

Synne Christin Colling

Bruk av programmering i undervisning om smittespredning og vaksinering

Utvikling av et undervisningsopplegg

Masteroppgave i naturfag. Grunnskolelærerutdanning 5.-10. trinn
Veileder: Arne Stormo

Mai 2024



NTNU

Kunnskap for en bedre verden

Synne Christin Colling

Bruk av programmering i undervisning om smittespredning og vaksiner

Utvikling av et undervisningsopplegg

Masteroppgave i naturfag. Grunnskolelærerutdanning 5.-10. trinn
Veileder: Arne Stormo
Mai 2024

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



Kunnskap for en bedre verden

SAMMENDRAG

I denne studien har det blitt utviklet og testet et undervisningsopplegg bestående av tre oppgavehefter fordelt på temaene radiokommunikasjon, smittespredning og vaksinerings med tilhørende lærerveiledning. Formålet med studien er å *bidra til at programmering kan integreres i undervisningen om smittespredning og vaksinerings*.

Jeg opplevde selv utfordringen med å kombinere programmering med naturfaglige fenomener. Programmering er et nytt fagområde i skolen, og dermed vil det være ulik programmeringskompetanse blant lærerne. Gjennom denne studien ønsker jeg å bidra til å gjøre dette programmeringsarbeidet enklere. Videre kan smittespredning og vaksinerings prege samfunnet i stor grad, og det er derfor viktig at elevene lærer om dette i skolen. Hensikten med denne studien er å utvikle et undervisningsopplegg som benytter programmering av micro:bit til å simulere smittespredning og vaksinerings som begrensende faktor.

Metoden jeg benyttet er inspirert av pedagogisk designforskning (Bjørndal, 2013). Undervisningsopplegget ble testet ut og forbedret i tre omganger, en gang på 8. trinn og to ganger på 10. trinn med to ulike naturfagslærere med ulik programmeringskompetanse. Mine egne observasjoner og samtaler med lærerne dannet grunnlag for datamaterialet som ble brukt til videreutvikling av undervisningsopplegget.

I den etterfølgende retrospektive analysen fant jeg at det er viktig at elever reflekterer over sine handlinger også ved programmeringsarbeid, både under uttestingen og ved etterfølgende diskusjoner. Siden lærere og elever vil ha ulik kompetanse på et nytt fagområde som programmering, er det også viktig å tenke på nivå-differensiering i både lærerveiledningen og programmeringsoppgavene. Det er videre viktig for utvikler å være åpen i tankene i forhold til mulige behov for endring av modeller underveis i prosessen. En god modell blir først til etter kontinuerlig videreutvikling og forbedring. Elevene på sin side bør forklares at modeller er en forenkling av virkeligheten med de begrensninger det medfører i prosessen.

ABSTRACT

In this study, a teaching plan has been developed and tested consisting of three exercise booklets divided into the topics of radio communication, the spread of infection and vaccination with associated teacher guidance. The purpose of the study is to *help ensure that programming can be integrated into teaching about the spread of infection and vaccination.*

I have myself experienced the challenge of combining programming with natural science phenomena. Programming is a new subject area in the school, and thus there will be different programming skills among the teachers. Through this study, I want to contribute to making this programming work easier. Furthermore, the spread of infection and vaccination can largely affect society, and it is therefore important that pupils learn about this at school. The purpose of this study is to develop a teaching plan that uses micro:bit programming to simulate the spread of infection and vaccination as a limiting factor.

The method I used is inspired by educational design research (Bjørndal, 2013). The teaching plan was tested and improved in three sessions, once in the 8th grade and twice in the 10th grade with two different science teachers with different programming skills. My own observations and conversations with the teachers formed the basis for the data material that was used for further development of the teaching plan.

In the subsequent retrospective analysis, I found that it is important that students reflect on their actions also during programming work, both during the testing and during subsequent discussions. Since teachers and students will have different competences in a new subject area such as programming, it is also important to think about level differentiation in both the teacher's guide and the programming tasks. It is also important for developers to be open-minded about possible needs to change models during the process. A good model only comes into existence after continuous further development and improvement. The students, for their part, should be explained that models are a simplification of reality with the limitations it entails in the process.

FORORD

I skrivende stund gjøres nå den siste finpussen på denne masteroppgaven som markerer slutten på min fem år lange reise gjennom lærerutdanningen. Denne studietiden har gitt meg mye glede, nye venner og ny kunnskap. SALT-Revyen har vært en stor del av denne studietiden, og jeg ville ikke hatt det på noen annen måte.

Når jeg nå sitter og tenker tilbake fem måneder i tid hadde jeg aldri forestilt meg å ha en ferdig masteroppgave klar til innlevering. Disse fem månedene har bestått av lange dager og netter, med både opp- og nedturer. På tross av dette kan jeg nå med stolthet si at jeg endelig har nådd målstreken, men dette hadde jeg ikke klart på egenhånd.

Jeg vil med dette rette en stor takk til mine foreldre for inspirasjon, støtte, hjelp og praktisk bistand i innspurten, og en spesiell takk til min kjære Emil (teknologimester) for all teknisk bistand. Ikke minst vil jeg takke Alpha for all tålmodigheten han har vist i innspurten, og at han har gitt meg så mye glede og oppmuntring på sin måte underveis.

Videre vil jeg også takke lesesalen for å ha spredd glede i hverdagen og for å ha løftet meg opp når jeg har vært nede i bølgedalene. Også resten av masterklassen må få en takk for de sosiale sammenkomstene som har gjort disse fem månedene så mye bedre.

Det må også gis en takk til naturfaglærerne på 8. og 10. trinn som velvillig meldte seg til å teste ut opplegget, og som ga meg verdifulle bidrag til endringer av undervisningsopplegget.

Avslutningsvis vil jeg rette en stor takk til veileder Arne Stormo for hjelp til å sortere tankene innledningsvis, og videre konstruktive og detaljrike tilbakemeldinger underveis i prosessen.

Trondheim, mai 2024

Synne Christin Colling

INNHALDSFORTEGNELSE

Sammendrag	v
Abstract	vi
Forord	vii
Figurliste	xi
Tabelliste	xii
1 Innledning	1
1.1 Bakgrunn for studien	1
1.2 Formålet med oppgaven	2
1.3 Hva tilfører dette undervisningsopplegget?	2
1.4 Oppgavens struktur	3
2 Litteraturgjennomgang	4
2.1 Programmering i naturfag	4
2.1.1 Bruk av micro:bit	4
2.2 Elevers utfordringer med programmering	6
2.3 Didaktiske grep i programmeringsundervisning	7
2.3.1 PRIMM	8
2.3.2 Parsons problemer	9
2.3.3 Bruk av flytskjemaer	9
2.3.4 Parprogrammering	10
2.4 Bruk av modeller og simuleringer i naturfag	11
2.4.1 Deltakende simuleringer	12
2.5 Det naturfaglige fenomenet smittespredning og vaksinerings	12
2.6 Andre pedagogiske rammeverk	13
2.6.1 Sosiokulturell læringsteori	13
2.6.2 John Dewey: erfaring og tenkning	14
2.6.3 Kognitiv belastningsteori	14
2.6.4 Mestringsforventning	14
2.6.5 Tilpasset undervisning	14
3 Metode	15
3.1 Pedagogisk designforskning	15

3.2	Forskningssted og forskningsdeltakere	16
3.2.1	Elevenes og lærernes forkunnskaper	16
3.3	Min utviklingsprosess	16
3.3.1	Forberedelse av designeksperimentet	18
3.3.2	Gjennomføring av designeksperimentet	18
3.3.3	Retrospektiv analyse av designeksperimentet	19
3.4	Metoder for datainnsamling	19
3.5	Sortering av datamateriale.....	20
3.6	Forskningsetikk	21
4	Resultater og diskusjon	22
4.1	Presentasjon av undervisningsopplegget.....	22
4.2	Forberedelse av undervisningsopplegget.....	26
4.2.1	Utvikling av programmene	26
4.2.2	Utviklingen av oppgaveheftene	31
4.2.3	Utvikling av lærerveiledningen	34
4.3	Gjennomføring og videreutvikling av undervisningsopplegget	35
4.3.1	Gjennomføring og videreutvikling på 8. trinn	35
4.3.2	Gjennomføring og videreutvikling etter første klasse på 10. trinn	38
4.3.3	Gjennomføring og videreutvikling etter andre klasse på 10. trinn	44
4.4	Retrospektiv analyse	50
5	Avslutning og veien videre.....	53
6	Litteraturliste	54
7	Vedlegg.....	60
7.1	Vedlegg 1: Oppgaveheftet del 1: Radiokommunikasjon	61
7.2	Vedlegg 2: Oppgaveheftet del 2: Smittespredning	66
7.3	Vedlegg 3: Oppgaveheftet del 3: Vaksinerings	76
7.4	Vedlegg 4: Lærerveiledning.....	85
7.5	Vedlegg 5: Godkjent Sikt-søknad	127

FIGURLISTE

Figur 1: Bilde av en micro:bit	5
Figur 2: Eksempel på et program lagd i Microsoft MakeCode, med det Scratch-lignende blokkmiljøet. Programmet illustrerer hvordan micro:bit kan benyttes til å simulere terningkast som videre summeres sammen.	5
Figur 3: Til venstre er programmet som fungerer som en sender, der tallet 1 bli sendt. Til høyre er mottakeren som mottar tallet 1 og viser det på skjermen sin. Både senderen og mottakeren er satt til radiogruppe 5, så mottakeren vil oppfatte at signalet er adressert til seg.	6
Figur 4: Eksempel på et program der en terning kastes og summeres sammen. Ved å bruke knappene skal man både se summen og nullstille. Her vist til venstre med blokkprogrammeringsspråk og til høyre med tekstprogrammering i Python.	7
Figur 5: Fornorsket modell hentet fra PRIMMportal (u.å.) som viser til hvordan eierskapet av et program blir gradvis større gjennom de fem ulike stadiene i rammeverket.	8
Figur 6: Eksempel på et Parsons problem uten distraktorer. Disse klossene skal lage et program der en terning kastes for så å summeres sammen. Ved å bruke knappene skal man både kunne se summen av alle kastene og nullstille summen. Til venstre er Parsons problemer som elevene hadde fått tildelt, og til høyre er det puslet sammen.	9
Figur 7: Eksempel på et flytskjema, bygget med de tre mest sentrale komponentene ifølge Gjøvik og Høyland (2022).	10
Figur 8: Modell av utviklingsprosessen i denne studien inndelt i de tre fasene forberedelse, gjennomføring og retrospektiv analyse og tidslinjen gjennom prosessen	17
Figur 9: programmet elevene skal forutse og kjøre i oppgave 1 i oppgaveheftet del 1: Radiokommunikasjon. Gjennom programmet skal micro:bitene kunne kommunisere «Ja» til hverandre.	22
Figur 10: oppgaven elevene får utdelt i oppgave 2 i oppgaveheftet del 1: Radiokommunikasjon. Dette er et Parsons problem uten distraktorer hvor elevene skal pusle sammen klossene slik at micro:bitene kan kommunisere «Ja» og «Nei».	23
Figur 11: bilde av programmet som elevene skal forutse og kjøre i oppgave 1 i oppgaveheftet Del 2: smittespredning. Programmet fungerer som en nedtellingsfunksjon når tiden stilles.	24
Figur 12: programmet som skal benyttes for å simulere smittespredning, som elevene skal arbeide med ved hjelp av et Parsons problem og flytskjemaer.	24
Figur 13: Parsons problem som elevene får utdelt, der vaksinerings skal implementeres i programmet for smittespredning.	25
Figur 14: programmet med vaksine integrert før endringer etter samtale med veileder. 28	
Figur 15: programmet med vaksine integrert etter endringer fra samtale med veileder. 29	

Figur 16: Flytskjemaene som elevene får tildelt for å sette sammen Programmet for smittespredning. 33

Figur 17: spørsmål knyttet til programmeringen av oppgave 2 i oppgaveheftet del 2 som tar utgangspunkt i elevenes konseptuelle kunnskap. 47

TABELLISTE

Tabell 1: Oversikt over endringene gjort etter diskusjon med veileder om programmene til smittespredning og vaksineringsprogrammet. 30

Tabell 2: Oversikt over oppbygningen til de tre oppgaveheftene. 31

Tabell 3: Resultater og endringer etter gjennomføring av oppgaveheftet del 1: Radiokommunikasjon med 8. trinn. 36

Tabell 4: Resultater og endringer i lærerveiledningen etter gjennomføring med læreren på 8. trinn. 37

Tabell 5: Resultater og endringer som følge av uttesting av oppgaveheftet del 2: Smittespredning med den første 10. klassen. 40

Tabell 6: Resultater og endringer som følge av uttesting av oppgaveheftet del 3: Vaksineringsprogrammet med den første 10. klassen. 43

Tabell 7: Resultater og mulige endringer etter gjennomføring av oppgaveheftet Del 2: Smittespredning med den andre 10. klassen. 45

Tabell 8: Resultater og endringer etter gjennomføring av oppgaveheftet Del 3: Vaksineringsprogrammet med den andre 10. klassen. 48

Tabell 9: Resultater og endringer i lærerveiledningen etter gjennomføring med læreren på 10. trinn. 49

1 INNLEDNING

Vi lever i en stadig mer digitalisert verden, og den raske utviklingen av ny teknologi er mye av grunnen til dette. At det vil være behov for ny og forbedret teknologi i fremtiden kan man heller ikke se bort fra. Å utvikle god digital kompetanse vil derfor være avgjørende både for å kunne utvikle ny teknologi, men også for å kunne benytte seg av teknologien (Sevik, 2016). Denne kompetansen mener European Schoolnet (Balanskat & Engelhardt, 2015) at elevene kan få gjennom programmeringsarbeid. Ifølge opplæringsloven (1998, §1-1) skal opplæringen i skolen åpne dørene mot verden og fremtiden, og elevene skal få kunnskap, ferdigheter og holdninger til å kunne mestre eget liv og bidra i arbeidslivet. Dermed vil gode opplegg og målbevisst programmeringsundervisning i skolen være nødvendig for at elevene skal kunne møte et samfunn preget av en stadig mer digitalisert hverdag.

I tillegg til digitaliseringen er mennesker også sosiale vesener, der fysiske møter mellom mennesker er svært viktig for å ivareta et grunnleggende menneskelig behov. En risiko ved tett menneskelig interaksjon, er at individet blir utsatt for smitte som kan føre til sykdom. Innføring av vaksinasjonsprogrammer har hatt som hensikt å redusere eller fjerne risiko for alvorlig sykdom ved smitte (Furuseth et al., 2023). I Norge har eksempelvis barnevaksinasjonsprogrammet ført til at sykdommer som polio, røde hunder, difteri og meslinger nærmest ikke forekommer (Furuseth et al., 2023). Flere enn 9 av 10 barn følger barnevaksinasjonsprogrammet ifølge folkehelseinstituttet (Furuseth et al., 2023), som taler for at man i Norge kan føle seg trygge mot spredningen av disse sykdommene. World Health Organization (2023) skriver imidlertid at det var en økning i antall land med utbrudd av meslinger fra 22 land i 2021 til 37 land i 2022. I Europa ble det i 2023 registrert over 30.000 nye tilfeller av meslinger (European Centre for Disease Prevention and Control., 2024), en sykdom vi har hatt vaksine mot i Norge siden 1969 (Folkehelseinstituttet, 2022). Både i Sverige og Danmark ses det nå i 2024 enkelte tilfeller av meslinger (Vik, 2024). Også i Norge ble det 26. april 2024 påvist to nye tilfeller av meslinger (Hansen, 2024).

Meslinger er en av de mest smittsomme sykdommene vi kjenner til og er den mest alvorlige barnesykdommen med høy barnedødelighet (Folkehelseinstituttet, 2022). Det finnes heller ingen spesifikk behandling dersom man først blir smittet av sykdommen (Folkehelseinstituttet, 2022). Det er derfor viktig at elever, som potensielt kommende foreldre, lærer om smittespredning og vaksinerings som en begrensende faktor (Plutzer & Warner, 2021). Fordi undervisning i skolen er en av de største arenaene for kunnskapsdeling til barn, er dette en god arena for å gi læring på området (Plutzer & Warner, 2021). I denne studien benyttes programmering for å lære elevene om betydningen av smittespredning og vaksinerings.

1.1 BAKGRUNN FOR STUDIEN

Programmering ble innført i skolen gjennom fagfornyelsen (LK20). Naturfag fikk, sammen med kunst og håndverk, musikk og matematikk, ansvaret for programmeringsarbeidet i skolen (Saabye, 2019) Da programmering er et nytt fagområde i skolen, vil det være varierende kompetanse blant lærere (Stenlund, 2021). For flere lærere vil det å undervise i et fagområde der de selv mangler kompetanse kunne oppleves som krevende. Likeså

skal lærerne jobbe for at elevene oppnår kompetansemålene, og et av kompetansemålene i naturfag etter endt 10. trinn er « bruke programmering til å utforske naturfaglige fenomener » (Kunnskapsdepartementet, 2019, s.10). Det vil derfor kunne være et godt bidrag å utarbeide et undervisningsopplegg som bistår lærere med dette kompetansemålet. Mitt bidrag ønsker jeg å gi med bakgrunn i at også jeg ved starten av min programmeringsreise opplevde arbeidet med naturfaglige fenomener gjennom programmering som krevende. For å hjelpe lærerne i arbeidet med enda et kompetansemål etter endt 10. trinn, nemlig « utforske, forstå og lage teknologiske systemer som består av en sender og mottaker » (Kunnskapsdepartementet, 2019, s.10) valgte jeg å benytte programmering av micro:bit, en liten programmerbar mikroprosessor, som utgangspunkt for utviklingen av undervisningsopplegget.

Det å benytte vaksiner som det naturfaglige fenomenet er basert på viktigheten av å lære elevene om betydningen vaksiner har for å begrense smittespredning. Etter å ha gjennomført et søk over tilgjengelige undervisningsopplegg om vaksiner på ungdomstrinnet, erfarte jeg i stor grad at oppleggene var preget av bruk av film, animasjoner, tekst og oppgaver. Undervisningsopplegget fra Viten.no er et eksempel på dette (Erlien et al., 2021). Ved å kombinere programmering av micro:bit med vaksiner er ideen at elevene via micro:biten, « fysisk » skal kunne merke at de blir smittet og hvilken betydning vaksinen har for smittespredningen. Gjennom utviklingen av dette undervisningsopplegget vil også dette kompetansemålet for etter endt 10. trinn kunne arbeides med « beskrive kroppens immunforsvar og hvordan vaksiner virker, og gjøre rede for hva vaksiner betyr for folkehelsen » (Kunnskapsdepartementet, 2019, s.10).

1.2 FORMÅLET MED OPPGAVEN

Basert på viktigheten av å lære programmering for å øke digital kompetanse, samt å undervise elevene på ungdomstrinnet om vaksiner som beskyttelse mot smittespredning har jeg definert dette som mitt formål med studien:

Bidra til at programmering kan integreres i undervisningen om smittespredning og vaksiner.

De to delmålene for denne studien er som følger:

1. Utvikle et undervisningsopplegg som integrerer programmering av micro:bit med vaksiner og smittespredning.
2. Benytte observasjoner og tilbakemeldinger etter gjennomføring for å legge til rette for videreutvikling av undervisningsopplegget.

1.3 HVA TILFØRER DETTE UNDERVISNINGSOPPLEGGET?

Etter et omfattende søk har jeg funnet et par undervisningsopplegg som grenser mot min egen studie, nemlig bruk av micro:bit til å simulere smittespredning. Imidlertid finner jeg ingen undervisningsopplegg som benytter programmering av micro:bit som, i tillegg til å simulere smittespredning, også inkluderer vaksiner som en tilleggsfaktor avgjørende for smittespredningen.

Parsons (2021) har tidligere laget et undervisningsopplegg som simulerer smittespredning av Covid-19 ved hjelp av micro:bit. Programmet er lagt opp slik at hver enkelt elev gis anledning til å trykke på en knapp (A) opptil 10 ganger for å spre smitte, etter at de selv har blitt smittet. Etter det tiende trykket, "dør" eleven. Det er elevene selv som velger når de skal spre smitte ved å trykke på knappen. Lærer skal stille inn micro:bitene i forkant av studien. Elevene får med dette ikke innsikt i programmeringen av micro:biten. Studien har heller ingen vaksinerings som smittebegrensende faktor.

Et annet undervisningsopplegg som kombinerer smittespredning med programmering av micro:bit er publisert på Lær Kidsa Koding (Schaathun, u.å.), som er en database med læringsressurser for programmering. Dette opplegget er kalt «smitte:bit» og her er poenget at avsendere sender smitte til ulike mottakere. Mottakeren samler smitte og etter hvert som kroppen har nådd en viss mengde smitte, blir mottakeren selv en sender av smitte. Ved enda mer smittesamling blir «kroppen syk» og til slutt «dør kroppen» etter å ha mottatt nok smitte. Denne studien har heller ikke fokus på overlevelse, og det vises ikke hvordan vaksinerings potensielt kan påvirke dødsutfallet.

Med min studie har jeg vært opptatt av å vise at smitte er vilkårlig og det ikke skal være opp til den enkelte å beslutte når smitten skal spres. Derfor passet Parsons (2021) opplegg dårlig siden elevene i denne studien selv beslutter når de ønsker å sende smitte. I tillegg ønsket jeg å vise frem på hvilken måte vaksinerings kan påvirke smittespredning, noe ingen av de to andre studiene tok for seg. Videre blir samtlige i mitt program friske igjen etter smitte, noe som ofte stemmer bedre med virkeligheten.

Etter å ha vurdert de to andre undervisningsoppleggene, fant jeg at ingen av disse var dekkende i forhold til min målsetting i studien. Derfor vurderte jeg det som mest hensiktsmessig å utvikle min egen modell med fokus på å vise frem vilkårlig smittespredning der eleven er varig smittsom i løpet av sykdomsperioden. Videre viser min modell hvordan immunforsvaret arbeider mot smitte og i tillegg hvordan vaksinasjon fungerer som smittebegrenser. Mitt undervisningsopplegg ble utarbeidet på en slik måte at elevene aktivt tok del i selve programmeringen av micro:biten, da jeg ønsket å bidra med læring også innenfor programmeringsområdet.

1.4 OPPGAVENS STRUKTUR

Denne studien består av to komponenter, denne rapporten og det fullstendige undervisningsopplegget som følge av forberedelse og revidering. Det fulle opplegget kan ses i vedlegg 1 til 4. Selve rapporten beskriver innledningsvis valg av tema og bakgrunnen for dette som igjen utgjorde rammene for studien. Dette er beskrevet i innledningen, kapittel 1. Studien er basert på tidligere forskning og teori, og det aktuelle grunnlaget for utvikling av undervisningsopplegget er beskrevet i kapittel 2, litteraturgjennomgang. Metoden benyttet i denne studien og valgene tatt i forbindelse med dette, presenteres i kapittel 3, metode. Kapittel 4, resultat og diskusjon presenterer undervisningsopplegget samt utviklingsprosessen. Avslutning og veien videre, presentert i kapittel 5, er en oversikt over aktuelle funn i denne studien samt forslag til videre arbeid.

2 LITTERATURGJENNOMGANG

I dette kapitlet vil teorien og tidligere forskning benyttet under utviklingen av dette undervisningsopplegget presenteres. Det er fordelt i seks delkapitler: programmering i naturfag, elevers utfordringer med programmering, didaktiske grep i programmeringsundervisningen, bruk av modeller og simuleringer i naturfag, smittespredning og vaksinerings, og avslutningsvis andre pedagogiske rammeverk.

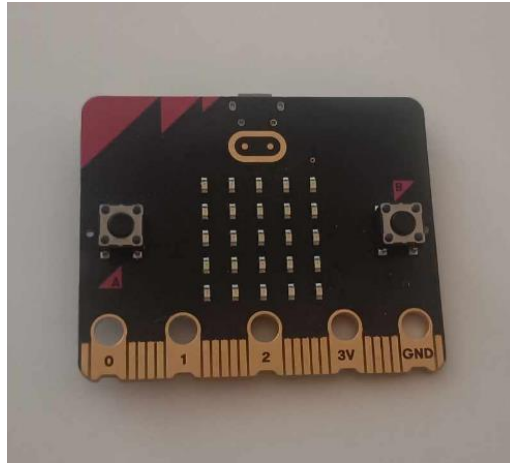
2.1 PROGRAMMERING I NATURFAG

Programmering har gjennom fagfornyelsen blitt en del av naturfagundervisningen, og senteret for IKT i utdanningen (Sevik, 2016, s.9) beskriver programmering som «prosessen fra å identifisere et problem og tenke ut mulige løsninger på problemet, til å skrive kode som kan forstås av en datamaskin, og å feilsøke og kontinuerlig forbedre denne koden».

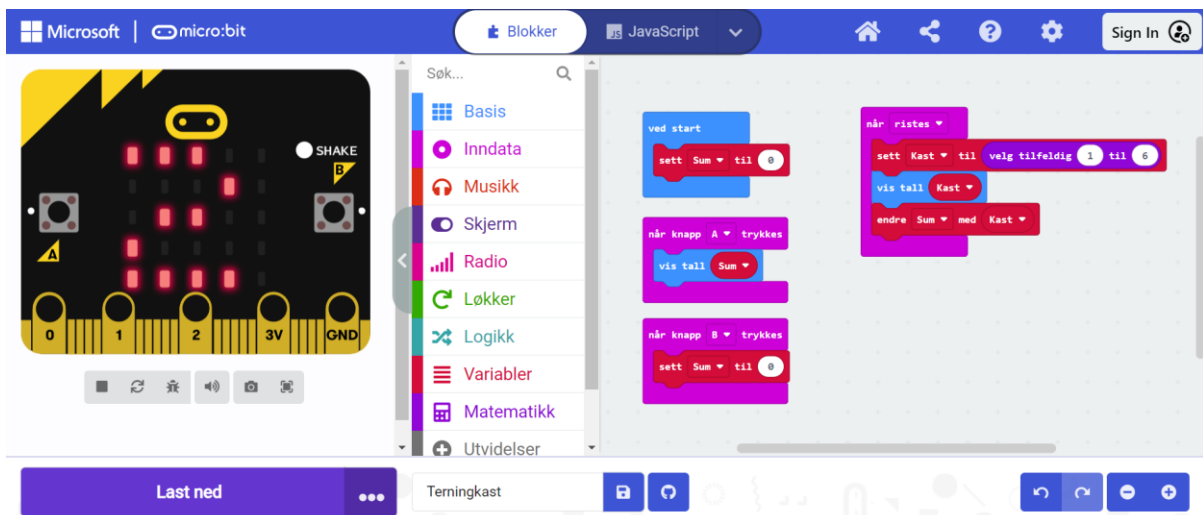
Mørch et al. (2019) fant gjennom sin studie at enkelte elever klarte å se en kobling mellom programmeringsarbeidet og andre skolefag. Videre har også Lai og Lai (2012) gjort funn som tilsier at 60% av elevene i en 5. klasse likte å bruke Scratch i naturfagundervisningen, og opplevde at det ga dem en bedre forståelse av naturfaglige fenomener. Caballero et al. (2014) observerte etter å ha gjennomført undervisningen der de kombinerer arbeidet med programmering og fysikk, at elevene opplevde denne kombinasjonen som krevende. Dette skyldtes at elevene syntes det var vanskelig å vite om utfordringene de møtte på skyldtes feil i programmeringsforståelsen eller deres faglige forståelse (Caballero et al., 2014).

2.1.1 BRUK AV MICRO:BIT

Micro:bit, som kan ses i figur 1 på neste side, er en liten programmerbar dataenhet med innebygd skjerm, programmerbare knapper, innebygd kompass, bevegelses-, temperatur- og lyssensorer, samt Bluetooth-tilkobling til andre micro:bit-er og enheter som mobil (Ball et al., 2016). Det er også mulig å koble micro:biten til andre eksterne sensorer og fysiske objekter (Ball et al., 2016). Micro:biten kan programmeres i flere ulike programmeringsmiljøer som Microsofts Touch Develop, JavaScript, MicroPython og et Scratch-lignende blokkmiljø (Sentance et al., 2017). Microsoft MakeCode kan benyttes til å programmere micro:biten, men den kan også programmeres i flere andre editorer deriblant Scratch (micro:bit, u.å.). I denne studien er det editoren Microsoft MakeCode som er benyttet. Figur 2 på neste side viser et eksempel på et program lagd i Microsoft MakeCode i det Scratch-lignende blokkmiljøet. Programmet viser hvordan micro:bit kan benyttes til å simulere terningkast der verdiene av kastene summeres sammen.



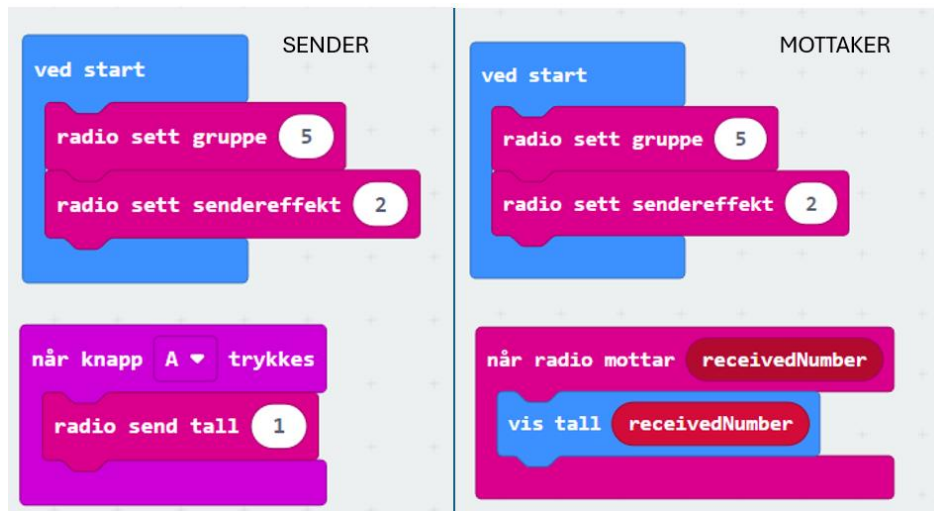
FIGUR 1: BILDE AV EN MICRO:BIT



FIGUR 2: EKSEMPEL PÅ ET PROGRAM LAGD I MICROSOFT MAKECODE, MED DET SCRATCH-LIGNENDE BLOKKMILJØET. PROGRAMMET ILLUSTRERER HVORDAN MICRO:BIT KAN BENYTTES TIL Å SIMULERE TERNINGKAST SOM VIDERE SUMMERES SAMMEN.

Micro:biten sin innebygde radio, som benytter Bluetooth protokollen, gjør at den kan benyttes til å sende og motta meldinger trådløst mellom micro:bitene (Rossing & Johansen, 2021). Denne kommunikasjonen kan foregå mellom to eller flere micro:bitere på en gang, hvilken eller hvilke micro:bitere som mottar meldingen bestemmes av hvilken radiogruppe micro:bitene er satt til (Rossing & Johansen, 2021). Når micro:biten mottar et radiosignal fra en sender vil den sjekke om meldingen er adressert til seg, altså om de er satt til samme radiogruppe (Rossing & Johansen, 2021). Slik som vist i figur 3 på neste side er både senderen og mottakeren satt til radiogruppe 5, som tilsier at dersom senderen trykker på knapp (A) vil mottakeren oppdage at meldingen er adressert til den, og vise tallet på sin skjerm. Senderen kan ses til venstre i figur 3 da den er programmert med programblokk *radio send tall*, mens mottakeren til høyre har *når radio mottar receivedNumber* (Rossing & Johansen, 2021). Micro:biten gir også mulighet for å sette

sendereffekten til micro:biten som avgjør hvor sterkt signal som sendes, der 7 er størst og 0 er minst (Rossing & Johansen, 2021).



FIGUR 3: TIL VENSTRE ER PROGRAMMET SOM FUNGERER SOM EN SENDER, DER TALLET 1 BLI SENDT. TIL HØYRE ER MOTTAKEREN SOM MOTTAR TALLET 1 OG VISER DET PÅ SKJERMEN SIN. BÅDE SENDEREN OG MOTTAKEREN ER SATT TIL RADIOGRUPPE 5, SÅ MOTTAKEREN VIL OPPFATTE AT SIGNALET ER ADRESSERT TIL SEG.

Sentance et al. (2017) oppsummerer fordelene ved bruk av fysiske programmeringsobjekter i fire hovedpunkter: motivasjon, håndfasthet, samarbeid og kreativitet. Det er nettopp denne fysiske håndfastheten som gjør at elevene kan ta, kjenne og føle på enheten, samt fysisk se hva som skjer og ikke kun bruke en dataskjerm (Sentance et al., 2017). Sentance et al. (2017) gjorde også funn som tilsa at elevene opplevde det å benytte micro:biten som enkelt, som igjen vil kunne gjøres arbeidet med micro:biten mer motiverende.

2.2 ELEVERS UTFORDRINGER MED PROGRAMMERING

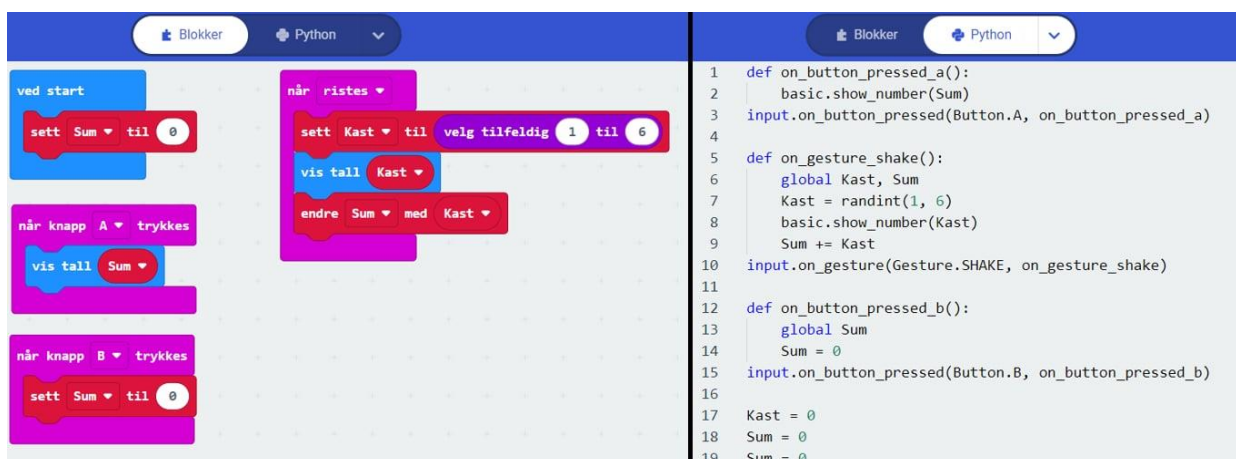
At programmering kan være utfordrende for elever er noe mange lærere har forståelse for (Sentance & Waite, 2017). Da programmering stiller store kognitive krav til elevene vil det være avgjørende å ta i betraktning at programmeringsarbeidet kan føre til kognitive overbelastningen (Berssanette & de Francisco, 2022).

Qian og Lehman (2017) presenterer flere utfordringer innen arbeid med programmering, og tiltak som kan hjelpe. Disse utfordringene knyttes i hovedsak til tre ulike typer kunnskap: syntaktisk, konseptuell og strategisk kunnskap (Qian & Lehman, 2017). Den syntaktiske kunnskapen bygger på forståelsen om hvordan programmeringsspråket er bygd opp og hvilke regler som må følges, som for eksempel bruk av innrykk, parenteser og semikolon i tekstbasert programmering (Qian & Lehman, 2017). Videre er den konseptuelle kunnskapen, forståelsen om hvordan ulike programmeringsprinsipper fungerer slik som løkker, variabler og vilkår (Qian & Lehman, 2017). Den strategiske

kunnskapen handler om å kunne benytte den konseptuelle- og syntaktiske kunnskapen til å kunne løse oppgaver (Qian & Lehman, 2017).

En strategi Qian og Lehman (2017) presenterer for å håndtere utfordringer blant nybegynnere innen programmering er bruk av gode, instruerende eksempeloppgaver. Fleury (1991) forteller at elever ofte kan ta med kunnskap fra disse eksemplene videre inn i arbeidet med andre oppgaver. Å jobbe med gjennomarbeidede eksempler ved å lese og følge programmet, samt å forklare dem, vil også bidra til å utvikle den strategiske kunnskapen innen programmering (Qian & Lehman, 2017).

En annen strategi er ifølge Qian og Lehman (2017) å fremheve eller fjerne vanlige syntaksfeil i programmeringen. Kelleher og Pausch (2005) forteller at bruk av programmeringsmiljø som kan fremheve syntaksfeil gjør det enklere for elever å identifisere og løse problemene. Videre kan blokkprogrammeringsspråk som Scratch fjerne disse problemene helt ved hjelp av forhåndsprogrammert grafikk (Resnick et al., 2009). Her kommer syntaksen frem i formen til selve klossene og hvordan de passer sammen, som gir en grafisk fremstilling av syntaksen. Figur 4 viser en sammenligning mellom blokkprogrammering og tekstprogrammering, der begge programmene vil gjøre det samme. Eksempelvis er syntaksbruken med kolon på slutten av enkelte linjer fjernet i blokkprogrammeringen.



FIGUR 4: EKSEMPEL PÅ ET PROGRAM DER EN TERNING KASTES OG SUMMERES SAMMEN. VED Å BRUKE KNAPPENE SKAL MAN BÅDE SE SUMMEN OG NULLSTILLE. HER VIST TIL VENSTRE MED BLOKKPROGRAMMERINGSSPRÅK OG TIL HØYRE MED TEKSTPROGRAMMERING I PYTHON.

2.3 DIDAKTISKE GREP I PROGRAMMERINGSUNDERVISNING

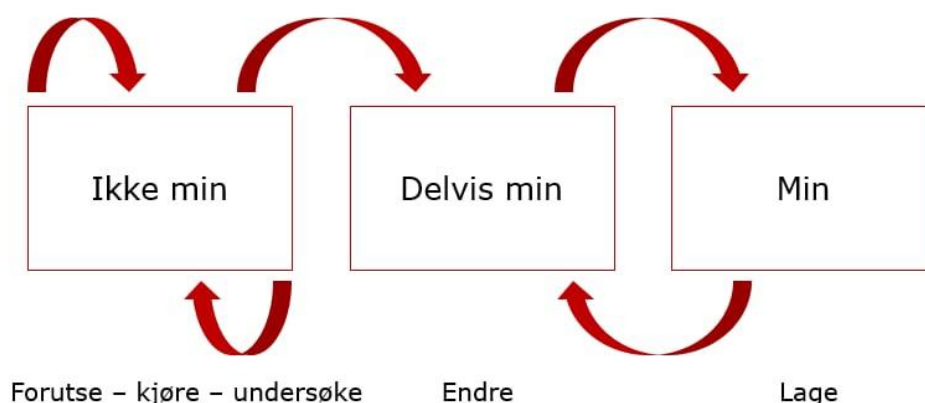
Det finnes flere didaktiske grep som kan benyttes i programmeringsundervisningen. De grepene som benyttes i denne studien er PRIMM (Sentance et al., 2019; Sentance & Waite, 2017), Parsons problem (Parsons & Haden, 2006), flytskjema (Charntaweekhun & Wangsiripitak, 2006) og parprogrammering (Denner et al, 2014).

2.3.1 PRIMM

PRIMM er et pedagogisk rammeverk basert på hvordan man lærer programmering, og med utgangspunkt i synet til Vygotsky rettet mot sosiokulturell læringsteori (Sentance et al., 2019). PRIMM bygger på en gradvis kompleksitet innenfor programmeringsarbeidet, og rammeverket er et akronym der de ulike begrepene har betydning som forklart i punktlisten under (Sentance et al., 2019; Sentance & Waite, 2017):

- **Predict (Forutse):** Elevene får her tildelt et program der de skal forutse hva programmet kommer til å gjøre uten å kjøre det. Her kan elevene gjerne skrive eller tegne hva de tror kommer til å skje.
- **Run (Kjøre):** I dette steget skal elevene kjøre programmet de fikk tildelt i forrige steg, for å se om antagelsen de gjorde stemte. Deretter skal antagelsene opp mot det som faktisk skjedde diskuteres.
- **Investigate (Undersøke):** Elevene skal på dette stadiet fremdeles arbeide med et tildelt program, men undersøke ulike elementer ved programmet, som for eksempel feilsøking og endring av variabler.
- **Modify (Endre):** På dette stadiet skal elevene endre et tildelt program. Elevene vil da kunne få ulike versjoner av programmet, og de vil oppleve et større eierskap til programmet.
- **Make (Lage):** På dette stadiet skal elevene lage sitt eget program, og de vil dermed få fullstendig eierskap til programmet. Her kan de benytte deler av programmene fra tidligere oppgaver.

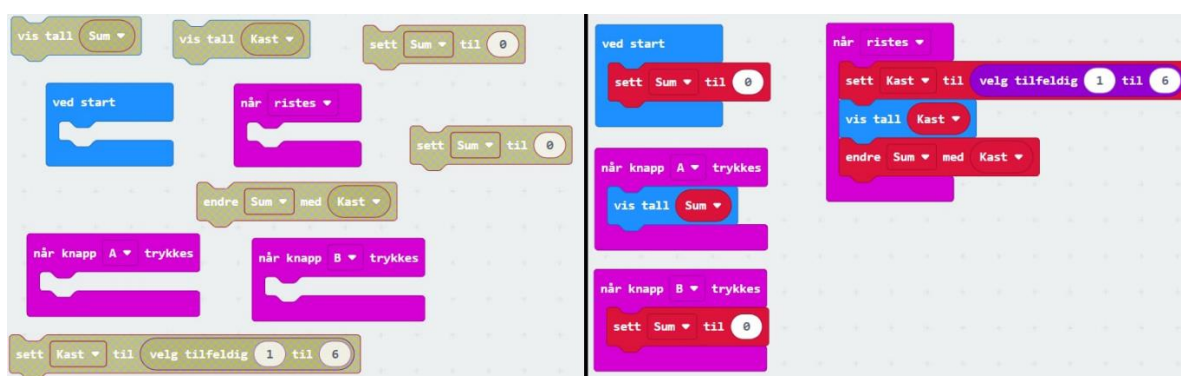
Alle fem stadier trenger ikke å benyttes hver gang, og de tidligere stadiene kan gjentas flere ganger før det arbeides videre med i de senere stadiene (PRIMMportal, u.å.). Figur 5 viser en modell av hvordan elevene gradvis får mer eierskap til et program gjennom de ulike stegene i PRIMM.



FIGUR 5: FORNORSKET MODELL HENTET FRA PRIMMPORTAL (U.Å.) SOM VISER TIL HVORDAN EIERSKAPET AV ET PROGRAM BLIR GRADVIS STØRRE GJENNOM DE FEM ULIKE STADIENE I RAMMEVERKET.

2.3.2 PARSONS PROBLEMER

Parsons problemer fungerer som et puslespill der fragmenter av et program skal pusles sammen til et fullstendig program (Parsons & Haden, 2006). Disse kan lages både med eller uten distraktorer, som er klosser som ikke benyttes i den endelige løsningen av programmet (Ericson et al., 2017). Figur 6 viser et eksempel på et Parsons problem uten distraktorer da alle klossene til venstre i figuren blir benyttet for å lage de ferdige programmet til høyre i figuren. Harms et al. (2016) fant i sin studie at Parsons problemer uten distraktorer er enklere å løse, og argumenterer for hvordan bruk av distraktorer kan påvirke motivasjonen for videre arbeid med programmering. Parsons problemer med færre klosser vil også være enklere å løse, da det er færre alternative løsninger for plassering av klossene (Ericson et al., 2017).



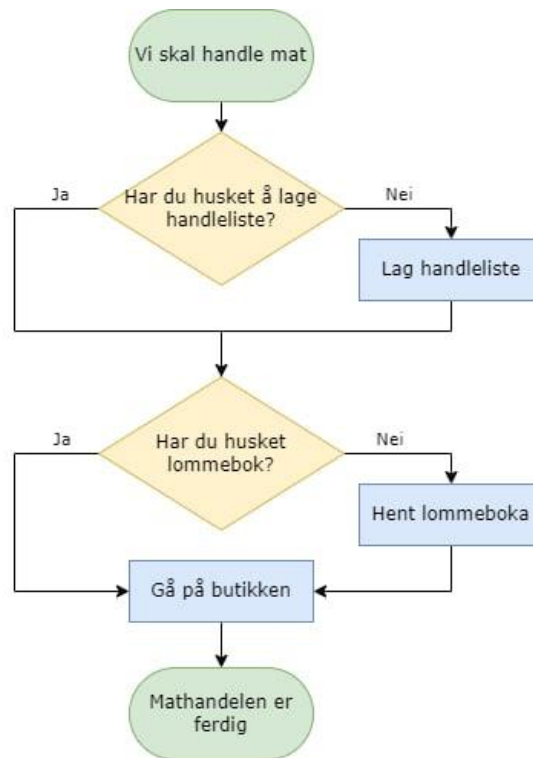
FIGUR 6: EKSEMPEL PÅ ET PARSONS PROBLEM UTEN DISTRAKTORER. DISSE KLOSSENE SKAL LAGE ET PROGRAM DER EN TERNING KASTES FOR SÅ Å SUMMERES SAMMEN. VED Å BRUKE KNAPPENE SKAL MAN BÅDE KUNNE SE SUMMEN AV ALLE KASTENE OG NULLSTILLE SUMMEN. TIL VENSTRE ER PARSONS PROBLEMER SOM ELEVENE HADDE FÅTT TILDELTE, OG TIL HØYRE ER DET PUSLET SAMMEN.

Ericson et al. (2017) og Zhi et al. (2019) fant begge gjennom sine studier at de som løste Parsons problemer brukte kortere tid enn de som skrev programmer fra bunnen av. Sistnevnte argumenterer for at årsaken til dette er at elevene bruker mindre tid på å lete etter klossene de skal benytte, teste ut klosser som ikke blir benyttet i det endelige programmet, og endre inputen til de ulike variablene (Zhi et al., 2019). Parson og Haden (2006) argumenterer også for at denne formen for problemløsning vil være mer engasjerende for elevene. Videre argumenterer Zhi et al. (2019) for at elevenes kognitive belastning vil være mindre ved løsning av Parsons problemer da elevene har færre klosser å forholde seg til. Både Ericson et al. (2017) og Zhi et al., (2019) fant i sine studier at elevene som løste Parsons problemer ikke hadde lavere læringsutbytte enn elevene som skrev programmer fra bunnen av.

2.3.3 BRUK AV FLYTSKJEMAER

Et flytskjema benyttes for å vise rekkefølgen og valgene som utføres i et program, der pilene i skjemaet følges stegvis (Charntaweekhun & Wangsiripitak, 2006). Når man kommer til symbolet som signaliserer at det skal tas en avgjørelse følges pilen som

stemmer overens med avgjørelsen som tas (Charntaweekhun & Wangsiripitak, 2006). Gjøvik og Høyland (2022) skriver at flytdiagram kan være omfattende, men at de pleier å ha en start- og sluttmarkør, rektangler som tilsier at noe skal skje i programmet og en rombe som benyttes for å stille spørsmål der det må tas en avgjørelse for hva som skal skje videre. Figur 7 viser et flytskjema som kun benytter disse tre sentrale komponentene.



FIGUR 7: EKSEMPEL PÅ ET FLYTSKJEMA, BYGGET MED DE TRE MEST SENTRALE KOMPONENTENE IFØLGE GJØVIK OG HØYLAND (2022).

Gjennom sin studie legger Hooshyar et al. (2015) frem et forslag om at flytskjemaer kan være betydningsfullt for å utvikle programmeringskompetanse, da det gir en visuell fremstilling av en algoritme. En algoritme er en bestemt rekkefølge av operasjoner for å oppnå det ønskede resultatet (Haraldsrud et al., 2020). Ved bruk av flytskjema kan nybegynnere enkelt lage en fremstilling av sin løsning på et problem (Charntaweekhun & Wangsiripitak, 2006; Hooshyar et al., 2015). Flytskjemaene kan også hjelpe nybegynnere med å få en forklaring på hvordan de kan skrive et program og for å skape en forståelse av hvordan et program er bygd opp (Charntaweekhun & Wangsiripitak, 2006).

2.3.4 PARPROGRAMMERING

Parprogrammering foregår ved at to personer samarbeider om et program ved å kun benytte én datamaskin (Denner et al., 2014). Begge skal delta aktivt i prosessen, men én person har ansvaret for å styre mus og tastatur og den andre bidrar med vurderinger og meninger rettet mot programmet (Williams & Kessler, 2000).

Hanks et al. (2011) har gjennomført en litteraturstudie omhandlende bruk av parprogrammering i nybegynnerundervisning på bachelornivå, og har funnet flere fordeler ved å benytte parprogrammering. Ved bruk av parprogrammering oppnår studentene bedre suksess innenfor programmeringsarbeidet, og kvaliteten på programmene blir bedre (Hanks et al., 2011). Videre blir også studentene mer selvsikre, de opplever samarbeidet som hyggeligere og mener at parprogrammering fører til bedre læring (Hanks et al., 2011). Observasjonene gjort av læringsutbyttet tyder også på at studentene oppnår en dypere forståelse gjennom parprogrammering (Hanks et al., 2011). Forskningen tyder også på at parprogrammering fører til at studentene vil fortsette med programmering, samt at det ikke vil ha noen negative konsekvenser når de senere skal arbeide med programmeringen på egenhånd (Hanks et al., 2011).

Denner et al. (2014) har gjort en studie angående bruk av parprogrammering blant elever mellom 10 og 14 år, og har kommet frem til flere av de samme funnene som Hanks et al. (2011). Et av disse er at elevene som arbeidet i par fikk et bedre læringsutbytte enn de som arbeidet alene og at elevene ble mer selvsikre i programmeringsarbeidet (Denner et al., 2014). Denner et al. (2014) fant også i sin studie at begge partnere hadde fordel av samarbeidet uavhengig av nivå, men at utbyttet ikke nødvendigvis var likt. Cao og Xu (2005) og Williams et al. (2008) fant derimot begge gjennom sine studier at studentene selv ønsket å arbeide med en motpart som er på likt eller høyere nivå enn dem selv.

2.4 BRUK AV MODELLER OG SIMULERINGER I NATURFAG

Modeller er mye benyttet i naturfagsklasserommet, både gjennom ferdiglagde modeller som bidrar med å skape forståelse av det naturfaglige fenomenet og ved at elevene selv lager modeller for å fremstille sine oppfatninger av fenomenet (Treagust et al., 2002). Modeller er en forenkling av virkeligheten, lagd for å skape forståelse av et naturfaglig fenomen (Gilbert et al., 2000). Schwartz et al. (2009) definerer en vitenskapelig modell som en forenkling av virkeligheten som fokuserer på hovedelementer som søker etter å forklare et vitenskapelig fenomen. Nia og de Vries (2017) skriver at modellene lages på en spesifikk måte ut fra hva målet med modellen er, altså hva modellen skal fremstille. Basert på naturen til modeller vil det aldri være mulig å unngå at de forenkles, og dette kan gjøres ved å utelate elementer fra den virkelige verden eller ved å gjøre små endringer som bidrar til å forenkle det naturfaglige fenomenet (Nia & de Vries, 2017). Nia og de Vries (2017) skriver at det bak en god forenklet modell ligger en gjentagende videreutvikling og kontinuerlig forbedring.

I arbeidet med vitenskapelige modeller er det viktig at elevene mestrer å kunne lage egne modeller, anvende eksisterende modeller, og evaluere og revidere modellene (Schwarz et al., 2009). Også Hallström & Schönborn (2019) poengterer viktigheten av at elevene har ferdigheten som gjør dem i stand til å vurdere modellens omfang og begrensninger.

En simulering er en interaktiv modell der brukeren kan endre variabler for å kunne se ulike scenarioer av en prosess (Peffer et al., 2016). Simuleringene kan benyttes til å vise hvordan et eksisterende fenomen fungerer i den virkelige verden, og for å teste ut mulige prediksjoner på hypotetiske utfall (Peffer et al., 2016). Ved å benytte simuleringer kan elevene se konsekvensen av en handling uten at de må oppleve konsekvensen i det virkelige liv (Dieleman & Huisingh, 2006).

2.4.1 DELTAKENDE SIMULERINGER

Når simuleringen ikke gjennomføres på dataskjermen, men gjennomføres i virkeligheten kalles det for en deltakende simulering (Colella, 2000). Undervisningen kan ved bruk av deltakende simuleringer gjøres mer interaktiv og realistisk (Parsons, 2021). Elevene blir selv en aktiv deltaker i simuleringen, og vil derfor bli påvirket av den og har selv interaksjoner med de andre deltakerne (Colella, 2000). Dette bygger på Dewey (2001) sin tanke om læring gjennom erfaring. Forskning viser også til den positive effekten deltakende simuleringer kan ha på engasjementet, motivasjonen, problemløsning og den selvstyrte læringen til elevene (Klopfer et al., 2005).

2.5 DET NATURFAGLIGE FENOMENET SMITTESPREDNING OG VAKSINERING

Målet med denne studien er å kunne bidra til at programmering integreres i undervisningen om smittespredning og vaksinerings. Da dette skal benyttes på ungdomstrinnet er teori angående smittespredning og vaksinerings benyttet i utviklingen av dette undervisningsopplegget hentet fra læreboka *Naturfag 10* (Steiniger & Wahl, 2021) og *Biologi for lærere* (Grindeland et al., 2020). Dette fordi disse bøkene er representative for hvilket nivå elevene skal lære om fenomenene på. Immunforsvaret spiller også en stor rolle når det kommer til smittespredning og vaksinerings, men da immunforsvaret er et stort og omfattende fenomen, vil kun det som vurderes essensielt for utviklingen av undervisningsopplegget i denne studien bli presentert. Dette innebærer blant annet at ikke alle celletypene knyttet til det indre immunforsvaret presenteres, samt hvordan de fungerer og dannes. Dersom dette er av interesse, kan det leses mer om på side 426 til 430 i Grindeland et al. (2020).

Mennesker utvikler sykdom når mikroorganismene kommer seg inn i kroppen og begynner å formere seg (Grindeland et al., 2020). Smittestoffene som kan gjøre oss syke kan være virus, bakterier, sopp og noen protister (Grindeland et al., 2020). Disse har tilpasset seg flere ulike spredningsmåter, som å spres via luft, via dråpe og ved fuktighet (Grindeland et al., 2020). Mange av de vanligste sykdommene spres via dråpesmitte, der dråpen som spres er infisert med smittestoffet (Grindeland et al., 2020). Disse bittesmå dråpene kan spres i lufta når mennesker hoster og nyser (Steiniger & Wahl, 2021), og kan overføres via fysiske gjenstander eller gjennom mat og drikke (Grindeland et al., 2020).

Immunforsvaret beskytter oss mot uvedkommende inntrengere, og består av det ytre uspesifikke forsvaret, det indre uspesifikke forsvaret og det indre spesifikke forsvaret (Grindeland et al., 2020). Det ytre forsvaret er vår første beskyttelse, og består av huden og slimhinner som, så lenge det er intakt, vil holde mange mikroorganismer ute av kroppen (Grindeland et al., 2020). Dersom mikroorganismer trenger forbi kroppens ytre barriere møter de det indre immunforsvaret, som blant annet består av hvite blodceller med ulike roller (Grindeland et al., 2020). Det indre uspesifikke immunforsvaret er det første inntrengerne møter som består av blant annet fagocytter, også kalt spiseceller, som spiser bakterier frem til de dør som følge av giftstoffene som hopper opp i cellen (Grindeland et al., 2020). Det indre spesifikke immunforsvaret slår muligens senere inn, og består av lymfocytter (Grindeland et al., 2020). Immunforsvaret må skille inntrengerne fra kroppens egne celler, dette skjer blant annet med dannelsen av antistoffer (Grindeland et al., 2020). Antistoffer, karbohydratholdige proteiner, skiller friske og syke celler samt inntrengere fra

hverandre (Grindeland et al., 2020). Antistoffet skilles ut i blodet, og det dannes hukommelsesceller fra en B-lymfocyt for at antigenet huskes, og gjør at kroppen bekjemper inntrengeren raskere hvis den kommer inn i kroppen på nytt (Grindeland et al., 2020).

Vaksiner blir tatt for å oppnå immunitet uten at man må bli syk (Grindeland et al., 2020). Dette kan gjøres ved at sykdomsfremkallende mikroorganismer som er svekkede eller døde sprøytes inn i kroppen (Grindeland et al., 2020). Noen av vaksinene kan også gis i drikkeform, som nesenspray eller i tablettform (Steiniger & Wahl, 2021). Når disse kommer inn i kroppen startes en immunrespons der antistoffer og enkelte hukommelsesceller dannes (Grindeland et al., 2020). Neste gang immunforsvaret oppdager disse mikroorganismene vil det komme raskere i gang slik at personen unngår å bli syk (Steiniger & Wahl, 2021). Varigheten til immuniteten etter tatt vaksine vil variere (Grindeland et al., 2020; Steiniger & Wahl, 2021). Blant de vaksinerte vil ikke vaksinen være 100% effektiv (Europeisk informasjonsportal om vaksinasjon, 2020).

2.6 ANDRE PEDAGOGISKE RAMMEVERK

For å kunne utvikle et undervisningsopplegg må det som kan påvirke og hjelpe til læring tas i betraktning. I denne delen vil sosiokulturell læringsteori, John Dewey: erfaring og tenkning, kognitiv belastningsteori, mestringsforventning og tilpasset opplæring presenteres.

2.6.1 SOSIOKULTURELL LÆRINGSTEORI

Den sosiokulturelle læringsteorien tar i betraktning hvilken betydning de sosiale og kulturelle interaksjonene mellom mennesker har for læring og utvikling, og Lev Vygotsky sitt arbeid står sentralt i utviklingen av denne teorien (Mahn, 1999). Vygotsky (1978) skriver at barnets språk spiller en viktig rolle når de skal utføre en oppgave. Språket fungerer som et verktøy for å løse vanskelige oppgaver, overvinne impulsive handlinger, planlegge løsninger på oppgaver før utførelse og mestre egen adferd (Vygotsky, 1978). Gjennom språket kan barnet vise hva de tenker og hvordan de oppfatter verden (Vygotsky, 1978).

Vygotsky (1978) presenterer også den nærmeste utviklingssonen. Denne utviklingssonen ligger mellom det aktuelle og potensielle utviklingsnivået hos en person (Vygotsky, 1978). Det aktuelle utviklingsnivået er det eleven klarer av selvstendig problemløsning, mens det potensielle utviklingsnivået er det eleven har mulighet til å løse i samarbeid med voksne eller andre som kan mer enn dem selv (Vygotsky, 1978). Vygotsky (1978) foreslår videre at læringen som skjer gjennom samarbeid i den nærmeste utviklingssonen i dag, vil neste dag være internalisert i det aktuelle utviklingsnivået.

2.6.2 JOHN DEWEY: ERFARING OG TENKNING

John Dewey (2001) skriver at læring skjer gjennom erfaring, men at aktiviteten alene ikke er nok. Det er først når erfaringen kombineres med refleksjon og tanker at den fører til læring (Dewey, 2001). Gjennom tanker og refleksjon vil man kunne danne en forbindelse mellom aktiviteten og konsekvensen den førte til (Dewey, 2001). Denne tankeprosessen vil igangsettes når personen ønsker å bestemme betydningen av en handling som skal eller har blitt gjennomført (Dewey, 2001).

2.6.3 KOGNITIV BELASTNINGSTEORI

Sweller et al. (1998) presenterer at informasjon lagres i langtidsmminnet som skjemaer, der informasjonen er kategorisert. For å konstruere skjemaene må informasjonen manipuleres og organiseres, og dette skjer i arbeidsminnet (Sweller et al., 1998). Arbeidsminnet er bare i stand til å håndtere enn viss mengde informasjon om gangen, og dette er mest sannsynlig kun to til tre ulike elementer av informasjon om gangen (Sweller et al., 1998). Det er den begrensede prosesseringsevnen til arbeidsminnet som er grunntanken bak den kognitive belastningsteorien, og som burde tas i betraktning under informasjonsformidling da arbeidsminnet kan bli overbelastet (Sweller et al., 1998).

2.6.4 MESTRINGSFORVENTNING

Bandura (1994) skriver at mestringsforventning er definert som menneskets tro på egne evner til å mestre en oppgave. Mestringsforventningen bestemmer hvordan man håndterer nederlag, hvor mye innsats man gir, og hvor lenge man holder ut gjennom motgang og hindring (Bandura, 1977). Dersom man erfarer dårlige opplevelser eller utfall, vil man da få en lavere mestringsforventning, mens større mestringsforventning vil gjøre at man takler motgang bedre (Bandura, 1977). Det er fire hovedelementer som kan påvirke mestringsforventningen (Bandura, 1994). Der den mest effektive er å selv mestre oppgaven, men også det å se andre på likt nivå som deg selv mestre en oppgave kan øke din mestringsforventning (Bandura, 1994). Videre vil også verbale oppmuntringer fra andre, samt å begrense personens stress og negative følelser knyttet til oppgaven kunne øke personens mestringsforventning (Bandura, 1994).

2.6.5 TILPASSET UNDERVISNING

Som det står i opplæringsloven (2008, §1-3) har alle elever krav på at opplæringen skal tilpasses deres evner og forutsetninger. Det finnes flere mulige didaktiske grep for å tilpasse undervisningen, og noen av disse er nevnt i delkapittel 2.3. Haraldsrud et al. (2020) skriver at de store forskjellene mellom elevene innen programmering mest sannsynlig ikke er å finne så kraftig i andre fagdisipliner. De anbefaler at de svakere elevene burde få hint tilkoblet oppgavene eller andre forenklinger, mens for de sterkeste elevene burde det gis oppgaver som tar lenger tid, er mer kompliserte, eller der de er avhengig av å søke på nett for å videreutvikle programmet (Haraldsrud et al., 2020).

3 METODE

Målet med denne studien er å kunne bidra til at programmering kan integreres i undervisningen om smittespredning og vaksinerings. Metoden benyttet i denne studien bygger på min tolkning av pedagogisk designforskning som blir redegjort for i Bjørndal (2013), van den Akker et al. (2006) og Gravemeijer og Cobb (2006). Metodekapittelet vil inneholde pedagogisk designforskning som metode, forskningssted og forskningsdeltakere, min utviklingsprosess, metoder for datainnsamling, samt sortering av dataen. Avslutningsvis vil forskningsetikken tilknyttet denne studien presenteres. Jeg vil opplyse om at deler av dette kapittelet er hentet fra en egenskrevet eksamensoppgave i MGLU5208 vitenskapsteori og metode skrevet høsten 2023, slik det tillates i sensurveiledningen for denne masteroppgaven.

3.1 PEDAGOGISK DESIGNFORSKNING

Pedagogisk designforskning øker forskningens relevans da den har en sterkere tilkobling til undervisningspraksisen, sett opp mot tidligere klasseromsforskning som har fått kritikk for å ha en svak tilkobling til praksis (van den Akker et al., 2006). Hovedtrekkene til pedagogisk designforskning går ut på å utvikle, deretter teste og justere i flere omganger, for avslutningsvis å evaluere undervisningsopplegg for å bedre undervisningen (van den Akker et al., 2006). Gjennom den sykliske prosessen er målet å danne ny forståelse og forbedring (van den Akker et al., 2006).

Ideen bak denne type forskning er å designe noe som kan ha betydning i den virkelige verden, samtidig som den skal være brukervennlig (van den Akker et al., 2006). Denne forskningen bør bidra til å skape et nyskapende undervisningsopplegg, og tilføre noe som ikke allerede eksisterer (Bjørndal, 2013). Når man gjennomfører prosessen, kan den også motbevise eventuelle antakelser, og nye kan dannes. Dette vil igjen kunne føre til ny videreutvikling og testing (Bjørndal, 2013). Funnene fra prosessen vil også bidra til å gi informasjon til kommende design (Bjørndal, 2013).

Det som også kjennetegner denne typen forskning er at ny kunnskap vokser frem som et samarbeid mellom forsker/lærer/elev i det naturlige klasserommiljøet der nye undervisningsmetoder testes ut (Bjørndal, 2013). Samarbeidet mellom forskeren og lærer/elev er avgjørende for å få gjennomført en god prosess og skape en felles forståelse av hva som leder til forbedringer (Bjørndal, 2013). Et av hovedpoengene med denne type forskning, er at alle parter skal gis anledning til å bidra i prosessen med sine meninger og refleksjoner (Bjørndal, 2013). Basert på dette samarbeidet vil denne studien ha et sosialkonstruktivistisk vitenskapelig ståsted. Dette tilsier altså at mennesker i sosiale settinger konstruerer kunnskap, i samråd med andre (Tjora, 2021).

Utviklingsprosessen i denne studien er basert på Gravemeijer og Cobb (2006) sine tre faser: forberedelse av designeksperimentet, gjennomføring av designeksperimentet og retrospektiv analyse. Hva som ligger i disse tre fasene, vil presenteres i delkapittel 3.3.

3.2 FORSKNINGSSTED OG FORSKNINGSDELTAKERE

Denne studien ble gjennomført ved to ulike skoler, med to ulike lærere og tre ulike klasser. Den ene læreren hadde en 8. klasse som gjennomførte del 1 av undervisningsopplegget. Lærer nummer to hadde to 10. klasser som gjennomførte del 2 og del 3 av undervisningsopplegget. Utvalgsriteriene var at læreren måtte være naturfagslærer, jobbe på ungdomsskole og ha en eller flere klasser i naturfag. For å finne aktuelle lærere ble det sendt en e-post til ungdomsskoler i nærområdet. E-posten ble sendt til både rektor og avdelingsleder. I e-posten var det informasjon om hensikten med studien, samt hva det ville si å bidra. Lærerne meldte selv sin interesse til å bidra i forskningen.

3.2.1 ELEVENES OG LÆRERNES FORKUNNSKAPER

Resultatene samlet inn under gjennomføring av undervisningsopplegget vil bli påvirket av elevenes og lærernes forkunnskaper, og det vil derfor være viktig å ta disse forkunnskapene i betraktning.

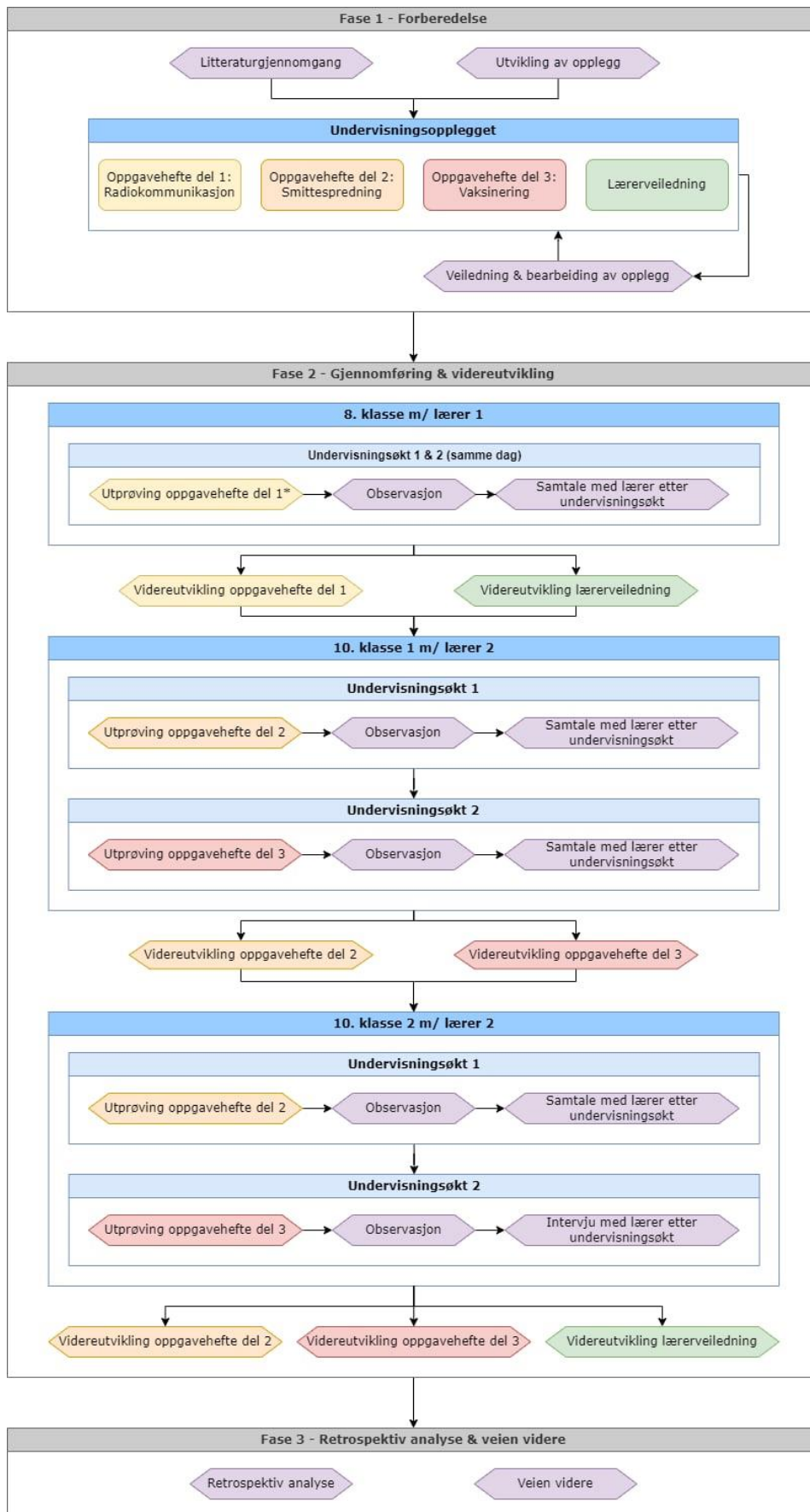
Elevene på 8. trinn hadde ikke tidligere arbeidet med programmering på generell basis i naturfagundervisningen på ungdomsskolen. Læreren hadde heller ikke kjennskap til hvor mye elevene hadde arbeidet med programmering tidligere, men visste at enkelte elever hadde programmering som valgfag. Læreren visste dermed at enkelte elever hadde mer kompetanse på området. Elevene på 8. trinn hadde heller ikke hatt noe undervisning om immunforsvaret, smittespredning eller vaksinerings.

Alle elevene i de to 10. klassene hadde nylig arbeidet med programmering, også da programmering av micro:bit, i naturfagundervisningen. Alle elevene på dette trinnet skulle dermed ha kjennskap til hvordan micro:biten fungerer og hvordan den programmeres. Noen få elever i den ene klassen hadde også programmering som valgfag, og hadde dermed arbeidet mer med fagområdet. Læreren hadde også nylig gjennomført undervisning om immunforsvaret, smittespredning og vaksinerings, som tilsier at alle elevene skal ha noen forkunnskaper.

Når det kommer til lærernes kompetanse hadde læreren på 8. trinn begrenset med programmeringskompetanse, og uttrykte selv at det var manglende kompetanse på dette området. Læreren på 10. trinn hadde derimot relativt god programmeringskompetanse, og følte seg trygg på å undervise i programmering. Denne kompetansen hadde læreren tilegnet seg gjennom noen kurs i programmering og egenarbeid.

3.3 MIN UTVIKLINGSPROSESS

Figur 8 på neste side gir en oversikt over min utviklingsprosess i denne studien. Den er fordelt inn i de tre fasene forberedelse, gjennomføring og retrospektiv analyse. Figuren viser hvordan de tre fasene ble gjennomført, og tidslinjen gjennom prosessen.



*1 elev i klassen jobbet videre med programmeringsarbeidet i del 2. 1 elev gjorde både programmeringsarbeidet i del 2 og 3.

FIGUR 8: MODELL AV UTVIKLINGSPROSESSEN I DENNE STUDIEN INNDELTE I DE TRE FASENE FORBEREDELSE, GJENNOMFØRING OG RETROSPEKTIV ANALYSE OG TIDSLINJEN GJENNOM PROSESSEN

3.3.1 FORBEREDELSE AV DESIGNEKSPERIMENTET

I denne fasen defineres mål for arbeidet og det besluttes hvilke produkter som skal utvikles på et nærmere bestemt spesifikt og begrenset område for å nå målsettingen (Gravemeijer & Cobb, 2006). Valg av området og mål begrunnes i tilgjengelig litteratur, og det etableres en plan for hvordan utprøvingen skal skje i praksis (Gravemeijer & Cobb, 2006). I denne fasen blir det også etablert hvilken datainnsamlingsmetode som skal benyttes og hvor og med hvem utprøvingen skal skje (Bjørndal, 2013). Eksisterende litteratur på området gjennomgås, både i form av tidligere forskning, læreplaner og andre elementer som er blitt vurdert som aktuelle (Bjørndal, 2013). Litteraturgjennomgangen benyttes deretter videre til å designe et førsteutkast (Bjørndal, 2013).

Forberedelsesfasen i denne studien startet med å beslutte valg av område, altså å kombinere programmering med smittespredning og vaksinerings. Basert på dette ble målet definert som *å bidra til at programmering kan integreres i undervisningen om smittespredning og vaksinerings*. Før videre utvikling, ble det gjennomført et søk der det ble gjort en vurdering angående hva som allerede eksisterer. Mer utdypende om bakgrunnen for studien, målet for studien og hva dette undervisningsopplegget tilfører, ble presentert i innledningen, kapittel 1. Videre ble litteraturen, definert relevant for utviklingen, gjennomgått. Slik det kan ses i figur 8 førte denne litteraturgjennomgangen og utviklingen til et undervisningsopplegg bestående av tre oppgavehefter og en lærerveiledning. Før gjennomføring ble det også gitt veiledning, som førte til bearbeiding av programmene for smittespredning og vaksinerings. Disse endringene blir presentert senere i delkapittelet 4.2.1. Etableringen av forskningssted og forskningsdeltakerne ble presentert under delkapittelet 3.2, og hvilke former for datainnsamling som blir benyttet i denne studien presenteres i delkapittelet 3.4.

3.3.2 GJENNOMFØRING AV DESIGNEKSPERIMENTET

Andre fase av det pedagogiske designeksperimentet er gjennomføringen, der formålet er å teste ut undervisningsopplegget (Gravemeijer & Cobb, 2006). Målet med testingen er å kunne gjøre forbedringer og få en forståelse av hvordan det fungerer, fremfor å gjøre en vurdering av om designet i seg selv fungerer (Gravemeijer & Cobb, 2006). Denne fasen er preget av en syklisk prosess der man etter hver endt test analyserer prosessen. Basert på denne forskningen vil det være mulig å endre designet før det igjen blir retestet (Gravemeijer & Cobb, 2006).

Gjennomføringsfasen av denne studien besto av totalt seks undervisningsøkter. Alle disse øktene ble gjennomført av naturfagslæreren elevene hadde til vanlig. Videre ble alle øktene observert av meg. Etter endt undervisning hadde læreren og jeg en samtale om økten. Etter siste undervisningsøkt hadde jeg også en utdypende samtale med læreren på 10. trinn. Nærmere om benyttet datainnsamlingsmetodene kommer jeg inn på i delkapittel 3.4. De to første øktene ble gjennomført på samme dag, med lærer på 8. trinn. Denne klassen gjennomførte kun oppgaveheftet del 1. Bortsett fra to elever som arbeidet videre. Én av elevene jobbet seg gjennom programmeringsarbeidet i oppgaveheftet del 2, mens den siste eleven i tillegg jobbet seg gjennom programmeringen i oppgaveheftet del 3. Slik figur 8 viser førte dataen fra 8. trinn til videreutvikling av oppgaveheftet del 1 og

lærerveiledningen. Observasjonen gjort av elevene som arbeidet videre med de to oppgaveheftene diskuteres nærmere i delkapittel 4.3.1.

De neste fire undervisningsøktene ble alle gjennomført med samme 10. trinn lærer. I begge klassene på 10. trinn arbeidet elevene med oppgaveheftet del 2 i første undervisningsøkt, og med oppgaveheftet del 3 i andre undervisningsøkt. I etterkant av de to undervisningsøktene i den første 10. klassen ble oppgaveheftet del 2 og oppgaveheftet del 3 videreutviklet. Disse to oppgaveheftene ble også videreutviklet etter undervisningsøktene i den andre 10. klassen. I tillegg førte også tilbakemeldingen fra læreren etter gjennomført utdypende samtale til en videreutvikling av lærerveiledningen.

3.3.3 RETROSPEKTIV ANALYSE AV DESIGNEKSPERIMENTET

Den tredje fasen, altså den retrospektive analysen, går ut på å gjennomføre en analyse av hele datasettet som ble samlet inn under gjennomføringen av designet (Gravemeijer & Cobb, 2006). Formålet med denne analysen er se undervisningsopplegget under ett, og vurdere hvordan funnene kan gjenbrukes videre (Gravemeijer & Cobb, 2006). Hovedfunnene fra studien har som mål å kunne benyttes i utviklingen av et lignende undervisningsopplegg ved et senere tidspunkt (Gravemeijer & Cobb, 2006).

Basert på dette har min retrospektive analyse i denne studien tatt for seg hovedfunnene fra alle utprøvingene, i hovedsak hva som fungerer bra og hva som burde forbedres også sett ut fra tidligere forskning. På denne måten er det tilrettelagt slik at neste utvikler kan ta disse funnene i betraktning før de gjennomfører sin egen fremtidige uttesting.

3.4 METODER FOR DATAINNSAMLING

Valg av metode for datainnsamling i denne studien er begrunnet i anbefalingen til Gravemeijer og Cobb (2006), som foreslår at forskeren bør være til stede i klasserommet under uttesting, for så å gjennomføre samtaler med læreren i etterkant av undervisningsøkten. I tillegg til de korte samtalene etter undervisningsøktene, anbefaler også Gravemeijer og Cobb (2006) at det tidvis bør gjennomføres lenger samtaler. Datamateriale i denne studien vil dermed basere seg på observasjonsnotatene tatt underveis i øktene, notatene fra de korte samtalene med lærerne og lydopptaket fra den utdypende samtalen med læreren på 10. trinn.

Gjennom observasjon får forskeren mulighet til å danne seg et bilde først, før de involverte får muligheten til å tolke situasjonen (Tjora, 2021). Slik som Tjora (2021) skriver er det under observasjon vanlig å både notere ned de direkte observasjonene man gjør, samt refleksjonene tilknyttet disse. Dette ble også gjort under min observasjon. Basert på anbefalingene til Postholm (2010) ble notatblokk delt i to på langs. På venstre siden noterte jeg de direkte observasjonene notert, og på høyre side refleksjonene tilknyttet disse. Planen min under observasjonen var å ha rolle som deltakende observatør. Dette innebærer at elevene og lærer visste at jeg var til stede i klasserommet, men at jeg ikke skulle ta en aktivt rolle i undervisningen (Gold, 1958). I alle de fire øktene på 10. trinn forble jeg en deltakende observatør, men gjennomføringen på 8. trinn resulterte i at jeg tok rollen som observerende deltaker. Dette tilsier da at jeg tok del i undervisningen, og

observerte samtidig (Gold, 1958). Denne endringen av observatørrolle førte også til endringer i lærerveiledningen som jeg kommer tilbake til i delkapittel 4.3.1.

I etterkant av hver undervisningsøkt ble det gjennomført en samtale med læreren. I disse samtalene kom både lærerens og mine refleksjoner frem. På denne måten kunne det bygges et felles forståelsesgrunnlag, som er essensielt i pedagogisk designforskning (van den Akker et al., 2006). Basert på observasjonene og tilbakemeldingene fra læreren ble det gjort endringer som følge av den felles forståelsen. Med læreren på 8. trinn ble denne samtalen vurdert som et godt nok grunnlag etter gjennomføring, da begge undervisningsøktene ble gjennomført på samme dag og i kun et klasserom. Med læreren på 10. trinn ble det også gjennomført en utdypende samtale i etterkant av de fire undervisningsøktene da disse ble gjennomført over fire ulike dager og i to ulike klasserom. Ved å gjennomføre denne utdypende samtalen ble det mulighet for ytterligere refleksjoner angående hva som fungerte eller hva som trengte revidering, samt hvilke forskjeller som ble oppdaget i de ulike klasserommene.

3.5 SORTERING AV DATAMATERIALE

Sortering av datamateriale handler om å redusere datamaterialet slik at det blir mer håndterlig, for å kunne skape en mening ut av det (Postholm, 2010). Sorteringen av datamaterialet i denne studien skjedde i fire omganger. Etter hver gjennomføring, ble datamaterialet sortert og deretter ble det sorterte materialet benyttet til å skape mening ved videreutvikling av undervisningsopplegget. Den fjerde og siste sorteringen av datamaterialet ble benyttet i forbindelse med den retrospektive analysen der hele datamaterialet ble sett under ett (Gravemeijer & Cobb, 2006).

Etter hver gjennomføring ble datamaterialet sortert og redusert. Hvor i prosessen dette skjedde kan ses i figur 8 der datamaterialet førte til en videreutvikling. Observasjonsnotatene fra gjennomføringene ble fordelt inn etter oppgavene i oppgaveheftet det ble arbeidet med i den bestemte økten. De observasjonene som ikke var direkte tilknyttet oppgavene ble satt utenom for å vurdere om dette var koblet til lærerveiledningen. Basert på denne sorteringen ble det gjort en reduksjon av datamaterialet der kjernen av observasjonene ble hentet ut. Samme sortering og reduksjon ble gjennomført av notatene fra samtalen med lærerne. Videre ble reduksjonen gjort av observasjonsnotatene og samtalen med læreren sett opp mot hverandre for å se om det samme kom frem i begge datamaterialene. Etter hver sortering og reduksjon av datamaterialet ble det tatt en vurdering av om det var behov for endring i undervisningsopplegget.

Etter den siste av de fire undervisningsøktene med læreren på 10. trinn utgjorde lydopptaket etter den utdypende samtalen enda et datamateriale. Dette lydopptaket ble lyttet til og gjenfortalt i skriftlig form. Dette var for videre å kunne trekke ut kjernen av samtalen, og gjennomføre en videre reduksjon av datamaterialet. Denne utdypende samtalen var med på å danne grunnlaget for den siste videreutviklingen av undervisningsopplegget. Den fjerde og siste sorteringen som ble gjennomført av datamaterialet var i forbindelsen med den retrospektive analysen. Her ble alle de tidligere gjennomførte sorteringene og reduksjonen av datamaterialet satt mot hverandre for å sortere ut om det var noe som var gjennomgående likt i hele datamaterialet.

3.6 FORSKNINGSETIKK

I ethvert forskningsprosjekt vil det være viktig å ta i betraktning de verdiene, normene og retningslinjer som påvirker den vitenskapelige praksis, og den nasjonale forskningsetikk komiteen for samfunnsvitenskap og humaniora (NESH) har ansvaret for å sikre dette (NESH, 2021). De mest grunnleggende verdiene er knyttet til menneskeverd, og bygger på tre prinsipper: respekt, beskyttelse mot skade og rettferdighet (NESH, 2021). Når det kommer til klasseromsforskning er det i hovedsak tre viktige prinsipper: frivillighet, tilrettelagt informasjon og samtykke (Sikt, u.å.). Dersom det ikke skal behandles personopplysninger som følge av forskningen kan elever over 15 år samtykke selv, men dersom elevene er under 15 år må foreldrene samtykke for dem (Sikt, u.å.). Da det ble vurdert unødvendig med innsamling av personopplysninger om elevene ble ikke denne delen av studien søkt om til Sikt.

I forkant av gjennomføringen på 8. trinn fikk læreren tilsendt et informasjonsskriv som ble sendt videre til elevenes foresatte. I dette skrevet fikk de foresatte informasjon om studien, hva det hadde å si for deres barn, samt barnas mulighet til å ikke bidra i forskningen. Før gjennomføringsstart på 8. trinn ble elevene også informert om forskningen, samt selv gitt anledning til å ikke ønske å bidra. Her ble det også informert om at elevene skulle ta del i undervisningen uavhengig av om de ønsket å bidra i forskningen.

For elevene på 10. trinn ble jeg i samråd med læreren enig i at det ikke var behov for å sende ut et informasjonsskriv til elevenes foresatte da alle elevene var over 15 år, og i stand til å samtykke selv. Dette da det ikke ble innhentet personopplysninger (Sikt, u.å.). En uke før gjennomføringen hadde læreren informert elevene om forskningen, samt gitt dem muligheten til å ikke delta. Før gjennomføringsstart ble elevene på nytt informert om forskningen for å sikre at de var bevisste på hva som skulle foregå. Også i dette tilfellet ble elevene informert om at de skulle delta i undervisningen uavhengig av deres ønske om å bidra i forskningen.

Det ble vurdert unødvendig å ha en utdypende samtale med læreren på 8. trinn. Det ble derimot vurdert nødvendig å ha en utdypende samtale med læreren på 10. trinn. I forbindelse med den utdypende samtalen ble det vurdert hensiktsmessig å ta lydopptak. Dette for å ikke miste essensiell informasjon. Da lydopptak er innsamling av personopplysninger, i denne forbindelse om lærer på 10.trinn, ble dette omsøkt til Sikt og søknaden ble godkjent, se vedlegg 5. Læreren samtykket til å bidra i starten av dette lydopptaket. Lydopptaket ble lagret på NTNU Personlig hjemmeområdet («M»:-disk), etter informasjonslagringsguiden ved NTNU (u.å.).

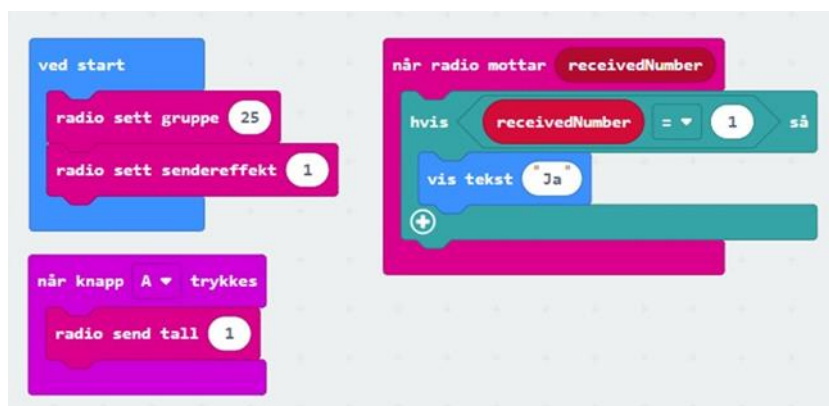
4 RESULTATER OG DISKUSJON

I dette kapittelet vil selve undervisningsopplegget presenteres, som har gjennomgått revidering som følge av utviklingsprosessen. Videre vil grunnlaget for forberedelsen av førsteutkastet og teorien benyttet som grunnlag vises frem. Grunnlaget etterfølges av gjennomføringen av undervisningsopplegget, med de eventuelle endringene som fortløpende ble gjort. Avslutningsvis vil den retrospektive analysen presenteres.

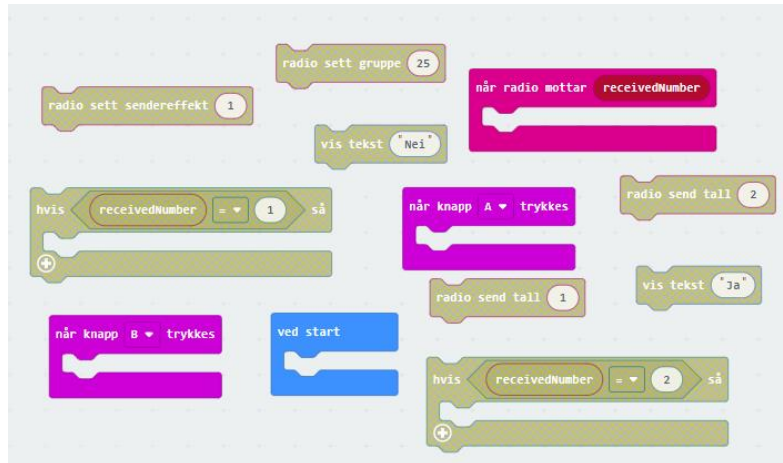
4.1 PRESENTASJON AV UNDERVISNINGSSOPPLEGGET

Undervisningsopplegget utviklet gjennom denne studien består av tre oppgavehefter fordelt på temaene radiokommunikasjon, smittespredning og vaksinerings, samt en lærerveiledning. Lærerveiledningen er utarbeidet for å hjelpe lærerne i arbeidet med undervisningsopplegget. Det fullstendige undervisningsopplegget er å finne i vedlegg 1 til 4.

Det første oppgaveheftet del 1: Radiokommunikasjon (Vedlegg 1) består av totalt tre oppgaver og en introduksjonstekst som introduserer at micro:bitene kan kommunisere med hverandre. De to første oppgavene er lagd etter PRIMM, der elevene i den første oppgaven først skal forutsi hva programmene kommer til å gjøre for deretter å kjøre den og se om antagelsen stemte. De har fått utdelt programmet i figur 9 som vil sende meldingen «Ja» til hverandre når knapp (A) trykkes dersom de er i samme radiogruppe. I oppgave 2 skal elevene undersøke programmet, altså det neste steget i PRIMM, og her får de utdelt et Parsons problem der klossene skal settes sammen til et program som gjør at micro:bitene kan kommunisere «Ja» og «Nei» til hverandre. Figur 10 på neste side viser oppgaven elevene får tildelt. Tilknyttet denne oppgaven er det også gitt hint til elevene som opplever oppgaven som krevende, samt videre arbeid. Det videre arbeidet bygger videre på PRIMM der elevene skal endre programmet ved å få micro:biten til å kommunisere andre ord og figurer. I den tredje og siste oppgaven skal elevene diskutere hvordan radiokommunikasjon kan benyttes til å simulere smittespredning.

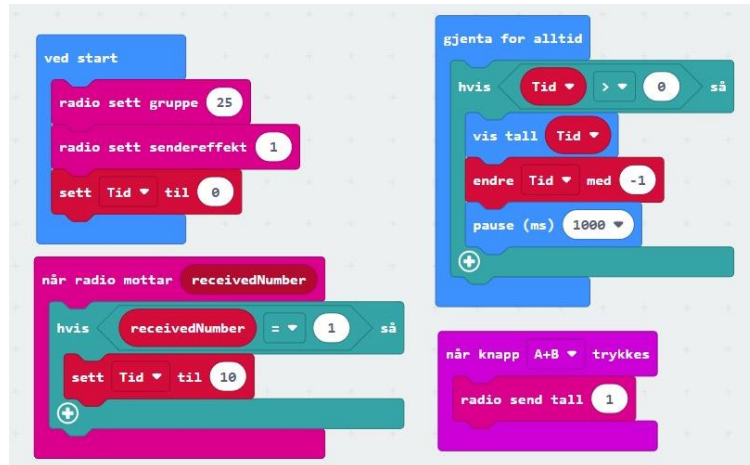


FIGUR 9: PROGRAMMET ELEVENE SKAL FORUTSE OG KJØRE I OPPGAVE 1 I OPPGAVEHEFTET DEL 1: RADIOKOMMUNIKASJON. GJENNOM PROGRAMMET SKAL MICRO:BITENE KUNNE KOMMUNISERE «JA» TIL HVERANDRE.

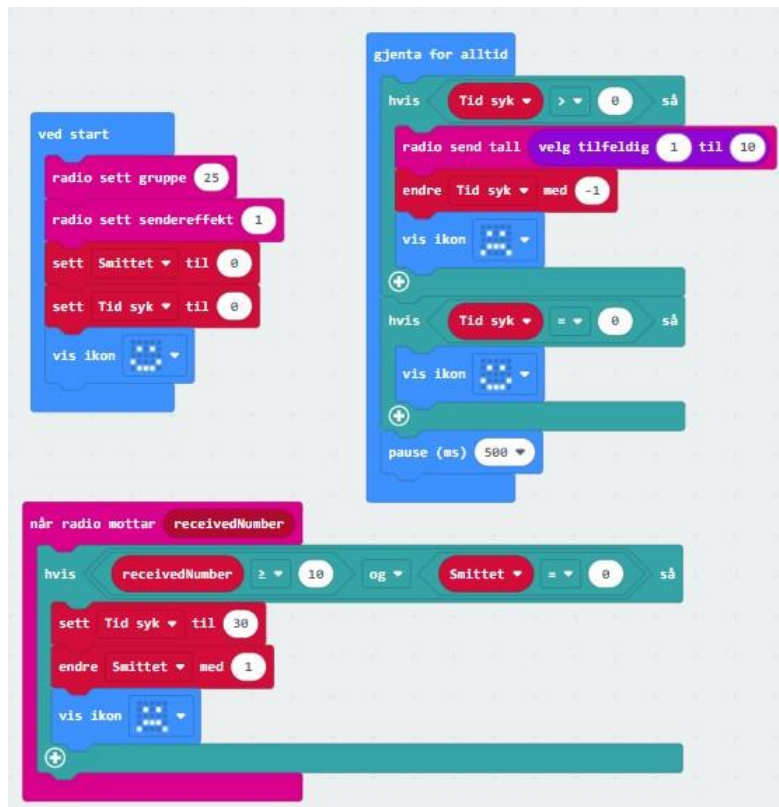


FIGUR 10: OPPGAVEN ELEVENE FÅR UTDILT I OPPGAVE 2 I OPPGAVEHEFTET DEL 1: RADIOKOMMUNIKASJON. DETTE ER ET PARSONS PROBLEM UTEN DISTRAKTORER HVOR ELEVENE SKAL PUSLE SAMMEN KLOSSENE SLIK AT MICRO:BITENE KAN KOMMUNISERE «JA» OG «NEI».

Oppgaveheftet del 2: smittespredning (Vedlegg 2) består av fem oppgaver og en introduksjonstekst. I introduksjonsteksten presenteres det at mikroorganismer kan gjøre oss syke, hvordan de sprer seg, samt hvilke deler immunforsvaret består av. Den første oppgaven bygger på de to første stegene i PRIMM, der elevene først skal forutse hva som skjer når programmet kjøres, for så å kjøre det og sjekke antagelsene. Programmet fungerer som en nedtellingsfunksjon, og kan ses i figur 11 på neste side. I oppgave 2 skal elevene programmere programmet som skal benyttes til å simulere smittespredning. Elevene skal i denne oppgaven undersøke programmet, altså det tredje steget i PRIMM. De får med denne oppgaven utdelt et Parsons problem, samt flytskjemaer for å se rekkefølgen i programmet. Figur 12 på neste side viser det fullstendige programmet elevene skal komme frem til gjennom denne oppgaven. Tilknyttet denne oppgaven er det også lagt til en forenkling for å hjelpe elevene som opplever oppgaven som utfordrende, samt videre arbeid med programmet dersom elevene opplever programmeringen som enkel. Det er også direkte spørsmål tilknyttet programmeringsarbeidet slik at elevene må tenke over hva de gjør. I oppgave 3 skal smittespredningen testes ut på tre ulike måter, og her benyttes programmet de laget i oppgave 2. I etterkant av alle tre gjennomføringene skal elevene besvare spørsmålene direkte tilknyttet uttestingen. Oppgave 4 er en diskusjonsoppgave der elevene skal diskutere hvordan simuleringen kan benyttes til å vise hvordan smittespredning og immunforsvaret fungerer. I den femte og siste oppgaven skal elevene diskutere svakhetene og styrkene til simuleringen, samt hva som kan være mulige feilkilder som følge av gjennomføringen.



FIGUR 11: BILDE AV PROGRAMMET SOM ELEVENE SKAL FORUTSE OG KJØRE I OPPGAVE 1 I OPPGAVEHEFTET DEL 2: SMITTESPREDNING. PROGRAMMET FUNGERER SOM EN NEDTELLINGSFUNKSJON NÅR TIDEN STILLES.

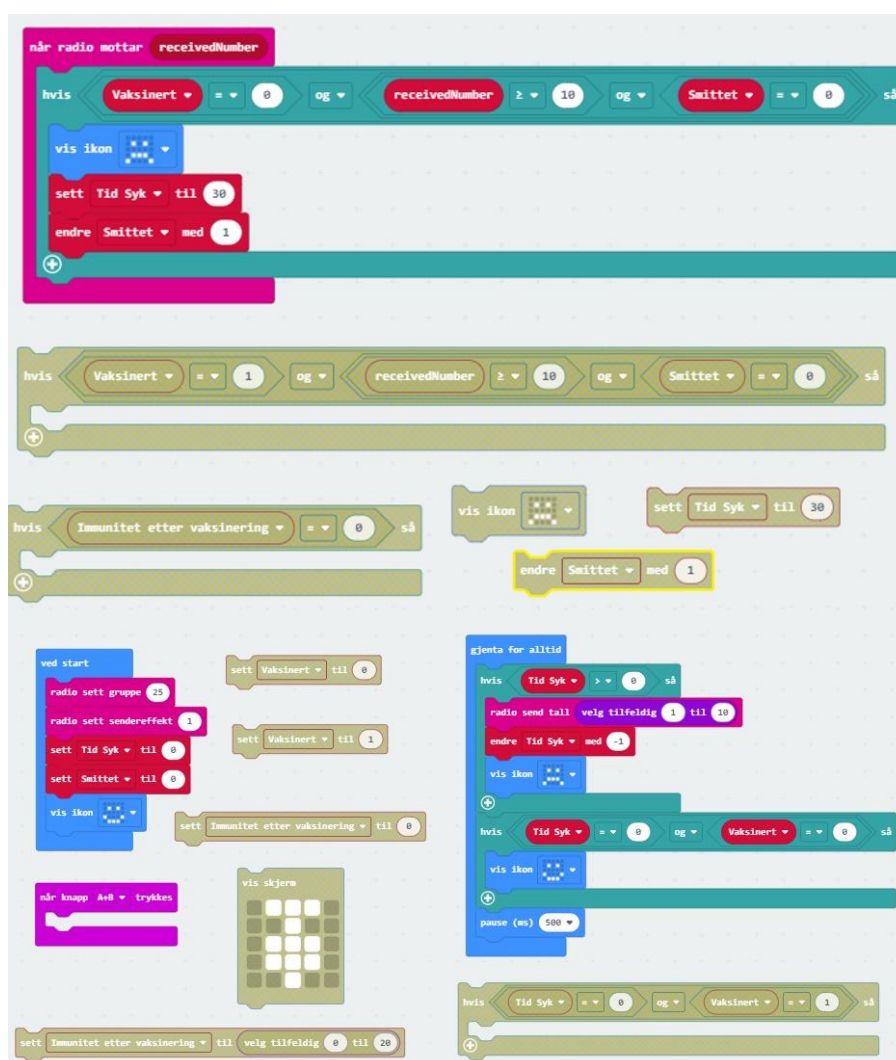


FIGUR 12: PROGRAMMET SOM SKAL BENYTTES FOR Å SIMULERE SMITTESPREDNING, SOM ELEVENE SKAL ARBEIDE MED VED HJELP AV ET PARSONS PROBLEM OG FLYTSKJEMAER.

I oppgaveheftet del 3: vaksinerer (Vedlegg 3) er det fire oppgaver og en introduksjonstekst der det forklares hva en vaksine er og hvordan den fungerer. I den første oppgaven i dette heftet skal elevene implementere en vaksine i programmet for smittespredning som de jobbet med i forrige oppgavehefte. Elevene får i denne oppgaven

tildelt nye klosser og nye flytskjemaer som skal benyttes for å lage det ferdige programmet. Slik programmet vil se ut når elevene får tildelt de nye klossene, kan ses i figur 13.

Hvordan dette programmet kommer til å se ut når det er puslet riktig sammen kan ses i figur 15 på side 29. Tilknyttet oppgaven er det også lagt til forenklinger for å hjelpe elevene som opplever oppgaven som utfordrende, samt videre arbeid med programmet dersom elevene opplever programmeringen som enkel. Det er også direkte spørsmål tilknyttet programmeringsarbeidet slik at elevene må tenke over hva de gjør. I oppgave 2 i dette heftet skal effekten til vaksinen testes ut. Dette gjentas 3 ganger med prosentvis økende vaksinedekning. Lærer utpeker de elevene som skal ta vaksine. I etterkant av alle de tre gjennomføringene skal elevene besvare spørsmålene direkte tilknyttet uttestingen. Oppgave 3 er en diskusjonsoppgave der elevene skal diskutere hvordan simuleringen kan benyttes for å vise hvordan vaksinerings kan beskytte både den vaksinerte og andre mot sykdom. Elevene skal i oppgave 4 diskutere svakhetene og styrkene til simuleringen, samt hva som kan være mulige feilkilder som følge av gjennomføringen.



FIGUR 13: PARSONS PROBLEM SOM ELEVENE FÅR UTDILT, DER VAKSINERING SKAL IMPLEMENTERES I PROGRAMMET FOR SMITTESPREDNING.

Lærerveiledningen (Vedlegg 4) består av tre hoveddeler som skal bidra med å støtte læreren i arbeidet med undervisningsopplegget. Delene består av en introduksjon til undervisningsopplegget, gjennomgang av micro:biten og gjennomgang av undervisningsopplegget. Den første hoveddelen, introduksjon til undervisningsopplegget, presenterer kompetansemålene som det kan arbeides med gjennom undervisningen, samt mulige læringsmål for elevene. Her presenteres også nødvendig utstyr, mulig vurderingsform og hvordan undervisningsopplegget bør arbeides med, samt noen forkunnskaper elevene bør ha før arbeidet med oppgaveheftene påbegynnes. Neste del er en introduksjon av micro:biten, og den tar for seg de relevante delene, hvordan programmet lastes opp og mulige årsaker til at micro:biten kanskje ikke fungerer som ønsket. Den siste delen av lærerveiledningen tar for seg en forklaring av alle oppgavene i oppgaveheftene, samt tips til gjennomføring og en gjennomgang av alle programmene som benyttes.

4.2 FORBEREDELSE AV UNDERVISNINGSSOPPLEGGET

Et av de første valgene som ble tatt i denne studien preger hele undervisningsopplegget, og det valget var å benytte micro:bit i programmeringsarbeidet. Det er micro:biten sin håndfasthet (Sentance et al., 2017), sammen med den innebygde radioen som gjør den til et egnet verktøy for å simulere smittespredning og vaksinerings (Rossing & Johansen, 2021). Siden micro:biten kan sende signaler, kan den simulere at smitte spres. Mottakeren tar videre til seg dette signalet, som kan simulere at en annen person har blitt utsatt for smitten. Samtidig vil micro:biten sin håndfasthet føre med seg smittespredningen ut fra dataskjermen og inn i klasserommet.

Med dette som utgangspunkt ble det i forberedelsesfasen produsert et førsteutkast av undervisningsopplegget, med bakgrunn i relevant teori og tidligere forskning. Utviklingen av dette førsteutkastet er delt inn i tre deler: utvikling av programmene benyttet i simuleringene, utvikling av oppgavene, oppbygningen til oppgaveheftene og utvikling av lærerveiledningen.

4.2.1 UTVIKLING AV PROGRAMMENE

Før oppgaveheftene kunne utvikles måtte programmene som skulle benyttes for å simulere smittespredning og vaksinerings komme på plass. I den forbindelse sto det første valget på hvilket programmeringsmiljø som skulle benyttes, siden micro:biten kan programmeres i flere ulike miljøer (Sentance et al., 2017). På bakgrunn av Lai & Lai (2012) sin studie som viste at elevene likte å benytte Scratch i undervisningen, samt utfordringen knyttet til syntaktisk kunnskap som Qian og Lehman (2017) presenterer, falt valget på det Scratch-lignende blokkmiljøet. Ved å benytte det Scratch-lignende blokkmiljøet vil utfordringer tilknyttet syntaktisk kunnskap minimeres da klossene allerede har en forhåndsprogrammert grafikk, som gir en grafisk fremstilling av samhandlingen mellom komponentene i programmet (Resnick et al., 2009). Ved å minimere utfordringer knyttet til den syntaktiske kunnskapen, vil det være færre momenter som kan gjøre elevene usikre på om utfordringene de møter skyldes feil i deres programmerings- eller fagforståelse (Caballero et al., 2014).

Programmet som simulerer smittespredningen, var det første som ble utviklet. Dette programmet ble så videreutviklet til å gi muligheten for en vaksine. Videre i dette delkapittelet vil programmet ferdigimplementert med vaksinen som det tas utgangspunkt i. Figur 12 på side 24 viser imidlertid hvordan det endelige programmet ser ut uten den implementerte vaksinen.

Når det kommer til utviklingen av programmet vil en modell alltid være en forenkling av virkeligheten (Gilbert et al., 2000; Schwartz et al., 2009; Nia & de Vries, 2017). Simuleringene av smittespredningen og vaksineringsen er en slik forenkling, og i disse er det følgende aspekter som det fokuseres på:

1. Immunforsvaret klarer å beskytte kroppen mot mange mikroorganismer, og vil ikke bli syk hver gang den er i kontakt med mikroorganismene (Grindeland et al., 2020).
2. Etter at en person har vært gjennom et sykdomsforløp vil kroppen kjenne igjen smittestoffene og dermed angripe fortere neste gang de kommer inn i kroppen, da kroppen har dannet antistoffer og hukommelsesceller (Grindeland et al., 2020).
3. Når mikroorganismene trenger inn i kroppen og starter å formere seg blir du syk. Immunforsvaret arbeider da for å fjerne inntrengerne, og dette kan ta tid (Grindeland et al., 2020).
4. Når en person er infisert av mikroorganismer, vil personen spre disse smittestoffene videre ved blant annet hosting og nysing (Steiniger & Wahl, 2021).
5. Vaksinen vil begrense smittespredning, siden kroppen allerede har antistoffene og hukommelsescellene som gjør den i stand til å bekjempe inntrengerne med én gang de trenger inn i kroppen (Grindeland et al., 2020).
6. En vaksine vil ikke være 100% effektiv, og ikke alle vil bli immune etter å ha tatt vaksinen (Europeisk informasjonsportal om vaksinasjon, 2020).

Videre ble programmene for smittespredning og vaksineringsen diskutert med veileder. Figur 14 på neste side viser hvordan programmet så ut før endringene, mens figur 15 på side 29 viser programmet etter at endringene har blitt gjennomført. Tabell 1 på side 30 viser en oversikt over endringene som ble gjort, og begrunnelsen for endringene. Endringene skyldes at det ligger en kontinuerlig forbedring og gjentakende videreutvikling bak en modell (Nia & de Vries, 2017). Dette tilsier at det ofte vil være rom for forbedring av modellen.

```

ved start
  radio sett gruppe 25
  radio sett sendereffekt 1
  sett Tid til 0
  sett Syk til 0
  sett Vaksine til 0
  sett Immunitet etter vaksinerings til 0
  vis ikon

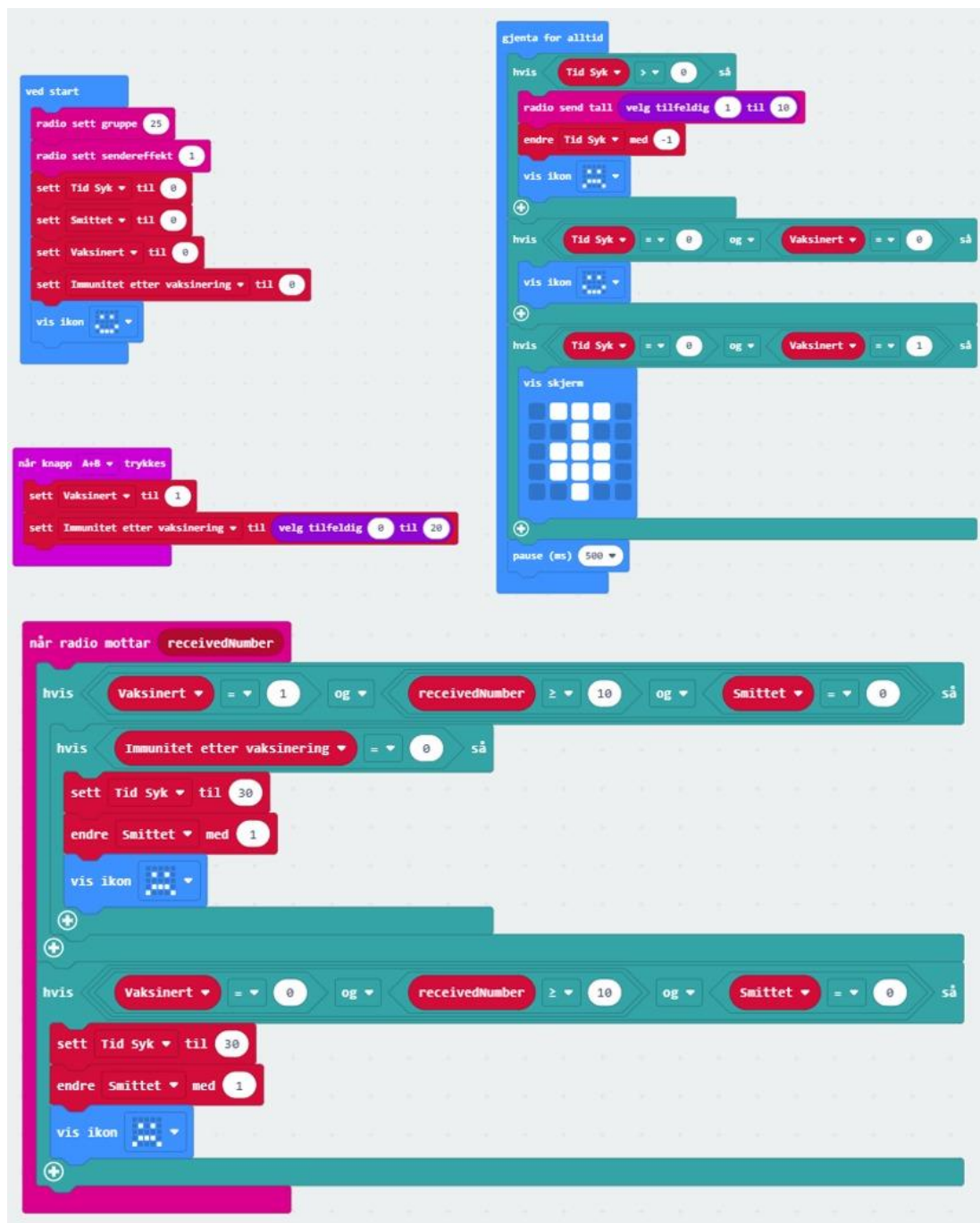
når knapp A+B trykkes
  sett Vaksine til 1

gjenta for alltid
  hvis Tid > 0 så
    radio send tall velg tilfeldig 0 til 10
    endre Tid med -1
    vis ikon
  +
  hvis Tid = 0 og Vaksine = 0 så
    vis ikon
  +
  hvis Tid = 0 og Vaksine = 1 så
    vis skjerm
  +
  pause (ms) 500

når radio mottar receivedNumber
  hvis Vaksine = 1 og receivedNumber = 1 og Syk = 0 så
    sett Immunitet etter vaksinerings til velg tilfeldig 0 til 20
    hvis Immunitet etter vaksinerings = 1 så
      vis ikon
      sett Tid til 30
      endre Syk med 1
    +
  hvis Vaksine = 0 og receivedNumber = 1 og Syk = 0 så
    vis ikon
    sett Tid til 30
    endre Syk med 1
  +

```

FIGUR 14: PROGRAMMET MED VAKSINE INTEGRERT FØR ENDRINGER ETTER SAMTALE MED VEILEDER.



FIGUR 15: PROGRAMMET MED VAKSINE INTEGRERT ETTER ENDRINGER FRA SAMTALE MED VEILEDER.

Tabell 1: Oversikt over endringene gjort etter diskusjon med veileder om programmene til smittespredning og vaksinerings.

Før veiledning	Etter veiledning	Hvorfor endringen blir gjort
En variabel hadde navnet «Syk»	Variabelen fikk navnet «Smittet»	Ved å endre variabelnavnet til «Smittet» vil det tilsa at micro:biten har blitt smittet av mikroorganismen.
En variabel hadde navnet «Tid»	Variabelen fikk navnet «Tid Syk»	Ved å endre variabelen til «Tid Syk» vil det tilsa varigheten man er syk og dermed smittsom.
En av variablene het «Vaksine»	Variabelen fikk navnet «Vaksinert»	Ved å endre til «vaksinert» betyr det at micro:biten er vaksinert og har tatt en vaksine.
ReceivedNumber = 1	ReceivedNumber større eller lik 10	Ved å endre ReceivedNumber til større eller lik 10 istedenfor 1, kan programmet illustrere at "kroppen kan tåle" mer. dvs. man ikke nødvendigvis blir syk hver gang man er i kontakt med smitte. Ved å ta større eller lik vil man også kunne endre dette tallet for å vise at noen har et svakere immunforsvar, og dermed tåler mindre smitte før man blir syk.
Vaksinen har ikke gitt effekt hvis variabelen "Immunitet etter vaksinerings" = 1	Vaksinen har ikke gitt effekt hvis variabelen "Immunitet etter vaksinerings" = 0	Ved å endre om vaksinen har gitt effekt fra 1 til 0 vil det bedre illustrere at vaksinen ikke har fungert for å gi immunitet mot sykdommen.
Vilkår etter ReceivedNumber som hver gang man ble utsatt for smitte avgjorde om du ble smittet eller ei.	Variabelen «immunitet etter vaksinerings» blir bestemt tilfeldig med en gang vaksinen settes, altså når knapp A trykkes.	Vaksinens effekt bestemmes når vaksinen settes, og ikke for hver gang man blir utsatt for smitte.

I forbindelse med programmet for smittespredning og vaksinerings er det også viktig å simulere at smitten har et begrenset spredningsområde med hensyn til avstand. For å kunne justere dette kan klossens «sendereffekt» benyttes for å justere styrken på signalet (Rossing & Johansen, 2021). Denne justeringen strekker seg fra 0 til 7 (Rossing & Johansen, 2021). Etter å ha testet ut ulike sendereffekter med to mikro:biten endte valget på å benytte sendereffekt 1, da signalet strakk seg omtrent 2 meter.

Det ble i tillegg laget to mindre komplekse programmer. Disse to programmene ble utviklet for å danne et grunnlag for elevene. Det første programmet brukes til å kommunisere «Ja» og «nei», og programmet sikter på å gi en forståelse av hvordan micro:bitens sender- og mottakerfunksjon fungerer. Samtidig er denne oppgaven en enklere inngang til

programmeringen. Dersom elevene klarer disse oppgavene kan det øke mestringsforventningen til å også mestre den neste som kommer (Bandura, 1977). Det andre programmet ble laget for å illustrere for elevene hvordan «Tid syk» teller ned i programmene som benyttes i simuleringen av smittespredning og vaksineringsprogrammet. Dette programmet fungerer som en nedtelling, og kan ses i figur 11 på side 24.

4.2.2 UTVIKLINGEN AV OPPGAVEHEFTENE

I utviklingen av oppgaveheftene var det flere avgjørelser som måtte tas. Det første valget var inndeling av oppgaveheftene som resulterte i tre separate hefter. Disse bygger på hverandre ut fra de tre hovedtemaene radiokommunikasjon, smittespredning og vaksineringsprogrammet. Avgjørelsen ble i første omgang tatt fordi et stort hefte kunne ført til kognitiv overbelastning da det vil være for mye å prosessere på en gang (Sweller et al., 1998). Elevene kunne oppleve dette som overveldende. Dette kan igjen ha en betydning for elevenes mestringsforventning da det kan medføre unødvendig stress (Bandura, 1994). Tabell 2 viser en oversikt over hvordan de tre oppgaveheftene er bygd opp og hvilke oppgaver som er utviklet med bakgrunn i den samme teorien.

Tabell 2: Oversikt over oppbygningen til de tre oppgaveheftene.

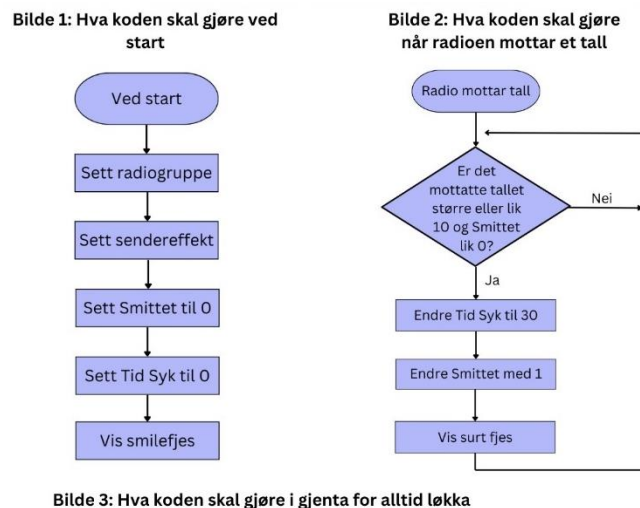
Oppgaveheftet	Oppbygningen av oppgaveheftene
Del 1: Radiokommunikasjon (Vedlegg 1)	Introduksjon av hvordan micro:bitene kan kommunisere med hverandre
	Oppgave 1: Forutse og kjøre programmet
	Oppgave 2: Undersøkelse ved bruk av Parsons problemer
	Oppgave 3: Diskusjonsoppgave
Del 2: Smittespredning (Vedlegg 2)	Introduksjon som omhandler hvordan smitte sprer seg og hvordan immunforsvaret arbeider mot det
	Oppgave 1: Forutse og kjøre programmet
	Oppgave 2: Undersøkelse ved bruk av Parsons problemer og flytskjemaer
	Oppgave 3: Testing av simuleringen med spørsmål etter uttesting
	Oppgave 4: Diskusjonsoppgave
Del 3: Vaksineringsprogrammet (Vedlegg 3)	Introduksjonstekst som omhandler hva en vaksine er og hvordan den fungerer
	Oppgave 1: Undersøkelse ved bruk av Parsons problemer og flytskjemaer
	Oppgave 2: Testing av simuleringen med spørsmål etter uttesting
	Oppgave 3: Diskusjonsoppgave

Det pedagogiske rammeverket PRIMM har blitt benyttet som en gradvis inngang til programmeringsarbeidet i oppgaveheftet del 1: Radiokommunikasjon. Oppgave 1 tar for seg de to første stegene *forutse* og *kjøre*. Elevene skal her først forutse hva programmet

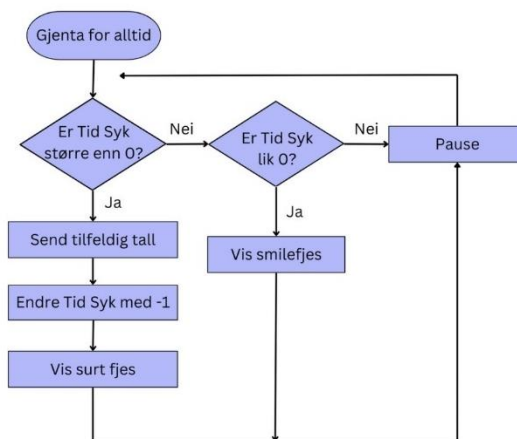
kommer til å gjøre, for så å kjøre den (Sentance et al., 2019; Sentance & Waite, 2017). Figur 9 på side 22 viser hvordan programmet i denne oppgaven ser ut. Oppgave 2 tar for seg det neste steget i PRIMM, altså *undersøke*, og elevene skal i denne oppgaven pusle sammen klossene slik at programmet kan benyttes til å svare både «ja» og «nei». Dette kalles et Parsons problem (Parsons & Haden, 2006), og oppgaven slik elevene får den tildelt kan ses i figur 10 på side 23. Det ble avgjort at Parsons problemet i denne oppgaven ikke skulle inneholde distraktorer siden det kan gjøre oppgaven vanskeligere (Harms et al., 2016). Hovedmålet med dette heftet er at elevene enten skal danne forståelse, eller friske opp forståelsen av hvordan en sender og en mottaker fungerer. Da dette er hovedmålet er det viktigere at elevene fokuserer på denne forståelsen fremfor at oppgaven i seg selv skal gjøres mer komplisert ved bruk av distraktorer. Begrunnelsen knyttet til oppgave 3 i dette oppgaveheftet vil komme avslutningsvis i dette delkapitlet.

Også oppgaveheftet del 2: smittespredning bygger på den gradvise inngangen til programmering som det pedagogiske rammeverket PRIMM presenterer (Sentance et al., 2019, Sentance & Waite, 2017). Slik som i oppgaveheftet del 1, skal eleven i den første oppgaven forutse og deretter kjøre programmet (Sentance et al., 2019; Sentance & Waite, 2017). Programmet elevene skal benytte i denne oppgaven fungerer som en nedtellingsfunksjon, og kan ses i figur 11 på side 24. Oppgave 2 bygger videre på PRIMM. I dette tilfellet benyttes steget *undersøke*, da oppgaven allerede her en ferdig løsning som elevene skal komme frem til (Sentance et al., 2019; Sentance & Waite, 2017). Elevene skal i denne oppgaven programmere programmet som skal benyttes for å simulere smittespredning. Det ferdige programmet kan ses i figur 12 på side 24. Elevene skal gjøre dette ved hjelp av et Parsons problem og flytskjema.

I utviklingen av denne oppgaven var det et par elementer som måtte tas i betraktning. For det første falt valget på å benytte Parsons problemer, siden Parson og Haden (2006) argumenterer for at dette er en spennende form for problemløsning. Etter å ha gjort oppgaven til et Parsons problem, oppsto utfordringen med å lage en tekst som forklarte hva elevene skulle komme frem til. Denne teksten ble fort lang og med mye informasjon, som skapte bekymringen for kognitiv overbelastning. Dette da elevene både måtte prosessere teksten og klossene i programmet samtidig (Bandura, 1977). Også utfordringene knyttet til elevenes strategiske kunnskap kunne vært en begrensende faktor, og ved å da heller gi elevene et flytskjema ble denne utfordringen begrenset (Qian & Lehman, 2017). Valget om å benytte flytskjemaet skyldes at dette er en visuell fremstilling av algoritmen, og elevene kan danne en forståelse av hvordan programmet skal bygges opp (Hooshyar et al., 2015; Charntaweekhun & Wangsiripitak, 2006). Flytskjemaene som elevene ble tildelt vises i figur 16 på neste side. Selv om oppgaven kunne blitt løst med kun flytskjemaet, valgte jeg å beholde Parsons problemet da det tar kortere tid enn dersom elevene skal lete etter, eller teste ut klossene (Ericson et al., 2017; Zhi et al., 2019) samt at bruk av Parsons problem vil redusere risiko for kognitiv overbelastning. (Zhi et al., 2019).



Bilde 3: Hva koden skal gjøre i gjenta for alltid løkka



FIGUR 16: FLYTSKJEMAENE SOM ELEVENE FÅR TILDELTE FOR Å SETTE SAMMEN PROGRAMMET FOR SMITTESPREDNING.

Videre er oppgave 3 i heftet om smittespredning en oppgave der simuleringen skal testes ut. Elevene skal i denne oppgaven teste ut hvordan smitten sprer seg, og har to mål. I den ene testen skal elevene prøve å ikke bli smittet, og i den andre skal elevene prøve å bli smittet. Denne fungerer da som en deltakende simulering, og vil dermed være mer interaktiv for elevene (Parsons, 2021). I disse simuleringene vil elevene kunne teste ut konsekvensen av å bli smittet uten å faktisk måtte bli syke som en konsekvens av det. Dette er det positive med å benytte simuleringer (Dieleman & Huisingh, 2006). Ved at simuleringen også er deltakende vil elevene selv bli påvirket, som igjen kan skape engasjement (Colella, 2000; Klopfer et al., 2005). De deltakende simuleringene bygger også på Dewey (2001) sine prinsipper om at elevene lærer gjennom erfaring, men han poengterer også at aktiviteten må tenkes over for å bli en erfaring. Det er derfor inkludert spørsmål i etterkant av simuleringene, som er direkte knyttet til hva elevene så og erfarte. Videre vil begrunnelsen knyttet til oppgave 4 i samme oppgavehefte komme avslutningsvis i dette delkapittelet.

I oppgaveheftet del 3: vaksinerer, er oppgave 1 innenfor rammeverket PRIMM, direkte på undersøkelsesstadiet (Sentance et al., 2019). Valget bak dette er at denne oppgaven er en direkte forlengelse av programmet for smittespredning. I denne oppgaven skal

elevene utvide programmet ved å implementere muligheten for en vaksine. Programmet for dette kan ses i figur 15 på side 29. Også i denne oppgaven får elevene utdelt et Parsons problem, og flytskjemaer for å sette sammen programmet. Det er kun klossene som ikke allerede er en del av programmet for smittespredning som elevene skal inkludere. Oppgaven med de nye klossene vises i figur 13 på side 25. Valgene bak å benytte Parsons problem og flytskjemaer er allerede begrunnet, og vil dermed ikke begrunnes på nytt. Også oppgave 2 i dette oppgaveheftet bygger på den samme teorien om deltakende simuleringer som i oppgaveheftet om smittespredning. Disse simuleringene har også spørsmål tilknyttet seg i etterkant av aktiviteten som oppfordring til refleksjon (Dewey, 2001). Simuleringene i denne oppgaven gjøres ved at elevene går rundt i klasserommet, og flere og flere vil få tildelt vaksine for å se hvordan dette kan begrense smittespredning.

Som det kan ses i tabell 2 på side 31, avsluttes både del 1, 2 og 3 med en diskusjonsoppgave. I del 1 skal elevene diskutere hvordan radiokommunikasjon kan benyttes til å simulere smittespredning. I del 2 skal elevene diskutere hvordan simuleringen fungerte til å simulere smittespredning og immunforsvarets arbeid. I del 3 skal elevene diskutere hvordan vaksinerer beskytter deg, men også andre, mot smitte og sykdom. Diskusjonsoppgavene bygger på Vygotskys (1978) tanke om å skape en forståelse gjennom språket og i samarbeid med andre. Det er også lagt opp til at disse oppgavene skal gjennomføres i et samarbeid mellom elevene.

4.2.3 UTVIKLING AV LÆRERVEILEDNINGEN

Utviklingen av lærerveiledningen kom i etterkant av de utviklede oppgaveheftene og er utviklet for å hjelpe lærerne i arbeidet med undervisningsopplegget. Lærerveiledningen er todelt, og det består av en introduksjon og en gjennomgang av undervisningsopplegget. Den første delen, introduksjonen til undervisningsopplegget, tar for seg hvilke oppgavehefter undervisningsopplegget består av, kompetansemålene det kan bidra til å arbeide med, mulige læringsmål, samt hvordan elevene burde arbeide med programmeringen. Elevene skal arbeide i par med programmeringen og på kun én datamaskin. Dette er kjent som parprogrammering (Denner et al., 2014). Grunnlaget for dette valget ligger i funnene fra studiene til Hanks et al., (2011) og Denner et al (2014). der elevene som arbeidet sammen fikk et bedre læringsutbytte, ble mer selvsikre og opplevde det som mer trivelig. Den andre delen av lærerveiledningen inneholder en gjennomgang av undervisningsopplegget, samt tips for gjennomføring. Denne delen består også av en grundig gjennomgang av alle programmene, for å prøve å imøtekomme læreres varierte kompetansenivå innenfor programmering (Stenlund, 2021).

4.3 GJENNOMFØRING OG VIDEREUTVIKLING AV UNDERVISNINGSSOPPLEGGET

I dette delkapittelet vil gjennomføringen, resultatene og endringene som følge av disse presenteres. Delkapittelet er inndelt etter de tre gjennomføringene, som også kan ses i figur 8 på side 17. Den første gjennomføringen ble gjort med 8. trinn, mens de to påfølgende gjennomføringene ble gjort på 10. trinn. En lærer gjennomførte undervisningsopplegget i 8. klassen, og en annen lærer gjennomførte undervisningsopplegget med begge 10. klassene.

4.3.1 GJENNOMFØRING OG VIDEREUTVIKLING PÅ 8. TRINN

En uke før gjennomføringen med 8. trinn fikk læreren tilsendt undervisningsopplegget, dette for å kunne løse opp i eventuelle misforståelser. I denne gjennomføringen var det satt av 2 undervisningsøkter på samme dag. Disse var på henholdsvis 60 og 90 minutter, som til sammen utgjorde en utprøvingstid på 150 minutter. Målet var i utgangspunktet å gjennomføre alle tre oppgaveheftene. Elevene slet imidlertid med å forstå oppgaveheftet del 1, med unntak av de elevene som hadde programmering som valgfag. I tillegg til at alle elevene brukte veldig mye tid på oppgaveheftet del 1, var det i tillegg tekniske problemer med skolens nettverk som førte til at avsatt tid i realiteten ble vesentlig redusert i forhold til planen. Utenom arbeid med oppgaveheftet del 1 var det én elev som gjennomførte programmeringsarbeidet i oppgaveheftet del 2, og én elev som gjennomførte programmeringsarbeidet i oppgaveheftet del 2 og del 3. Denne erfaringen tilsa at oppgavene var overkommelige for elevene som hadde bedre programmeringskompetanse. Da elevene klarte å gjennomføre disse oppgavene, kan det tyde på at de nødvendige programmeringsprinsippene er en del av disse to elevenes aktuelle utviklingsnivå. (Vygotsky, 1978).

Som nevnt i metoden under delkapittel 3.4, var planen å være en observerende deltaker i klasserommet. Altså å ikke ta en aktiv rolle i undervisningen (Gold, 1958). Da det oppsto mye utfordringer tilknyttet opplastning av program på micro:biten, samt andre utfordringer tilknyttet at micro:biten ikke fungerte som ønsket, endte jeg med å påta meg rollen som deltakende observatør. Dette tilsier altså at jeg tok en aktiv rolle i undervisningen (Gold, 1958). Dette var i første omgang fordi læreren trengte hjelp med de tekniske utfordringene tilknyttet micro:biten, men også for å hjelpe elevene med programmeringsarbeidet da nesten hele klassen trengte mye hjelp og veiledning. Hvilke endringer dette medførte kommer jeg tilbake til avslutningsvis i dette delkapittelet.

Som følge av gjennomføringen på 8. trinn ble det notert ned observasjoner og tilbakemeldinger fra læreren. Essensen fra disse observasjonene og tilbakemeldingene er presentert i tabell 3 på side 36 for oppgaveheftet del 1 og i tabell 4 på side 37 for lærerveiledningen. Også endringene gjort som følge av disse resultatene presenteres i tabell 3 og tabell 4.

Tabell 3: Resultater og endringer etter gjennomføring av oppgaveheftet del 1: Radiokommunikasjon med 8. trinn.

Oppgavehefte Del 1: Radiokommunikasjon		
Observasjon, tilbakemelding fra læreren eller både observert og tilbakemelding fra læreren.	<p>Observed</p> <p>Tilbakemelding fra lærer</p> <p>Observasjon og tilbakemelding fra lærer</p>	<p>Endringer som følge av observasjon, tilbakemelding fra lærer eller begge.</p> <p>(-----) tilsier at det ikke har blitt gjort en endring da behovet ikke var der.</p>
Oppgave 1 var overkommelig for alle elevene	Både observert og tilbakemelding fra lærer.	-----
Oppgave 2 var for krevende for elevene med lavere programmeringskompetanse, men for enkel for de med høyere programmeringskompetanse	Både observert og tilbakemelding fra lærer	Legge til mulig nivådifferensiering av oppgavene.
Oppgave 3 fungerte bra som en introduksjonsoppgave.	Både observert og tilbakemelding fra lærer	-----

Som tabell 3 viser, var det behov for endringer særskilt i oppgave 2. Oppgave 1 og 3 fungerte i stor grad etter hensikten. Dette underbygges både av tilbakemelding fra lærer og egne observasjoner. Resultatene fra oppgave 2 tilsa at det var behov for mer nivådifferensierte oppgaver. Dette er viktig å ta i betraktning da det i opplæringsloven (1998, §3) står at elevene har rett på tilpasset opplæring. Gjennom dette poengteres viktigheten av å gi lærerne verktøy til å kunne nivådifferensiere oppgaven. Endringene gjort i oppgave 2 som følge av dette bygger på Haraldsrud et al. (2020) sine anbefalinger og å gi tips til de som opplever programmeringen som krevende, og gi mer tidkrevende oppgaver til elevene som finner programmeringen enkel. Elevene som opplever oppgaven som krevende får tips om å se på den forrige oppgaven de gjorde, og finne likhetene. For elevene som opplever oppgaven som enkel, ble det neste steget i PRIMM, *endre*, implementert som en videreføring av oppgaven (Sentance et al., 2019). Elevene får her beskjed om å benytte kreativiteten gjennom å få micro:bitene til å sende andre meldinger eller symboler til hverandre. Da elevene kan komme opp med flere ulike løsninger, vil de også oppleve et større eierskap til programmet (Sentance et al., 2019).

Tabell 4: Resultater og endringer i lærerveiledningen etter gjennomføring med læreren på 8. trinn.

Lærerveiledning		
Observasjon, tilbakemelding fra læreren eller både observert og tilbakemelding fra læreren.	<p>Observed</p> <p>Tilbakemelding fra lærer</p> <p>Både observert og tilbakemelding fra lærer</p>	<p>Endringer som følge av observasjonen, tilbakemeldingen eller begge.</p> <p>(-----) tilsier at det ikke har blitt gjort en endring da behovet ikke var der.</p>
Den var utfyllende og tydelig, og med god gjennomgang av programmene	Tilbakemelding fra lærer	-----
Utfordringer med tilkobling av Micro:biten og utfordringer knyttet til at micro:biten ikke fungerte som ønsket	Observed	Legge til en beskrivelse over knappene og delene som er aktuelle for opplegget, samt de to tilkoblingsmulighetene og mulige årsaker til at den ikke fungerer.
Elevene med veldig ulik programmeringskompetanse hadde ikke like godt utbytte av hverandre.	Observed	Legge til en anbefaling om at elevene burde være på et nærmere kompetansenivå innenfor programmering for større utbytte.
Elevene med lite til ingen programmeringskompetanse hadde utfordringer med oppgavene i del 1, noe som gjør det vanskeligere å gjennomføre de senere oppgavehefter.	Både observert og tilbakemelding fra lærer	Legge til en anbefaling om forkunnskaper elevene burde ha før gjennomføring av undervisningsopplegget.
Savnet en ide om hvordan undervisningsopplegget kunne vurderes	Tilbakemelding fra lærer	Legge til mulig vurderingsform

Overordnet, som vist i tabell 4 opplevde læreren på 8. trinn lærerveiledningen som utfyllende og tydelig. Læreren savnet en tanke om hvordan undervisningsopplegget kunne vurderes. I denne forbindelse ble det lagt til to mulige alternativer. Den første er å benytte diskusjonen som et vurderingsgrunnlag, dette bygger på Vygotsky (1978) sin tanke om at elevene danner mening gjennom språket. Elevene vil gjennom å sette ord på tankene sine kunne vise hva de har lært og forstått. Det andre alternativet er at elevene skriver

ned noen tanker rundt spørsmålene som stilles, samt tegner eller skriver hva programmet kommer til å gjøre. Dette er noe som blant annet anbefales av Sentance et al. (2019) at elevene gjør på *Forutse* stadiet i PRIMM rammeverket.

Videre kom det frem både gjennom observasjonene og tilbakemeldingene fra læreren, at elevene uten eller med lite programmeringskompetanse, hadde utfordringer med å løse oppgavene i dette første oppgaveheftet. Basert på observasjon og tilbakemelding ble det lagt til en anbefaling om hvilke forkunnskaper elevene burde inneha for å arbeide med dette undervisningsopplegget, altså en grunnleggende forståelse om hvordan løkker, vilkår og variabler fungerer. Dette burde være innenfor elevenes aktuelle utviklingsnivå, slik at de kan løse oppgavene på egenhånd (Vygotsky, 1978) eller i elevenes nærmeste utviklingszone, slik at det skal mindre hjelp til for at elevene skal klare det (Vygotsky, 1978). Når det kommer til den nærmeste utviklingssonen, vil det også være aktuelt å plassere elever med relativt nært kompetansenivå sammen i par. Dette var også en annen observasjon som ble gjort som følge av gjennomføringen. Parene med et veldig stort sprang mellom sin programmeringskompetanse fikk lite utbytte av hverandre. Eleven med lite eller ingen kompetanse ble sittende og se på at eleven med nødvendig forkunnskap gjennomførte oppgavene alene. Dette kan kobles opp til funnene fra Cao og Xu (2005) og Williams et al. (2008) som tilsa at elevene ønsket å arbeide med en partner på høyere eller likt nivå som seg selv, slik at de unngår å bli preget av en partner på lavere nivå.

Den siste observasjonen tilknyttet lærerveiledningen i denne gjennomføringen, har grunnlag i at jeg måtte påta meg rollen som deltagende observatør. Denne baseres på at det var mye utfordringer tilknyttet tilkobling av micro:biten, og at den ikke alltid fungerte som ønsket. Denne observasjonen var grunnlaget for en helt ny del i lærerveiledningen, nemlig gjennomgang av micro:biten. Utfordringene var tilknyttet lærerens kompetanse om micro:bitens funksjon, og den nye delen i lærerveiledningen forsøkte å ta høyde for mer variert lærerkompetanse (Stenlund, 2021).

Utover de observasjonene og tilbakemeldingene som var direkte knyttet til oppgaveheftet del 1 og lærerveiledningen, hadde lærer på 8. trinn en siste overordnet kommentar på undervisningsopplegget: *«Jeg kunne tenkt meg å benytte dette undervisningsopplegget med den andre klassen jeg har i naturfag når de skal lære om smittespredning og vaksinerings»*.

4.3.2 GJENNOMFØRING OG VIDEREUTVIKLING ETTER FØRSTE KLASSE PÅ 10. TRINN

Læreren på 10. trinn fikk tildelt dette undervisningsopplegget en uke før gjennomføring for å kunne nøste opp i eventuelle misforståelser. Dette var den andre gjennomføringen i studien. Det var i denne klassen satt av 2 undervisningsøkter på henholdsvis 60 minutter, som utgjør en utprøvingstid på totalt 120 minutter. Basert på antall timer som var til rådighet, var det ikke et mål å gjennomføre alle oppgaveheftene. Læreren hadde i forkant gitt meg informasjon om at elevene nylig hadde arbeidet med micro:bit, og radiokommunikasjon, som tilsier at elevene skulle ha kjennskap til hvordan en sender og mottaker fungerer. Da dette er hovedmålet med oppgaveheftet del 1, startet elevene på 10. trinn direkte med oppgaveheftet del 2, og arbeidet deretter med oppgaveheftet del 3 påfølgende dag. I dette klasserommet forble jeg en deltagende observatør, da det ikke

var behov for å ta en aktiv rolle (Gold, 1958). Dette kan være grunnet lærerens relativt gode kompetanse innenfor programmering.

Notater fra observasjonen og samtalen med læreren ble også i denne gjennomføringen grunnlaget for resultatene presentert i tabell 5 på side 40 og tabell 6 på side 43. I tabell 5 presenteres essensen fra gjennomføringen av oppgaveheftet del 2, samt endringene som førte til videreutviklingen av samme oppgavehefte. Resultatene etter gjennomføringen av oppgaveheftet del 3 er vist i tabell 6. Også i denne tabellen presenteres endringer som førte til videreutvikling av dette oppgaveheftet.

Tabell 5: Resultater og endringer som følge av uttesting av oppgaveheftet del 2: Smittespredning med den første 10. klassen.

Oppgavehefte Del 2: Smittespredning		
Observasjon, tilbakemelding fra læreren eller både observert og tilbakemelding fra læreren.	<p>Observert</p> <p>Tilbakemelding fra lærer</p> <p>Både observert og tilbakemelding fra lærer</p>	<p>Endringer som følge av observasjonen, tilbakemeldingen eller begge.</p> <p>(-----) tilsier at det ikke har blitt gjort en endring da behovet ikke var der.</p>
Oppgave 1 var overkommelig for alle elevene	Både observert og tilbakemelding fra lærer	-----
Gjennomsnittet av elevene klarte å gjøre oppgave 2. Den var for krevende for elevene med lavere programmeringskompetanse og for enkel for elevene med høyere programmeringskompetanse.	Både observert og tilbakemelding fra lærer	Lage mer nivå-differensieringer av oppgaven.
Simuleringen i oppgave 3 fungerte bra.	Både observert og tilbakemelding fra lærer	-----
Spørsmålene i etterkant av simuleringen i oppgave 3 fungerte bra.	Observert	-----
Diskusjonsspørsmålene i oppgave 4 fungerte bra.	Både observert og tilbakemelding fra lærer	-----
Behov for en diskusjon om simulering, og hvordan den fungerer.	Både observert og tilbakemelding fra lærer	Legge til en vurderingsoppgave på slutten.
Tidsmessig var 60 minutter for kort tid for arbeid med heftet.	Både observert og tilbakemelding fra lærer	Legge inn at det er behov for mer enn 60 minutter for å gjennomføre dette oppgaveheftet.

Som det kan ses i tabell 5 fungerte de fleste oppgavene etter hensikten, og det var derfor ikke behov for revidering av oppgave 1,3, 4 eller 5. I oppgave 2 var det derimot behov for nivåddifferensiering av oppgaven. Også tilpasningene gjort i dette oppgaveheftet baseres på Haraldsrud et al. (2020) sine anbefalinger som nevnt tidligere, under videreutviklingen av oppgaveheftet del 1.

Det ble i oppgaveheftet lagt til en forenkling av oppgaven, som skal gjøre den mer overkommelig for elevene som opplever utfordringer. Denne forenklingen var en lenke til et Parsons problem der klossene var mer puslet sammen. Denne tilpasningen gjøres på grunnlag av at Ericson et al. (2017) sier at det er lettere å løse Parsons problemer med færre klosser da det er færre alternative løsninger. Dersom elevene fremdeles opplever utfordringer knyttet til oppgaven, er neste forenkling å gi elevene et bilde av hvordan det ferdige programmet skal se ut. Det ble avgjort at bildet ikke skal inkluderes i oppgaveheftet, da elevene som ikke har behov for dette, kan benytte det som en enkel løsning. Det å gi elevene bilde av det ferdige programmet bygger på at det er enklere for elevene å lese og kopiere allerede ferdige eksempler (Qian & Lehman, 2017). Denne forenklingen vil også ta oppgaven tilbake til det første steget i PRIMM, altså *forutse*, som er enklere enn å *undersøke* da PRIMM rammeverket bygger på en gradvis kompleksitet (Sentance et al., 2019). Gjennom arbeidet med det ferdige programmet vil også elevene utvikle sin strategiske kunnskap (Qian & Lehman, 2017), som igjen kan overføres til andre oppgaver (Fleury, 1991).

For elevene med større læringspotensial er det anbefalt av Haraldsrud et al. (2020) å gi oppgaver som tar lenger tid, er mer komplisert, eller der elevene må søke på nett for å få inspirasjon. Det er i oppgaveheftet gitt to oppgaver som bygger videre på programmet etter at elevene har gjort ferdig oppgaven. Den første oppgaven er en ny oppgave som ligger på *undersøke* stadiet i PRIMM der elevene skal endre variabler i programmet (Sentance et al., 2019). Elevene skal i denne oppgaven gjøre endringer i det ferdige programmet som kan benyttes til å simulere at en person har et dårligere immunforsvar. Den mer kompliserte oppgaven går videre til det neste steget i PRIMM, nemlig *endre*, her skal de gjøre endringer i det allerede eksisterende programmet (Sentance et al., 2019). Elevene får her spørsmål om å videreutvikle programmet slik at det kan benyttes til å simulere at ytterst få dør som følge av sykdommen.

Den andre endringen som ble gjort på bakgrunn av resultatene var å legge til en vurderingsoppgave i slutten av oppgaveheftet. Elevene skal i denne oppgaven vurdere svakhetene og styrkene ved simuleringen, samt hvilke feilkilder som kan ha forekommet. At elevene skal vurdere svakhetene og styrkene til simuleringer er et viktig element som både Schwarz et al. (2009) og Hallström & Schönborn (2019) poengterer. Det er viktig at elevene er i stand til å vurdere omfanget og begrensningen modellen har (Hallström & Schönborn, 2019), da den er en forenkling av virkeligheten (Gilbert et al., 2000). Basert på denne nye tilførte oppgaven i oppgaveheftet ble det også lagt til et nytt kompetansemål i undervisningsopplegget, nemlig « bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensninger » (Kunnskapsdepartementet, 2019, s9).

I tillegg til å vurdere modellens styrker og svakheter skal elevene også vurdere mulige feilkilder som kan ha forekommet i gjennomføringen av simuleringen. Simuleringer er interaktive modeller, og dermed kan variablene endres (Renken et al., 2016). Elevene kan ha endret variablene i programmet til å gjøre dem mer eller mindre smittsomme, samt

endre tallene slik at de ikke kan bli smittet. Dette ville kunne ha påvirket utfallet av simuleringen.

Den siste endringen er knyttet til tidsbruken av dette oppgaveheftet, både læreren og jeg observerte at 60 minutter ble litt snaut med tid for å kunne gjennomføre oppgaveheftet på en god måte. Dermed er det lagt inn i lærerveiledningen at det er behov for mer enn 60 minutters gjennomføringstid, samt at tidsbruken kan påvirkes av elevenes forkunnskaper og hvordan undervisningen gjennomføres.

Tabell 6: Resultater og endringer som følge av uttesting av oppgaveheftet del 3: Vaksinerings med den første 10. klassen.

Oppgavehefte del 3: Vaksinerings		
Observasjon, tilbakemelding fra læreren eller både observert og tilbakemelding fra læreren.	<p>Observed</p> <p>Tilbakemelding fra lærer</p> <p>Både observert og tilbakemelding fra lærer</p>	<p>Endringer som følge av observasjonen, tilbakemeldingen eller begge.</p> <p>(-----) tilsier at det ikke har blitt gjort en endring da behovet ikke var der.</p>
Gjennomsnittet av elevene klarte å gjøre oppgave 1. Den var for krevende for elevene med lavere programmeringskompetanse og for enkel for elevene med høyere programmeringskompetanse.	Både observert og tilbakemelding fra lærer	Legge til nivåddifferensiering av oppgaven.
Når elevene gikk rundt i klasserommet oppgave 2, var det ikke mulig å se effekten på en god måte. Samtidig som de vaksinerte ble plassert i et område av klasserommet	Både observert og tilbakemelding fra lærer	Simuleringen gjennomføres mens elevene sitter på plassene sine eller står stille, og vaksinen spres rundt i rommet.
Spørsmålene i etterkant av simuleringen i oppgave 2 fungerte bra	Observed	-----
Diskusjonsspørsmålene i oppgave 3 fungerte bra	Både observert og tilbakemelding fra lærer	-----
Behov for en diskusjon om simulering, og hvordan den fungerer.	Både observert og tilbakemelding fra lærer	Legge til en vurderingsoppgave på slutten.
Tidsmessig var 60 minutter nok tid til å gjennomføre oppgaveheftet.	Både observert og tilbakemelding fra lærer	Legge til i lærerveiledningen at det burde settes av omtrent 60 minutter til arbeidet.

Som det kan ses i tabell 6 var det behov for flere revideringer som følge av denne gjennomføringen. Unntakene var spørsmålene i etterkant av simuleringsuttestingen, og diskusjonsspørsmålene avslutningsvis i oppgaveheftet om vaksinerings.

Også i dette oppgaveheftet, slik som i de to tidligere oppgaveheftene var det behov for nivåddifferensiering av programmeringsoppgaven, altså oppgave 1. På lik måte som i oppgaveheftet om smittespredning, er det lagt til en lenke til et Parsons problem som er mer puslet sammen da dette skal være enklere for elevene (Ericson et al., 2017). Dersom elevene fremdeles har utfordringer tilknyttet oppgaven er det neste steget å gi elevene bilde av det ferdige programmet. Dette bygger på den samme begrunnelsen som gitt for denne tilpasningen i oppgaveheftet om smittespredning. For elevene med større læringsutbytte er det også i dette oppgaveheftet gitt to oppgaver. Begge benytter det samme oppsettet som i oppgaveheftet om smittespredning. Den første oppgaven er enda en *undersøkeoppgave* innenfor PRIMM rammeverket (Sentance et al., 2019), der elevene skal gjøre endringer i programmet for å gjøre vaksinen mer eller mindre effektiv. Den andre oppgaven går videre i rammeverket PRIMM, altså *endre*, elevene skal her endre programmet slik at det kan benyttes til å simulere at effekten til vaksinen øker når en person tar den andre dosen av vaksinen.

Videre vil en modell være i kontinuerlig forbedring og videreutvikling (Nia & de Vries, 2017). Det vil derfor som følge av resultatene gjøres endringer. Da dette er en deltakende simulering vil elevene fungere som variabler som spiller inn på hvordan simuleringen fungerer. Dette da elevene har en aktiv rolle i simuleringen (Colella, 2000). Det er gjennom denne simuleringen et mål å vise elevene hvordan vaksinerings kan bidra til å begrense smittespredning. Læreren og jeg opplevde at denne simuleringen ikke viste dette tydelig da elevene gikk rundt i klasserommet fremfor å sitte stille. Samtidig var de vaksinerte kun plassert i en del av klasserommet. Det ble på bakgrunn av dette gjort en endring slik at simuleringen skulle ha ønsket utfall. Elevene skal nå bli sittende på plassene sine, og læreren fordeler de vaksinerte rundt i rommet. På denne måten blir det enklere å se hvor de vaksinerte er og hvordan smitten sprer seg.

På lik linje som i oppgaveheftet om smittespredning, var det behov for en vurderingsoppgave også av denne simuleringen. Elevene skal også her vurdere svakhetene og styrkene ved simuleringen, da dette er en viktig ferdighet som elevene trenger (Hallström & Schönborn, 2019). Også feilkildene knyttet til om elever som ikke skal være vaksinert, vaksinerer seg, eller om de har lastet opp et program med ulik effekt, har en betydning for utfallet av simuleringen, og kan dermed være en feilkilde.

Avslutningsvis erfarte vi at en 60 minutters økt var tilstrekkelig til å få gjennomført dette oppgaveheftet på en ordentlig måte. Dermed har det blitt lagt inn i lærerveiledningen at 60 minutter skal være dekkende. Også her vil tidsbruken være påvirket av elevenes forkunnskaper og hvordan undervisningen gjennomføres.

4.3.3 GJENNOMFØRING OG VIDEREUTVIKLING ETTER ANDRE KLASSE PÅ 10. TRINN

Den tredje og siste gjennomføringen som ble gjort i denne studien, skjedde i den andre 10. klassen. Læreren hadde allerede fått utdelt undervisningsopplegget da det var samme lærer som gjennomførte i begge 10. klassene. På lik linje med den andre 10. klassen var det også her satt av 2 undervisningsøkter på 60 minutter. Også denne klassen hadde

arbeidet med radiokommunikasjon tidligere, og startet direkte med oppgaveheftet del 2, og deretter oppgaveheftet del 3. Også i denne klassen forble jeg observerende deltaker. (Gold, 1958).

Notater fra observasjonen og påfølgende samtale med læreren ble også i denne gjennomføringen grunnlaget for resultatene presentert i tabell 7, tabell 8 på side 48 og tabell 9 på side 49. Resultatene og de mulige endringene som følge av utprøvingen av oppgaveheftet del 2 er presentert i tabell 7, i tabell 8 presenteres resultatene og mulige endringer i oppgaveheftet del 3. I tabell 9 presenteres resultatene og videreutviklingen av lærerveiledningen.

Tabell 7: Resultater og mulige endringer etter gjennomføring av oppgaveheftet Del 2: Smittespredning med den andre 10. klassen.

Oppgavehefte Del 2: Smittespredning		
Tilbakemelding, observasjon eller begge	<p>Observert</p> <p>Tilbakemelding fra lærer</p> <p>Både observert og tilbakemelding fra lærer</p>	<p>Mulige endringer som følge av observasjonen, tilbakemeldingen eller begge.</p> <p>(-----) tilsier at det ikke har blitt gjort en endring da behovet ikke var der.</p>
Oppgave 1 var overkommelig for alle elevene	Både observert og tilbakemelding fra lærer	-----
Oppgave 2 fungerte bra med nivåddifferensiering av oppgavene	Både observert og tilbakemelding fra lærer	-----
Elevene gjorde oppgave 2 uten å tenke over hva programmet gjorde eller hvorfor de plasserte klossene slik.	Tilbakemelding fra lærer	Legge inn spørsmål elevene må tenke på under programmeringsarbeidet.
Fikk ikke ønsket resultat ved simuleringen, da alle elevene som var i nærheten av smittesprederen ble syke samtidig.	Både observert og tilbakemelding fra lærer	Endre programmet slik at tilfeldigheten for om elevene blir smittet er hos mottaker og ikke hos senderen (ikke endret).
Elevene burde se hvordan smitten sprer seg rundt i klasserommet også.	Tilbakemelding fra lærer	Legge til enda en uttesting der elevene står stille, for å se hvordan smitten sprer seg.

Smitten spredde seg for langt og fort i uttestingen i oppgave 3	Både observert og tilbakemelding fra lærer	Legge til i lærerveiledningen at sendereffekten må tilpasses størrelsen på klasserommet, eller at størrelsen på rommet må tilpasses den opprinnelige sendereffekten.
Spørsmålene i etterkant av simuleringen i oppgave 3 fungerte bra	Observed	-----
Diskusjonsspørsmålene i oppgave 4 fungerte bra	Både observert og tilbakemelding fra lærer	-----
Diskusjon om simuleringen fungerte bra	Både observert og tilbakemelding fra lærer	-----
Vurderingsoppgaven fungerte bra	Både observert og tilbakemelding fra lærer	-----
Tidsmessig var 60 minutter litt kort til for arbeid med heftet	Både observert og tilbakemelding fra lærer	Dette var vi bevist på fra forrige gjennomføring med 10. trinn, så ingen endring som følge av dette.

Som det kan ses i tabell 7 fungerte endringene som ble gjort etter forrige uttesting av oppgavehefte del 2 etter hensikten. Denne gjennomføringen bekreftet at oppgave 1, spørsmålene etter simuleringen og diskusjonsspørsmålene i oppgave 4 fremdeles fungerte som tenkt. I etterkant av denne andre gjennomføringen av oppgaveheftet del 2 er det fremdeles behov for justering av flere områder.

Basert på den første tilbakemeldingen fra læreren, som tilsa at elevene kun gjennomførte programmeringsarbeidet i oppgave 2 uten å tenke over det, ble det utarbeidet spørsmål til oppgaven. Dette er noe Dewey (2001) poengterer ved å si at det ikke er nok å kun gjennomføre en handling, men den må tenkes over for å skape mening. Spørsmålene som ble utarbeidet for at elevene måtte tenke over arbeidet er knyttet til elevenes konseptuelle kunnskap, altså forståelsen om programmeringsprinsipper som løkker, vilkår og variabler (Qian & Lehman, 2017). Dette på bakgrunn av oppgavens oppbygning der det Scratch-lignende blokkmiljøet minimerer utfordringen med syntaktisk kunnskap, og flytskjemaet som minimerer utfordringen med den strategiske kunnskap. Figur 17 på neste side viser spørsmålene som ble en del av oppgave 2. Gjennom disse spørsmålene skal elevene finne igjen programmeringsprinsippene, og tenke over hvorfor de benyttes på den måten.

Spørsmål til programmeringen

1. Hvilke løkker finner dere i dette programmet, og hvordan brukes de?
2. Hvilke vilkår finner dere i dette programmet?
3. Hvorfor har dere valgt å plassere vilkårene der dere har gjort?
4. Hva forteller disse vilkårene oss?
5. Hvilke variabler finner dere i programmet?
6. Hvorfor har dere valgt å plassere variablene der dere har gjort?
7. Hva brukes disse variablene til, og hva skal de hjelpe med å vise?

FIGUR 17: SPØRSMÅL KNYTTET TIL PROGRAMMERINGEN AV OPPGAVE 2 I OPPGAVEHEFTET DEL 2 SOM TAR UTGANGSPUNKT I ELEVENES KONSEPTUELLE KUNNSKAP.

Basert på den samme tanken om at modeller er i kontinuerlig endring (Nia & de Vries, 2017), er det også behov for endring av simuleringsgjennomføringen etter denne utprøvingen.

Læreren ga tilbakemelding om et savn i forbindelse med en utprøving av smittespredningen der elevene står stille, og ser at smitten sprer seg. Dette ble derfor lagt til som enda en uttesting av smittespredningssimuleringen. På denne måten ville elevene kunne se at smitten sprer seg. Også denne uttestingen av simuleringen har fått et spørsmål i etterkant, slik at elevene må tenke over opplevelsen (Dewey, 2001).

Den andre utfordringen som ble oppdaget er knyttet til sendereffekten, som var satt til 1 i programmet. Dette er det nest laveste nivået effekten kan settes til (Rossing & Johansen, 2021). Vi erfarte i denne gjennomføringen at smitten spredte seg raskere da klasserommet var av mindre størrelse og som følge av kortere distanse mellom elevene. På bakgrunn av dette ble det derfor lagt til i lærerveiledningen at sendereffekten må tilpasses distansen mellom elevene og størrelsen på klasserommet, alternativt må området tilpasses sendereffekten.

En annen observasjon var at alle elevene ble syke helt likt når de var i kontakt med smittebærer. Her er en mulig tanke å flytte tilfeldigheten for å bli syk over til mottakeren. Dette er en mulighet, men krever en programendring. Grunnet tidsperspektivet ble ikke denne muligheten undersøkt i denne studien. Dette er likevel noe som kan ses på ved et senere tidspunkt.

Tabell 8: Resultater og endringer etter gjennomføring av oppgaveheftet Del 3: Vaksinerings med den andre 10. klassen.

Oppgavehefte del 3: Vaksinerings		
Tilbakemelding, observasjon eller begge	Observert Tilbakemelding fra lærer Både observert og tilbakemelding fra lærer	Endringen som følge av observasjonen, tilbakemeldingen eller begge. (-----) tilsier at det ikke har blitt gjort en endring da behovet ikke var der.
Oppgave 1 fungerte bra med nivåddifferensiering av oppgavene	Både observert og tilbakemelding fra lærer	-----
Flere av elevene gjorde oppgave 1 uten å tenke over hvorfor de tok valgene de tok	Tilbakemelding fra lærer	Legge inn spørsmål elevene må tenke på under programmeringsarbeidet.
Simuleringen fungerte bedre når elevene satt på plassene, og de vaksinerte ble spredd utover.	Både observert og tilbakemelding fra lærer	-----
Smitten spredde seg lenger enn ønsket.	Både observert og tilbakemelding fra lærer	Størrelse på rom må tilpasses sendereffekt, eller at gjennomføringen må skje på et større område
Spørsmålene i etterkant av simuleringen i oppgave 2 fungerte bra	Observert	-----
Diskusjonsspørsmålene i oppgave 3 fungerte bra	Både observert og tilbakemelding fra lærer	-----
Vurderingsoppgaven fungerte bra	Både observert og tilbakemelding fra lærer	-----
Tidsmessig var 60 minutter nok tid til å gjennomføre oppgaveheftet.	Både observert og tilbakemelding fra lærer	Dette var vi bevist på fra forrige gjennomføring med 10. trinn, så ingen endring som følge av dette.

Endringene i oppgaveheftet del 3 som ble implementert etter første gjennomføring på 10. trinn, viste seg å fungere etter hensikten ved andre gjennomføring på samme trinn. Også

etter denne gjennomføringen, er det behov for noen revideringer. Den samme tilbakemeldingen knyttet til at elevene kun gjennomførte programmeringsarbeidet i oppgave 2, i oppgaveheftet del 3 ble også erfart i gjennomføringen av dette oppgaveheftet. Det ble også her lagt til lignende spørsmål som i figur 17. Videre opplevde vi også at sendereffekten må tilpasses i dette oppgaveheftet på lik linje med i oppgaveheftet om smittespredning.

Tabell 9: Resultater og endringer i lærerveiledningen etter gjennomføring med læreren på 10. trinn.

Lærerveiledningen		
Observasjon, tilbakemelding fra læreren eller både observert og tilbakemelding fra læreren.	<p>Observert</p> <p>Tilbakemelding fra lærer</p> <p>Observasjon og tilbakemelding fra lærer</p>	<p>Endringer som følge av observasjonen, tilbakemelding fra lærer eller begge.</p> <p>(-----) tilsier at det ikke har blitt gjort en endring da behovet ikke var der.</p>
Den var utfyllende	Tilbakemelding fra lærer	-----
Savnet en forklaring på hva en simulering er.	Tilbakemelding fra lærer	Legge til en forklaring av hva en simulering er.
Ønsket at lærerveiledningen skulle tilpasses mer til lærernes ulike kompetansenivå.	Tilbakemelding fra lærer	Legge ulike bakgrunnsfarger på de ulike programforklaringene for å forenkle for de med god kompetanse.
Programmering i par fungerte bra for mange, men elevene burde få en forklaring på hva deres roller er i samarbeidet.	Tilbakemelding fra lærer	Legge til en forklaring på de ulike oppgavene elevene har i programmeringsarbeidet og anbefalt læreren å gjennomgå dette med elevene.

Tilbakemeldingene knyttet til lærerveiledningen fra læreren på 10. trinn kan ses i tabell 9, også denne læreren opplevde lærerveiledningen som utfyllende. Likevel opplevde læreren at det i lærerveiledningen var litt vanskelig å skille mellom hva som var forklaringen på programmet, oppgaven og micro:biten. Læreren utrykte også at på bakgrunn av relativt god programmeringskompetanse var det ikke behov for å gjennomgå programmene så nøye. For å tilpasse lærerveiledningen bedre til lærernes varierte programmeringskompetanse (Stenlund, 2021), ble det lagt en lilla bakgrunn bak forklaringen til programmene og en grønn bakgrunn bak forklaringen av micro:biten for å lettere skille disse områdene fra hverandre. Videre savnet læreren en forklaring på hva en simulering er, da dette skal forklares til elevene. På bakgrunn av denne tilbakemeldingen ble denne forklaringen lagt til i lærerveiledningen.

Den siste tilbakemeldingen fra læreren var at elevene burde få en forklaring på hva deres rolle i programmeringssamarbeidet går ut på. I lærerveiledningen ble dermed de positive aspektene ved parprogrammering med tanke på bedre læringsutbytte, mer selvsikkerhet

og den mer trivelige opplevelsen lagt til (Denner et al., 2014; Hanks et al., 2011). Videre ble også elevenes roller i samarbeidet presentert, ved at begge aktivt skal bidra, men én har ansvar for mus og tastatur mens den andre skal bidra med innspill (Williams & Kessler, 2000). Avslutningsvis anbefales det også at disse rollene blir gjennomgått med klassen før de starter programmeringsarbeidet.

Utover de observasjonene og tilbakemeldingene som var direkte knyttet til oppgaveheftet del 2 og 3, samt lærerveiledningen, hadde læreren på 10. trinn en siste overordnet kommentar på opplegget: «*Spennende, kunne tenkt meg å bruke det senere og en fin måte å få inn programmeringsmålet med sender og mottaker på en litt annen måte*».

4.4 RETROSPEKTIV ANALYSE

Denne oppgaven har to delmål. Det første delmålet knytter seg opp mot å utvikle et undervisningsopplegg som integrerer programmering i undervisning om smittespredning og vaksiner, og det andre delmålet er å benytte observasjoner og tilbakemeldinger underveis i prosessen for å legge til rette for videreutvikling av undervisningsopplegget. I de to tidligere delkapitlene har valgene i forbindelse med utviklingen av oppgaveheftet blitt presentert, samt de endringene som har blitt gjennomført som følge av observasjoner og tilbakemeldinger. Analysen i dette delkapittelet vil fokusere på erfaringer underveis i prosessen, hvilke tilbakemeldinger som er blitt implementert, og hvilke funn som fremdeles står igjen som uavklarte. Erfaringer gjort i prosessen kan benyttes som inspirasjon og kan tas i betraktning ved utarbeidelse av et tilsvarende undervisningsopplegg ved en senere anledning.

Gjennomgående gjennom hele studien har det vært tre elementer som har fungert. Den første er å bruke *forutse* og *kjøre* som er de to første stegene i rammeverket PRIMM (Sentance et al., 2019; Sentance & Waite, 2017). Det fungerte bra som en felles start på programmeringsarbeidet, der alle elevene uavhengig av nivå vil kunne lykkes. Basert på dette vil jeg anbefale de neste som skal utarbeide lignende undervisningsopplegg ved et senere tidspunkt å ta dette i betraktning. De to andre elementene knytter seg til bruk av spørsmål direkte knyttet til uttesting av simuleringen, og bruk av diskusjonsoppgaver til refleksjon. Dette tyder på at Vygotsky (1978) sin tanke om at språk er et godt verktøy å benytte for å få elevene til å knytte flere elementer opp mot hverandre, slik som det er gjort i min studie.

I alle oppgaveheftene var det behov for nivådifferensiering innenfor programmeringsarbeidet. Haraldsrud et al. (2020) presenterer at det innenfor programmering er store forskjeller mellom elevene. I utvikling av et lignende undervisningsopplegg ved en senere anledning vil det være greit å ta i betraktning disse nivåforskjellene allerede fra start. Ofte er det slik at tilpasningen skjer i forhold til den svake elev. Min erfaring gjennom denne studien er imidlertid at det er viktig å tenke begge veier ved utvikling av opplegget. At elevene skal få opplæring tilpasset sine evner er også lovfestet gjennom opplæringsloven (1998, §1-3). Denne loven skal nettopp sikre at elevene uavhengig av forkunnskaper skal få utbytte av undervisningen. Ved å legge til nivådifferensiering av oppgavene i oppgaveheftene vil dette hjelpe lærerne. De kan benytte undervisningsopplegget direkte også i relasjon til differensiering uten selv å måtte tenke ut egne løsninger.

Det er også viktig å ta hensyn til lærernes ulike kompetanse (Stenlund, 2021), både med hensyn til de med lav og høy kompetanse, da særlig innen et nytt fag som programmering. En lærer med høy kompetanse vil ønske å bli veiledet direkte til essensen i hva programmet skal utføre, uten behov for videre utdypninger, mens en lærer med lav programmeringskompetanse har et større behov for bakgrunnsforklaringer. Det vil i utviklingen av et tilsvarende undervisningsopplegg være greit å ta dette i betraktning allerede i forberedelsesfasen.

Et annet funn, er at jeg er blitt oppmerksom på viktigheten av å være åpen for endringer i modellene som ligger til grunn for undervisningsopplegget i denne studien. Underveis fikk jeg flere tilbakemeldinger og observerte selv hvordan opplegget kunne forbedres for bedre å bidra til elevenes læring. Utvikling av modeller er en kontinuerlig forbedringsprosess (Nia & de Vries, 2017), og den som arbeider med modeller bør ha et åpent sinn i forhold til andre innspill. Et eksempel på dette i min egen studie, er tilbakemeldingen om at samtlige elever ble smittet samtidig. I den forbindelse kan det være hensiktsmessig å vurdere å endre modellen slik at tilfeldigheten for å bli smittet bør ligge hos mottaker fremfor sender som i dagens modell. Denne tanken kan være inspirasjon til eventuell videreutvikling av dette undervisningsopplegget.

Det er videre viktig å få tydelig kommunisert at modeller er en forenkling av virkeligheten, og ikke en direkte kopi av verden (Gilbert et al., 2000; Nia & de Vries, 2017; Schwarz et al., 2009). Et funn i denne studien, poengterer hvor viktig dette er for elevens forståelse. Det bør derfor legges opp til at elevene selv må vurdere modellene de bruker, som igjen danner et større grunnlag for denne forståelsen (Hallström & Schönborn, 2019; Schwarz et al., 2009). Det foreslås med dette at når elevene arbeider med undervisningsopplegg der modellen er en stor del av undervisningen som i denne studien, at det legges opp til vurdering for å unngå at elevene får feilaktig forståelse av at modellen er en kopi av virkeligheten.

I løpet av denne studien, fant jeg også at flere elever utførte oppgavene som beskrevet, men i mindre grad reflekterte over hva som faktisk skjedde i forhold til de ulike valgene som ble tatt. På bakgrunn av dette er det lagt til spørsmål underveis i oppgavene slik at elevene må reflektere over erfaringene. Dette stemmer overens med Dewey (2001) som poengterer viktigheten av å tenke over aktivitetene som gjennomføres. I denne studien ble det benyttet spørsmål for å få elevene til å tenke over simuleringene de gjennomførte, men det var ikke tenkt på i programmeringsarbeidet. I min egen studie la jeg derfor til spørsmål til programmeringsarbeidet. Det vil ofte være enkelt å tenke over at elevene må diskutere fysiske opplevelser, men ofte i mindre grad andre former for oppgaver. For fremtidige studier og undervisningsopplegg, der målsettingen innebærer å skape rom for refleksjon hos elevene, kan det være fornuftig å heller ta med denne faktoren allerede ved utarbeidelse av undervisningsopplegget.

Et mer spesifikt funn tilknyttet bruk av micro:bit i undervisningssammenheng, er at sendereffekten mellom micro:bitene bør tas med som en faktor. I denne studien fant jeg at sendereffekten hadde betydning for hvor godt simuleringen fungerte, og ved feil bruk av sendereffekt kan elevene få mindre læring da resultatene blir annerledes enn tiltenkt. Dette påvirker hensikten med simuleringen, og vil dermed ikke kunne vise det naturfaglige fenomenet på den tiltenkte måten som er poenget med en modell (Schwarz et al., 2009). Jeg la opplysninger om denne utfordringen i lærerveiledningen da dette ble oppdaget. Ideelt sett bør det imidlertid utarbeides en matrise over hvilke innstillinger som er korrekt

i forhold til romstørrelse og antall elever som deltar i opplegget. Ovenstående innspill kan danne inspirasjon til eventuell videreutvikling av dette undervisningsopplegget.

5 AVSLUTNING OG VEIEN VIDERE

Formålet med denne studien er å bidra til at programmering kan integreres i undervisningen om smittespredning og vaksinerings. Studien har benyttet metoden pedagogisk designforskning med de tre fasene utvikling, gjennomføring og retrospektiv analyse. Gjennom denne studien har et undervisningsopplegg bestående av tre oppgavehefter og en lærerveiledning blitt utviklet og revidert.

Hovedfunnene i denne studien gir grunnlag for videreutvikling både spesifikt opp mot dette undervisningsopplegget, men funnene bør også tas i betraktning dersom andre skal utvikle et liknende undervisningsopplegg. Det som har fungert gjennomgående i denne studien er å benytte diskusjon der flere elementer trekkes sammen, og å starte programmeringsarbeidet med en felles oppgave uavhengig av nivå. Det anbefales å vurdere å bruke tilsvarende metodikk i fremtidige utviklinger.

Studien viser at det på generelt grunnlag er viktig å ta hensyn til nivå-differensiering både hos lærer og elev allerede i utviklingsfasen, viktigheten av innspill på modellen for å oppnå nødvendige forbedringer, og hvor viktig det er allerede innledningsvis å legge opp til refleksjon hos elevene i forhold til oppgaven de skal utføre. Det kan være fornuftig å tenke over om disse funnene bør tas i betraktning allerede i forberedelsesfasen, fremfor i revideringen som ble gjort i denne studien.

Tilbakemeldingene i løpet av studien vitner om at undervisningsopplegget utviklet gjennom denne studien fremstår både interessant og nyttig: Lærer på 8. trinn «*Jeg kunne tenkt med å benytte dette undervisningsopplegget med den andre klassen jeg har i naturfag når de skal lære om smittespredning og vaksinerings*» Lærer på 10. trinn: «*Spennende, kunne tenkt meg å bruke det senere og en fin måte å få inn programmeringsmålet med sender og mottaker på en litt annen måte*» og fremstår som gode tilbakemeldinger.

Disse uttalelsene peker i retning av at dette undervisningsopplegget egner seg godt til å bidra til økt læring av både programmering, smittespredning og vaksinerings. Det er dog ikke gjennomført en reell uttesting av læringsutbyttet i denne studien. Det kan være hensiktsmessig å foreta en slik uttesting for å vurdere i hvor stor grad dette undervisningsopplegget har for elevenes læringsutbytte.

Det kan videre vurderes om modellen som ligger til grunn for undervisningsopplegget bør endres, med tanke på om mottaker som smittespreder vil gi en mer realistisk fremstilling. Det kan videre være hensiktsmessig å ta for seg sendereffekten i selve micro:biten for å få en så realistisk utprøving som mulig. Dette er kommentert i lærerveiledningen, men det kan være fornuftig å vurdere om det bør utarbeides en matrise som viser innstillingene i forhold til både romstørrelse og elevmasse slik at hver enkelt lærer unngår problemstillingen.

Denne studien er lisensiert med CC BY-SA 4.0 DEED. Dette innebærer at undervisningsopplegget i denne studien kan deles og bearbeides (creative commons, u.å.). Jeg oppfordrer med dette andre til å bruke og videreutvikle dette undervisningsopplegget.

6 LITTERATURLISTE

- Balanskat, A., & Engelhardt, K. (2015). *Computing our future: Computer programming and coding—Priorities, school curricula and initiatives across Europe*. European Schoolnet.
http://www.eun.org/documents/411753/817341/Computing+our+future_final_2015.pdf/d3780a64-1081-4488-8549-6033200e3c03
- Ball, T, Protzenko, J, Bishop, J, Moskal, M, J. de Halleux, Braun, M, Hodges, S, & Riley, C. (2016). Microsoft Touch Develop and the BBC micro:bit. I *ICSE '16: Proceedings of the 38th International Conference on Software Engineering Companion* (s. 637–640). IEEE. <https://doi.org/10.1145/2889160.2889179>
- Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, 84(2), 191–215. <https://doi.org/10.1037/0033-295X.84.2.191>
- Bandura, A. (1994). Self-efficacy. I V. S. Ramachaudran (Red.), *Encyclopedia of human behavior* (Bd. 4, s. 71–81). Academic Press.
- Berssanette, J. H. & de Francisco, A. C. (2022). Cognitive Load Theory in the Context of Teaching and Learning Computer Programming: A Systematic Literature Review. *IEEE Transactions on Education*, 65(3), 440–449.
<https://doi.org/10.1109/TE.2021.3127215>
- Bjørndal, K. E. W. (2013). Pedagogisk designforskning—En forskningsstrategi for å fremme bedre undervisning og læring. I M. Brekke & T. Tiller (Red.), *Læreren som forsker: Innføring i forskningsarbeidet i skolen* (s. 245–259). Universitetsforlaget.
- Caballero, M. D., Burk, J. B., Aiken, J. M., Thoms, B. D., Douglas, S. S., Scanlon, E. M., & Schatz, M. F. (2014). Integrating Numerical Computation into the Modeling Instruction Curriculum. *The Physics Teacher*, 52(1), 38–42.
<https://doi.org/10.1119/1.4849153>
- Cao, L., & Xu, P. (2005). Activity Patterns of Pair Programming. I *Proceedings of the 38th Annual Hawaii International Conference on System Sciences: 2005 (Hawaii International Conference on System Sciences: Proceedings)* (s. 88a–88a). IEEE.
<https://doi.org/10.1109/HICSS.2005.66>
- Charntaweekhun, K., & Wangsiripitak, S. (2006). Visual Programming using Flowchart. I *2006 International Symposium on Communications and Information Technologies* (s. 1062–1065). IEEE. 10.1109/ISCIT.2006.339940
- Colella, V. (2000). Participatory Simulations: Building Collaborative Understanding Through Immersive Dynamic Modeling. *Journal of the Learning Sciences*, 9(4), 471–500. https://doi.org/10.1207/S15327809JLS0904_4
- creative commons. (u.å.). CC BY-SA 4.0 DEED. Hentet 8. mai fra <https://creativecommons.org/licenses/by-sa/4.0/>.

- Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair Programming: Under What Conditions Is It Advantageous for Middle School Students? *Journal of Research on Technology in Education*, 46(3), 277–296. <https://doi.org/10.1080/15391523.2014.888272>
- Dewey, J. (2001). Erfaring og læring. I E. L. Dale (Red.), *Om utdanning: Klassiske tekster* (s. 53–66). Gyldendal Akademisk.
- Dieleman, H., & Huisingh, D. (2006). Games by which to learn and teach about sustainable development: Exploring the relevance of games and experiential learning for sustainability. *Journal of Cleaner Production*, 14(9–11), 837–848. <https://doi.org/10.1016/j.jclepro.2005.11.031>
- Ericson, B., Margulieux, L., & Rick, J. (2017). Solving parsons problems versus fixing and writing code. I *Koli Calling '17: Proceedings of the 17th Koli Calling International Conference on Computing Education Research* (s. 20–29). Association for Computing Machinery. <https://doi.org/10.1145/3141880.3141895>
- Erlien, W., Sørborg, Ø., Gregers, T., Haug, B., & Korsager, M. (2021, mai). *Virus og vaksine*. Viten. <https://www.viten.no/filarkiv/virus/#/>
- European Centre for Disease Prevention and Control. (2024). *Measles on the rise in the EU/EEA: Considerations for public health response*. European Centre for Disease Prevention and Control. <https://data.europa.eu/doi/10.2900/064162>
- Europeisk informasjonsportal om vaksinasjon. (2020, 13. mars). *Hvor effektiv er vaksinen?* <https://vaccination-info.europa.eu/nb/om-vaksiner/hvordan-vaksiner-virker/hvor-effektiv-er-vaksinen>
- Fleury, A. E. (1991). Parameter passing: The rules the students construct. *ACM SIGCSE Bulletin*, 23(1), 283–286. <https://doi.org/10.1145/107004.107066>
- Folkehelseinstituttet. (2022, 27. januar). *Meslinger*. HelseNorge. <https://www.helsenorge.no/sykdom/infeksjon-og-betennelse/meslinger/>
- Furuseth, E., Rydland, K. M., Berg, A. S., Hansen, B. L. T. T., Daae, A. O., Greve-Isdahl, M., & Solberg, S. L. (2023, 7. februar). *Vaksiner i Norge*. Folkehelseinstituttet. <https://www.fhi.no/he/folkehelse rapporten/smitte/vaksiner-i-norge/?term=>
- Gilbert, J. K., Boulter, C. J., & Elmer, R. (2000). Positioning Models in Science Education and in Design and Technology Education. I J. K. Gilbert & C. J. Boulter (Red.), *Developing Models in Science Education* (s. 3–17). Springer Netherlands. https://doi.org/10.1007/978-94-010-0876-1_1
- Gjøvik, Ø., & Høyland, J. (2022). *Kloss for Kloss: Blokkprogrammering for lærere*. Universitetsforlaget.
- Gold, R. L. (1958). Roles in Sociological Field Observations. *Social Forces*, 36(3), 217–223. <https://doi.org/10.2307/2573808>

- Gravemeijer, K., & Cobb, P. (2006). Design research from a learning design perspective. I J. van den Akker, K. Gravemeijer, S. Mckenney, & N. Nieveen (Red.), *Educational design research* (s. 17–51). Routledge.
- Grindeland, J. M., Staberg, R. L., & Tandberg, C. (2020). *Biologi for lærere: Naturfag i grunnskolelærerutdanninga* (2. utg.). Gyldendal Akademisk.
- Hallström, J., & Schönborn, K. J. (2019). Models and modelling for authentic STEM education: Reinforcing the argument. *International Journal of STEM Education*, 6(1), 22. <https://doi.org/10.1186/s40594-019-0178-z>
- Hanks, B., Fitzgerald, S. C., McCauley, R., Murphy, L., & Zander, C. (2011). Pair programming in education: A literature review. *Computer Science Education*, 21(2), 135–173. <https://doi.org/10.1080/08993408.2011.579808>
- Hansen, A. (2024, 1. mai). Meslingutbrudd—Påvist i Norge. *Dagbladet*. <https://www.dagbladet.no/nyheter/pavist-i-norge/81337338>
- Haraldsrud, A. D., Sveinsson, H. A., & Løvold, H. H. (2020). *Programmering i skolen*. Universitetsforlaget.
- Harms, K. J., Chen, J., & Kelleher, C. L. (2016). Distractors in Parsons Problems Decrease Learning Efficiency for Young Novice Programmers. I *ICER '16: Proceedings of the 2016 ACM Conference on International Computing Education Research* (s. 241–250). Association for Computing Machinery. <https://doi.org/10.1145/2960310.2960314>
- Hooshyar, D., Ahmad, R., Md Nasir, M. H. N., Samshirband, S., & Horng, S.-J. (2015). Flowchart-Based Programming Environments for Improving Comprehension and Problem-Solving Skill of Novice Programmers: A Survey. *International Journal of Advanced Intelligence Paradigms*, 7(1), 24–56. <https://doi.org/10.1504/IJAIP.2015.070343>
- Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys*, 37(2), 83–137. <https://doi.org/10.1145/1089733.1089734>
- Klopfer, E., Yoon, S. A., & Perry, J. (2005). Using Palm Technology in Participatory Simulations of Complex Systems: A New Take on Ubiquitous and Accessible Mobile Computing. *Journal of Science Education and Technology*, 14(3), 285–297. <https://doi.org/10.1007/s10956-005-7194-0>
- Kunnskapsdepartementet. (2019). *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>
- Lai, M. -H., & Lai, C.-S. (2012). Using Computer Programming to Enhance Science Learning for 5th Graders in Taipei. I *2012 International Symposium on Computer, Consumer and Control* (s. 146–148). IEEE. <https://doi.org/10.1109/IS3C.2012.45>

- Mahn, H. (1999). Vygotsky's Methodological Contribution to Sociocultural Theory. *Remedial and Special Education, 20*(6), 341–350. <https://doi.org/10.1177/074193259902000607>
- micro:bit. (u.å.). *Let's code*. Hentet 19. februar 2024 fra <https://microbit.org/code/>.
- Mørch, A. I., Litherland, K. T., & Andersen, R. (2019). End-User Development Goes to School: Collaborative Learning with Makerspaces in Subject Areas. I A. Malizia, S. Valtolina, A. Morch, A. Serrano, & A. Stratton (Red.), *End-User Development: 7th International Symposium, IS-EUD 2019, Hatfield, UK, July 10–12, 2019, Proceedings* (s. 200–208). Springer International Publishing. <https://doi.org/10.1007/978-3-030-24781-2>
- NESH. (2021) *Forskningsetiske retningslinjer for samfunnsvitenskap og humaniora* (5 utg., redigert 2023). De nasjonale forskningsetiske komiteene. <https://www.forskningsetikk.no/globalassets/dokumenter/4-publikasjoner-som-pdf/forskningsetiske-retningslinjer-for-samfunnsvitenskap-og-humaniora.pdf>
- Nia, M. G., & de Vries, M. J. (2017). Models as artefacts of a dual nature: A philosophical contribution to teaching about models designed and used in engineering practice. *International Journal of Technology and Design Education, 27*(4), 627–653. <https://doi.org/10.1007/s10798-016-9364-1>
- NTNU. (u.å.). *Lagringsguide*. Hentet 3. april 2024 fra <https://i.ntnu.no/wiki/-/wiki/Norsk/Lagringsguide>.
- Opplæringslova. (1998). *Lov om grunnskolen og den vidaregåande opplæringa (LOV-1998-07-17-61)*. Lovdata. <https://lovdata.no/dokument/NL/lov/1998-07-17-61>
- Parsons; D. (2021). Mobile Participatory Simulation of COVID-19 Transmission Using the micro:bit. I *2021 IEEE International Conference on Engineering, Technology & Education (TALE)* (s. 1–4). IEEE. <https://doi.org/10.1109/TALE52509.2021.9678867>
- Parsons, D., & Haden, P. (2006). Parson's programming puzzles: A fun and effective learning tool for first programming courses. I D. Tolhurst & S. Mann (Red.), *ACE '06: Proceedings of the 8th Australasian Conference on Computing Education* (s. 157–163). Australian Computer Society, Inc.
- Peffer, M., Renken, M., Girault, I., Chiocciariello, A., & Otreel-Cass, K. (2016). Distinctions Between Computer Simulations and Other Technologies for Science Education. I M. Renken, M. Peffer, K. Otreel-Cass, I. Girault, & C. Augusto (Red.), *Simulations as Scaffolds in Science Education, in SpringerBriefs in Educational Communications and Technology*. Springer Cham. <https://doi.org/10.1007/978-3-319-24615-4>
- Plutzer, E., & Warner, S. B. (2021). A potential new front in health communication to encourage vaccination: Health education teachers. *Vaccine, 39*(33), 4671–4677. <https://doi.org/10.1016/j.vaccine.2021.06.050>
- Postholm, M. B. (2010). *Kvalitativ metode: En innføring med fokus på fenomenologi, etnografi og kasusstudier* (2. utg.). Universitetsforlaget.

- PRIMMportal. (u.å.). *What is PRIMM?* Hentet 4. mars 2024 fra <https://primmportal.com/>.
- Qian, Y., & Lehman, J. D. (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>
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for All. *Communications of the ACM, 52*(11), 60–67. <https://doi.org/10.1145/1592761.1592779>
- Rossing, N. K., & Johansen, A. (2021). MICRO:BIT Radiokommunikasjon – Forslag til undervisningsopplegg. *Skolelaboratoriet ved NTNU*. <https://www.ntnu.no/documents/2004699/12108297/MicroBit+Radiokommunikasjon+-+forslag+til+undervisningsopplegg.pdf/9e0e6a25-1cc6-5051-4820-1dcfdb801205?t=1642515239732>
- Schwarz, C. V., Reiser, B. J., Davis, E., Kenyon, L., Achér, A., Fortus, D., Shwartz, Y., Hug, B., & Krajcik, J. (2009). Developing a learning progression for scientific modeling: Making scientific modeling accessible and meaningful for learner. *Journal of Research in Science Teaching, 46*(6), 632–654. <https://doi.org/10.1002/tea.20311>
- Schaathun, S. (u.å.). *PXT: Smitte:Bit*. Lær kidsa koding. Hentet 6. februar 2024 fra https://oppgaver.kidsakoder.no/microbit/smitte_bit/smitte_bit.
- Sentance, S., & Waite, J. (2017). PRIMM: Exploring pedagogical approaches for teaching text-based programming in school. I E. Barendsen & P. Hubwieser (Red.), *WiPSCE '17: Proceedings of the 12th Workshop on Primary and Secondary Computing Education* (s. 113–114). Association for Computing Machinery. <https://doi.org/10.1145/3137065.3137084>
- Sentance, S., Waite, J., Hodges, S., MacLeod, E., & Yeomans, L. (2017). «Creating Cool Stuff»: Pupils' Experience of the BBC micro:bit. I *SIGCSE '17: Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (s. 531–536). Association for Computing Machinery. <https://doi.org/10.1145/3017680.3017749>
- Sentance, S., Waite, J., & Kallia, M. (2019). Teaching computer programming with PRIMM: a sociocultural perspective. *Computer Science Education, 29*(3), 1–41. <https://doi.org/10.1080/08993408.2019.1608781>
- Sevik, K. (2016). *Programmering i skolen. Notat fra senter for IKT i utdanningen*. https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf
- Sikt. (u.å.). *Barnehage- og skuleforskning*. Hentet 2. april 2024 fra <https://sikt.no/tjenester/personverntjenester-forskning/personvernhandbok-forskning/barnehage-og-skuleforskning>.
- Steineger, E., & Wahl, A. (2021). *Naturfag 10*. Cappelen Damm.

- Stenlund, E. (2021). *Programmering og Fagfornyelsen: En casestudie om lærernes nye situasjon og holdninger til programmering i skolen* [Masteroppgave, Universitetet i Oslo]. DOU Vitenarkiv.
https://www.duo.uio.no/bitstream/handle/10852/87187/Masteroppgave_Erlend_Stenlund.pdf?sequence=1&isAllowed=y
- Sweller, J., van Merriënboer, J. J. G., & Paas, F. G. W. C. (1998). Cognitive Architecture and Instructional Design. *Educational Psychology Review*, 10(3), 251–296.
<https://doi.org/10.1023/A:1022193728205>
- Saabye, M. (Red.). (2019). *Læreplanverket for Kunnskapsløftet 2020—Grunnskolen*. Pedlex.
- Tjora, A. (2021). *Kvalitative forskningsmetoder i praksis* (4. utg.). Gyldendal.
- Treagust, D. F., Chittleborough, G. D., & Mamiala, T. L. (2002). Students' understanding of the role of scientific models in learning science. *International Journal of Science Education*, 24(4), 357–368.
<https://doi.org/10.1080/09500690110066485>
- van den Akker, J., Gravemeijer, K., Mckenney, S., & Nieveen, N. (2006). Introducing educational design research. I J. van den Akker, K. Gravemeijer, S. Mckenney, & N. Nieveen (Red.), *Educational Design Research* (s. 3–7). Routledge.
- Vik, J. N. (2024, 19. mars). Meslingutbrudd: - Vil kunne komme tilfeller i Norge. TV2.
<https://www.tv2.no/nyheter/innenriks/vil-kunne-komme-tilfeller-i-norge/16549383/>
- Vygotsky, L. S. (1978). *Mind in Society* (M. Cole, V. John-Steiner, S. Scribner, & E. Souberman, Red.). Harvard University Press. <https://doi.org/10.2307/j.ctvjf9vz4>
- Williams, L. A., & Kessler, R. R. (2000). All I really need to know about pair programming I learned in kindergarten. *Communications of the ACM*, 43(5), 108–114. <https://doi.org/10.1145/332833.332848>
- Williams, L., Mccrickard, D. S., Layman, L., & Hussein, K. (2008). Eleven Guidelines for Implementing Pair Programming in the Classroom. I G. Melnik, P. Kruchten, & M. Poppendieck (Red.), *Agile 2008 Conference* (s. 445–452). IEEE.
<https://doi.org/10.1109/Agile.2008.12>
- World Health Organization. (2023, 16. november). *Global measles threat continues to grow as another year passes with millions of children unvaccinated*.
<https://www.who.int/news/item/16-11-2023-global-measles-threat-continues-to-grow-as-another-year-passes-with-millions-of-children-unvaccinated>
- Zhi, R., Chi, M., Barnes, T., & Price, T. W. (2019). Evaluating the Effectiveness of Parsons Problems for Block-based Programming. I *ICER '19: Proceedings of the 2019 ACM Conference on International Computing Education Research* (s. 51–59). Association for Computing Machinery.
<https://doi.org/10.1145/3291279.3339419>

7 VEDLEGG

Vedlegg 1: Oppgaveheftet del 1: Radiokommunikasjon

Vedlegg 2: Oppgaveheftet del 2: Smittespredning

Vedlegg 3: Oppgaveheftet del 3: Vaksinerings

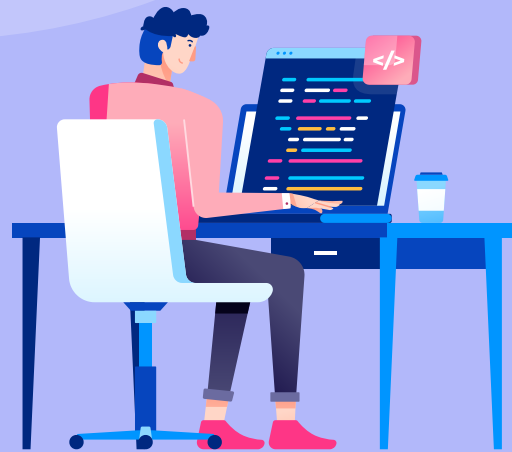
Vedlegg 4: Lærerveiledning

Vedlegg 5: Godkjent Sikt-søknad

7.1 VEDLEGG 1: OPPGAVEHEFTET DEL 1: RADIOKOMMUNIKASJON

SMITTESPREDNING OG VAKSINERING

PROGRAMMERING AV MICRO:BIT



OPPGAVEHEFTE

DEL 1: RADIOKOMMUNIKASJON



SYNNE CHRISTIN COLLING

Smittespredning og vaksinerings - Programmering av micro:bit

Oppgavehefte del: 1: Radiokommunikasjon

Synne Christin Colling

Trondheim 2024

Bilde av micro:bit:

micro:bit Board, 2017, av SparkFun Electronics (<https://n9.cl/7xtm1>) CC BY 2.0 DEED. Retusjering: fjernet bakgrunn.

Illustrasjoner fra Canva

Designet i Canva

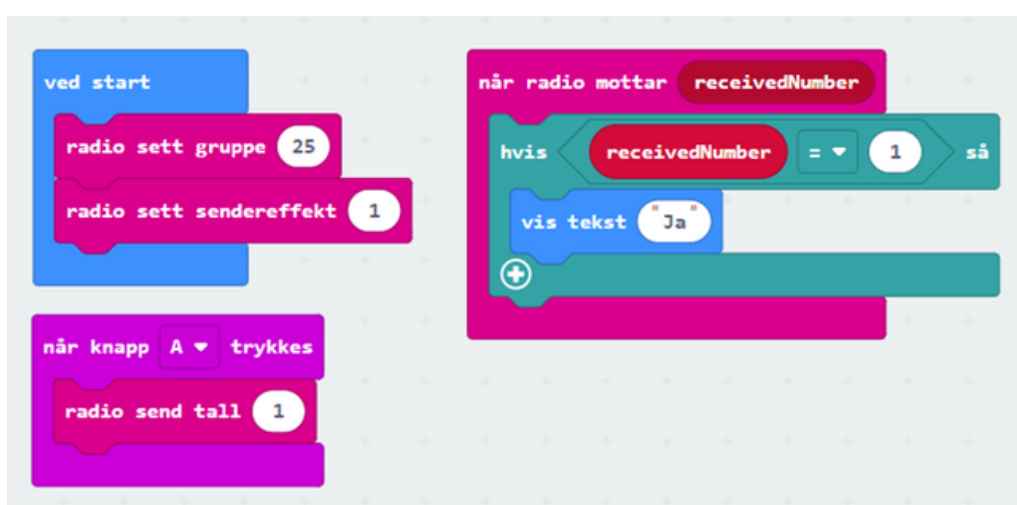
Micro:bitene kan kommunisere med hverandre

Dette skjer gjennom radiokommunikasjon. For at de skal kunne kommunisere med hverandre må de være i samme radiogruppe. Alle parene som samarbeider vil få sin egen radiogruppe av læreren som skal brukes i alle oppgavene i heftet.

For å kunne laste ned og redigere prosjektene må dere trykke på rediger øverst i høyre hjørne etter å ha trykket på lenken.

Oppgave 1

A) Diskuter sammen. Hva tror dere programmet under kommer til å gjøre?



B) Last ned programmet i lenken under på micro:biten og test den ut. Husk å bytte til deres radiogruppe. Stemte det dere trodde kom til å skje?

- Lenke til prosjektet: https://makecode.microbit.org/_Pky81zLbe7r5

Oppgave 2

I lenken vil dere finne et program med noen løse klosser. Pusle sammen klossene slik at dere får et program der dere kan svare både ja og nei med micro:biten.

- Lenke til prosjektet: https://makecode.microbit.org/_aqz07o72K5WK

Ferdig?

Bruk kreativiteten og få micro:bitene til å sende andre meldinger eller symboler til hverandre.

Trenger dere litt hjelp?

Se på programmet fra oppgave 1, husker dere hva denne gjorde?

Oppgave 3

Hvordan kan radiokommunikasjon brukes til å simulere smittespredning?

Litteraturliste

- Rossing, N. K., & Johansen, A. (2021). MICRO:BIT Radiokommunikasjon – Forslag til undervisningsopplegg. Skolelaboratoriet ved NTNU.
<https://www.ntnu.no/documents/2004699/12108297/MicroBit+Radiokommunikasjon++forslag+til+undervisningsopplegg.pdf/9e0e6a25-1cc6-5051-4820-1dcfdb801205?t=1642515239732>

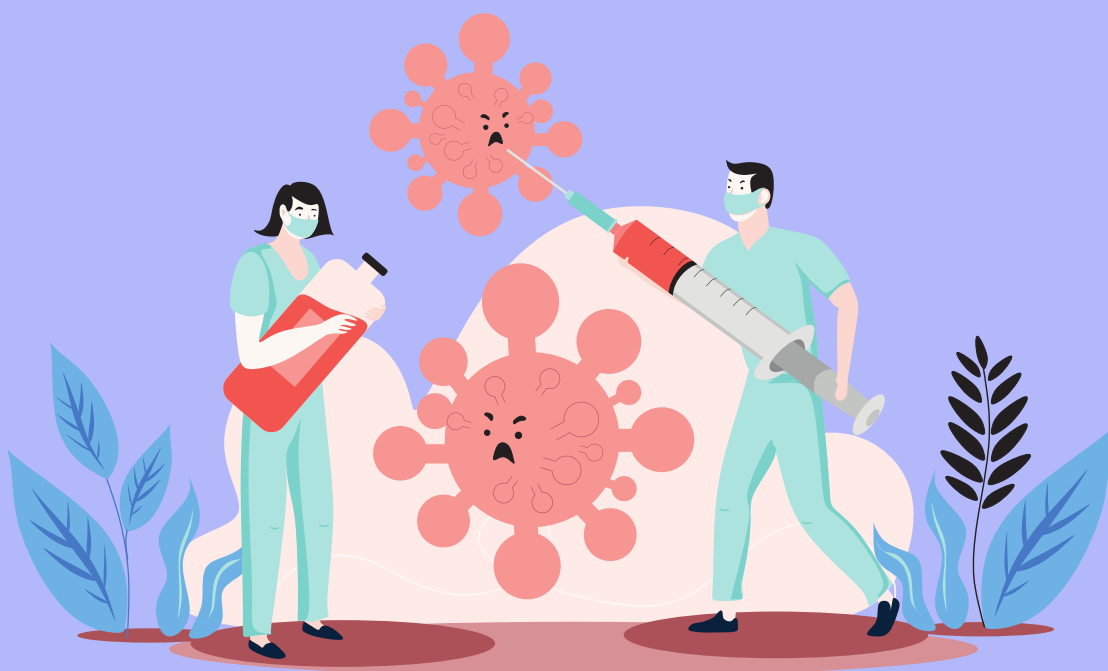
7.2 VEDLEGG 2: OPPGAVEHEFTET DEL 2: SMITTESPREDNING

SMITTESPREDNING OG VAKSINERING

PROGRAMMERING AV MICRO:BIT



OPPGAVEHEFTE DEL 2: SMITTESPREDNING



SYNNE CHRISTIN COLLING

Smittespredning og vaksinerings - Programmering av micro:bit

Oppgavehefte del: 2: Smittespredning

Synne Christin Colling

Trondheim 2024

Bilde av micro:bit:

micro:bit Board, 2017, av SparkFun Electronics (<https://n9.cl/7xtm1>) CC BY 2.0 DEED. Retusjering: fjernet bakgrunn.

Illustrasjoner fra Canva

Designet i Canva

Mikroorganismer er en grunn til at mennesker blir syke

Mennesker kan bli syke gjennom at mikroorganismer kommer seg inn i kroppen vår og begynner å formere seg. At vi blir syke er et tegn på at immunforsvaret har satt i gang. Immunforsvaret prøver da å fjerne de fremmede mikroorganismene som ikke hører til i kroppen vår. Disse mikroorganismene kan være virus, bakterier og sopp, og er så små at man ikke kan se dem uten mikroskop. Hvis en person er smittsom vil personen ved å nyse, hoste eller snakke spre smitte i lufta. Bittesmå dråper som kommer ut fra munnen og nesa inneholder mikroorganismene som spres i lufta. Vi blir utsatt for smitte nesten overalt, og derfor er immunforsvaret viktig.



Ikke alle mikroorganismer gjør deg syk

Immunforsvaret består av tre hoveddeler

Disse tre er det medfødte ytre immunforsvaret, det medfødte indre immunforsvaret og det tilpassede indre immunforsvaret.

Det medfødte ytre immunforsvaret

Det medfødte ytre immunforsvaret beskytter kroppen vår slik at mikroorganismer ikke kommer inn. Dette består av huden, flimmerhår, slimhinner og magesyren. Den sure magesyren dreper bakteriene som vi svelger. Slimhinnene finner vi i kroppsåpningene, slik som i munnen og nesa, og slipper ut slim med stoffer som gjør at bakteriene har vanskeligheter med å komme inn i kroppscellene. I nesa og på luftrøret sitter det også flimmerhår som beveger seg og flytter partiklene vi puster inn videre til svelget. Huden beskytter kroppen mot inntrengere så lenge den er hel og ikke har noen sår.

Det medfødte indre immunforsvaret

Dersom mikroorganismene kommer seg forbi det ytre immunforsvaret setter det indre medfødte immunforsvaret i gang. Dette består av fagocytter som er ulike hvite blodceller som "spiser" og ødelegger uvedkomne inntrengere.

Det tilpassede indre immunforsvaret

Det ytre og medfødte indre immunforsvaret klarer å stoppe de fleste fremmede mikroorganismene. Men noen kommer forbi og da må det tilpassede indre immunforsvaret ut i kamp. Dette består av lymfocytter som er hvite blodceller.

Vi har to ulike hovedtyper lymfocytter: T-celler og B-celler. Disse kjenner igjen hvert sitt smittestoff. De fungerer litt forskjellig, men begge har i oppgave å kjenne igjen og angripe uvedkommende mikroorganismer. Dette skal de gjøre uten å røre kroppens egne celler.

Antistoffer er noe mange av B-cellene kan lage. Dette er proteiner som kjenner igjen og ødelegger uvedkommende mikroorganismer. Antistoffene kan være lenge i blodet, og neste gang de samme mikroorganismene kommer seg inn i kroppen din vil de ikke gjøre deg syk. Dette er fordi du får en immunitet mot sykdommen i en viss periode.

Vi har også en spesiell type B-celle som kalles hukommelsescelle, som kan være lenge i blodet. Noen av dem kan vi ha hele livet. Disse har som oppgave å kjenne igjen uvedkommende mikroorganismer, og angriper raskt neste gang de kommer seg inn i kroppen.



For å kunne laste ned og redigere prosjektene må dere trykke på rediger øverst i høyre hjørnet etter å ha trykket på lenken.

Oppgave 1

A) Diskuter sammen. Hva tror dere programmet under kommer til å gjøre?

A screenshot of a Scratch script editor showing a program with the following logic:

- ved start** (when green flag clicked):
 - radio sett gruppe 25
 - radio sett sendereffekt 1
 - sett Tid til 0
- når radio mottar receivedNumber** (when radio receives number):
 - hvis receivedNumber = 1 så (if receivedNumber equals 1 then):
 - sett Tid til 10
- gjenta for alltid** (repeat forever loop):
 - hvis Tid > 0 så (if Tid is greater than 0 then):
 - vis tall Tid
 - endre Tid med -1
 - pause (ms) 1000
 - hvis Tid = 0 så (if Tid equals 0 then):
 - vis ikon [grid icon]
- når knapp A+B trykkes** (when A+B button is pressed):
 - radio send tall 1

B) Last ned programmet fra lenken og kjør det. Husk å bytte radiogruppe til den dere har fått utdelt av læreren for å teste programmet. Stemte det dere trodde kom til å skje?

- Lenke til prosjektet: https://makecode.microbit.org/_1UwHpjad1Yv9

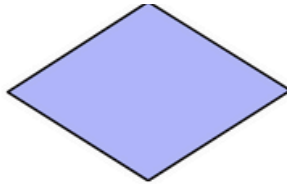
Oppgave 2

I denne oppgaven skal dere programmere micro:biten slik at vi kan bruke dem til å simulere smittespredning. I programmeringsfeltet vil dere finne mange løse klosser som dere skal sette sammen. Dere skal kun bruke disse klossene, og alle skal brukes opp. For å sette disse sammen i riktig rekkefølge har dere fått litt hjelp. På neste side vil dere finne tre flytskjema som forteller hva programmet skal gjøre.

- Lenke til prosjektet: https://makecode.microbit.org/_VT16fTHMU3ku



Der koden starter



Her spør koden et spørsmål.
Spørsmålet besvares enten ja eller nei, og bestemmer hva som skjer videre



Det koden faktisk skal gjøre

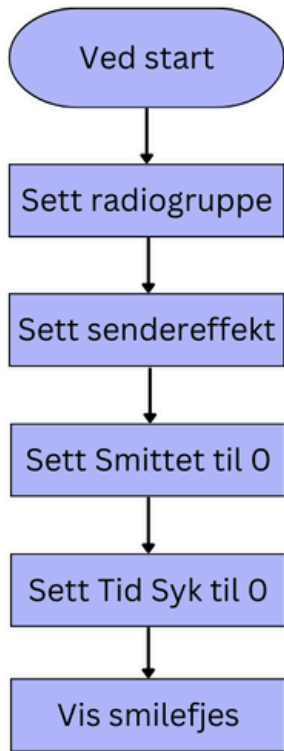
Bildene på neste side består av de tre symbolene ovenfor. Under hvert symbol ser dere hva symbolet betyr. Pilene mellom symbolene forteller dere hva som skal komme etter.

Trenger dere litt hjelp?

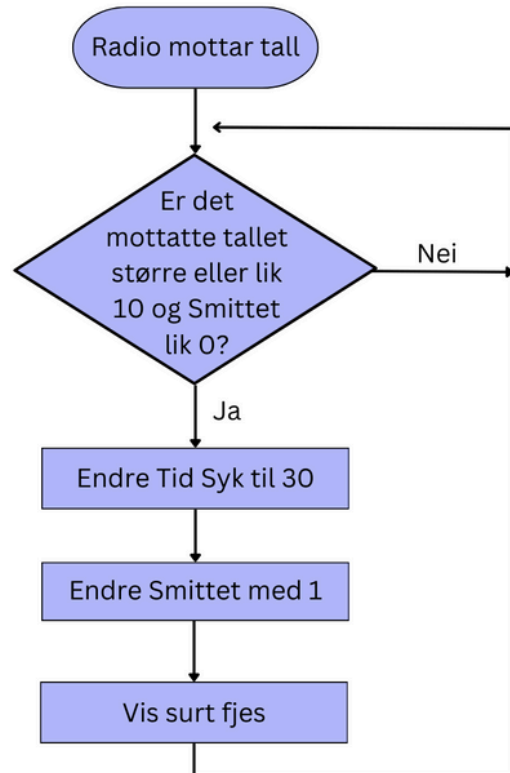
Trykk på denne lenken og jobb videre med dette programmet:

https://makecode.microbit.org/_C7WaF0fk7HML

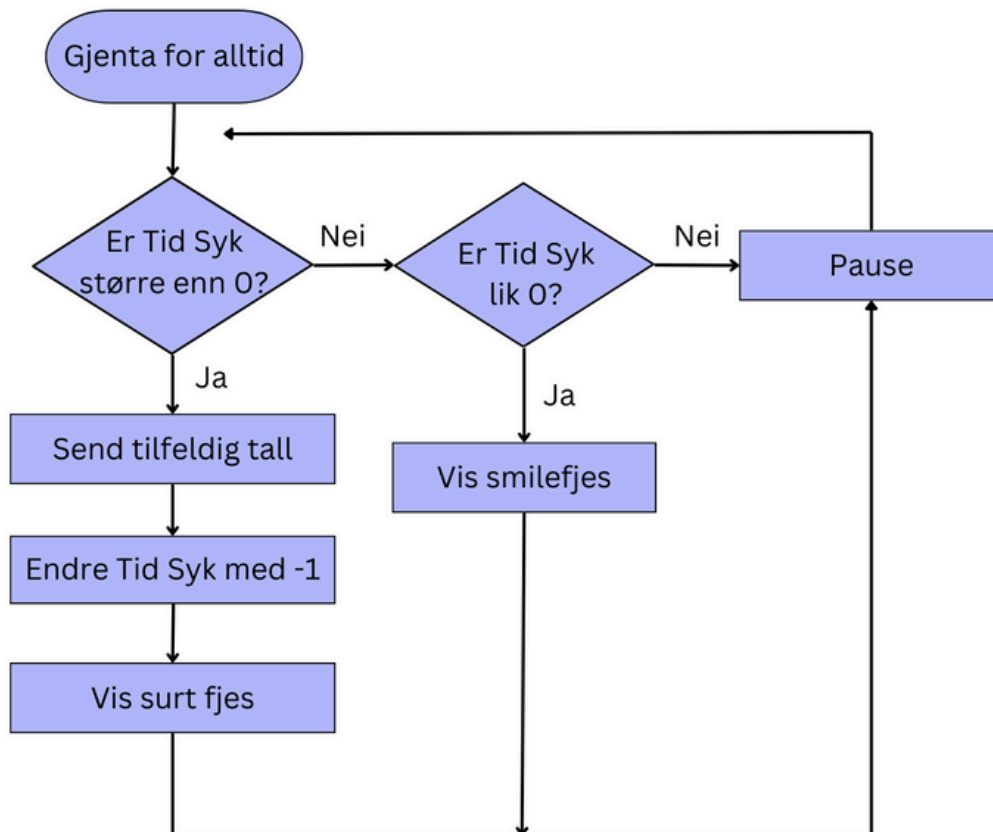
Flytskjema 1: Hva programmet skal gjøre ved start



Flytskjema 2: Hva programmet skal gjøre når radioen mottar et tall



Flytskjema 3: Hva programmet skal gjøre i “gjenta for alltid”-løkka



Spørsmål til programmeringen

1. Hvilke løkker finner dere i dette programmet, og hvordan brukes de?
2. Hvilke vilkår finner dere i dette programmet?
3. Hvorfor har dere valgt å plassere vilkårene der dere har gjort?
4. Hva forteller disse vilkårene oss?
5. Hvilke variabler finner dere i programmet?
6. Hvorfor har dere valgt å plassere variablene der dere har gjort?
7. Hva brukes disse variablene til, og hva skal de hjelpe med å vise?

Ferdig?

1. Kan dere endre tall i programmet for å simulere at noen har et svakere immunforsvar. Finner dere mer enn én måte?
2. Kan dere endre programmet slik at det kan simulere at det er mulig, men veldig liten sjanse for å dø av sykdommen?

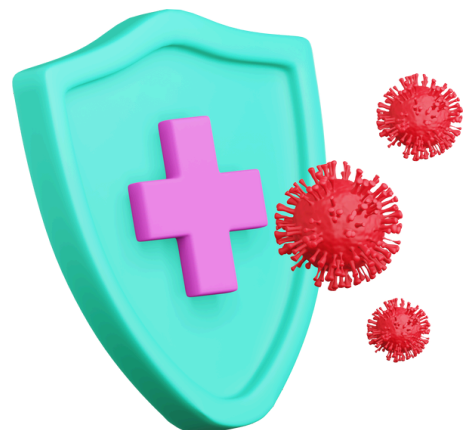
Oppgave 3

Vi skal nå teste smittespredning med hele klassen. Vi skal bruke programmet som dere lagde i oppgave 2. Dere får vite av læreren hvem som er smittsom fra start. Vi skal gjennomføre tre tester:

1. Stående stille. Dere skal her se hvordan smitten sprer seg.
2. Gående rundt. Målet er å ikke bli syk.
3. Gående rundt. Målet er å bli fortest mulig syk.

Spørsmål til etter testingen

- A) Hvordan spredde smitten seg da dere sto stille?
- B) Hva gjorde dere for å prøve å ikke bli syke? Og hvorfor?
- C) Ble dere syke selv om dere prøvde å ikke bli syke?
- D) Hva gjorde dere for å prøve å bli syke? Og hvorfor?
- E) Tok det lang tid før dere ble syke når dere prøvde å bli det?

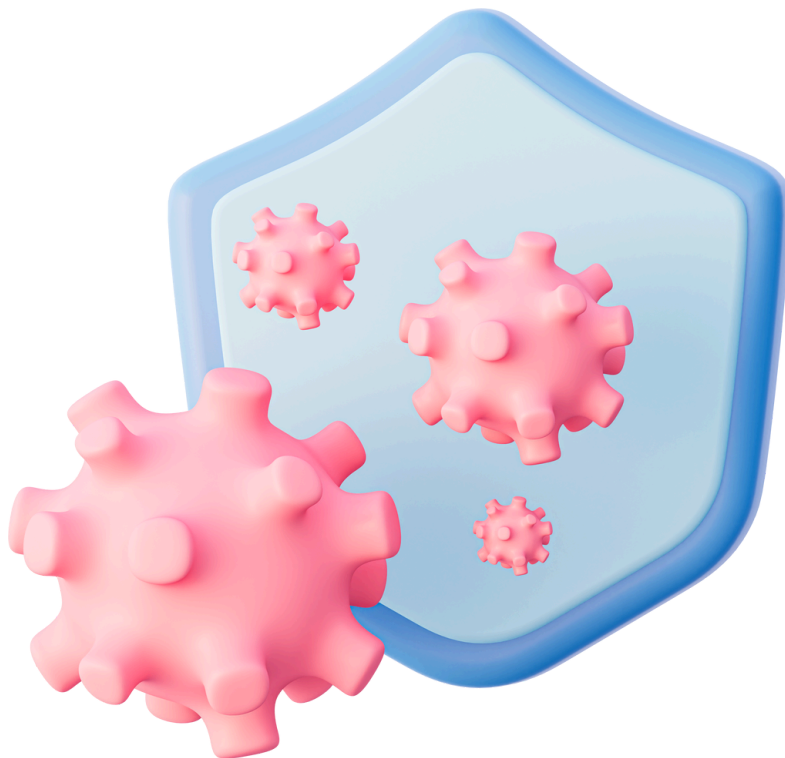


Oppgave 4

- A) Diskuter hvordan denne simuleringen kan være med på å vise hvordan sykdom sprer seg.
- B) Diskuter hvordan denne simuleringen kan være med på å vise hvordan immunforsvaret fungerer.

Oppgave 5

- A) Simuleringer har sine svakheter og styrker. Hva tenker dere er denne simuleringen sine styrker og svakheter?
- B) Fungerte alt som det skulle under testingen av simuleringen? Hva kan eventuelt være grunnen til at noe ikke fungerte som det skulle?



Litteraturliste

- Grindeland, J. M., Staberg, R. L., & Tandberg, C. (2020). Biologi for lærere (2. utg.). Gyldendal Akademisk.
- Steineger, E., & Wahl, A. (2021). Naturfag 10. Cappelen Damm.

7.3 VEDLEGG 3: OPPGAVEHEFTET DEL 3: VAKSINERING

SMITTESPREDNING OG VAKSINERING

PROGRAMMERING AV MICRO:BIT



OPPGAVEHEFTE DEL 3: VAKSINERING



SYNNE CHRISTIN COLLING

Smittespredning og vaksinerings - Programmering av micro:bit

Oppgavehefte del: 3: Vaksinerings

Synne Christin Colling

Trondheim 2024

Bilde av micro:bit:

micro:bit Board, 2017, av SparkFun Electronics (<https://n9.cl/7xtm1>) CC BY 2.0 DEED. Retusjering: fjernet bakgrunn.

Illustrasjoner fra Canva

Designet i Canva

Hva er en vaksine og hvordan fungerer den?

Når immunforsvaret ødelegger og angriper uvedkommende inntrengere vil du som regel bli syk. Som dere lærte i det forrige oppgaveheftet kan immunforsvaret “huske” smittestoffene, og man er da immun. Ideen bak en vaksine er at immunforsvaret skal kunne “huske” smittestoffene uten at du først må ha vært syk. Når du blir vaksinert mot en spesifikk sykdom tilføres det et stoff som setter i gang immunforsvaret. Det er vanligst at det skjer gjennom at man får en sprøyte, men det kan også skje gjennom drikke, tablett eller nes spray. Vaksinen inneholder et virkestoff som kan bestå av deler av viruset eller andre mikroorganismer som er sterkt svekket eller ødelagt. Når vi får tilført dette virkestoffet vil kroppen lage antistoffer og hukommelsesceller mot den “ekte” mikroorganismen. På denne måten vil immunforsvaret komme raskere i gang hvis du senere blir smittet av mikroorganismen, slik at du ikke vil bli syk.

Hvor lenge man blir immun av vaksinen vil variere, og man må ta noen flere ganger



For å kunne laste ned og redigere prosjektene må dere trykke på rediger øverst i høyre hjørnet etter å ha trykket på lenken.

Oppgave 1

I denne oppgaven skal dere jobbe videre med programmet dere lagde om smittespredning. Dere vil i lenken til prosjektet finne det ferdige programmet til smittespredning. I dette programmet skal dere legge til at man kan vaksineres mot sykdommen. Dere vil også finne mange nye løse klosser, som skal brukes for å lage ferdig programmet. Dere skal igjen kun bruke disse klossene og alle skal brukes opp. Dere har igjen fått hjelp nedenfor til hva programmet skal gjøre. På de tre neste sidene vil dere se fire bilder som forteller hva som skal skje i programmet.

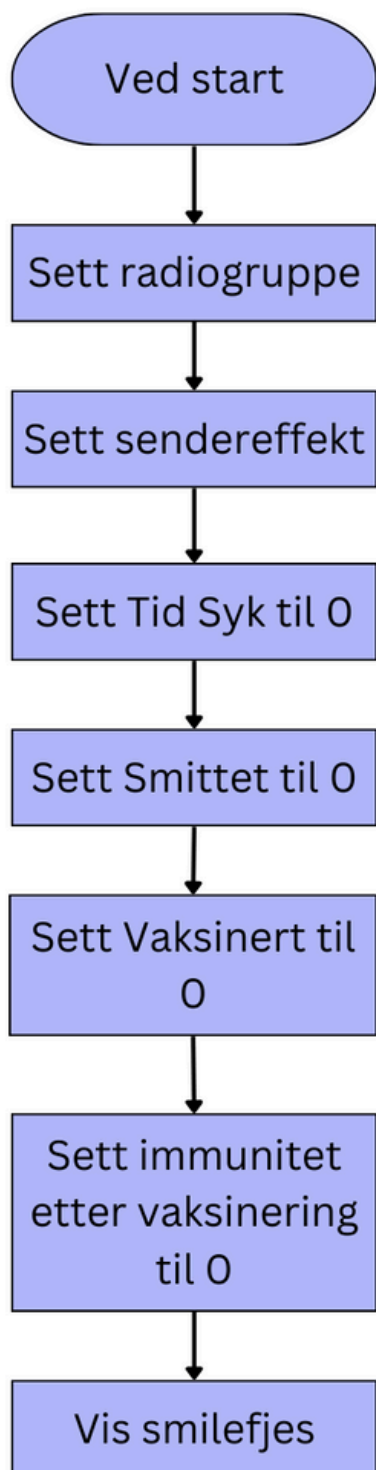
- Lenke til prosjektet: https://makecode.microbit.org/_KsCdD20t0fv2

Trenger dere litt hjelp?

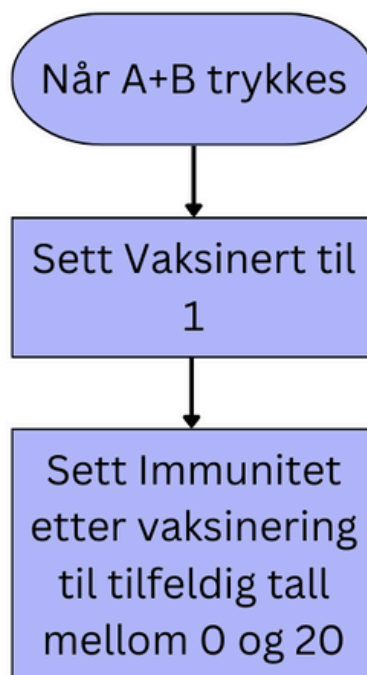
Trykk på denne lenken og jobb videre med dette programmet:

https://makecode.microbit.org/_C7WaF0fk7HML

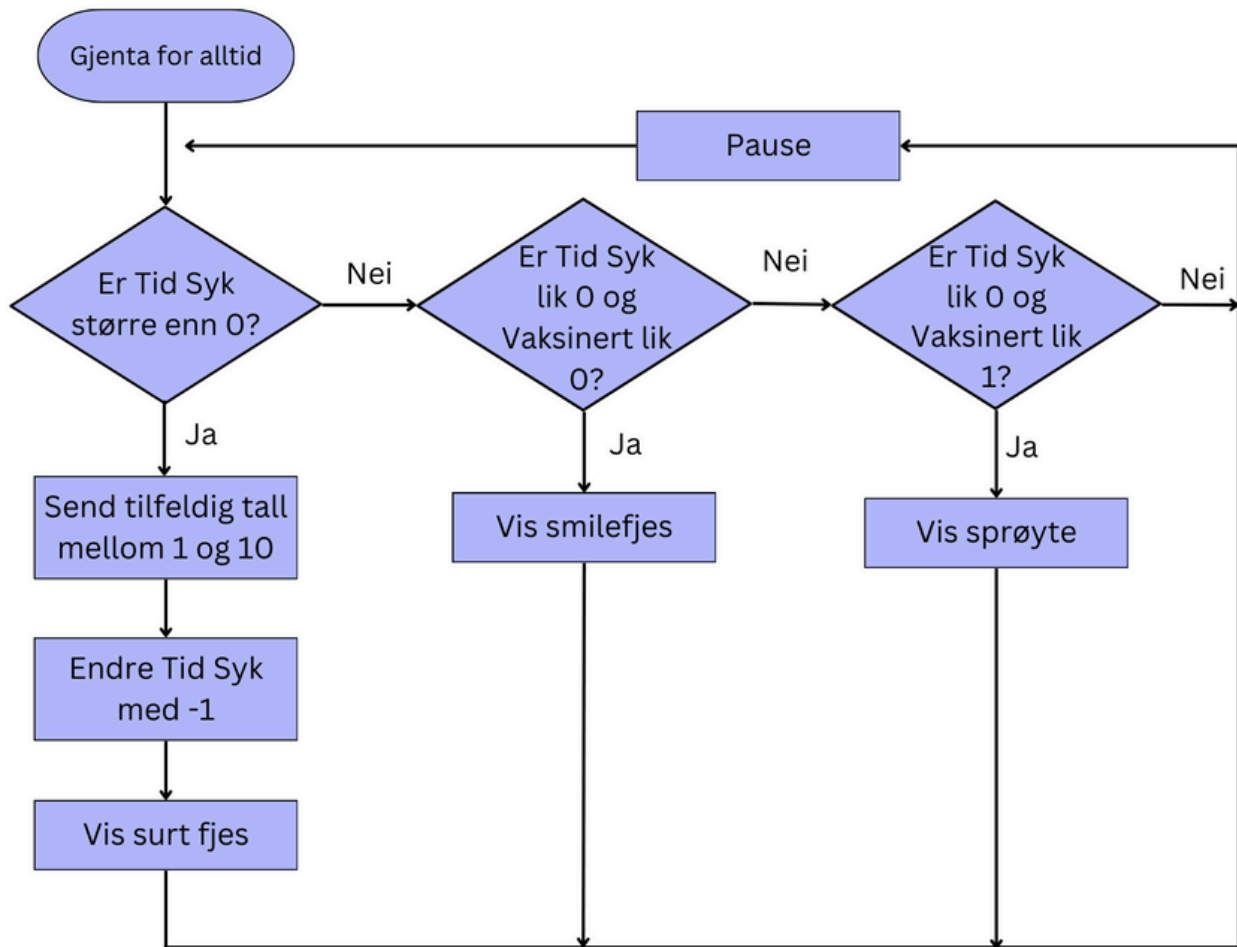
Flytskjema 1: Hva programmet skal gjøre ved start



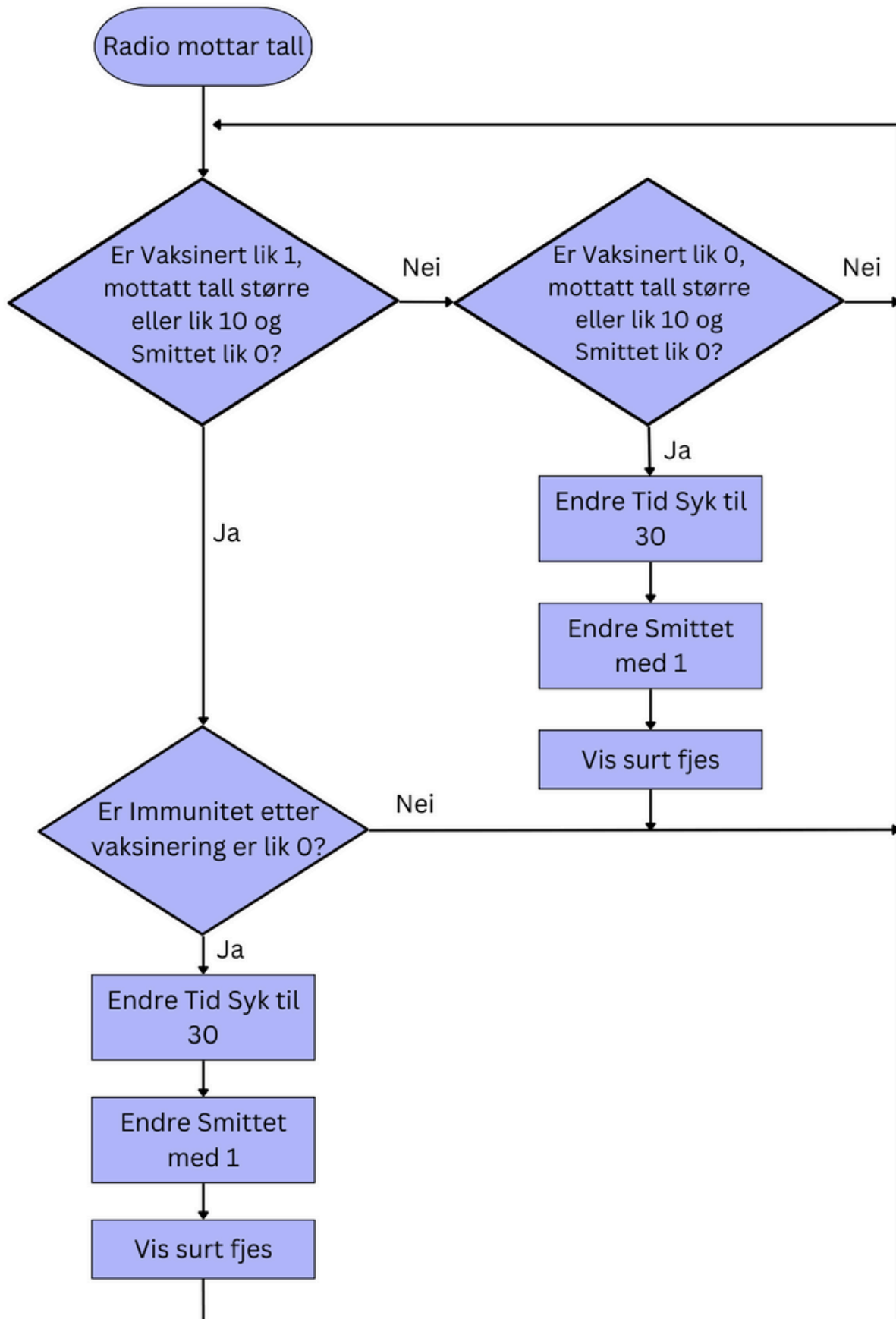
Flytskjema 2: Hva programmet skal gjøre når A+B knappen trykkes



Flytskjema 3: Hva programmet skal gjøre i gjenta for alltid løkka



Flytskjema 4: Hva programmet skal gjøre når radioen mottar et tall



Spørsmål til programmeringen

1. Hvilke nye vilkår finner dere i dette programmet, som ikke var satt sammen fra før?
2. Hva forteller disse vilkårene oss?
3. Hvorfor har dere valgt å plassere vilkårene der dere har gjort?
4. Hvilke nye variabler finner dere i dette programmet, som ikke var satt sammen fra før?
5. Hva brukes disse variablene til, og hva skal de hjelpe med å vise?
6. Hvorfor har dere valgt å plassere variablene der dere har gjort?

Ferdig?

1. Kan dere gjøre endringer i programmet som gjør at vaksinen ikke er like effektiv eller er mer effektiv?
2. Kan dere gjøre endringer i programmet som kan simulere at vaksinen har bedre effekt hvis det settes enda en dose av vaksinen?

Oppgave 2

Vi skal nå teste hvordan vaksinerings kan beskytte mot sykdom og begrense smittespredning. Sammen med hele klassen skal det gjøres tre tester der flere og flere tar vaksine. Læreren bestemmer hvem som skal ta vaksine i de ulike testene, og det er KUN de som får beskjed om å ta den som skal ta den.

Spørsmål til etter testingen:

- A) Så dere en forskjell i hvor mange som ble syke mellom hver testing?
- B) Ble noen som hadde tatt vaksine syke?

Oppgave 3

- A) Diskuter hvordan denne simuleringen kan være med på å vise hvordan vaksinerings kan beskytte deg mot sykdom.
- B) Diskuter hvordan denne simuleringen kan være med på å vise hvordan vaksinerings kan være med på å beskytte andre mennesker mot smitte.

Oppgave 4

- A) Simuleringer har sine svakheter og styrker. Hva tenker dere er denne simuleringen sine styrker og svakheter?
- B) Fungerte alt som det skulle under testingen av simuleringen? Hva kan eventuelt være grunnen til at noe ikke fungerte som det skulle?



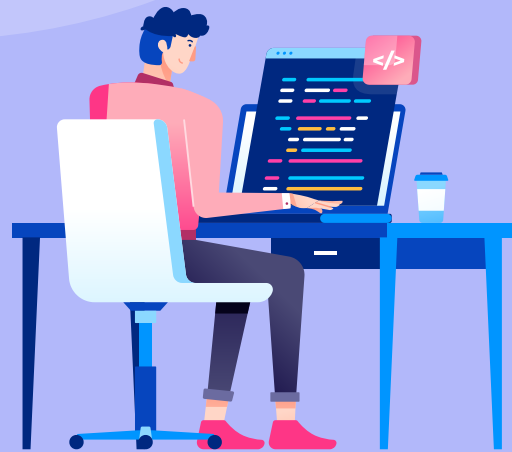
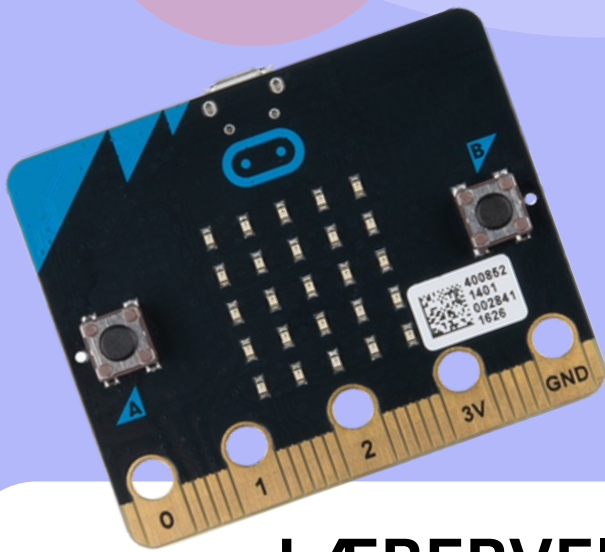
Litteraturliste

- Grindeland, J. M., Staberg, R. L., & Tandberg, C. (2020). Biologi for lærere (2. utg.). Gyldendal Akademisk.
- Steineger, E., & Wahl, A. (2021). Naturfag 10. Cappelen Damm.

7.4 VEDLEGG 4: LÆRERVEILEDNING

SMITTESPREDNING OG VAKSINERING

PROGRAMMERING AV MICRO:BIT



LÆRERVEILEDNING

MED GJENNOMGANG AV UNDERVISNINGSSOPPLEGGET, TIPS OG
GJENNOMGANG AV PROGRAMMENE



SYNNE CHRISTIN COLLING

Smittespredning og vaksinerings - Programmering av micro:bit

Lærerveiledningen

Synne Christin Colling

Trondheim 2024

Bilde av micro:bit:

micro:bit Board, 2017, av SparkFun Electronics (<https://n9.cl/7xtm1>) CC BY 2.0 DEED. Retusjering: fjernet bakgrunn.

micro:bit Board, 2017, av SparkFun Electronics (<https://n9.cl/7xtm1>) CC BY 2.0 DEED. Retusjering: fjernet bakgrunn og redigert inn i nytt bilde.

micro:bit Board, 2017, av SparkFun Electronics (<https://n9.cl/tsj41>) CC BY 2.0 DEED. Retusjering: fjernet bakgrunn og redigert inn i nytt bilde.

Illustrasjoner fra Canva.

Designet i Canva.

Innholdsfortegnelse

Introduksjon til undervisningsopplegget	4
Hvem er undervisningsopplegget for?.....	4
Anbefalte forkunnskaper	5
Læringsmål til elevene	5
Utstyr	5
Hvordan jobbe med undervisningsopplegget	6
Mulig vurderingsform.....	6
Forklaring av fargekodene videre i heftet	6
Gjennomgang av micro:biten	7
Relevante innganger og knapper for dette undervisningsopplegget	7
Relevante områder på makecode	7
Tilkobling av micro:bit	8
Dersom den vanlige tilkoblingsmetoden ikke fungerer	9
Tips dersom micro:biten ikke funker som den skal	10
Gjennomgang av undervisningsopplegget.....	11
Del 1: Radiokommunikasjon.....	11
Om oppgaveheftet	11
Før oppstart av arbeid med oppgaveheftet	11
Oppgave 1	12
Oppgave 2	14
Oppgave 3	16
Del 2: Smittespredning.....	17
Om oppgaveheftet	17
Før oppstart av arbeid med oppgaveheftet	17
Oppgave 1.....	17
Oppgave 2.....	20
Oppgave 3.....	26
Oppgave 4	27
Oppgave 5	28
Del 3: Vaksinerings	29
Om oppgaveheftet	29
Før oppstart av arbeid med oppgaveheftet	29
Oppgave 1.....	29
Oppgave 2.....	36
Oppgave 3.....	38
Oppgave 4	39

Introduksjon til undervisningsopplegget

Dette er et undervisningsopplegg som kombinerer programmering av micro:bit med temaene smittespredning, immunforsvaret og vaksinerings.

Undervisningsopplegget går ut på å programmere micro:biten slik at man i klasserommet kan simulere hvordan smittespredning foregår og hvordan vaksinerings kan være med på å begrense smittsomheten og hvorvidt man blir smittet eller ikke.

Undervisningsopplegget er tredelt og består av tre ulike oppgavehefter til elevene:

- Del 1: Radiokommunikasjon
- Del 2: Smittespredning
- Del 3: Vaksinerings

Undervisningsopplegget er delt i tre deler for å enklere kunne tilpasse opplegget til egen timeplan og de varierte lengdene på øktene.

Omtrent tidsbruk

Oppgavehefte del 1: Radiokommunikasjon: 30 min

Oppgavehefte del 2: Smittespredning 1 time og 30 minutter

Oppgavehefte del 3: Vaksinerings: 1 time

Dette er regnet som omtrentlig tidsbruk, men det vil variere basert på hvordan undervisningsopplegget gjennomføres og elevenes forkunnskaper.

Hvem er undervisningsopplegget for?

Undervisningsopplegget er laget for ungdomstrinnet, og kan kobles opp til fire kompetansemål som elevene skal ha tilnærmet seg etter 10. trinn. Disse fire kompetansemålene er hentet fra læreplanen i naturfag:

- Utforske, forstå og lage teknologiske systemer som består av en sender og en mottaker.
- Bruke programmering til å utforske naturfaglige fenomener.
- Beskrive kroppens immunforsvar og hvordan vaksiner virker, og gjøre rede for hva vaksiner betyr for folkehelsen.
- Bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensninger.

Anbefalte forkunnskaper

Det er anbefalt at elevene har arbeidet noe med micro:bit tidligere. Det er greit om de har en viss forståelse om hvordan løkker, vilkår og variabler fungerer. Dette er for at elevene enklere skal kunne koble programmeringen opp til smittespredning, immunforsvaret og vaksineringsen. Det kan være mulig å gjennomføre dette undervisningsopplegget selv om elevene ikke har jobbet mye med programmering av micro:bit før, men da må du belage deg på å bruke mer tid. Samt jobbe med elevene om hvordan de ulike funksjonene i programmene fungerer.

Læringsmål for elevene

Basert på kompetansemålene og undervisningsopplegget er mulige læringsmål for elevene:

- Kunne forklare hvordan en sender og en mottaker fungerer.
- Simulere hvordan smittespredning foregår ved programmering av micro:bit.
- Kunne forklare hvordan smitte sprer seg, og hvordan immunforsvaret beskytter deg mot smitte.
- Simulere hvordan vaksineringsen har en betydning for sykdom og smittespredning ved programmering av micro:bit.
- Kunne forklare hvordan vaksiner fungerer, og hvordan den kan beskytte både seg selv og andre mot sykdommen.
- Kunne vurdere svakheter og styrker ved en modell.

Utstyr

Det hver elev må ha:

- Micro: bit
- USB- kabel
- Batteripakke
- 2 AAA Batterier
- Oppgaveheftene digitalt

Det hvert par må ha:

- PC med internett

Anbefalt utstyr til elevene:

- Utskrift av flytskjemaene



Hvordan jobbe med undervisningsopplegget

Det er anbefalt at elevene arbeider med programmeringsoppgavene i par, ved å kun benytte én datamaskin. Dette kalles for parprogrammering og har flere fordeler:

1. Bedre læringsutbytte
2. Elevene blir mer selvsikre
3. Elevene opplever det som mer trivelig.

På denne måten vil elevene kunne spille hverandre gode. Det kan da være greit å tenke over parene som jobber sammen, og deres kompetanse innenfor programmering. Elevene burde være på et ganske likt nivå innenfor programmering. Dette er for å unngå at den ene eleven blir sittende og se på fordi eleven med mer kompetanse bare gjennomfører alt, eller at elevene med lavere kompetanse opplever at de ikke ønsker å bidra.

Det kan også være nyttig å gå gjennom med elevene hva deres rolle i samarbeidet er. Begge skal aktivt delta i prosessen, men én person har ansvaret for å styre mus og tastatur og den andre bidrar med vurderinger og meninger rettet mot programmet.

Når det kommer til diskusjonsoppgaven vil det være smart å la elevene diskutere seg mellom først, så ta det opp i plenum med hele klassen.

Mulig vurderingsform

Diskusjonen elevene har rundt undervisningsopplegget vil være en mulig vurderingsform. Gjennom diskusjon må elevene sette ord på det de selv har observert og lært. Her vil det være essensielt å få forklart hva programmet gjør og hvordan den kan vise prosessene smittespredning, immunforsvaret og vaksinerings.

En annen form er at elevene skriver ned svarene på spørsmålene som stilles underveis i oppgaveheftene. Da er det viktig å merke at elevene fort trenger lenger tid til å arbeide med oppgavene.

Forklaring av fargekodene videre i heftet

Hvit bakgrunn tilsier at det er en forklaring av opplegget

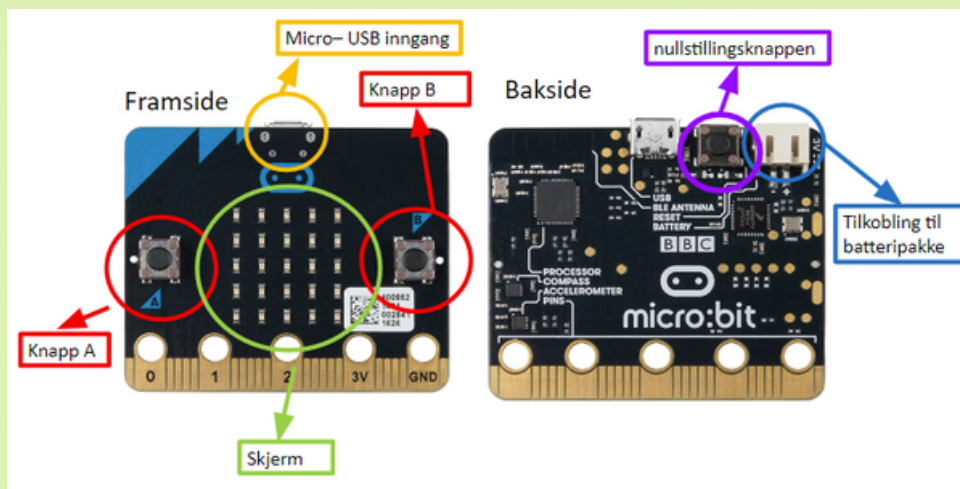
Grønn bakgrunn er forklaring av micro:biten

Lilla bakgrunn er forklaring av programmene

Gjennomgang av micro:biten

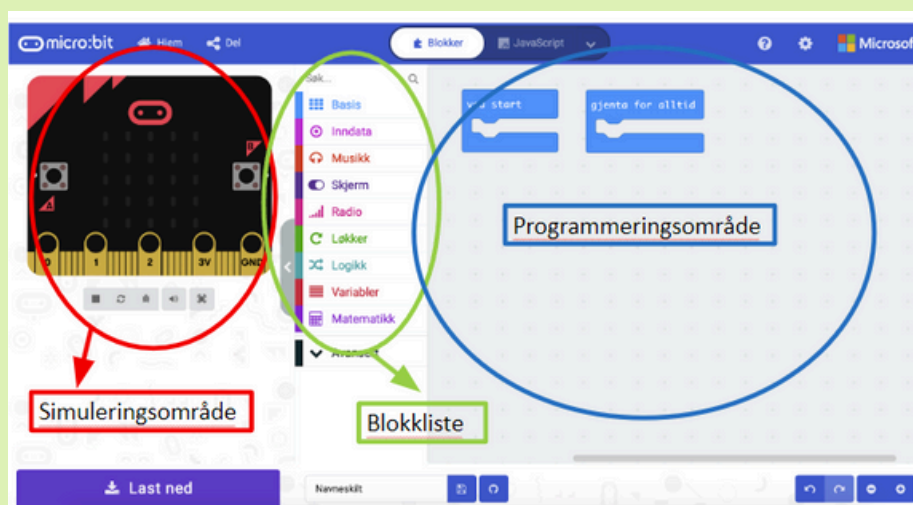
Denne delen av lærerveiledningen vil gå kort gjennom de elementene av micro:biten som skal benyttes i dette undervisningsopplegget. Videre vil det være en gjennomgang av hvordan man kobler micro:biten til pc-en, og hva man kan gjøre hvis det ikke fungerer på den vanlige måten. Til slutt vil den inneholde noen tips dersom micro:biten ikke fungerer som den skal.

Relevante innganger og knapper på micro:biten for dette undervisningsopplegget



Dette er de relevante inngangene og knappene du trenger å ha kontroll på for å benytte dette undervisningsopplegget.

Relevante områder på makecode

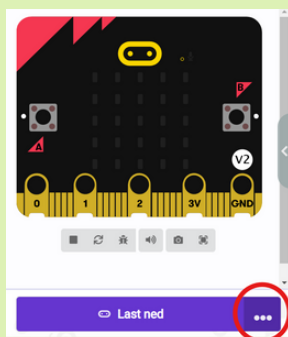


Dette er de relevante områdene på makecode med tanke på programmeringen av micro:bitene. Dette inkluderer ikke tilkobling og nedlastning av programmer.

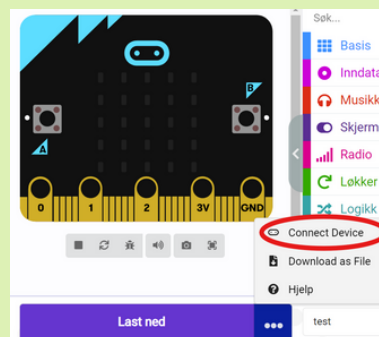
Tilkobling av micro:biten

Tilkobling av micro:biten til PC-en gjøres på denne måten:

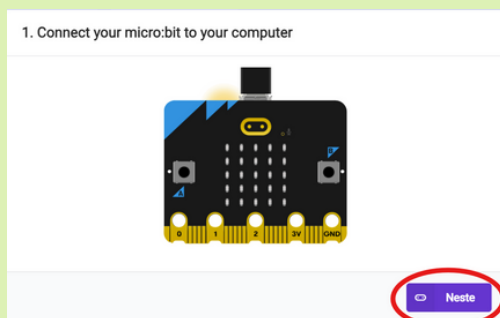
1. Koble micro:biten til PC-en med en Micro-USB-kabel.
2. Trykk på de tre prikkene ved siden av “Last ned”.
3. Det kommer så en meny. Trykk på det øverste alternativet “Connect Device”.
4. På det neste vinduet som kommer opp så trykker du på “neste”.
5. Når det nye vinduet har kommet opp trykker du på “Pair”.
6. Deretter kommer det opp enda et nytt vindu. Her trykker du der det står BBC micro:bit før du trykker på “Koble til”.
7. Micro:biten skal nå være koblet opp til PC-en, og man kan laste ned programmet på den.



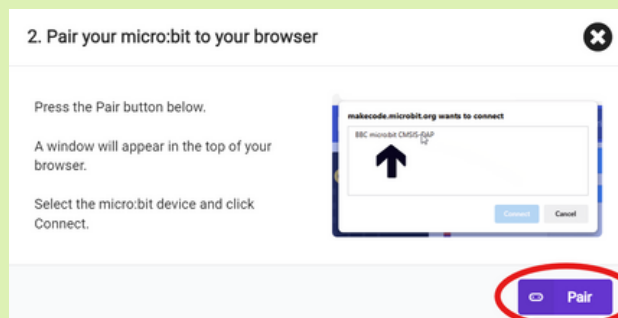
Steg 2



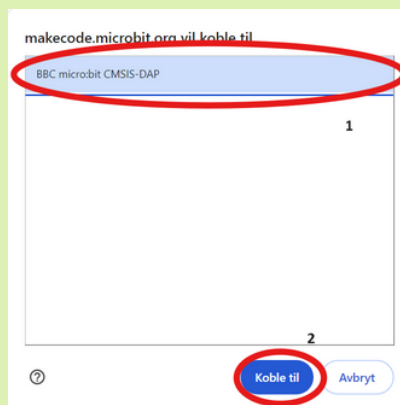
Steg 3



Steg 4



Steg 5

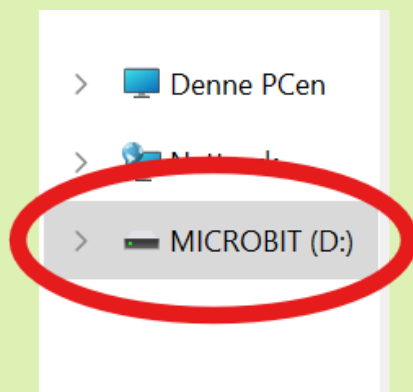


Steg 6

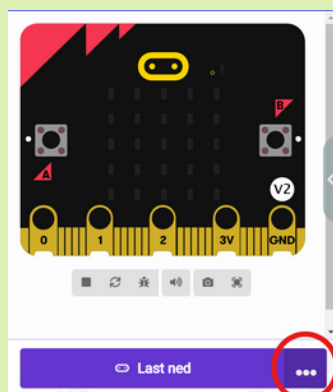
Dersom den vanlige tilkoblingsmetoden ikke fungerer

Dersom du ikke klarer å få koblet til micro:biten med den forrige er det en annen måte å få lastet opp programmet på micro:biten:

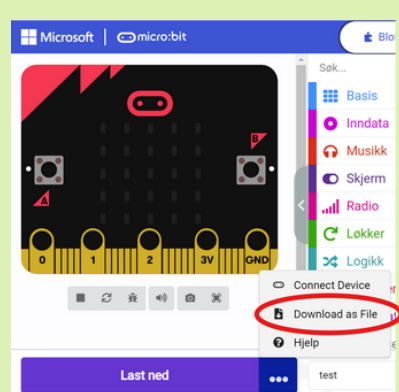
1. Koble micro:biten til PC-en med en Micro-USB-kabel.
2. Sjekk at micro:biten er koblet til i filutforskeren.
3. Trykk på de tre prikkene ved siden av “Last ned”.
4. Det kommer så en meny. Trykk på det mellomste alternativet “Download as File”.
5. Last så filen over til micro:biten som du finner i filutforskeren.
6. La filen laste seg helt over. Programmet er nå lastet ned på micro:biten. OBS! Merk at filen ikke vil synes i mappen til micro:biten selv om programmet er lastet ned.



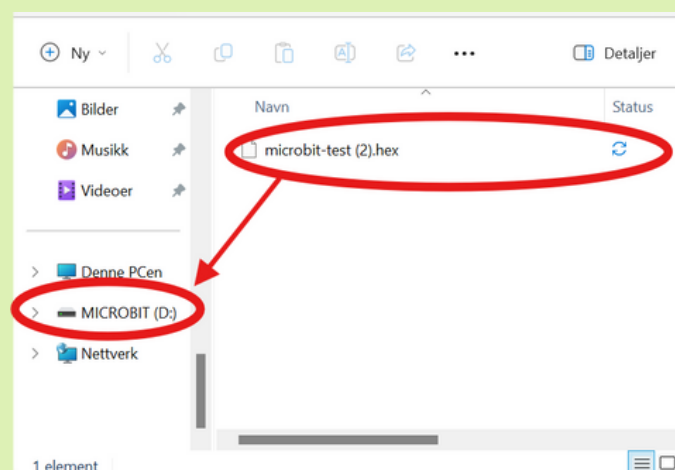
Steg 2



Steg 3



Steg 4



Steg 5

Tips dersom micro:biten ikke funker som den skal

Dersom micro:biten ikke vil koble seg til PC-en

Dersom du ikke får koblet micro:biten til PC-en kan det skyldes en dårlig micro-USB-kabel eller USB-inngang. Prøv en annen USB-inngang på PC-en. Hvis dette ikke fungerer prøv å koble til med en annen micro-USB-kabel. Hvis dette ikke løser problemet prøv å koble micro:biten til en annen PC.

Dersom micro:biten fungerer når den er koblet til PC-en, men ikke til batteripakken

Den vanligste grunnen er enten at det ikke er batterier i batteripakken eller at batteriene er tomme for strøm. Dersom du har byttet batterier, og vet at de sitter i ordentlig kan det skyldes kabelen til batteripakken. Dersom dette er tilfellet prøv å bytt ut batteripakken og se om det fungerer da.

Dersom micro:biten ikke fungerer etter å ha testet ulike kabler, batterier, batteripakker

Da er det mest sannsynlig noe feil med micro:biten, så prøv en annen.

Gjennomgang av undervisningsopplegget

Denne delen av lærerveiledningen inneholder en gjennomgang av undervisningsopplegget med forklaring av opplegget, tips til gjennomføring og forklaring av programmet som inngår i de ulike oppgavene.

Del 1: Radiokommunikasjon og programmeringsoppstart

Om oppgaveheftet

Dette oppgaveheftet er tenkt som en oppstart til programmering og radiokommunikasjon. Dette er for det første for at elevene skal danne en forståelse av hvordan en sender og en mottaker fungerer. Men også for å ha en grei oppstart av programmeringen. Dersom elevene nettopp har jobbet med programmering av micro:bit, der de har benyttet sender og mottaker er det mulig å hoppe over de to første oppgavene. Det er likevel anbefalt å benytte oppgave 3 som et innledende spørsmål før oppstart av arbeidet med oppgavehefte del 2 og del 3.

Før oppstart av arbeid med oppgaveheftet

Før elevene starter arbeidet med oppgaveheftet må du allerede ha fordelt radiogrupperne til parene. Dette for at elevene skal kunne komme i gang med oppgavene så fort og enkelt som mulig. Radiogrupperne strekker seg fra 1 til 256. I undervisningsopplegget har radiogruppe 25 blitt benyttet som standard, så det kan være greit å unngå denne i utdelingen. Hvordan du fordeler parene og radiogrupperne er opp til deg. En mulighet er å ha ferdiglagde par og fordelte radiogrupper klart til starten av timen. Da kan elevene enten få et dokument med fordelingen av radiogrupperne eller det kan tas opp på tavla. Det kan også være nyttig å ha en påminnelse om å bytte radiogruppe ofte på tavla. Denne fordelingen må også gjøres selv om du velger å hoppe over oppgave 1 og 2 i dette oppgaveheftet, siden elevene også vil måtte benytte disse radiogrupperne i det neste oppgaveheftet.

Videre er det anbefalt at du tar en oppstart av hele undervisningsopplegget samlet under ett. Altså at de skal jobbe med programmering av micro:bit der de skal lage en simulering av smittespredning og vaksinerings, samt se på hvilken rolle immunforsvaret spiller for dette. Videre er det anbefalt at du tar en gjennomgang av hva en simulering er da dette er et begrep som benyttes i alle tre oppgaveheftene.

En simulering er en interaktiv modell der brukeren kan endre variablene for å kunne se ulike scenarioer av en prosess. Det er viktig her å få frem at en simulering slik som en modell er en forenkling av virkeligheten.

Oppgave 1

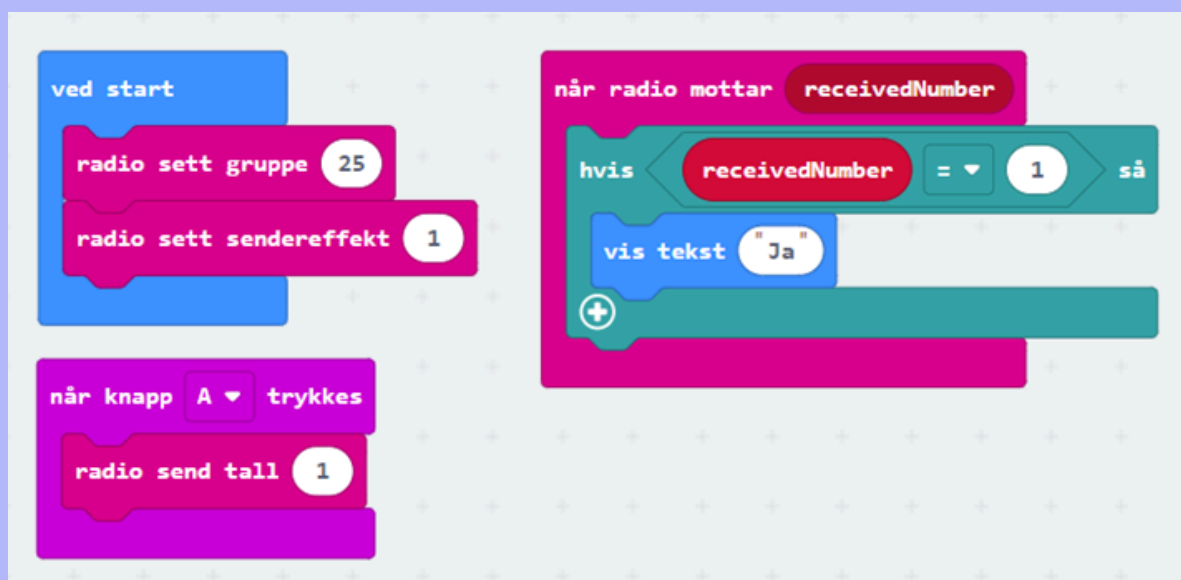
Den første oppgaven i dette oppgaveheftet består av to deloppgaver. I deloppgave A skal elevene forutsi hva som kommer til å skje når programmet kjøres. Når elevene skal forutsi hva et program gjør må de tenke over poenget med klossene og hvilken betydning rekkefølgen har. I den andre deloppgaven skal elevene kjøre programmet, og vurdere om antagelsene deres stemte med det som faktisk skjedde.

Denne oppgaven er fin å benytte for å se at alle får koblet opp micro:biten. En anbefaling er å be elevene starte med å gjøre oppgave 1a, mens du deler ut micro:bitene. Etter at micro:bitene er utdelt og elevene har diskutert hva de tror programmet kommer til å gjøre kan du få alle elevene til å koble til micro:biten og laste ned programmet samtidig. Her kan det være smart å gjennomgå tilkoblingen av micro:biten på tavla, mens elevene gjør det samme. På denne måten kan du muligens unngå å benytte så mye tid på å få koblet opp alle micro:bitene.

Gjennomgang av programmet i oppgaven

Oversikt over hele programmet

programmet er lagd for å bruke micro:biten til å kunne si ja til hverandre.



Programmet består av tre hoveddeler. Delenes oppgave består av:

- En oppstartsfunksjon som tilsier hva som skal skje når programmet startes opp
- En funksjon som sender et tall
- En mottakerfunksjon som avgjør hva som skal skje når et tall mottas.

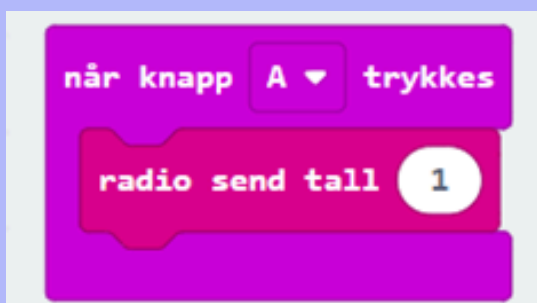
Oppstartsfunksjonen



Oppstartsfunksjonen setter betingelsene ved start, der de ulike klossene gjør følgende:

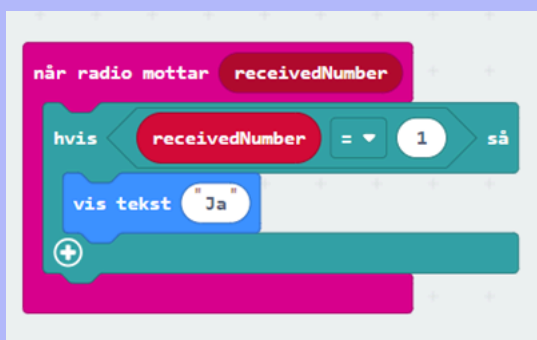
- radio-klossen er kommunikasjonskanalen mellom micro:bitene. Gruppe 25 er den valgte kanalen, men denne må elevene huske å bytte ved testing. Micro:bitene i samme radiogruppe kan kommunisere med hverandre.
- «sendereffekt» er en valgt innstilling for hvor langt signalene sendes i rommet

Sender et tall



- Når knapp A trykkes vil radio sende tallet 1 til en annen micro:bit som har samme radiogruppe.

Mottar et tall



Mottakerfunksjonen er den som mottar signalet som sendes. I dette tilfellet vil det som skjer inne i løkka skje når radioen mottar et tall.

- Hvis tallet micro:biten mottar er tallet 1 så vil teksten "Ja" vises på skjermen til micro:biten.

Oppgave 2

I denne oppgaven skal elevene jobbe videre med programmet fra oppgave 1. De har her fått utdelt en lenke med mange løse klosser som de skal pusle sammen slik at micro:biten både kan benyttes til å si "Ja" og "Nei". Gjennom å allerede ha gitt elevene klossene som skal benyttes for å lage programmet vil de bruke mindre tid på å lete etter de riktige klossene. Dette er også en metode som kommer til å bli benyttet senere i oppgaveheftene, og det vil derfor være gunstig at elevene ser dette i en mindre skala først.

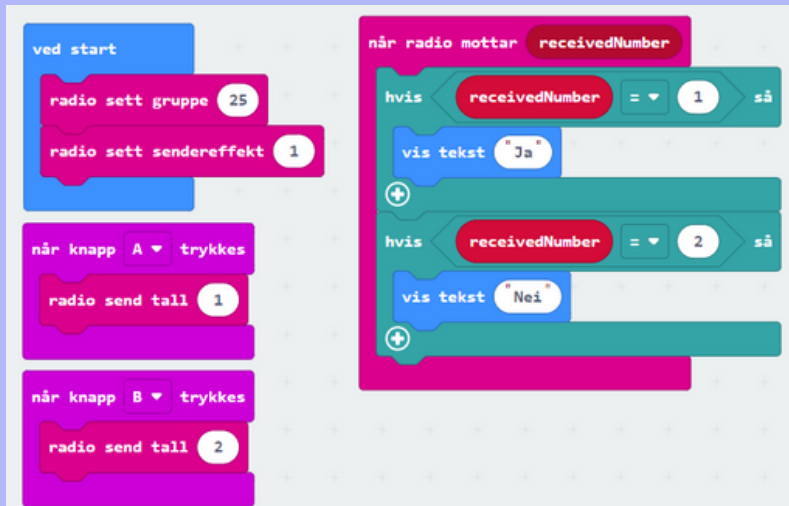
- Lenke til det ferdige prosjektet:

https://makecode.microbit.org/_TeKhcbMKFP5E

I oppgaveheftet er det også lagt til tips for elevene som synes oppgaven er krevende, og videre arbeid for elevene som opplever oppgaven som enkel. Det er også mulig å benytte denne oppgaven til å la elevene undersøke sendereffekt dersom de ikke er kjent med dette fra før. Og la dem utforske micro:biten mer.

Gjennomgang av programmet i oppgaven

Oversikt over hele programmet



programmet består av fire hoveddeler. Delenes oppgave består av:

- En oppstartsfunksjon som tilsier hva som skal skje når programmet startes opp
- To funksjoner som sender et tall
- En mottakerfunksjon som avgjør hva som skal skje når et tall mottas.

MERK! Det vil også være mulig at når man trykker knapp A så vil tall 2 sendes eller at man får teksten "Nei" når tall 1 sendes. Dette har ikke noe å si for programmet. Men det er viktig å være obs på at begge veier fungerer.

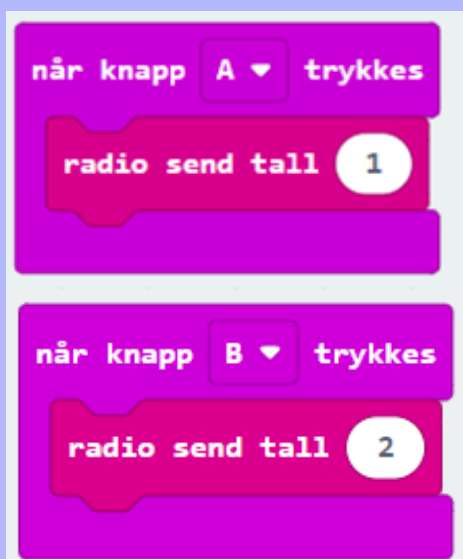
Oppstartsfunksjonen



Oppstartsfunksjonen setter betingelsene ved start, der de ulike klossene gjør følgende:

- radio-klossen er kommunikasjonskanalen mellom micro:bitene. Gruppe 25 er den valgte kanalen, men denne må elevene huske å bytte ved testing. Micro:bitene i samme radiogruppe kan kommunisere med hverandre.
- «sendereffekt» er en valgt innstilling for hvor langt signalene sendes i rommet

Sender et tall



- Når knapp A trykkes vil radio sende tallet 1 til en annen micro:bit som har samme radiogruppe.
- Når knapp B trykkes vil radio sende tallet 2 til en annen micro:bit som har samme radiogruppe.

Mottar et tall



Mottakerfunksjonen er den som mottar signalet som sendes. I dette tilfellet vil det som skjer inne i løkka skje når radioen mottar et tall.

- Hvis tallet micro:biten mottar er tallet 1 så vil teksten "Ja" vises på skjermen til micro:biten.
- Hvis tallet micro:biten mottar er tallet 2 så vil teksten "Nei" vises på skjermen til micro:biten.

Oppgave 3

Siste oppgave i dette heftet er en diskusjonsoppgave. Det anbefales her å gå gjennom hva en simulering er. Dette er et begrep som blir benyttet i de to neste oppgaveheftene. Etter dette kan oppgave 3 diskuteres. Gjennom spørsmålet må elevene tenke over hvordan radiokommunikasjon fungerer samtidig som det vil fungere som en oppstart av del 2.

Her er hovedpoenget dere kommer frem til at når mennesker er smittsomme så vil de gjennom å hoste, nyse og snakke spre små dråper som kan inneholde mikroorganismer og overføres til andre mennesker. Ettersom micro:bitene kan kommunisere med hverandre vil de kunne simulere at en micro:bit for eksempel hoster eller nyser og dermed overfører smitte til den andre micro:biten som gjør at den har blitt syk.

Del 2: Smittespredning

Om oppgaveheftet

Dette oppgaveheftet inneholder fem oppgaver. De to første oppgavene er knyttet til programmering, der oppgave 2 er selve programmeringen av smittespredningen. Oppgave 3 går ut på å teste simuleringen av smittespredningen med hele klassen, mens oppgave 4 er en diskusjonsoppgave som knytter simuleringen til smittespredning og immunforsvaret. I oppgave 5 skal eleven vurdere simuleringen. Før oppgavene er det en liten introduksjonstekst om hvordan mennesker kan bli smittet, hva som kan gjøre oss syke og hvordan immunforsvaret jobber mot mikroorganismene.

Før oppstart av arbeid med oppgaveheftet

Før oppstart av arbeidet med oppgavene i heftet anbefales det å lese gjennom introduksjonsteksten til heftet i plenum. Dette er for å sikre at alle elevene har vært gjennom introduksjonsteksten før de starter på oppgavene. Grunnen til at introduksjonsteksten er satt i oppgaveheftet til elevene er fordi elevene kan bruke denne informasjonen i arbeidet med diskusjonsspørsmålene senere i oppgaveheftet.

Oppgave 1

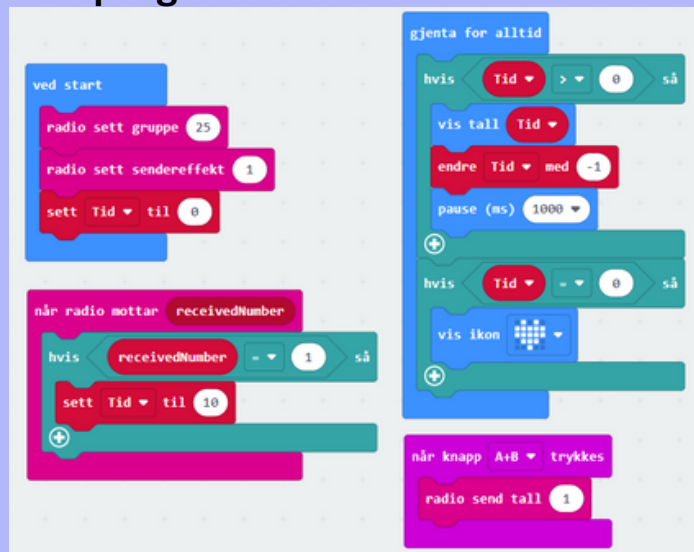
Denne oppgaven består av to deloppgaver, og fungerer på lik måte som oppgave 1 fra oppgaveheftet om radiokommunikasjon. Elevene skal i den første deloppgaven forutsi hva de tror kommer til å skje når programmet kjøres. Dette er for at elevene skal måtte tenke gjennom hvilken rolle de ulike klossene har og hvilken rekkefølge klossene er satt i. Selv om dette er en lignende oppgave vil den likevel være av betydning for elevene da de vil møte på nye klosser som de også vil benytte i oppgave 2 i dette oppgaveheftet. Den andre deloppgaven går ut på at elevene skal kjøre programmet for å se om antagelsene de gjorde stemte med det som faktisk skjedde.

Denne oppgaven er igjen fin å benytte for å se at alle får koblet opp micro:biten. En anbefaling er å be elevene starte med å gjøre oppgave 1a, mens du deler ut micro:bitene. Etter at micro:bitene er utdelt og elevene har diskutert hva de tror programmet kommer til å gjøre, kan du få alle elevene til å koble til micro:biten og laste ned programmet samtidig. Her kan det være smart å gjennomgå tilkoblingen av micro:biten på tavla, mens elevene gjør det samme. På denne måten kan du muligens unngå å bruke unødvendig mye tid på å få koblet opp alle micro:bitene.

Gjennomgang av programmet til oppgaven

Denne programmet vil fungere som en nedtelling fra 10 til 0 når nedtellingen får beskjed om å starte.

Oversikt over hele programmet



programmet består av fire hoveddeler. Delenes oppgave er:

- En oppstartsfunksjon, som gjør klart til oppstart
- En nedtellingsfunksjon, som teller ned tiden det settes til
- En tidsettingsfunksjon som bestemmer at tiden skal settes
- En senderfunksjon som sender et tall som radioen mottar

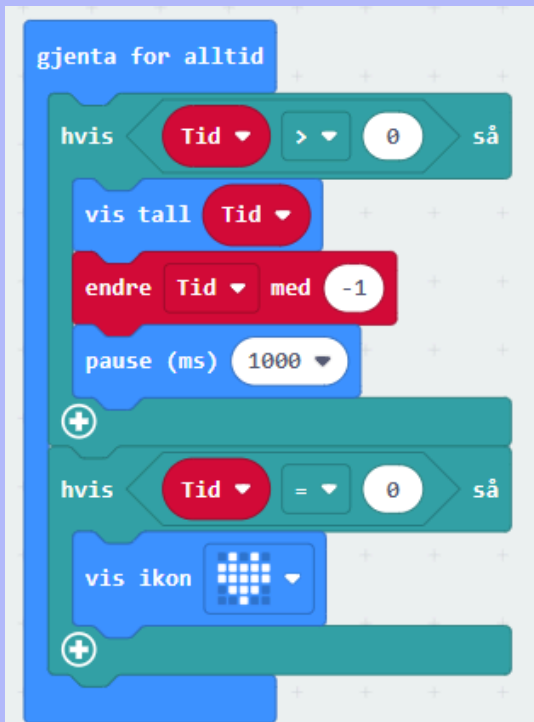
Oppstartsfunksjon



Oppstartsfunksjonen setter betingelsene ved start av programmet:

- radio-klassen er kommunikasjonskanalen mellom micro:bitene. Gruppe 25 er den valgte kanalen. Alle micro:bitene satt i denne gruppen kan da kommunisere med hverandre. Denne må elevene bytte ut i uttestingen.
- «sendereffekt» er en valgt innstilling for hvor langt signalene sendes i rommet.
- Variabelen “Tid” settes til 0 ved start for å signalisere at nedtelling ikke er satt i gang.

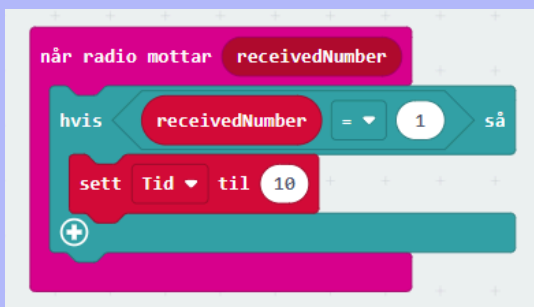
Nedtellingsfunksjon



Denne er kalt nedtellingsfunksjonen da denne vil telle ned, og klossene i den fungerer på denne måten:

- Dette gjentas for alltid, så lenge programmet kjører.
- Først sjekker den om variabelen "Tid" er større enn 0, for å se om den skal telle ned eller ikke.
- Hvis tid er større enn 0 vil den vise tallet som variabelen "Tid" tilsvarer på skjermen.
- Deretter vil variabelen "Tid" endres med -1 for å telle ned.
- Det er deretter en liten pause for at nedtellingen ikke skal gå for fort.
- Videre i løkka blir det sjekket om variabelen "Tid" er lik 0. Dersom dette stemmer vil det vises et hjerte på skjermen til micro:biten.

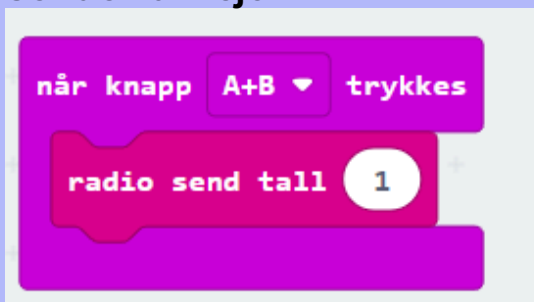
Tidssettingsfunksjon



Denne tidssettingsfunksjonen setter tiden på nedtellingen, og klossene fungerer på denne måten:

- Radioen mottar et tall, og sjekker så om tallet er lik 1.
- Hvis tallet er lik 1 vil tid settes til 10, og nedtellingen vil så starte.

Senderfunksjon



Denne senderfunksjonen avgjør om nedtellingen skal startes, og klossene fungerer på denne måten:

- Dersom knapp A og B trykkes samtidig vil radioen sende tallet 1, som ber mottakeren om å sette tiden til 10.

Oppgave 2

Elevene skal i denne oppgaven programmere selve smittespredningen. Det har i denne oppgaven blitt benyttet to ulike teknikker for å lette elevenes arbeid med programmeringen. For det første har elevene fått en lenke med et prosjekt med mange løse klosser. Her er elevenes oppgave å pusle disse klossene sammen slik at programmet gjør det samme som flytskjemaene tilsier. Den andre teknikken som er benyttet er flytskjemaene. Flytskjemaene forteller elevene hva som skal skje i programmet, og dette vil fungere som en slags oppskrift slik at elevene vet hva som skal skje i programmet og når. Deres oppgave blir dermed å skjønne hvilke klosser de skal pusle sammen i riktig rekkefølge for å få programmet til å gjøre det samme som flytskjemaene tilsier.

- Lenke til det ferdige prosjektet:
https://makecode.microbit.org/_J4fawWJh20aT

En anbefaling til denne oppgaven vil også være at arket med flytskjemaene er skrevet ut slik at elevene slipper å gå frem og tilbake mellom fanene. Dermed kan de lettere fokusere på å pusle sammen klossene slik at det samsvarer med det flytskjemaet tilsier.

- Lenke til arket med flytskjemaene:
https://drive.google.com/file/d/1eL1Mdx_SuHeGuDNJXheOYnsyYsmKz89J/view?usp=sharing

Det har i oppgaveheftet blitt lagt til to oppgaver for elevene som blir fort ferdig eller som trenger enn større utfordring. Det er også lagt til en forenkling av oppgaven for å hjelpe elevene som opplever programmeringsarbeidet som krevende. Dersom de fremdeles trenger hjelp gir du dem bildet av den ferdige programmet:

- <https://docs.google.com/document/d/1qY3pYQkRyKnmJ4C362EJwuzgGTEk3YGWwylbTnTUbpo/edit?usp=sharing>

Tilknyttet oppgaven har også elevene fått utdelt spørsmål som skal svares på. Disse kan godt tas i plenum i etterkant av programmeringsarbeidet. Her vil det også være smart å ta opp programmet på tavla slik at elevene får sjekket om sine programmer stemmer. På dette måten vil micro:bitene også gjøres klare til neste oppgave, nemlig simuleringen av smittespredningen. Da det i den oppgaven er viktig at alle elevene har lastet opp samme programmet.

Svar på spørsmålene som elevene får tildelt:

Hvilke løkker finner dere i dette programmet, og hvordan brukes de?

- Dette programmet har kun én løkke, og den vises som en “gjenta for alltid”-løkke. Den brukes til å konstant sjekke om micro:biten har blitt smittet eller ikke.

Hvilke vilkår finner dere i dette programmet?

- I programmet brukes det tre vilkår, og alle de tre vilkårene kommer i form av “hvis”-klosser.

Hvorfor har dere valgt å plassere vilkårene der dere har gjort?

- Hvis ReceivedNumber er større eller lik 10 og smittet =0
 - Plassert under ReceivedNumber fordi vi vil kontrollere tallet *etter* at det er mottatt.
- Hvis “Tid Syk” er større enn 0
 - Plassert under “gjenta for alltid” for å sjekke om micro:biten er smittsom
 - Det er irrelevant om det er dette vilkåret eller neste som plasseres først, siden de ikke kan oppfylles samtidig i samme iterasjon av løkka.
- Hvis “Tid Syk” er lik 0
 - Plassert under “gjenta for alltid” for å sjekke om ikke er smittsom

Hva forteller disse vilkårene oss?

- Hvis ReceivedNumber er større eller lik 10 og smittet =0
 - Dersom dette stemmer vil micro:biten bli syk
- Hvis “Tid Syk” er større enn 0
 - Dersom dette stemmer er micro:biten smittsom
- Hvis “Tid Syk” er lik 0
 - Dersom dette stemmer er micro:biten ikke smittsom

Hvilke variabler finner dere i programmet?

- “Smittet”
- “Tid Syk”

Hvorfor har dere valgt å plassere variablene der dere har gjort?

- Begge er satt til 0 ved start for å nullstille variablene
- “Tid Syk” = 30 under ReceivedNumber tilsier at personens sykdomsforløp har blitt stilt til starten av forløpet
- “Tid Syk” = -1 under gjenta for alltid tilsier at sykdomsforløpet kortes ned ett steg
- “Smittet” = 1 tilsier at personen har vært eller er smittet av sykdommen

Hva brukes disse variablene til, og hva skal de hjelpe med å vise?

- “Smittet” viser til om personen har vært smittet av mikroorganismen eller ikke
- “Tid Syk” viser til om personen er smittsom eller ikke, og fungerer som en nedtelling for hvor lenge personen er syk.

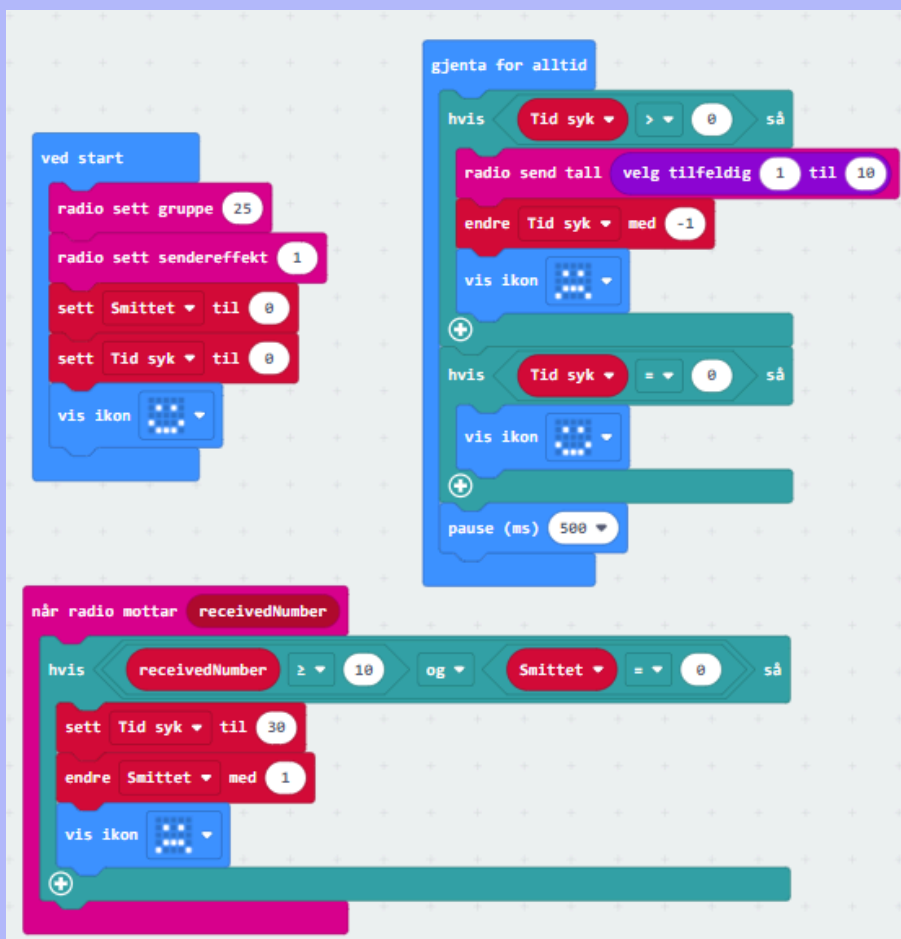
Gjennomgang av programmet i oppgaven

programmet simulerer et smitteforløp i en populasjon ved en fiktiv sykdom.

Det er bestemt en generell motstandsdyktighet mot smitten, der man har 1/10 sjanse for å kunne smittes av smittebærer. Dette skal simulere hvordan immunforsvaret kan motstå smitte selv om man er innenfor rekkevidden til smitten.

Alle micro:biter vil kun være smittebærere én gang i en gitt periode. Dette simulerer at kroppen ofte bygger opp en immunitet mot samme sykdom innenfor en viss periode.

Oversikt over hele programmet



Programmet består av 3 hoveddeler. Delenes oppgaver er:

- En oppstartsfunksjon, som kan bestemme hvem som starter som syke
- En funksjon som utspiller smitteforløpet
- En mottakerfunksjon som mottar smitten dersom personen er frisk

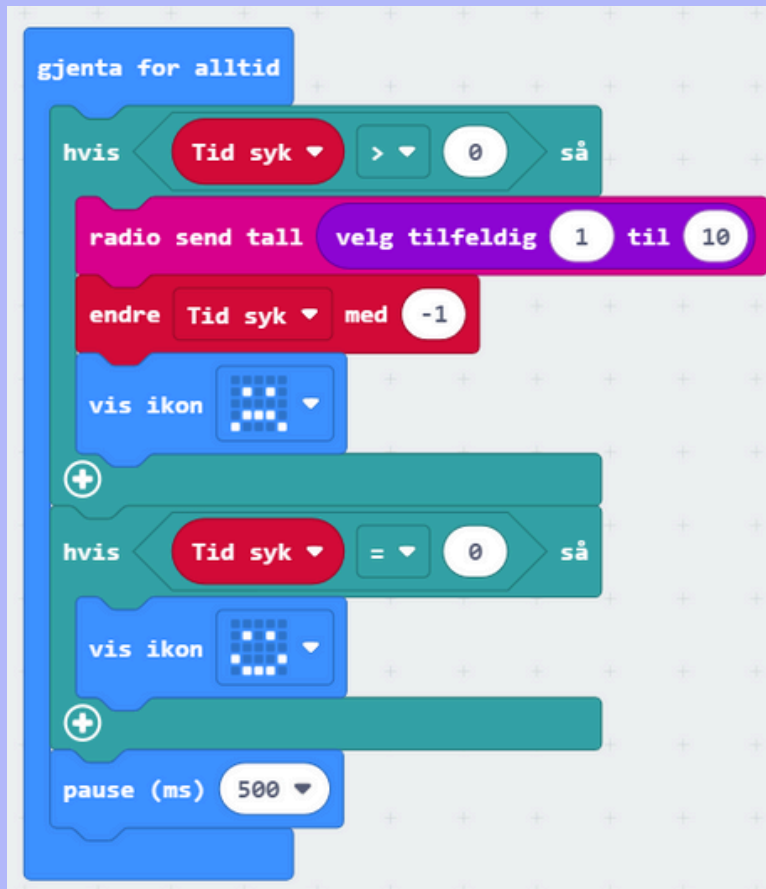
Oppstartsfunksjonen



Oppstartsfunksjonen setter betingelsene ved start av simuleringen, der de ulike klossene gjør følgende:

- radio-klossen er kommunikasjonskanalen mellom micro:bitene. Gruppe 25 er den valgte kanalen. Alle micro:bitene satt i denne gruppen kan da kommunisere med hverandre.
- «sendereffekt» er en valgt innstilling for hvor langt signalene sendes i rommet. Etter uttesting er den satt til 1 da dette er ca. 2 m. Hvis dette ikke samsvarer i ditt klasserom bør du endre sendereffekten.
- Variabelen «smittet» viser til om micro:biten starter som smittsom eller frisk, bestemt med henholdsvis verdiene 1 og 0. Dersom verdien er 0 har micro:biten ikke vært smittet før.
- Variabelen «Tid Syk» viser startverdi for varigheten man er smittsom. Denne er 0 for friske micro:biten og er valgt til å starte på 30 for de som er smittsomme.
- Det vises et smilefjes for friske micro:biten, og et surt fjes dersom micro:biten starter som smittsom.

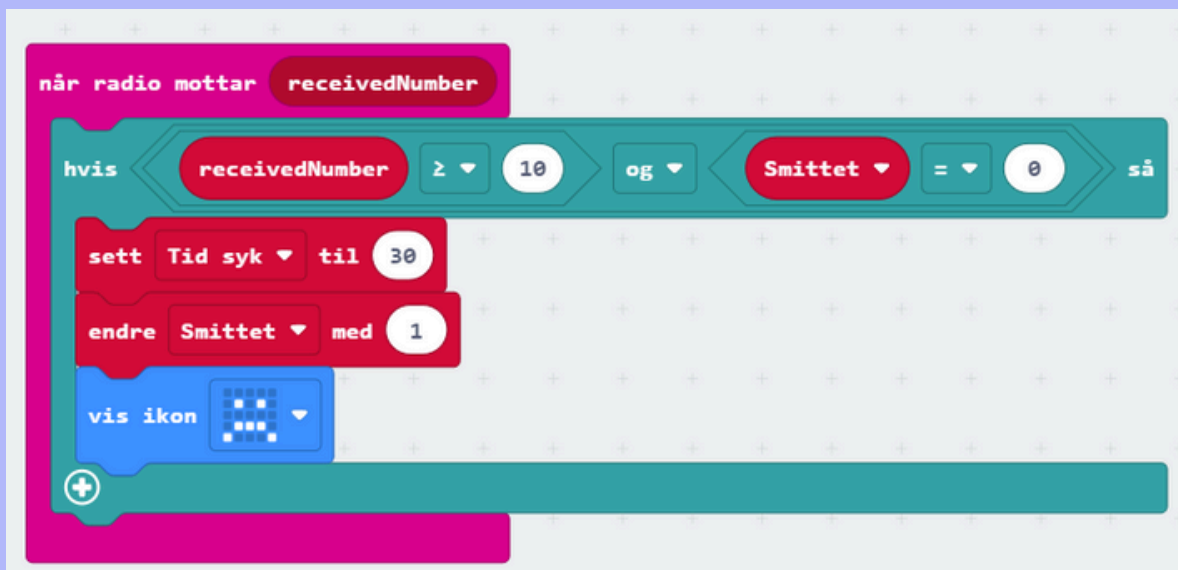
Smitteforløpet



De ulike klossene i funksjonen for smitteforløpet gjør følgende:

- Dette gjentas for alltid, så lenge programmet kjører.
- Først sjekkes det om micro:biten er smittsom, ved variabelen «Tid Syk». Dersom den er større enn 0, er sykdomsforløpet satt i gang og den er smittsom.
- Da sendes et tilfeldig tall mellom 1 og 10 med «radio-klossen». Senere forklares det at kun ett av disse tallene vil føre til smittespredning. Dette simulerer at immunforsvaret ofte kan stå imot smitten – man blir ikke alltid syk.
- Når smitte-signalet er sendt ut, reduseres lengden på sykdomsforløpet med 1, helt til variabelen «Tid Syk» blir 0, og den tidligere «hvis-sjekken» ikke lenger er oppfylt.
- Så lenge tid er over 0, vises surt fjes på skjermen til micro:biten for å signalisere at micro:biten er smittsom.
- Hvis «Tid Syk» er 0 er man ikke smittsom, og et smilefjes vises på micro:bit-skjermen.
- Før hele denne løkka gjentas, er det lagt inn en pause på 500 ms.

Smittemottaker



Mottakerfunksjonen simulerer immunforsvarets motstand mot smitte, og klossene fungerer på følgende måte:

- Signalet som micro:biten mottar fra smittebærer er definert som et tall med den innebygde micro:bit-klossen «receivedNumber». Som forklart tidligere er dette et tilfeldig tall mellom 1 og 10 for hvert signal som sendes fra smittebærer.
- Hvis dette tallet er større eller lik 10 («receivedNumber» større eller lik 10) og mottaker er frisk («smittet» = 0), har mottakeren nå blitt smittet av dette signalet. Det er altså 1/10 sjanse for å smittes av smittebærere når mottakeren er innenfor signalrekkevidden definert som «sendereffekt» = 1. Tallet 10 er valgt for å simulere at mottakeren har blitt utsatt for så mye smitte at den nå er syk.
- Ved denne smitten settes sykdomsforløpet til utgangspunktet på 30.
- Deretter endres/økes «Smittet» med 1, slik at denne hvis-setningen ikke kan oppfylles enda en gang. Man kan altså kun smittes én gang etter oppstarten.
- Deretter vises surt fjes på micro:bit-skjermen.

Simuleringsslutt

Simuleringen vil kjøre evig. Etter hvert vil alle micro:bitene sannsynligvis ha blitt smittet, og alle skjermene vil ha vist både surt fjes og deretter smilefjes 1 gang hver.

Oppgave 3

I denne oppgaven skal simuleringen av smittespredningen testes ut med hele klassen. Det er to essensielle forutsetninger for at det skal bli en vellykket test. Den første er at sendereffekten er tilpasset størrelsen på klasserommet, eller at området testen gjennomføres i er tilpasset sendereffekten. Den andre forutsetningen er at elevene har programmert smittespredningen riktig og at alle tallene i programmene er like hos alle.

For å kunne gjennomføre testingen av simuleringen vil det også være avgjørende å ha en micro:bit som allerede er smittet. Her kan du selv avgjøre om det er du som lærer som har den smittsomme micro:biten ved start eller om en av elevene skal laste ned programmet som allerede er smittsomt. Dersom du velger at en av elevene skal være smittsom ved start kan det være greit å tenke over hvilken elev det skal være.

- Lenke til allerede smittsomt program:
https://makecode.microbit.org/_EewdM47WA2wt

Nå som micro:bitene er klare til å gjennomføre simuleringene kan testene startes, men først må elevene få en forklaring på hva målene deres er og hvilke retningslinjer de må forholde seg til. For det første er ikke målet at elevene skal løpe rundt, og vekk fra den smittede, men de kan gå unna. Den som er smittet ved start skal gå rundt til de andre som en normalt ville gjort. Dette er grunnen til at det er greit å tenke over hvem som er smittet ved start.

Det skal gjennomføres tre tester av simuleringen:

- Stående stille. Elevene skal her se hvordan smitten sprer seg.
- Gående rundt. Målet til elevene er å ikke bli syk.
- Gående rundt. Målet til elevene er å bli fortest mulig syk.

Husk å nullstille micro:bitene mellom simuleringene ved å trykke på nullstillingsknappen. Sørg for at micro:biten som er smittet ved start ikke nullstilles før alle de andre micro:bitene er klare til å gjennomføre neste simulering.

Simuleringene er ferdig etter at mange/alle elevene har vært smittet.

Når begge testene er gjennomført skal spørsmålene for etter testingen besvares. Her kan du la elevene diskutere spørsmålene sammen to og to før dere gjennomgår spørsmålene i plenum. Elevene må her tenke over hvordan smitten spredde seg, hvilke valg de gjorde for å unngå å bli syk og eller for å bli syk. Tanken med at elevene diskuterer alle spørsmålene på én gang er at målet ved spørsmålene er at elevene skal kunne se likhetene og ulikhetene ved de tre testene.

Oppgave 4

Denne oppgaven består av to diskusjonsoppgaver der elevene må bruke erfaringene fra testingen av simuleringen, selve programmeringsarbeidet og informasjonen fra introduksjonsteksten til å diskutere de to oppgavene. Det anbefales å ta én oppgave av gangen. Gi elevene begrenset tid til å diskutere en oppgave to og to før dere tar den sammen i plenum.

Her kan det være en ide å ha det ferdige programmet oppe på tavla, og gi elevene beskjed om at de også skal bruke programmet til å diskutere de to spørsmålene. På denne måten vil du også kunne se om elevene har forstått programmet og programmet bak simuleringen.

I deloppgave A skal elevene diskutere simuleringen opp mot smittespredning. Relevante poenger å komme inn på er:

- Distanse har noe å si for hvor utsatt man blir for smittespredning
- Når man er smittsom kan man smitte andre ved å spre mikroorganismer gjennom nysing, hosting og snakking.
- Man er smittsom over en viss periode.

I deloppgave B skal elevene diskutere simuleringen opp mot immunforsvaret. Relevante poenger å komme inn på er:

- Hvordan immunforsvaret kan beskytte deg mot smitte selv om du blir utsatt for smitte.
- Hvordan kroppen ved hjelp av antistoffer og lymfocytter kan gjøre kroppen immun mot sykdommen for en viss periode da den vil kjenne den igjen og ødelegge mikroorganismene før du blir syk.

Oppgave 5

I denne oppgaven skal eleven diskutere hva som er svakhetene og styrkene ved simuleringen, samt hva som kan være mulige feilkilder i simuleringen.

I deloppgave A skal elevene diskutere svakhetene og styrkene ved simuleringen. Relevante poenger å komme inn på er:

- **Styrker:** Immunforsvaret klarer å beskytte kroppen mot mange mikroorganismer. Kroppen vil kjenne igjen smittestoffene neste gang mikroorganismen kommer inn i kroppen og du ikke blir syk. En smittet person vil spre smitte ved å blant annet hoste og nyse. Immunforsvaret arbeider mot sykdommen som gjør at du blir syk i en viss periode.
- **Svakheter:** Det kan hende alle smittes veldig raskt, eller stikk motsatt. Dette kan gjøre simuleringen mindre tydelig. Smitten kan også oppleves som sporløs, og det er ikke tydelig hvor den kommer fra.

I deloppgave B skal elevene diskutere hva som kan være årsaken til at simuleringen ikke nødvendigvis fungerer som ønsket. Relevante poenger å komme inn på er:

- At noen har endret variablene i programmet.
- Programmet mangler komponenter eller hadde feil startverdier på tallene.
- Det var noe feil med selve tilkoblingen av micro:biten som gjorde at det riktige programmet ikke er lastet opp.

Del 3: Vaksinerings

Om oppgaveheftet

Dette oppgaveheftet inneholder tre oppgaver. Den første oppgaven er programmeringsoppgaven der elevene skal bruke programmet de lagde om smittespredning, og videreutvikle den slik at vaksinerings legges til i programmet. Den andre oppgaven går ut på å teste selve simuleringen av vaksinerings, før elevene i den siste oppgaven skal gjennomføre to diskusjonsoppgaver. I starten av oppgaveheftet er det også en introduksjonstekst om hva vaksiner er og hvordan de fungerer.

Før oppstart av arbeid med oppgaveheftet

Dette vil være avhengig av hvordan dere jobber med oppgaveheftene. Dersom de ikke arbeides med på samme dag kan det være greit å ta en repetisjon av hva de gjorde i oppgavehefte del 2.

Før oppstart av arbeidet med oppgavene i heftet anbefales det å lese gjennom introduksjonsteksten til heftet i plenum. Dette er for at alle elevene skal ha vært gjennom introduksjonsteksten før de starter rett på oppgavene. Grunnen til at introduksjonsteksten er inkludert i oppgaveheftet til elevene er så de kan benytte denne informasjonen i arbeidet med diskusjonsoppgavene senere i oppgaveheftet.

Oppgave 1

Elevene skal i denne oppgaven videreutvikle programmet de lagde for å simulere smittespredning. I denne oppgaven skal de videreutvikle programmet slik at en vaksine blir implementert. De vil i denne oppgaven få det ferdige programmet de lagde om smittespredningen, samt nye løse klosser som de skal pusle sammen. De har også fått flytskjemaer som sier hva programmet skal gjøre. Disse vil igjen fungere som en type oppskrift om hva som skal skje i programmet. Deres oppgave her blir å legge til de nye klossene, og gjøre en vurdering av hvilke klosser som mangler på hvilke steder, for å sette sammen den nye programmet slik at den fungerer som flytskjemaene tilsier.

- Lenke til det ferdige prosjektet:

https://makecode.microbit.org/_gVaXXygW6C2t

En anbefaling til denne oppgaven vil også være at arkene med flytskjemaene er skrevet ut og delt ut til elevene slik at de slipper å gå frem og tilbake mellom fanene. På denne måten kan de da lettere fokusere på å pusle sammen klossene slik at det samsvarer med det flytskjemaet tilsier.

- Lenke til arket med flytskjemaene:

<https://drive.google.com/file/d/13uBGYQWtP43FTrtKsz1BQ74Zz5b8vN6e/view?usp=sharing>

Det er lagt til to oppgaver for elevene som blir fort ferdige eller som trenger en større utfordring. Det er også lagt til en forenkling av oppgaven for å hjelpe elevene som opplever programmeringsarbeidet som krevende. Dersom de fremdeles trenger hjelp gir du dem bildet av den ferdige programmet:

- https://docs.google.com/document/d/1w_tQM-bceLChpB7SOKANkJwSBx7EN-ZdjioxTRv6ll/edit?usp=sharing

Tilknyttet oppgaven har også elevene fått utdelt spørsmål som skal svares på. Disse kan godt tas i plenum i etterkant av programmeringsarbeidet. Her vil det også være smart å ta opp programmet på tavla slik at elevene får sjekket om sin stemmer. På dette måten vil micro:bitene også gjøres klare til neste oppgave, nemlig simuleringen av smittespredningen.

Svar på spørsmålene som elevene får tildelt:

Hvilke nye vilkår (som *ikke* ble brukt i del 2: Smittespredning) finner dere i dette programmet?

- “Hvis tid syk = 0 og vaksinert = 0 så”
- “Hvis tid syk = 0 og vaksinert = 1 så”
- “Hvis Vaksinert = 1 og receivedNumber >= 10 og Smittet = 0 så”
- “Hvis immunitet etter vaksinerings = 0 så”
- “Hvis Vaksinert = 0 og receivedNumber >= 10 og Smittet = 0 så”

Hva forteller disse vilkårene oss?

- “Hvis tid syk = 0 og vaksinert = 0 så” og “Hvis tid syk = 0 og vaksinert = 1 så”
 - Sjekker om micro:biten er henholdsvis “syk og uvaksinert” eller “syk og vaksinert”
- “Hvis Vaksinert = 1 og receivedNumber >= 10 og Smittet = 0 så”
 - Sjekker om micro:biten er “vaksinert, mottar smitte og samtidig ikke allerede har vært smittet”
- “Hvis immunitet etter vaksinerings = 0 så”
 - Sjekker om micro:biten ikke har oppnådd immunitet etter vaksinerings

- “Hvis Vaksineret = 0 og receivedNumber >= 10 og Smittet = 0 så”
 - Sjekker om micro:biten er “uvaksineret, mottar smitte og samtidig ikke allerede har vært smittet”

Hvorfor har dere valgt å plassere vilkårene der dere har gjort?

- Rekkefølgen på samtlige vilkår unntatt “hvis immunitet etter vaksinerings = 0 så”, som er plassert inni et annet vilkår.
- Det viktige er at operasjonene de skal utføre hører til og er plassert inni riktig løkke eller funksjon.
- Når det gjelder “hvis immunitet etter vaksinerings = 0 så” er dette en ekstra sjekk som gjøres for å kontrollere om vaksinen virket. Den kunne like gjerne tilhørt samme vilkår som klossen dette vilkåret er plassert inni, men er plassert her for å synliggjøre den uvisse effekten av vaksinen.

Hvilke nye variabler finner dere i dette programmet?

- “Vaksineret”
- “Immunitet etter vaksinerings”

Hva brukes disse variablene til, og hva skal de hjelpe med å vise?

- “Vaksineret” viser til om micro:biten har tatt vaksine eller ikke
- “Immunitet etter vaksinerings” viser til om micro:biten har blitt immun etter vaksinen ble tatt

Hvorfor har dere valgt å plassere variablene der dere har gjort?

- Variablene er alle plassert i vilkårlig rekkefølge inni “ved start”-klossen for å sette startverdiene til variablene.
- Videre brukes de i vilkår-setningene for å sjekke verdiene/tilstandene til variablene.
- Variablene er også plassert inni vilkår- og “når knapp A+B trykkes”-klossene der variablene tilegnes nye verdier.

Gjennomgang av programmet

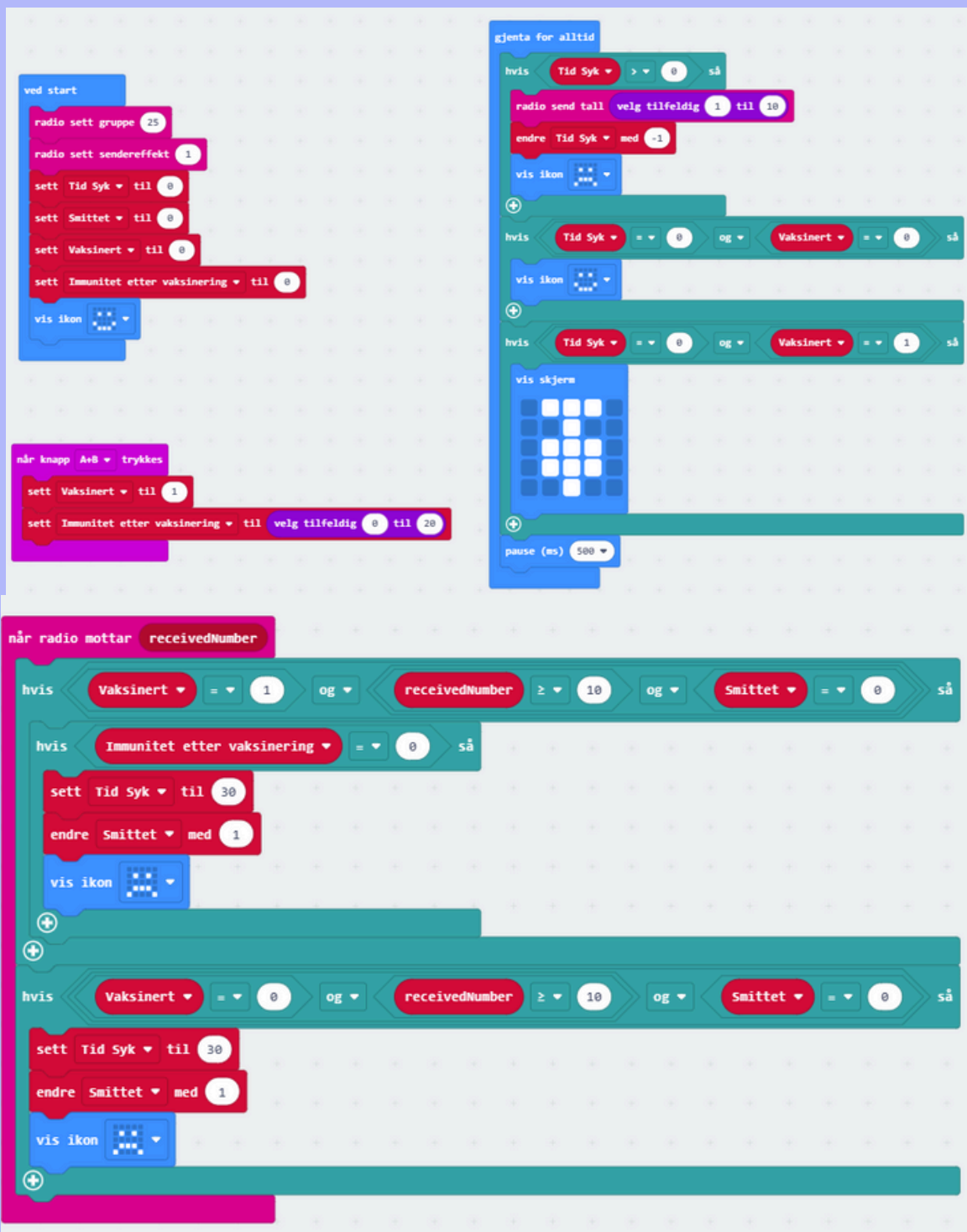
programmet simulerer et smitteforløp i en populasjon ved en fiktiv sykdom.

Det er bestemt en generell motstandsdyktighet mot smitten, der man har 1/10 sjans for å kunne smittes av smittebærer. Dette skal simulere hvordan immunforsvaret kan motstå smitte selv om man er i rekkevidden til smitten.

Det er også inkludert en vaksine, som drastisk minimerer sjansen for å smittes av smittebærere.

Alle micro:biten vil kun være smittebærere i en gitt periode. Dette simulerer at kroppen ofte bygger opp en immunitet mot samme sykdom innenfor en viss periode.

Oversikt over hele programmet



programmet består av 4 hoveddeler. Delenes oppgaver er:

- En oppstartsfunksjon, som kan bestemme hvem som starter som syke, friske og vaksinerte.
- En funksjon som vaksinerer micro:biten og avgjør om vaksinen gjorde deg immun mot sykdommen.

- En funksjon som utspiller smitteforløpet
- En mottakerfunksjon som kan motta smitten dersom personen er frisk

Oppstartsfunksjon



Oppstartsfunksjonen setter betingelsene ved start av simuleringen, der de ulike klossene gjør følgende:

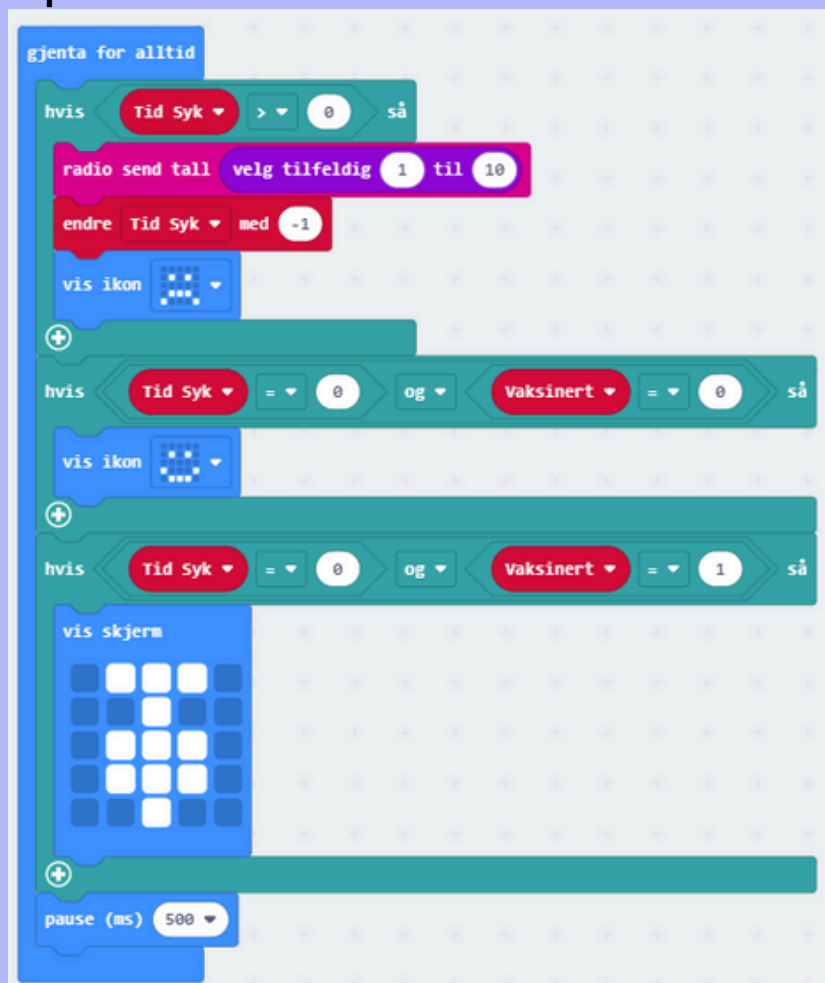
- radio-klossen er kommunikasjonskanalen mellom micro:bitene. Gruppe 25 er den valgte kanalen. Alle micro:bitene satt i denne gruppen kan da kommunisere med hverandre.
- «sendereffekt» er en valgt innstilling for hvor langt signalene sendes i rommet.
- Variabelen «Tid Syk» viser startverdi for varigheten man er smittsom. Denne er 0 for friske micro:bitere og er valgt til å starte på 30 for de som er smittsomme.
- Variabelen «Smittet» viser til om micro:biten starter som smittsom eller frisk, bestemt med henholdsvis verdiene 1 og 0.
- Variabelen «Vaksinert» viser til om micro:biten starter som vaksinert eller ikke, bestemt med henholdsvis verdiene 1 og 0.
- Variabelen «Immunitet etter vaksinerings» er en nullstilling av variabelen på starten, og den brukes for å bestemme om vaksinen har effekt mot smitten.
- Det vises et smilefjes for friske micro:bitere, og et surt fjes dersom micro:biten starter som smittsom.

Vaksinering



- Når knappene A og B trykkes inn samtidig, settes variabelen «Vaksinert» til 1, som fungerer som at micro:biten er vaksinert i resten av programmet.
- Videre vil variabelen “Immunitet etter vaksinerings” settes til et tilfeldig tall mellom 0 og 20. Dette er for å finne ut om vaksinen gjorde micro:biten immun mot sykdommen eller ikke.

Smitteforløpet



De ulike klossene i funksjonen for smitteforløpet gjør følgende:

- Dette gjentas for alltid, så lenge programmet kjører.
- Først sjekkes det om micro:biten er smittsom, ved variabelen «Tid Syk». Dersom den er større enn 0, er sykdomsforløpet satt i gang og den er smittsom.
- Da sendes et tilfeldig tall mellom 1 og 10 med «radio-klossen». Senere forklares det at kun ett av disse tallene vil føre til smittespredning. Dette simulerer at immunforsvåret ofte kan stå imot smitten – man blir ikke alltid syk.
- Når smitte-signalet er sendt ut, reduseres lengden på sykdomsforløpet med 1, helt til variabelen «Tid syk» blir 0, og den tidligere «hvis-sjekken» ikke lenger er oppfylt.
- Så lenge «Tid syk» er over 0, vises surt fjes på skjermen til micro:biten.
- Hvis «Tid syk» er 0 og «Vaksinert» er 0 er man hverken smittsom eller vaksinert, og et smilefjes vises på micro:bit-skjermen siden man er frisk.
- Hvis «Tid syk» er 0 og «Vaksinert» er 1 er man ikke smittsom men vaksinert, og en sprøyte vises på micro:bit-skjermen siden man er frisk og vaksinert.
- Før hele denne løkka gjentas, er det lagt inn en pause på 500 ms.

Smittemottaker

```
når radio mottar receivedNumber
hvis << Vaksinert = 1 og receivedNumber ≥ 10 og Smittet = 0 >> så
  hvis Immunitet etter vaksinerings = 0 så
    sett Tid Syk til 30
    endre Smittet med 1
    vis ikon
  +
  +
  hvis << Vaksinert = 0 og receivedNumber ≥ 10 og Smittet = 0 >> så
    sett Tid Syk til 30
    endre Smittet med 1
    vis ikon
  +
  +
```

Mottakerfunksjonen simulerer immunforsvarets motstand mot smitte, og klossene fungerer på følgende måte:

- Signalet som micro:biten mottar fra smittebærer er definert som et tall, med den innebygde micro:bit-klossen «receivedNumber». Som forklart tidligere er dette et tilfeldig tall mellom 1 og 10 for hvert signal som sendes fra smittebærer.
- Hvis dette tallet er større eller lik 10 («receivedNumber» er større eller lik 10) og mottaker er frisk (“Smittet” = 0) og mottaker er vaksinert (“Vaksinert” = 1), sjekkes det om vaksinen klarer å beskytte mot sykdommen.
- Dersom Immunitet etter vaksinerings er lik 0 har ikke vaksinen lyktes i å gjøre deg immun mot sykdommen, og man kan derfor bli syk likevel.
- Dersom man likevel smittes vises surt fjes på micro:bit-skjermen.
- Deretter settes sykdomsforløpet til utgangspunktet på 30.
- Deretter endres/økes «syk» med 1, slik at denne hvis-setningen ikke kan oppfylles enda en gang. Man kan altså kun smittes én gang etter oppstarten.
- Hvis variablene «receivedNumber» er større eller lik 10, «Vaksine» er 0 og «Syk» er 0, er man ikke vaksinert, har derfor ingen ekstra beskyttelse, og man blir smittet. Det samme sykdomsforløpet som tidligere blir satt i gang.

Simuleringslutt

Simuleringen vil kjøre evig. Etter hvert vil nok de fleste micro:bitene (som ikke er vaksinert) sannsynligvis ha blitt smittet, og alle skjermene vil ha vist både surt fjes og deretter smilefjes 1 gang. Med en sannsynlighet på 5% for at vaksinen ikke fungerte vil det tenkes at simuleringen heller må stanses basert på tid.

Oppgave 2

I denne oppgaven skal simuleringen av hvordan vaksinerings kan være med på å begrense smitte testes ut med hele klassen. Det er to essensielle forutsetninger for at det skal bli en vellykket test. Den første er at sendereffekten er tilpasset størrelsen på klasserommet, eller at området testen gjennomføres i er tilpasset sendereffekten. Den andre forutsetningen er at elevene har programmert vaksinen riktig og at alle tallene i programmene er like hos alle.

For å kunne gjennomføre testingen av simuleringen vil det også være avgjørende å ha en micro:bit som allerede er smittet.

- Lenke til allerede smittsomt program:
https://makecode.microbit.org/_P1aWutTLfeuT

Det skal gjennomføres tre tester der flere og flere blir vaksinert:

1. Det skal i første testen være 20% av klassen som er vaksinert.
2. Det skal i andre testen være 60% av klassen som er vaksinert.
3. Det skal i tredje testen være 90% av klassen som er vaksinert.

Elevene skal i disse testene sitte på plassene sine eller stå med en viss avstand mellom seg. Dette må avgjøres basert på sendereffekten og størrelsen på klasserommet. Har må du finne det som fungerer i ditt klasserom. Når alle elevene er klare skal du dele ut hvem som skal vaksinere seg. Det er her viktig at du sprer de vaksinerte ut i rommet slik at alle ikke er på samme plass. Dette vil gjøre at simuleringen ikke fungerer etter hensikten.

Etter hver test skal micro:bitene nullstilles ved å trykke på nullstillingsknappen. Her er det viktig å tenke på at micro:biten som er smittet ved start ikke nullstilles før alle de andre micro:bitene er klare. Dette inkluderer at alle de som skal ta vaksine, har tatt den. Etter hver test øker du antallet vaksinerte i klassen i samsvar med prosentandelen. Her kan det være greit å tenke at elevene som kanskje kan finne på å vaksinere seg selv om de ikke skal blir vaksinert tidlig i testene.

Det kan være litt vanskelig å vurdere når denne simuleringen er ferdig, så her må du enten vurdere om simuleringen avsluttes basert tid eller etter at mange elevene har blitt smittet.

Når begge testene er gjennomført skal spørsmålene for etter testingen besvares. Her kan du la elevene diskutere alle sammen to og to før dere gjennomgår spørsmålene i plenum. Elevene må her tenke over hvilken rolle vaksineringsen hadde for å minimere at folk ble syke.

Oppgave 3

Denne oppgaven består av to diskusjonsoppgaver der elevene må bruke erfaringene fra testingen av simuleringen, selve programmeringsarbeidet og informasjonen fra introduksjonsteksten til å diskutere de to oppgavene. Det anbefales her at elevparene først diskuterer den ene oppgaven sammen, før det tas i plenum. Hvordan dere tar diskusjonen i plenum vil være helt opp til deg.

Her kan det være en ide å ha det ferdige programmet oppe på tavla, og gi elevene beskjed om at de også skal bruke programmet til å diskutere de to spørsmålene. På denne måten vil du også kunne se om elevene har forstått programmet og programmet bak simuleringen.

I deloppgave A skal elevene diskutere hvordan vaksinerings kan beskytte dem selv mot smitte. Her er relevante poenger å komme inn på:

- At vaksinen fungerer slik at lymfocytene og antistoffer dannes uten at man har vært gjennom en sykdom, og at dette vil beskytte deg mot å bli syk dersom du senere blir utsatt for smitte. Det kan ses i programmet ved at det er mye mindre sannsynlighet for å bli smittet.

I deloppgave B skal elevene diskutere hvordan det at man selv er vaksinert kan beskytte andre mot smitte. Her er relevante poenger å komme inn på:

- Når det er færre syke som går rundt vil det være en mindre sannsynlighet for at folk blir syke. Når du har tatt vaksine mot sykdommen vil du mest sannsynlig ikke bli syk og dermed ikke kunne smitte andre. Her er også flokkimmunitet et sentralt begrep.

Oppgave 4

I denne oppgaven skal eleven diskutere hva som er svakhetene og styrkene ved simuleringen, samt hva som kan være mulige feilkilder i simuleringen.

I deloppgave A skal elevene diskutere svakhetene og styrkene ved simuleringen. Her er relevante poenger å komme inn på være:

- **Styrker:** Viser hvordan en vaksine kan begrense smittespredning og viser at vaksinene ikke vil være 100% effektiv.
- **Svakheter:** Det kan godt hende at ingen med vaksine blir smittet, og at den i praksis da vil være 100% effektiv til tross for programmet.

I deloppgave B skal elevene diskutere hva som kan være årsaken til at simuleringen ikke nødvendigvis fungerer som ønsket. Her kan relevante poenger å komme inn på være:

- Elevene som skulle være uvaksinerte kan ha trykket på A+B.
- Eleven oppfattet ikke at den hadde blitt smittet og vært gjennom sykdomsforløpet.
- Elevene kan ha endret variablene i programmet før de lastet det ned på micro:biten.
- Programmet manglet komponenter eller hadde feil startverdier på tallene.
- Det var noe feil med selve tilkoblingen av micro:biten som gjør at riktig program ikke er lastet opp.

Litteraturliste

- Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair Programming: Under What Conditions Is It Advantageous for Middle School Students? *Journal of Research on Technology in Education*, 46(3), 277–296.
<https://doi.org/10.1080/15391523.2014.888272>
- Gjøvik, Ø., & Høyland, J. (2022). *Kloss for Kloss: Blokkprogrammering for lærere*. Universitetsforlaget.
- Grindeland, J. M., Staberg, R. L., & Tandberg, C. (2020). *Biologi for lærere* (2. utg.). Gyldendal Akademisk.
- Hanks, B., Fitzgerald, S. C., McCauley, R., Murphy, L., & Zander, C. (2011). Pair programming in education: A literature review. *Computer Science Education*, 21(2), 135–173. <https://doi.org/10.1080/08993408.2011.579808>
- Haraldsrud, A. D., Sveinsson, H. A., & Løvold, H. H. (2020). *Programmering i skolen*. universitetsforlaget.
- Nia, M. G., & de Vries, M. J. (2017). Models as artefacts of a dual nature: A philosophical contribution to teaching about models designed and used in engineering practice. *International Journal of Technology and Design Education*, 27(4), 627–653. <https://doi.org/10.1007/s10798-016-9364-1>
- Parsons, D. (2021). Mobile Participatory Simulation of COVID-19 Transmission Using the micro:bit. I 2021 IEEE International Conference on Engineering, Technology & Education (TALE) (s. 1–4). IEEE.
<https://doi.org/10.1109/TALE52509.2021.9678867>
- Pepper, M., Renken, M., Girault, I., Chiocciariello, A., & Otreel-Cass, K. (2016). Distinctions Between Computer Simulations and Other Technologies for Science Education. I M. Renken, M. Pepper, K. Otreel-Cass, I. Girault, & C. Augusto (Red.), *Simulations as Scaffolds in Science Education*, in SpringerBriefs in Educational Communications and Technology. Springer Cham. <https://doi.org/10.1007/978-3-319-24615-4>
- Rossing, N. K., & Johansen, A. (2021). MICRO:BIT Radiokommunikasjon – Forslag til undervisningsopplegg. Skolelaboratoriet ved NTNU.
<https://www.ntnu.no/documents/2004699/12108297/MicroBit+Radiokommunikasjon+++forslag+til+undervisningsopplegg.pdf/9e0e6a25-1cc6-5051-4820-1dcfdb801205?t=1642515239732>
- Steineger, E., & Wahl, A. (2021). *Naturfag 10*. Cappelen Damm.
- Vygotsky, L. S. (1978). *Mind in Society* (M. Cole, V. John-Steiner, S. Scribner, & E. Souberman, Red.). Harvard University Press.
<https://doi.org/10.2307/j.ctvjf9vz4>

Litteraturliste

- Williams, L., & Kessler, R. (2000). All I really need to know about pair programming I learned in kindergarten. *Communications of the ACM*, 43, 108–114. <https://doi.org/10.1145/332833.332848>
- Zhi, R., Chi, M., Barnes, T., & Price, T. W. (2019). Evaluating the Effectiveness of Parsons Problems for Block-based Programming. I ICER '19: Proceedings of the 2019 ACM Conference on International Computing Education Research (s. 51–59). Association for Computing Machinery. <https://doi.org/10.1145/3291279.3339419>

7.5 VEDLEGG 5: GODKJENT SIKT-SØKNAD



Vurdering av behandling av personopplysninger

Referansenummer

609316

Vurderingstype

Automatisk

Dato

09.04.2024

Tittel

Kombinering av programmering, smittespredning og vaksinerings i naturfagsundervisningen

Behandlingsansvarlig institusjon

Norges teknisk-naturvitenskapelige universitet / Fakultet for samfunns- og utdanningsvitenskap (SU) / Institutt for lærerutdanning

Prosjektansvarlig

Arne Stormo

Student

Synne Christin Colling

Prosjektperiode

01.01.2024 - 27.05.2026

Kategorier personopplysninger

Alminnelige

Lovlig grunnlag

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

Behandlingen av personopplysningene er lovlig så fremt den gjennomføres som oppgitt i meldeskjemaet. Det lovlige grunnlaget gjelder til 27.05.2026.

[Meldeskjema](#) **Grunnlag for automatisk vurdering**

Meldeskjemaet har fått en automatisk vurdering. Det vil si at vurderingen er foretatt maskinelt, basert på informasjonen som er fylt inn i meldeskjemaet. Kun behandling av personopplysninger med lav personvernulempe og risiko får automatisk vurdering. Sentrale kriterier er:

- De registrerte er over 15 år
- Behandlingen omfatter ikke særlige kategorier personopplysninger;
 - Rasemessig eller etnisk opprinnelse
 - Politisk, religiøs eller filosofisk overbevisning
 - Fagforeningsmedlemskap
 - Genetiske data
 - Biometriske data for å entydig identifisere et individ
 - Helseopplysninger
 - Seksuelle forhold eller seksuell orientering
- Behandlingen omfatter ikke opplysninger om straffedommer og lovovertridelser
- Personopplysningene skal ikke behandles utenfor EU/EØS-området, og ingen som befinner seg utenfor EU/EØS skal ha tilgang til personopplysningene
- De registrerte mottar informasjon på forhånd om behandlingen av personopplysningene.

Informasjon til de registrerte (utvalgene) om behandlingen må inneholde

- Den behandlingsansvarliges identitet og kontaktopplysninger
- Kontaktopplysninger til personvernombudet (hvis relevant)
- Formålet med behandlingen av personopplysningene
- Det vitenskapelige formålet (formålet med studien)
- Det lovlige grunnlaget for behandlingen av personopplysningene
- Hvilke personopplysninger som vil bli behandlet, og hvordan de samles inn, eller hvor de hentes fra
- Hvem som vil få tilgang til personopplysningene (kategorier mottakere)
- Hvor lenge personopplysningene vil bli behandlet
- Retten til å trekke samtykket tilbake og øvrige rettigheter

Vi anbefaler å bruke vår [mal til informasjonsskriv](#).

Informasjonssikkerhet

Du må behandle personopplysningene i tråd med retningslinjene for informasjonssikkerhet og lagringsguider ved behandlingsansvarlig institusjon. Institusjonen er ansvarlig for at vilkårene for personvernforordningen artikkel 5.1. d) riktighet, 5. 1. f) integritet og konfidensialitet, og 32 sikkerhet er oppfylt.

