at de vedlagte tilpasningene skulle hjelpe elevene gjennom oppgaveheftet, da ved å minimere tapt tid til programmeringsvanskeligheter. Hvis elevene sto fast kunne de velge en tilpasning som passet deres forkunnskaper og hjelpe dem i programmeringsprosessen. Denne tilnærmingen til gjennomføringen var basert på de stillasbyggende idéene til Bjursten et al. (2022), hvor elevene skal oppleve å bli møtt med oppgaver hvor de skaper koblinger med programmerings problemløsning med mer hverdagslig problemløsning. Dermed skulle utregningen og oppgavebesvarelsen være den største utfordringen, og det som tok lengst tid. I sum skulle elevene oppleve micro:bit og programmering som verktøy for å lære et annet tema.

Originalt hadde vi med vei, fart, tid-trekanten. Vi anså denne som et hjelpemiddel for elevenes utregning, og noe læreren kunne henvise til i veiledningen av elevene. Basert på tilbakemelding fra veiledere og senere undersøkelse i teorien knyttet til dette valgte vi å fjerne denne. Marshall og Carrejo (2008) skriver at elevene må få utforske variablene og utvikle en forståelse som enten bekrefter eller avkrefter deres hverdagsforestillinger av dem. Deretter kan de eventuelt anvende en slik standard modell for å hjelpe dem gjennom utregningen (Hestenes, 1992; Marshall & Carrejo, 2008). Dermed kan vei, fart og tid trekanten fungere mot sin hensikt, da elevene ikke ser sammenhengen mellom variablene, men bruker dette kun som utregning.

4.3.3 Utprøving: Bruk av oppgaveheftet

Herunder, i likhet med strukturen for delmål 1, vil vi gå over til fase 2 og 3, og dermed fokusere på resultater knyttet til utprøvingen og diskusjon av funnene.

Vi spurte Kim om hva hen tenkte om læringsutbytte elevene satt med, sett bort i fra programmeringserfaringen de gjorde seg, da sa hen følgende:

Jeg hørte jo her at "når du skal kjøre, og så må du kjøre litt for langt, og så må du justere tallene litt ned", så de jobber jo med matte sånn ubevisst, de tenker ikke på det. "Her var det 14 000 millisekunder, nei vi må ned til 13 500", de snakker matte uten at de tenker på det. Og det er veldig fint, og de måtte justere hastigheten og prosent, så det var mye matteelementer i dette også, uten at de tenkte over det.

Her blir det forklart at elevene diskutere i gruppene hvordan tiden påvirker lengden bilen kjører. Siden de fleste gruppene hadde konstant fart på 100% av bit:botens motoreffekt, måtte de endre tiden bilen kjørte for å få ønsket lengde. Dette ble diskutert på flere grupper, og dermed kan det tenkes at elevene implisitt satt med en større forståelse av vei, fart og tid. Hen la også til:

Noen grupper tok jo tida, det sto jo hvor lang tid de skulle kjøre og forsåvidt, men ja, så det er bra mye utbytte. Dekker mange mål.

Dette var noe vi også observerte og undret litt på, siden de selv plotter inn tiden bit:boten skal kjøre. Men det kan virke som gruppene gjorde dette for å senere gjøre om sekunder til millisekunder, som det skal skrives i Makecode. Dette underbygger påstanden vår om at programmering ofte er tverrfaglig, da det er elementer fra matematikk i programmeringen.

Samtidig observerte vi at svært få elever skrev noe i oppgaveheftet. Dette kommer direkte fra tidsbegrensningen læreren opplevde. Hen mente at om det var mer tid hadde hen fokusert mer på introduksjon av vei, fart, tid, og samtidig krevd mer notering av svar til oppgavene. På spørsmål om hvordan hen opplevde læringsutbytte innenfor vei, fart, tid, svarer hen:

Ja, nå tror jeg, jeg er usikker på, for jeg har spurt om de har hatt dette i matte, og da sa vel mange nei. Så det ble litt ukjent for dem. Nå skal de jo være god på ganging og deling og den biten der, men hadde vi hatt lengre tid, så kunne jeg vel brukt litt tid til å gå gjennom den modellen.

Dette utdraget kan støttes av teoriene til Tekna (2021) og Pörn et al. (2021) som forteller

oss at kunnskap som skal læres med programmering som verktøy må jobbes med eksplisitt og være noe kjent for elevgruppen fra før. Basert på intervjuet, observasjonene og teorien kan man se et behov for å utprøve opplegget i en klasse som har blitt introdusert for tema tidligere. Det nevnes i utdragene fra intervjuet over at elevene gjorde og tenkte matematisk, og dermed brukte mange begrep som kan knyttes til utvikling innenfor forståelse av vei, fart, tid. Likevel ser man gjennom observasjoner en mangel på helhetlig forståelse tross mangelfull notering og bruk av oppgaveheftet. Selv om det er tydelig gjennom intervjuet og observasjonene at elevene forsto at tiden er en variabel som gjør utslag på strekningen, var det svært få som endret på farten for å nå målet. Det vi dermed ser er at opplegget viser tegn til at elevene kan utvikle forståelse av variablene og hvordan de fungerer som en funksjon på hverandre, men at de ignorerer den ene variabelen. Dette kan påstås å være en del av utforskningen som Marshall og Carrejo (2008) skriver om som nødvendig, hvor elevene får utfordret sine hverdagsforestillinger før de begynner utregningen, men at tidsbegrensningen gjorde at elevene ikke fikk begynt på neste steg i progresjonen.

Om vi skulle gjort dette igjen og dermed gitt oppgaven vår en ny syklus ville vi dermed tilsett at elevene faktisk har møtt dette temaet før og er kjent med alle variablene. Alternativt, kan man gi opplegget bedre tid og faktisk gjennomføre en omfattende introduksjonsøkt. Om det var tilfellet, at elevene fikk introduksjonen de trengte og likevel fikk god tid til å utforske programmeringsbiten, er det tenkelig at gjennomføringen hadde sett annerledes ut, og at elevene dermed hadde notert svar og diskutert strategier i plenum, på tvers av parene. Begge alternativene som nevnes ovenfor bør også gjøre det mulig for elevene å notere utregninger og svar, og dermed være mer fokusert på oppgavebesvarelsen. De vil samtidig oppdage at vi ber dem om å variere farten i oppgaveheftet, noe ingen la merke til, og dermed får de mulighet til å utvikle en helhetlig forståelse av alle variablene.

4.3.4 Utprøving: Elevenes opplevde læringsutbytte

I lærerveiledningen blir det lagt opp til en faglig diskusjon i klasserommet mot slutten av undervisningsopplegget, for å kartlegge hva elevene har lært, og hvilke tanker de har gjort seg gjennom utforskningen de har bedrevet. Dette ble nedprioritert da elevene var såpass motiverte, og vanskelige å løsrives fra programmeringen. Det ble istedenfor tatt opp igjen på slutten av dagen, der elevene fikk dele sine tanker om programmering i naturfag.

De svarte blant annet at de synes det var spennende å programmere i naturfag, og ikke bare i matematikk, og at de klarte å se en sammenheng med hva de hadde lært tidligere i andre fag, og knyttet dette opp til denne timen. Dette aspektet samsvarer med 5e rammeverket til Benton et al. (2016), under punktet bridgE, sammenheng mellom fag og programmering. Elevene syntes også det var spennende med micro:bit, siden det var nytt at de kunne programmere og gjøre noe fysisk og ikke bare på skjerm, som de hadde gjort tidligere.

Generelt viste elevene stor glede, utforskertrang og evne til å arbeide med programmeringen. Derimot mangler vi en del svar når det kommer til oppleggets evne til å undervise i vei, fart, tid. Det ligger også observasjoner til grunn for å påstå at programmeringsverktøyet er godt egnet til å undervise om tema, men teorien om integrering av fag i programmering er klar på at dette må være eksplisitt (Pörn et al., 2021), noe vi opplever at gjennomføringen ikke fikk til. Dermed anbefaler vi også på bakgrunn av dette aspektet å tilegne opplegget lengre tid eller å gjennomføre økten etter en grundig gjennomgang av vei, fart og tid.

4.4 Målet: Bidra til at lærere uten programmeringskompetanse kan knytte sammen undervisning i fagtemaet vei, fart og tid med programmering.

Hovedmålet for denne oppgaven, og årsaken til hvorfor vi har laget produktet vårt baserer seg både på et lærer- og elevfokus. Læreren skulle kunne bruke produktet, og elevene skulle få utbytte av økten. Teorien vi brukte for å designe produktet skulle legge til rette for en gjennomførbarhet for både lærer og elever. Begge partene skulle oppleve at de fikk utbytte av opplegget. Ser man kun på forskningen og manglene av svar vi sitter på, kan man oppleve at økten ikke var suksessfull, men spør du informanten får man et annet bilde. Selv om tilpasningene, de oppsummerende samtalene og oppgaveheftet ikke fikk plassen vi så for oss i undervisningen, var informanten fornøyd. Hen uttaler at hen gjerne kunne gjort dette igjen, og at hen ble tryggere på både maskin- og mykvaren. På spørsmål om hen ville endret noe i veiledningen svarer hen: nei. Hen skylder på dårlig forberedelse på usikkerheten hen følte på innledningsvis og sier at etter et par utprøvinger at hen ble trygg på opplegget og opplevde dette som "skuddsikker". Dermed kan man si at hens oppfatning av veiledningen var godt, og vi kan påstå at lærerveiledningen fungerer for læreren uten eller med manglende kompetanse innenfor programmering.

Forskningens innsyn i elevenes oppfatning av opplegget baserer seg på våre observasjoner og indirekte innblikk gjennom informantens oppfatninger. Funnene herfra gir et bilde av oppnåelse av rammeverket til Benton et al. (2016). Elevene fikk gjennom økten utforsket, forklart for partner, de fikk sett for seg utfall og utdypet seg i partnerens strategier. Og, riktignok implisitt, fikk de oppleve koblingen mellom programmering og fag. Det er gjort observasjoner av samtlige elever som diskuterer aspekter ved vei, fart, tid, og elevene kan dermed forventes å ha utviklet sin forståelse av variablene. Likevel mangler bevis for utvikling på grunn av at dette ikke ble gjennomført eksplisitt. Elevenes frihet til å kunne velge bort regning og notering gjør forskningen på dette delmålet mangelfull. Ser man dette opp mot tidligere nevnt teori, som krever tydelighet (Pörn et al., 2021), kan man diskutere at elevene ikke har opplevd kobling mellom programmering og fag.

Både observasjonene gjort, og uttalelsene fra intervjuet krever bedre tid. Til forberedelse og til gjennomføring. Informanten ser for seg at opplegget kunne truffet målene våre bedre om vi hadde doblet tiden tilgjengelig i klasserommet. Og vi inviterer dermed til videre forskning hvor gjennomføringene blir tildelt god tid til introduksjon av tema, elevs utforskning av programmeringen, oppgavebesvarelse og refleksjoner i klasserommet. Vi erfarer og at lærere oppfatter programmering mer meningsfullt dersom de klarer å anvende dette gjennom å undervise fag, og skape dybdelæring.

5 Avslutning og veien videre

Hensikten med oppgaven var å bidra til at lærere uten programmeringskompetanse kan knytte sammen undervisning i fagtemaet vei, fart og tid med programmering. Vi har utviklet en utfyllende lærerveiledning og et oppgavehefte i tråd med pedagogisk designforskning, og forankret disse i tidligere forskning for et best mulig resultat.

Det må også komme frem at læringsressursen vi har utviklet ikke er et ferdig produkt, men ment som noe som kan forskes på videre og utvikles. Et steg videre kunne vært og testet dette ut ytterligere med de funnene vi har gjort oss, og eventuelt gjort lærerveiledningen mer brukervennlig ved å introdusere programmering gjennom en kort film som tar for seg de viktigste funksjonene. En kort film kan være nyttig for både lærere og elever da ord og bilde gir bedre læring enn ord alene eller bilde alene. Det kommer frem i resultatene våre at det var begrenset med tid både til gjennomføring, og informantene hadde begrenset tid til forberedelser, noe som flere lærere kan kjenne seg igjen i. Tidsaspektet gjorde det også utfordrende å gå inn på de forskjellige tilpasningene vi la ved hver oppgave. Disse la vi stor vekt på i utviklingen av produktene, og vi skulle gjerne fått bedre innblikk i hvordan de faktisk kunne fungert.

Gjennom noen uformelle samtaler med andre lærere er problematikken med programmering å knytte dette opp mot fag noe som går igjen, vi håper dermed at vår læringsressurs kan utvikles videre slik at det kan generaliseres og tas i bruk blant lærere i naturfag og andre fag som skal undervise programmering. Som erfart gjennom utprøving er det mye engasjement å spore blant elevene gjennom å programmere, men læring er mer enn blide og tilfredse elever. En annen teori vi har utarbeidet oss gjennom erfaringer og gjennom uformelle samtaler med flere lærere, er at lærere ofte kan føle programmering blir mer meningsfullt dersom de klarer å anvende programmering til å undervise fag, å skape dybdelæring. Det krever innsikt i forkunnskapene til fagemnet som skal integreres i programmeringsundervisningen. I denne studien viste det seg at elevene ikke hadde forutsetningene til både å lære programmeringsverktøyet og anvende dette til å utvikle en forståelse for utregning av vei, fart, tid. Om dette kun skyldes dårlig tid, vet vi ikke, men vi mener at økten kan utvides til å gi elevene et bedre innblikk i hvordan man regner på disse variablene.

Selv om oppgaven har gitt funn som kan være viktige for videre forskning, er det viktig å erkjenne studiens begrensninger. Da må vi trekke frem dens begrensninger innenfor

omfang. Det er kun én lærer som har testet opplegget. Dette gjør det utfordrende å trekke konklusjoner om hvorvidt resultatene våre kan generaliseres. Samtidig må vi igjen vise til begrenset tid til gjennomføring. Selv om det er realistisk for en undervisningssituasjon, gjør det at vi mangler svar på spørsmål vi har, da spesielt knyttet til delmål 2.

Likevel viser studien at informanten opplevde undervisningen som svært god, og hen selv mener at opplegget er noe som kan bli brukt igjen. Bare dette forteller oss at vi har truffet på delmål 1, hvor målet var å lage et produkt som egner seg for læreren uten programmeringskompetanse. Selv om våre observasjoner og opplevelser av økten gjorde det mulig å komme med anbefalinger til flere endringer, mener Kim at opplegget var bra som det var.

Litteratur

- Ball, D. L., Thames, M. H. & Phelps, G. (2008). Content Knowledge for Teaching: What Makes It Special?. *Journal of Teacher Education*, 59(5), 389-407. <https://doi.org/10.1177/0022487108324554>
- Bau, D., Gray, J., Kelleher, C., Sheldon, J., & Turbak, F. (2017). Learnable programming. *Communications of the ACM*, 60(6), 72-80. <https://doi.org/10.1145/3015455>
- Bell, T.C., & Vahrenhold, J. (2018). CS Unplugged - How Is It Used, and Does It Work? *Adventures Between Lower Bounds and Higher Altitudes*.
- Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging primary programming and mathematics: Some findings of design research in England. *Digital Experiences in Mathematics Education*, 3(2), 115-138. <https://doi.org/10.1007/s40751-017-0028-x>
- Bjursten, E.-L., Nilsson, T., & Gumaelius, L. (2022). Computer programming in primary schools: Swedish Technology Teachers' pedagogical strategies. *International Journal of Technology and Design Education*, 33(1), 1345-1368. <https://doi.org/10.1007/s10798-022-09786-7>
- Bjørndal, K. E. (2013). Pedagogisk designforskning – en forskningsstrategi for å fremme bedre undervisning og læring. I M. Brekke, & T. Tiller (Red.), *Læreren som forsker. Innføring i forskningsarbeid i skolen* (s. 245-260). Universitetsforlaget.
- Bjørndal, C. R. P (2011). *Det vurderende øyet: Observasjon, vurdering og utvikling i undervisning og veiledning*. (2.utg.). Gyldendal.
- Brattetaule, I. (2022). *Få tegning tilbake i naturfagsundervisning: Utvikling av en didaktisk ressurs*. [Masteroppgave]. Norges teknisk-naturvitenskapelige universitet.
- Brown, N. C. C., & Wilson, G. (2018). Ten quick tips for teaching programming. *PLoS Computational Biology*, 14(4). <https://doi.org/10.1371/journal.pcbi.1006023>
- Bråting, K., & Kilhamn, C. (2021). The integration of programming in Swedish School Mathematics: Investigating Elementary Mathematics textbooks. *Scandinavian Journal of Educational Research*, 66(4), 594-609. <https://doi.org/10.1080/00313831.2021.1897879>
- Bybee, R. W., Taylor, J., Gardner, A., Scotter, P., Carlson, J., Westbrook, A., & Landes, N. (2006). *The BSCS 5E instructional model: Origins and effectiveness* (s. 1-80). Office of Science Education, National Institutes of Health.
- Carroll, M. & Ward, H. (2016) Using Computers in Science. I: H. Ward & J. Roden (Red.), *Teaching Science in the Primary School* (3.utg., s. 174- 192). Sage.
- Cobb, P., Confrey, J., diSessa, A., Lehrer, R. & Schauble, L. (2003). Design Experiments in Educational Research. *Educational researcher*, 32(1), 9-13. <https://doi.org/10.3102/0013189X032001009>
- Crotty, M. (1998). *Foundations of Social Research: Meaning and perspectives in the research process*. Routledge.
- Digital learning institute. (u.å.). *Mayer's 12 principles of multimedia learning | DLI*. Digital Learning Institute. <https://www.digitallearninginstitute.com/blog/mayers-principles-multimedia-learning>

- Finnish National Board of Education (u.å.). *National core curriculum for basic education 2014*. Hentet 16. April 2024 fra <https://www.oph.fi/en/education-and-qualifications/national-core-curriculum-primary-and-lower-secondary-basic-education>
- Fleury, S. C., & Bentley, M. L. (1991). Educating Elementary Science Teachers: Alternative Conceptions of the Nature of Science. *Teaching Education*, 3(2), 57–67. <https://doi.org/10.1080/1047621910030207>
- Gjøvik, Ø., Torkildsen, H. A. (2019). Algoritmisk tekning. *Tangenten – tidsskrift for matematikkundervisning*, 30(3). 31–37.
- Gomes, A., & Mendes, A. J. (2007). Learning to program-difficulties and solutions. *Paper presented at the International Conference on Engineering Education–ICEE*. Hentet fra: https://www.researchgate.net/publication/228328491_Learning_to_program_-_difficulties_and_solutions
- Haraldsrud, A. D., Sveinsson, H. A. & Løvold, H. H. (2020) Programmering i skolen. Universitetsforlaget.
- Hazelkorn, E., Beernaert, Y., Constantinou, C. P., Deca, L., Grangeat, M., Karikorpi, M., Lazoudis, A., Casulleras, R. P., & Welzel-Breuer, M. (2015). *Science education for responsible citizenship: Report to the European Commission of the expert group on science education* (Rapport nr. 26893). Europeakommisjonen.
- Hestenes, D. (1992). Modeling games in the Newtonian world. *American Journal of Physics*, 60(8), 732-748. <https://doi.org/10.1119/1.17080>
- Humble, N. (2022). Teacher observations of programming affordances for K-12 mathematics and technology. *Education and Information Technologies*, 27(4), 4887–4904. <https://doi.org/10.1007/s10639-021-10811-w>
- Johannessen, A. & Christoffersen, L. (2012). *Forskningsmetode for lærerutdanningene*. Abstrakt forlag.
- Johannessen, A., Christoffersen, L. & Tufte, P. A. (2020). *Forskningsmetode for økonomisk-administrative fag* (4.utg.). Abstrakt forlag.
- Kunnskapsdepartementet (2013). Læreplan i naturfag (NAT01-03). Fastsatt som forskrift. Læreplanverket for Kunnskapsløftet 2013. <https://data.udir.no/kl06/nat1-03.pdf>
- Kunnskapsdepartementet (2017). Overordnet del – verdier og prinsipper for grunnopplæringen. Fastsatt som forskrift ved kongelig resolusjon. Læreplanverket for Kunnskapsløftet 2020. <https://www.regjeringen.no/no/dokumenter/verdier-og-prinsipper-for-grunnopplaringen/id2570003/>
- Kunnskapsdepartementet (2019.a). Læreplan i naturfag (NAT01-04). Fastsatt som forskrift. Læreplanverket for Kunnskapsløftet 2020. <https://data.udir.no/kl06/v201906/laereplaner-1k20/NAT01-04.pdf?lang=nob>
- Kunnskapsdepartementet (2019.b). Læreplan i matematikk (MAT01-05). Fastsatt som forskrift. Læreplanverket for Kunnskapsløftet 2020. <https://data.udir.no/kl06/v201906/laereplaner-1k20/MAT01-05.pdf?lang=nob>
- Makki, T. W., O’Neal, L. J., Cotten, S. R., & Rikard, R. V. (2018). When first-order barriers are high: A comparison of second- and third-order barriers to classroom computing

- integration. *Computers & Education*, 120(1), 90–97.
<https://doi.org/10.1016/j.compedu.2018.01.005>
- Meld. St. 21 (2016-2017). *Lærelyst - tidlig innsats og kvalitet i skolen*. Kunnskapsdepartementet. <https://www.regjeringen.no/no/dokumenter/meld.-st.-21-20162017/id2544344/?ch=3>
- Misfeldt, M., Szabo, A., & Helenius, O. (2019). Surveying teachers' conception of programming as a mathematical topic following the implementation of a new mathematics curriculum. I: U. Jankvist, M. Van den Heuvel-Panhuizen, & M. Veldhuis (Red.), *Proceedings of the Eleventh Congress of the European Society for Research in Mathematics Education, CERME11* (s. 2713–2720). Freudenthal Group & Freudenthal Institute, Utrecht University and ERME
- Nagappan, N., Williams, L., Ferzli, M., Wiebe, E., Yang, K., Miller, C., & Balik, S. (2003). Improving the CS1 experience with pair programming. *SIGCSE Bulletin*, 35(1), 359-362. <https://doi.org/10.1145/792548.612006>
- NESH (2021). *Forskningsetiske retningslinjer for samfunnsvitenskap og humaniora*. (5.utg.). <https://www.forskningsetikk.no/globalassets/dokumenter/4-publikasjoner-som-pdf/forskningsetiske-retningslinjer-for-samfunnsvitenskap-og-humaniora.pdf>
- Newhall, T., Meeden, L., Danner, A., Soni, A., Ruiz, F., & Wicentowski, R. (2014). A support program for introductory CS courses that improves student performance and retains students from underrepresented groups. I: *Proceedings of the 45th ACM technical symposium on Computer science education (SIGCSE '14)* (s. 433–438). Association for Computing Machinery. <https://doi.org/10.1145/2538862.2538923>
- NTNU (2023). *Lagringsguide*. Innsida – NTNUs intranett. <https://i.ntnu.no/wiki/-/wiki/Norsk/Lagringsguide>
- Popat, S. & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers & Education*, 128, 365-376.
<https://doi.org/10.1016/j.compedu.2018.10.005>
- Postholm, M. B. & Jacobsen, D. I. (2018) *Forskningsmetode: for masterstudenter i lærerutdanning*. Cappelen Damm Akademisk
- Prediger, S. & Zwetschler, L. (2013). Topic-specific design research with a focus on learning processes: The case of understanding algebraic equivalence in grade 8. I: T. Plomp, & N. Nieveen (Red.), *Educational design research – Part B: Illustrative cases* (s. 407-424). SLO.
- Qian, Y., & Lehman, J. (2017). Students' misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education*, 18(1), 1-24. <https://doi.org/10.1145/3077618>
- Roalsvik, T. (2023). *Programmering for læring og motivasjon: En casestudie om hvordan undervisning med micro:bit kan skape læring og motivasjon i matematikk*. [Masteroppgave]. Universitetet i Stavanger.
- Roalsvik, T. (2023). *Programmering for læring og motivasjon: En casestudie om hvordan undervisning med micro:bit kan skape læring og motivasjon i matematikk*. [Masteroppgave]. Universitetet i Stavanger.

- Sentance, S., Waite, J., & Kallia, M. (2019). Teaching computer programming with PRIMM: a sociocultural perspective. *Computer Science Education*, 29(2-3), 136-176. <https://doi.org/10.1080/08993408.2019.1608781>
- Sikt (2023.a). *Barnehage- og skuleforskning*. Sikt: Kunnskapssektorens tjenesteleverandør. <https://sikt.no/tjenester/personverntjenester-forskning/personvernhandbok-forskning/barnehage-og-skuleforskning>
- Sikt (2023.b). *Vurdering av innsendte meldeskjema*. Sikt: Kunnskapssektorens tjenesteleverandør. <https://sikt.no/tjenester/personverntjenester-forskning/fylle-ut-meldeskjema-personopplysninger/vurdering-av-innsendte-meldeskjema>
- Skaalvik, E. M., & Skaalvik, S. (2015). *Motivasjon for læring: teori og praksis*. Universitetsforlaget.
- Staberg, R. L., Tandberg, C. & Grindeland, J. M. (2020). *Biologididaktikk for lærere*. Gyldendal.
- Statped (2021, 1. mars) *Programmering i skolen*. Statped. <https://www.statped.no/laringsressurser/teknologitema/programmering-for-barn-med-saerskilte-behov/programmering/programmering-i-skolen/>
- Statped (2023, 30. November) *Micro:bit*. Statped. <https://www.statped.no/laringsressurser/teknologitema/microbit/>
- Sunde, B. H. & Øen, E (2022). *Bruk av programmering i matematikkfaget på mellomtrinnet: En kvalitativ studie av læreres tilrettelegging og erfaringer med ressurser fra Super:bit*. [Masteroppgave]. Høgskulen på Vestlandet.
- Super:bit (u.å.a) *Hva er super:bit?*. Super:bit. <https://www.superbit.no/hva-er-superbit/>
- Super:bit (u.å.b) *Hvordan Få Bit:Bot til å kjøre en gitt lengde?*. Super:bit. <https://www.superbit.no/undervisningsopplegg/kjoer-en-meter-med-bitbot>
- Super:bit (u.å.c) *Kom i gang med micro:bit*. Super:bit. <https://www.superbit.no/undervisningsopplegg/kom-i-gang-med-microbit/kom-i-gang-med-microbit-elev/>
- Swensen, H. (2014). *Omvendt undervisning og tilpasset opplæring*. I T. H. Giæver, M. Johannesen & L. Øgrim (red.), *Digital praksis i skolen*. Oslo: Gyldendal Akademisk.
- Tekna. (2021, March 22). *Programmeringsdidaktikk*. Tekna – Teknisk-naturvitenskapelig forening. <https://www.tekna.no/fag-og-nettverk/realfag-og-utdanning/realfagsbloggen/programmeringsdidaktikk/>
- The Micro:bit Educational Foundation (u.å.) *Features*. Micro:bit. <https://microbit.org/get-started/features/overview/>
- UiB. (2023, 15. februar). *8 viktige yrker for Fremtiden*. Universitetet i Bergen. <https://www.uib.no/realfag/140960/8-viktige-yrker-fremtiden>
- Utdanningsdirektoratet. (2019. 13. mars). *Dybdelæring*. Utdanningsdirektoratet. <https://www.udir.no/laring-og-trivsel/dybdelaring/>
- Van den Akker, J., Gravemeijer, K., McKenny, S. & Niveen, N. (2006). *Introducing educational design research*. I J. van den Akker, K. Gravemeijer, S. McKenney & N. Niveen (Red.), *Educational design research* (s. 3-7). Routledge.

Vinnervik, P. (2023). Programming in school technology education. *Programming and Computational Thinking in Technology Education*, 173-195.
https://doi.org/10.1163/9789004687912_008

Wordpress. (2018, August). PRIMM. PRIMM. <https://primmportal.com/>

Øgreid, A.K. (2021) Intervensjonsbegrepet i fire kvalitative forskningsdesig. I E. Andersson-Bakken & C. P. Dalland (Red.), *Metoder i klasseromsforskning: Forskningsdesign, datainnsamling og analyse*. (s. 209-238). Universitetsforlaget.

Vedlegg

Vedlegg 1: Lærerveiledningen

Vedlegg 2: oppgaveheftet

Vedlegg 3: Intervjuguide

Vedlegg 4: Observasjonsplan

Vedlegg 5: Prosessdokument

Vedlegg 6: Samtykkeskjema – Elever

Vedlegg 7: Samtykkeskjema - Lærer

Vedlegg 1: Lærerveiledningen

Vi har valgt å legge ved lærerveiledningen i form av et google Docs dokument. Dette gjør vi for syns skyld, da vi opplever et vedlegg på omtrent 25 sider som forstyrrende på oppgaveintrykket. Informanten mottok et ordinært dokument da vi sendte hen lærerveiledningen, men vi har sørget for at innholdet og utseende er det samme.

Lærerveiledningen

<https://docs.google.com/document/d/1o3RVDUJRNxBhZO3Mp7R1EWUW1yYRXOYqUgJTt5HGbk/edit?usp=sharing>

Vedlegg 2: Oppgaveheftet

Oppgaveheftet blir også lagt ved som en google docs-link. I dette tilfellet er det i samme format som det informanten, og senere elevene, mottok.

Oppgaveheftet

<https://docs.google.com/document/d/180YDRVPVHW3AGt42aeioA8hSfo7q5bDVBYePdZI8P4/edit?usp=sharing>

Vedlegg 3: Intervjuguide

Bakgrunn

1. Hvor lenge har du vært lærer?
2. Hvilken lærerutdanning har du?
 - a. Hadde du opplæring i programmering i utdanningsløpet ditt?
3. Hvor lenge har du arbeidet på denne skolen?
4. Hvilken stilling har du på denne skolen?

5. Har du undervist programmering tidligere?
6. Hvilken kjennskap har du til denne elevgruppen?
7. Har elevgruppen blitt undervist i programmering tidligere?

Generelle spørsmål

1. Har du vurdert å ha programmeringsundervisning tidligere?
2. Hvordan opplevde du at økten gikk?
3. Var det noen faktorer som gjorde det utfordrende å gjennomføre økten slik den var planlagt?
4. Hvordan synes du det var å gjennomføre en økt som var planlagt for deg?
5. Hvilket læringsutbytte tror du elevene fikk gjennom denne økten?
6. Hvordan synes du denne økten fungerte som en innledning til videre arbeid med programmering?
7. Hvordan synes du det fungerte med ferdigprogrammerte oppgaver?
8. Hva tror du er årsaken til at så mange lærere vegrer seg for å undervise i eller med programmering?

9. Skulle du ønske at programmeringsundervisningen kommer separat fra naturfag og andre fag, og at det dermed fungerer som et eget emne?

Opplegget

1. Hvordan synes du opplegget fungerte?
 - a. Hva fungerte godt?
 - b. Hva fungerte dårlig?
2. Er det endringer du ville gjort?
3. Hvordan kan vi legge opp til at du kan tilpasse opplegget for din elevgruppe?
4. Er det noen mangler du oppdager i lærerveiledningen slik den er nå?
5. Ville du brukt dette opplegget slik det er nå i videre undervisning?

Ut ifra observasjonene

Vei fart tid

1. Temaet i økten var vei fart tid, hvordan synes du elevene diskuterte dette?
2. Hvordan kan vi sørge for at vei, fart, tid faktisk blir implementert i opplegget?
3. Du endret opplegget midtveis ved at elevene skulle fokusere mer på oppgavene enn programmeringsdesign, hvorfor og hvordan funket dette?
4. Helt generelt, hvis vi ikke hadde vært der. Hvordan tror du øktene ville blitt annerledes?
5. Hvis vi ikke var der, ville du fokusert mer på tilpasningene vi har lagt ved?

6. Og er dette noe du kunne fullført på egenhånd?
7. Er opplegget noe du kunne anbefalt til en kollega, andre lærere?
8. Du hadde flere som mente de kunne dette fra før. Hvordan ville du, som uerfaren på fagfeltet, gitt de nok utfordring?
9. Du delte opp elevene i arbeidspar og trioer basert på hvem de ville jobbe med, hvordan fungerte dette? Og tror du utfallet hadde blitt annerledes dersom du delte inn gruppene tilfeldig?
10. I opplegget oppfordrer vi til å samle elevgruppen etter de har gjennomført hver oppgave. Du gjorde ikke det. Hvorfor?
11. Hvordan synes du programmering står seg i læreplan?

Vedlegg 4: Observasjonsplan 4 gjennomføringer

Observasjon av lærer	Observasjon av elevgruppen
<p>Bruk av tilpasninger: Introduserte med "ferdig kode".</p> <p>Ingen andre koder introdusert av lærer</p>	<p>Bruk av tilpasninger:</p> <p>Mange elever bruker "ferdig kode" i oppstarten.</p> <p>Da ingen brukte tilpasningene, forsøkte jeg å gi en gruppe "debugging".</p> <p>- Opplever at de ble motiverte og påkoblet, men de slet veldig. De konkluderte korrekt på noen klosser, at de ikke burde være der, men helhetlig – veldig vanskelig</p>
<p>Selvstendighet: Lite selvstendighet til å begynne med, usikker på opplegget, observatører måtte bistå med introduksjon.</p>	<p>Utforskning:</p> <p>Elevene er generelt påkoblet og finner de riktige klossene etter en del prøving og feiling.</p>
<p>Programmeringsdidaktikk: Inndeling av grupper etter relasjoner og ønsker fra elevene.</p>	<p>Utfordringer knyttet til programmering:</p> <p>Noe utfordrende å komme i gang for noen. De som sliter mest blir oppfordret til å bruke "ferdig kode" - da kommer de i gang.</p> <p>Problem knyttet til maskinvaren – Elevene irriterer seg over at bilen ofte svinger svakt til en av sidene. Introduserer klossen *Juster høyre/venstre* - den fungerer godt! Og elevene kobler seg igjen på.</p>
<p>Valg: Satt en gruppe til å teste ferdigheter gjennom "debugging".</p> <p>Endret underveis fra designfokus til oppgavefokus. Lite frihetsgrader ved første gjennomføring. Elevene fikk bestemme blant annet hvor langt de fikk kjøre de siste gjennomføringene.</p>	<p>Utfordringer knyttet til utregning:</p> <p>Ingen regner i oppgaveheftet... De snakker om variablene, men ingen klarer å koble det de gjør til utregningen.</p>

<p>Bruker elevene aktivt under gjennomgangen</p>	<p>Etter å ha fått innblikk i elevenes forforståelse, tyder alt på at de ikke har lært hvordan de skal regne ut vei/fart/tid.</p> <p>Hvordan kan vi gjøre dette annerledes?</p>
<p>Bortvalg: Lite tid til avsluttende refleksjoner.</p> <p>Ingen gjennomgang av andre elevtilpasninger enn "ferdig kode".</p>	<p>Samarbeid:</p> <p>Generelt godt samarbeid i gruppene – god inndeling.</p> <p>Noen grupper fordeler tydelig ansvarsområder mellom seg. Noen koder, noen gjennomfører utprøving. Noen foreslår hva den andre skal gjøre på pcen.</p> <p>Samtalene er faglige!</p> <p>Noen få kobler seg av. Én gruppe har kun én som er delaktig i opplegget. Den andre eleven lurte seg bak skjermen i hjørnet og spiller spill. - lærer lar dette gå</p>

Vedlegg 5: Prosessdokument

I dette dokumentet skal vi presentere samskrivingsprosessen gjennom denne masteroppgaven. Vi skal forklare hvordan vi har jobbet med oppgaven, hvordan vi har fordelt arbeidsmengden, og hvordan samarbeidet har fungert gjennom oppgaven.

Da vi startet med oppgaven, var vi klare over premissene for arbeidet. Vi var forberedte på at dette kom til å bli en krevende oppgave. Vi var allerede godt kjent fra tidligere, og dette har virket positivt på samarbeidet gjennom oppgaven. Vi startet med få tanker om hva vi ville skrive om, og hvilke metoder vi ville bruke. Etter en dialog sammen og med veiledere falt vi på en utviklingsrettet master med programmering som tema, noe vi begge syntes å virke spennende.

En utviklingsrettet master hvor vi skulle utvikle en læringsressurs var avhengig av tett samarbeid gjennom starten av prosessen. Etter at vi ble enige om hvordan vi ville gjøre dette fordelte vi teori oss imellom, som skulle være utgangspunktet i prosessen. Både litteraturgjennomgang og skriving har blitt fordelt likt og etter ønske, slik at vi begge kunne aktivt ta del i oppgaven. For at begge skulle føle en tilhørighet til hele oppgaven, har vi skrevet enkelte deler hver for oss, som vi har gått gjennom og drøftet sammen.

Arbeidsfordelingen ble gjort slik at vi fordelte noen temaer som vi skulle ha ansvaret for, og reflektert dette sammen, hva som skal med og hva vi kan unnlate. Det har vært viktig for oppgaven vår å reflektere over og diskutere læringsressursen og teste ut dette, da vi får sett dette på forskjellige måter. Vi har alltid kunne diskutert og reflektert valg i utviklingen av læringsressursen gjennom hele prosessen.

I en masteroppgave hvor man er to vil en alltid støte på utfordringer. Deler er skrevet hvor vi ikke har vært sammen, men disse delene ble nøye diskutert og reflektert over i ettertid. Det har vært en åpen kommunikasjon, hvor vi alltid har funnet løsninger begge har vært fornøyde med. Innstillingen og skrivestilen våre er ganske like, så vi har lett for å finne et språk som flyter gjennom hele oppgaven. Det er selvsagt vært noen ulikheter i oppgavens struktur, noe vi også har kommet til enighet om hva som er det beste for oppgaven, så det skal være likt gjennom hele. Dette er noe vi selv ikke har bemerket oss som en utfordring, men heller sett på som en styrke. En annen styrke gjennom oppgaven var å teste ut produktet vårt hvor begge fikk observere, som ikke hadde blitt det samme dersom det hadde blitt gjennomført hver for oss.

Vil du delta i forskningsprosjektet

Implementering av programmering i ordinær naturfagsundervisning.

Dette er et spørsmål til deg om å delta i et forskningsprosjekt hvor formålet er å utvikle en lærerveiledning for programmering i naturfag. I dette skrevet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

Formål

Gjennom forskningen ønsker vi å utvikle en lærerressurs for implementering av programmering i ordinær undervisning. Ved å se hvordan vårt undervisningsopplegg fungerer i klasserommet vil vi i etterkant gjøre endringer slik at den i etterkant kan fungere bedre som en fullstendig veiledning til lærere som ønsker å ta i bruk programmering.

Denne gjennomføringen vil være hoveddelen av vår masteroppgave, og er helt essensiell for at vi kan levere en god besvarelse.

Hvem er ansvarlig for forskningsprosjektet?

NTNU (Norges teknisk-naturvitenskaplige universitet) er ansvarlig for prosjektet.

Hvorfor får du spørsmål om å delta?

Årsaken til at vi gjennomfører forskningen på denne skolen er at en av oss har tidligere kjennskap til skolen og har gjennom en praksisperiode blitt kjent med flere ansatte. Vi har tatt kontakt med læreren vi skal samarbeide med fordi hen faller under våre kriterier for en optimal deltaker i studien. Elevene som deltar er med fordi de er lærerens elever og hen har kjennskap til klassen og klassemiljøet. Vi ønsket å plassere studien i en så naturlig lærersituasjon som mulig og derfor følger elevene med læreren.

Hva innebærer det for deg å delta?

Det er læreren som er i fokus for denne forskningen. Derfor skal vi gjennomføre observasjoner av hvordan hen bruker vår lærerveiledning og bruke påfølgende intervju for å se hvordan denne fungerer i en reell undervisningssituasjon.

Hvis eleven deltar i prosjektet, innebærer det at hen deltar i en ordinær undervisningsøkt hvor det er to studenter som observerer og noterer lærerens handlinger. Eleven vil samtidig bli observert for å gi oss en forståelse av hvor godt vårt undervisningsdesign fungerer. Dette blir tatt opp i nevnte intervju, men vil ikke fungere som resultatgrunnlag i masterbesvarelsen. Vi skal med andre ord ikke samle inn elevens personopplysninger eller på noen som helst måte innhente informasjon som kan være identifiserende.

Det er frivillig å delta

Det er frivillig å delta i prosjektet. Hvis eleven/foresatte velger å avstå fra

deltakelse, kan man når som helst trekke samtykket tilbake uten å oppgi noen grunn. Det vil ikke ha noen negative konsekvenser hvis det velges å ikke delta eller man senere velger å trekke seg.

Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger

Vi vil ikke samle inn, lagre eller bruke noen personopplysninger fra elevene. Det er derfor fullstendig anonymitet for elevene. Våre observasjoner/notater vil ikke være knyttet til noen form for personopplysninger.

Med vennlig hilsen

Prosjektansvarlig

Elisabeth Inge Romjin

Student

Sindre Lines Arntzen

Lasse Husby

Samtykkeerklæring

Hvis det ikke er ønskelig å delta i forskningsprosjektet ber vi deg/dere ta kontakt med avsender.

Hvis det ikke blir tatt kontakt før dato for gjennomføring (8. mars 2024) anser vi det som mottatt samtykke for elevens deltakelse i prosjektet.

Vil du delta i forskningsprosjektet

Programmering i naturfag?

Formålet med prosjektet

Dette er et spørsmål til deg om du vil delta i et forskningsprosjekt hvor formålet er å

- Formålet med prosjektet er å lage et undervisningsopplegg i naturfag, hvor elevene skal lære om fart gjennom programmering i Micro: bit. Målet for prosjektet er å se hvordan undervisningsopplegget kan tilpasset slik at alle lærere kan ta dette i bruk i undervisningen.
- Dette er en utviklingsbasert masteroppgave.

Hvorfor får du spørsmål om å delta?

Du får denne forespørselen fordi

- Du er valgt ut som potensiell informant til dette prosjektet fordi du er lærer på barneskolen og har kompetansen som vi ønsker for å utvikle vår masteroppgave. Det er trolig at du blir vår eneste informant, da fokuset på vår besvarelse er på utviklingen av undervisningsopplegget.

Hvem er ansvarlig for forskningsprosjektet?

NTNU er ansvarlig for personopplysningene som behandles i prosjektet.

Det er frivillig å delta

Det er frivillig å delta i prosjektet. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg.

Hva innebærer det for deg å delta?

- Metodene som blir brukt for å samle inn data er intervju og observasjoner av gjennomføringen av undervisningsopplegget
- Et Semistrukturert intervju, med lydopptak
- Informanten skal anonymiseres så navn og personopplysninger skal ikke være med.
- Intervjuet skal registreres gjennom lydopptak, og observasjoner skal noteres ned.

Kort om personvern

Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrivet. Vi behandler personopplysningene konfidensielt og i samsvar med personvernregelverket. Du kan lese mer om personvern under*.

Med vennlig hilsen

Prosjektansvarlig
Elisabeth Inge Romijn

Studenter
Lasse Husby & Sindre Lines Arntzen

Utdypende om personvern – hvordan vi oppbevarer og bruker dine opplysninger

- Personopplysningene til deltakerne i dette prosjektet vil kun være tilgjengelig for oss og vår veileder mens prosjektet pågår. Etter endt forskningsperiode vil informasjonen slettes.
- Personopplysningene vil bli lagret på NTNUs sentrale server. Gjennom vårt personlige hjemområde på serveren vil informasjonen lagres kryptert og sikkert.
- Ingen personopplysninger vil bli publisert, og det vil ikke være gjenkjennelig hvem som står bak informasjonen vi publiserer i oppgavebesvarelsen. Gjennom anonymisering vil informantens identitet og arbeidsplass være ugjenkjennelig.

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag fra NTNU har personverntjenestene ved Sikt – Kunnskapssektorens tjenesteleverandør, vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

Dine rettigheter

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

- å be om innsyn i hvilke opplysninger vi behandler om deg, og få utlevert en kopi av opplysningene,
- å få rettet opplysninger om deg som er feil eller misvisende,
- å få slettet personopplysninger om deg,
- å sende klage til Datatilsynet om behandlingen av dine personopplysninger.

Vi vil gi deg en begrunnelse hvis vi mener at du ikke kan identifiseres, eller at rettighetene ikke kan utøves.

Hva skjer med personopplysningene dine når forskningsprosjektet avsluttes?

Prosjektet vil etter planen avsluttes 27. Mai 2024

Opplysningene vil da slettes etter prosjektet er avsluttet

Spørsmål

Hvis du har spørsmål eller vil utøve dine rettigheter, ta kontakt med:

- Elisabeth Inge Romijn
 - Tlf: 93253288
 - Mail: elisabeth.romijn@ntnu.no
- Personvernombud: Thomas Ørnulf Helgesen
 - thomas.helgesen@ntnu.no

Hvis du har spørsmål knyttet til Sikts vurdering av prosjektet, kan du ta kontakt på e-post: personverntjenester@sikt.no, eller på telefon: 73 98 40 40.

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

- q Å delta i Intervju
- r Å delta i observasjon

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

(Informant/Lærer)

