

Katarina Sørhøy og Rikke-Helen Aakernes Nielsen

# Programmering og naturlig seleksjon kombinert i naturfagundervisning

Utvikling av en læringsressurs

Masteroppgave i naturfag. Grunnskolelærerutdanning 5.-10.trinn

Veileder: Dag Atle Lysne

Medveileder: Torunn Smevik

Mai 2023



Katarina Sørhøy og Rikke-Helen Aakernes Nielsen

# **Programmering og naturlig seleksjon kombinert i naturfagundervisning**

Utvikling av en læringsressurs

Masteroppgave i naturfag. Grunnskolelærerutdanning 5.-10.trinn  
Veileder: Dag Atle Lysne  
Medveileder: Torunn Smevik  
Mai 2023

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



Kunnskap for en bedre verden





## Sammendrag

Denne studien presenterer utviklingen og utprøvingen av en læringsressurs som kombinerer arbeid med programmering og naturlig seleksjon for å imøtekomme utfordringene knyttet til disse temaene i skolen. Utviklingsarbeidet tar utgangspunkt i følgende kompetansemål i naturfag etter 10.trinn som sier at elever skal *bruke programmering til å utforske naturfaglige fenomener* (Kunnskapsdepartementet, 2019). Målet med studiet er å utvikle en læringsressurs som læreren kan ta i bruk i egen undervisning for å kombinere arbeidet med programmering med arbeidet med evolusjon og naturlig seleksjon. Vi ønsker dermed at vår studie kan bidra til bedre praksis innenfor arbeidet med programmering i naturfag.

Metoden som er brukt i utviklingen er inspirert av pedagogisk designforskning (Bjørndal, 2013). I utviklingsprosessen har det blitt gjennomført en kvalitetsutvikling og to sykluser med utprøving. Utprøvingen ble gjennomført i to klasser ved en ungdomsskole, hvor observasjon og samtale med en lærer utgjorde erfaringsgrunnlaget for videreutvikling av læringsressursen. Læringsressursen som har blitt utviklet består av to ulike simuleringer som illustrerer naturlig seleksjon, to nivådelte oppgavehefter og en lærerveiledning.

Vi har utviklet læringsressursen på bakgrunn av tidligere forskning om misoppfatninger i naturlig seleksjon og programmering i skolen. Basert på erfaringene fra utprøving har vi indikasjoner på at vår læringsressurs imøtekommer disse utfordringene. Vår studie viser at læringsressursen fungerer godt på ungdomsskolen. Gjennom utprøvingene har vi indikasjoner på at arbeid med læringsressursen, og dermed kombinasjonen av programmering og naturlig seleksjon, er motiverende og kan bidra til at elevene forstår evolusjon bedre.

## Abstract

This study presents the development and testing of a learning resource that combines programming and natural selection to address the challenges associated with these topics in schools. The development is based on the following competence aims in science for 10th grade, which state that students should use programming to explore scientific phenomena (Kunnskapsdepartementet, 2019). The aim of the study is to develop a learning resource that teachers can use in their own teaching to combine programming with the study of evolution and natural selection. Along these lines, our study aims to contribute to improved practices in programming within the field of science education.

The method used in the development is inspired by pedagogical design research (Bjørndal, 2013). In the development process, a quality development and two cycles of testing were conducted. The testing was carried out in two science classes at a secondary school, where observation and conversation with a teacher formed the data acquisition for further development of the learning resource. The learning resource consists of two different simulations that illustrate natural selection, two different levelled workbooks, and a teacher's guide.

We have developed the learning resource based on previous research on misconceptions in natural selection and programming in schools. Based on the experiences from testing, we have indications that our learning resource addresses these challenges. Our study demonstrates that the learning resource works well in secondary schools. Through the testing, we have indications that working with the learning resource, and hence the combination of programming and natural selection, is motivating and can contribute to better understanding of evolution among students.

## Forord

I det vi leverer denne masteroppgaven er vi ferdige med lærerutdanningen og ferdige utdannet lærere. Det er med skrekkblandet fryd at vi starter som lærere, men nå har vi i hvert fall et undervisningsopplegg ordnet til høsten.

Studietiden på lærerstudiet har inneholdt artige fester, mye lesing, lange lunsjer, eksamener, prokrastinering, ekskursionser, strikkekvelder, praksis over hele Midt-Norge og ikke minst utrolig fine folk. Tusen takk til alle lærervevenne våre, det blir rart og trist å ikke se dere hver dag lengre.

En stor takk til vår veileder Dag Atle Lysne for god, morsom, oppmuntrende og hyppig veiledning det siste halvåret. Spesielt er vi glade for at du har vist så stort engasjement for vårt prosjekt. Videre takk til vår medveileder Torunn Smevik, eller som vi vil kalle det vår programmeringsveileder, for å guide oss programmeringsnybegynnere gjennom utviklingen av denne noe optimistiske læringsressursen. Videre vil vi takke skolen og læreren som var til stor hjelp for å gjøre læringsressursen vår så god som mulig.

En liten oppmerksomhet må også sendes i retning programmeringsgeniet Edvard som har støttet oss når programmeringen ble vanskeligere enn antatt. Takk for gode tips til hvordan vi kunne løse problemene og fremdeles lage simuleringen slik vi ønsket. En god vin kommer din vei!

Takk også til gode romkamerater her i Trondheim for avbrekk, og familien vår hjemme for oppmuntrende ord.

Til slutt vil vi gi hverandre en stor takk. Det beste valget vi har tatt denne studietiden må være å skrive denne masteren sammen. Det å ha noen å tenke sammen med hele det siste året har vært veldig deilig.

Trondheim, mai 2023

Katarina Sørhøy og Rikke-Helen Aakernes Nielsen

## Figurer

Figur 1: Tilpassing til Bjørdals (2013) modell av pedagogisk forskningsdesign, som viser oversikt over prosessene for utviklingen av vår læringsressurs. Utgavene er læringsressursen før vi gjør nye endringer grunnet nytt erfaringsgrunnlag. ....	12
Figur 2: Vår tilpasning til Sandoval (2014) sitt antakelseskart for pedagogisk designforskning. ....	13
Figur 3a og b: Skjermbilder av simuleringen naturlig seleksjon ved ulike miljøer. ....	17
Figur 4: Utklipp fra programmeringen i simuleringen. ....	18
Figur 5: Eksempel på oppgave fra oppgaveheftet Naturlig seleksjon. ....	19
Figur 6: Eksempel hentet fra lærerveiledningen. I venstre kolonne er et utklipp av en kodesekvens i Scratch. I høyre kolonne beskrives det hva kodene gjør og skal illustrere om naturlig seleksjon. ....	20
Figur 7: I denne kodesekvensen parres en hvit og en brun hare. Variabelen «farge på pels» velger tilfeldig et tall mellom 1-6. Om tallet er 1,2 eller 3 er avkommet brunt, ved 4 eller 5 er avkommet hvitt, og om tallet er 6 er pelsfargen til avkommet blandet. ....	22
Figur 8a og b: Funksjonen for reproduksjon for to hvite harer. Inne i denne funksjonen benyttes det en løkke. Ved å undersøke denne funksjonen kan det være mer forståelig hvordan og hvorfor en løkke brukes. I dette eksempelet brukes løkker for å illustrere hvor mange avkom som fødes. Figuren viser også at vi bruker begrepet «avkom er hvit hare» istedenfor «lag kone av meg».....	23
Figur 9: Liste og forklaring av begreper fra økologi som brukes i oppgaveheftet Naturlig seleksjon. ....	25
Figur 10: Klossene som elevene skal bruke for å ferdigstille programmet er hentet frem. ....	28
Figur 11: Variabel som viser antall harer i simuleringen. ....	29
Figur 12: Eksempel på en melding som forteller hva som har hendt i programmet. ....	30
Figur 13a og b: Viser endringer som ble gjort i oppgaveheftet. Istedenfor å bruke hele kodesekvensen brukes bare en begrenset del som elevene skal fokusere på i oppgaven. ....	30

## Innholdsfortegnelse

1. Innledning .....	1
1.1 Bakgrunn for utviklingen av læringsressursen .....	2
1.2 Avgrensninger .....	2
1.3 Formålet med oppgaven .....	2
1.4 Hva skiller vår læringsressurs fra det som allerede er tilgjengelig?.....	3
1.5 Oppgavens struktur.....	3
2. Teori og tidligere forskning.....	4
2.1 Teori .....	4
2.1.1 Modeller i naturfag .....	4
2.1.2 Pedagogiske metoder og verktøy .....	4
2.1.2.1 PRIMM.....	5
2.2 Tidligere forskning .....	6
2.2.1 Programmering i naturfag .....	6
2.2.2 Misoppfatninger i programmering.....	7
2.2.3 Modeller av naturlige fenomener .....	8
2.2.4 Misoppfatninger i evolusjon og naturlig seleksjon.....	8
3. Metode og metodeteori.....	10
3.1 Vitenskapsteoretisk ståsted .....	10
3.2 Pedagogisk designforskning .....	10
3.2.1 Forberedelse av designeksperimentet .....	11
3.2.2 Gjennomføring av designeksperimentet.....	11
3.2.3 Retrospektive analyser.....	12
3.3 Antagelseskart for pedagogisk designforskning .....	12
3.4 Forskningssted og forskningsdeltakere .....	13
3.5 Kvalitet i forskningen.....	14
3.6 Forskningsetikk .....	14
4. Presentasjon av læringsressurs.....	16
4.1 Målet med læringsressursen .....	16
4.2 Simuleringer .....	17
4.3 Oppgavehefter .....	18
4.4 Lærerveiledning.....	19
5. Resultat og diskusjon .....	21
5.1 Utvikle en læringsressurs som kombinerer arbeid med programmering og naturlig seleksjon.....	21
5.1.1 Utviklingsfasen .....	21
5.1.1.1 Simulering .....	21
5.1.1.2 Oppgavehefte .....	24
5.1.1.3 Lærerveiledning .....	25

5.1.2	Utprøvningsfasen .....	26
5.1.3	Retrospektiv analysen .....	27
5.1.3.1	Simulering .....	28
5.1.3.2	Oppgavehefte .....	30
5.1.3.3	Lærerveiledningen .....	31
5.1.4	Har vi nådd målet i utviklingsprosessen? .....	32
5.2	Delmål 2: Utvikle en læringsressurs som imøtekommer utfordringene i skolen knyttet til kombinasjonen av programmering og naturlig seleksjon .....	33
5.2.1	Utfordringer med arbeid i programmering .....	33
5.2.2	Utfordringer med naturlig seleksjon .....	34
5.2.3	Utfordringer med å kombinere programmering og naturlig seleksjon .....	35
5.2.4	Har vi nådd målet om å utvikle en læringsressurs som imøtekommer utfordringer i skolen knyttet til kombinasjonen av programmering og naturlig seleksjon? .....	37
5.3	Studiens begrensninger .....	37
5.3.1	Vår rolle som forskere og utviklere .....	37
5.3.2	Begrensninger som følge av tid .....	38
5.3.3	Refleksjoner rundt utvalg .....	38
6.	Forslag til videreutvikling av læringsressursen .....	39
7.	Konklusjon .....	40
8.	Litteraturliste .....	41
	Vedlegg .....	45

## 1. Innledning

Ved innføringen av Kunnskapsløftet (LK20) i 2020 ble programmering inkludert i skolen. Programmering i skolen innebærer at elever skal lære seg å kode og programmere teknologiske systemer. Samtidig skal programmering ha en overføringsverdi i form av å utvikle elevers kompetanse innenfor problemløsning, samarbeid, kreativ tenkning og utforskning. Ytterligere skal programmering utvikle elevers nysgjerrighet, selvtillit og skaperglede i arbeid med faglige temaer (Statped, 2021).

LK20 beskriver at elever skal utvikle kompetanse i programmering gjennom kompetansemål og kjerneelementer i fagene: kunst og håndverk, musikk, matematikk og naturfag. I læreplanen i naturfag inkluderes programmering i kompetansemålene etter 7.trinn: *utforske, lage og programmere teknologiske systemer som består av deler som virker sammen* og etter 10.trinn: *bruke programmering til å utforske naturfaglige fenomener og utforske, forstå og lage teknologiske systemer som består av en sender og en mottaker* (Kunnskapsdepartementet, 2019). Videre inkluderes programmering i kjerneelementet *teknologi* ved at *elevene skal forstå, skape og bruke teknologi, inkludert programmering og modellering, i arbeid med naturfag, og at kunnskap om og kompetanse innenfor teknologi er viktig i et bærekraftsperspektiv* (Kunnskapsdepartementet, 2019). Etter innføringen av LK20 har programmering fått en sentral rolle i matematikk-, kunst og håndverk-, musikk- og naturfagundervisningen i skolen, og det kreves at lærere har kompetanse i programmering og hvordan dette kan tas i bruk i deres fagfelt.

For å få en oversikt over kompetansenivået til lærere innen programmering, gjennomførte vi høsten 2022 en pilotundersøkelse på en grunnskole i Trondheim. Undersøkelsen besto av et spørreskjema knyttet til implementeringen av programmering i undervisningen. Pilotundersøkelsen viser at majoriteten av lærerne på denne skolen hadde lav kompetanse i programmering. Funnene indikerer at den manglende kompetansen gjør det utfordrende for lærere å inkludere programmering i undervisning. Piloten viser videre at majoriteten av lærerne på den valgte skolen, ønsket seg høyere kompetanse i programmering og flere læringsressurser for å lettere ta i bruk programmering i undervisningen. Vår pilotstudie samsvarer med Berggren og Jom (2019) sin studie som beskriver at lærere er positive til å innføre programmering i skolen, men mangler programmeringskompetanse. Studien viser at den manglende programmeringskompetansen gjør lærere usikre på hvordan innføringen av programmering vil påvirke undervisningen (Berggren & Jom, 2019).

På bakgrunn av innføringen av programmering i skolen og behovet for å inneha kompetanse for å gjennomføre opplæring i henhold til LK20, ønsker vi som fremtidige naturfagslærere å utfordre og utvikle oss innenfor programmering. Til tross for innføringen av programmering i skolen har det vært mangel på programmeringsundervisning på lærerstudiet. Dermed ønsker vi å utvikle egen kompetanse innenfor fagfeltet, som en del av masteroppgaven. Vårt mål er å få erfaringer i arbeid med å utvikle undervisning som kombinerer programmering med et annet faglig tema i naturfag. I tillegg ønsker vi å utvikle en læringsressurs som skal bidra til å støtte lærere i deres arbeid med å inkludere programmering som en del av naturfag.

## 1.1 Bakgrunn for utviklingen av læringsressursen

I læreplanen i naturfag beskrives det i kompetansemålet etter 10.trinn at elever skal *bruke programmering til å utforske naturfaglig fenomener* (Kunnskapsdepartementet, 2019). Til tross for dette viser studien til Caballero et al. (2014) at det er utfordrende å koble programmering til andre faglige temaer. Funn fra studien beskriver hvordan elever som jobber samtidig med programmering og et annet naturfaglig tema opplever utfordringer med å forstå om feil i programmet ligger i programmeringen eller den faglige forståelsen. I tillegg beskriver studien til Bungum et al. (2014) hvordan veiledning og styring er avgjørende for at elever skal lære og anvende naturfaglig kunnskap og kompetanse i teknologiarbeid. På bakgrunn av disse studiene ønsker vi å utvikle en læringsressurs som tar utgangspunkt i programmering for å utforske naturfaglige fenomener.

## 1.2 Avgrensninger

For å avgrense masteroppgaven og læringsressursen har vi tatt utgangspunkt i et tema som elever finner utfordrende og har misoppfatninger om. Vi ønsker at programmering brukes som en alternativ undervisningsmetode, som kan bidra til å forhindre og avklare misoppfatninger innenfor temaet. Med dette som utgangspunkt har vi valgt å se nærmere på hvordan programmering kan brukes i arbeid med evolusjon og naturlig seleksjon. Avgrensningen er gjort på bakgrunn av en mengde studier som viser at befolkningen generelt har manglende forståelse av og innehar misoppfatninger om naturlig seleksjon (Harms & Reiss, 2019; Gregory, 2009; Reydon, 2021).

Ytterligere har vi avgrenset læringsressursen til å fokusere på blokkprogrammering og derav programmeringsspråket Scratch. Denne avgrensningen er gjort på bakgrunn av litteratur som beskriver at blokkprogrammering reduserer elevenes kognitive belastning (Resnick et al., 2009).

## 1.3 Formålet med oppgaven

Gjennom arbeidet med masterstudien utvikler vi en læringsressurs i form av to simuleringer av naturlig seleksjon i blokkprogrammeringsspråket Scratch, to digitale oppgavehefter og en lærerveiledning. Læringsressursen er tiltenkt å brukes som en del av temaet evolusjon, hvor fokuset er naturlig seleksjon. Læringsressursen er utviklet med mål om å støtte og inspirere lærere i arbeidet med programmering samtidig med et annet naturfaglig tema. Det overordnede målet for vårt utviklingsarbeid er å: *utvikle en læringsressurs som bruker programmering til å utforske naturlige fenomener.*

I arbeidet med å nå målet for utviklingsarbeidet har vi definert to delmål:

1. Utvikle en læringsressurs som kombinerer arbeid med programmering og naturlig seleksjon.
2. Utvikle en læringsressurs som imøtekommer utfordringene i skolen knyttet til kombinasjonen av programmering og naturlig seleksjon.

Delmål 1 fokuserer på prosessen og utviklingen av læringsressursen, mens delmål 2 fokuserer på hvordan læringsressursen kan bidra til å forhindre misoppfatninger i programmering og naturlig seleksjon. I arbeidet for å nå målet og delmålet i masteroppgaven, vil den utviklede læringsressursen utprøves, før produktet evalueres, revideres og forbedres.



## 1.4 Hva skiller vår læringsressurs fra det som allerede er tilgjengelig?

Læringsressursen som utvikles vil være en læringsressurs bestående av to simuleringer, to oppgavehefter som kombinerer programmering og naturlig seleksjon og en lærerveiledning. Vårt hovedfokus har vært å utvikle en simulering av naturlig seleksjon, til tross for at vi er kjent med at det finnes flere datasimuleringer som illustrerer dette fenomenet, slik som PhET (u.å.) sin.

PhET sin simulering, består i likhet med vår simulering av naturlig seleksjon av harer. I deres simulering kan brukeren utforske naturlig seleksjon ved å forandre på enkelte vilkår (PhET Interactive Simulations, u.å.). Vi har hentet inspirasjon for designet og innholdet fra PhET-simuleringen i utviklingen av vår egen simulering. Forskjellen fra vår simulering er at i PhET-simuleringen er programmeringen skjult og dermed lærer ikke elevene noe om programmering når de bruker denne simuleringen. Det som gjør vår simulering unik er at elevene lærer om programmering samtidig som de lærer om seleksjon. Så langt vi vet finnes det ikke noen didaktiske ressurser med samme formål innen temaet naturlig seleksjon.

## 1.5 Oppgavens struktur

Ettersom vårt masterprosjekt er en utviklingsrettet master, vil oppgaven bestå av to komponenter. Det ene er selve læringsressursen vi har utviklet: som består av to oppgavehefter, to simuleringer og en lærerveiledning. Læringsressursen presenteres i kapittel 4 og er i tillegg vedlagt i sin helhet som vedlegg 1, 2 og 3. Det andre komponenten er selve rapporten som gir innsikt i prosessen og teorien som er brukt i utviklingen av læringsressursen.

Masteroppgaven er delt inn i syv kapitler. I kapittel 1 presenteres innledningen til oppgaven. Videre presenteres teori og tidligere forskning som er relevant for oppgaven i kapittel 2. I kapittel 3 beskrives vår forståelse av pedagogisk designforskning (Bjørndal, 2013), som er metoden masteroppgaven og utviklingen baserer seg på.

Læringsressursen som er utviklet presenteres i kapittel 4. Ettersom vår masteroppgave er en utviklingsbasert master har vi ikke benyttet en tradisjonell analyse. Resultatene fra vår forskning er erfaringene fra gjennomføringene som vi bruker for å endre på forskningsdesignet. Disse erfaringene presenteres i kapittel 5. I tillegg presenterer kapittel 5 valg som er tatt i utviklingen av læringsressursen og videreutvikling av læringsressursen. Igjennom kapittelet diskuteres vårt utviklingsarbeid opp mot teori og erfaringer fra gjennomføring. I kapittel 6 beskrives forslag til videreutvikling av læringsressursen. Avslutningsvis oppsummerer vi prosessen med utviklingsarbeidet og konkludere om studien har nådd sine mål i kapittel 7.

## 2. Teori og tidligere forskning

I kapittelet om teori og tidligere forskning, presenteres relevant litteratur som utgjør grunnlaget for diskusjonen i masteroppgaven. Kapittelet inneholder litteratur innen fagfeltene programmering, modeller, pedagogiske metoder og verktøy, misoppfatninger og evolusjon.

### 2.1 Teori

I teoridelen presenteres teori om algoritmisk tenkning, modeller i naturfag og pedagogiske metoder og verktøy. Denne teorien er sentral i masterprosjektet for å utvikle en læringsressurs innenfor programmering som er tilpasset mangfoldet i skolen.

#### 2.1.1 Modeller i naturfag

I læreplanen er modeller en sentral del av naturfag og forklares med at *forståelse, bruk og arbeid med å lage modeller skal bidra til at den naturfaglige forståelsen utvikles* (Kunnskapsdepartementet, 2019). I naturfag brukes modeller for å forenkle og beskrive objekter, ideer, hendelser, prosesser og strukturer i systemer (Gilbert et al., 2000). Modeller kan brukes som et verktøy for å forklare og forutsi naturfaglige fenomener (Osborne, 2014). Det er viktig å poengtere at modeller er forenklinger av virkeligheten. Noen deler av fenomenet fremheves i modellen, mens andre deler samsvarer ikke med virkeligheten (Hannisdal & Ringnes, 2015).

Modellering er prosessen med å produsere en modell. Datasimuleringer gir elever mulighet til å modellere naturlige fenomener gjennom å endre betingelser og måle resultater. Denne type simulering kan brukes for å interagere med et naturlig fenomen og hvor elevene kan teste ut "hva-hvis"- senarioer. Modeller som datasimuleringer er hensiktsmessig å bruke til naturfaglige fenomener som er vanskelig å observere direkte (German, 2019).

Modellering og modeller er felles for disiplinene forskning, teknologi, ingeniørarbeid og matematikk og kan derfor brukes som et naturlig bindeledd mellom disiplinene. Evnen til å lage, bruke, evaluere og revidere modeller er essensielle for å kunne tilegne seg dybdekunnskap om både teknologiske utviklingsprosesser og naturfaglige praksiser. Å være kritisk til modellers formål og svakheter vil lede til en mer autentisk naturfagundervisning (Hallstøm & Schönborn, 2019).

#### 2.1.2 Pedagogiske metoder og verktøy

I utviklingen av læringsressursen har vi tatt utgangspunkt i flere pedagogiske metoder og verktøy. Læringsressursen baserer seg på Vygotsky (1978) sin sosiokulturelle teori, om at læring skjer i felleskapet med andre og at kunnskap utvikles gjennom å utveksle ideer mellom personer. Vygotsky skiller mellom to former for læring: det som barnet mestrer uten hjelp og det som barnet mestrer med støtte. Han definerer den læringen barnet mestrer med støtte som *den nærmeste utviklingssonen*. Gjennom å støtte eleven i læringen får eleven utvidet sitt utviklingspotensial og dermed utviklet kompetanse for å løse lignende problemer på egenhånd (Vygotsky 1978, henvist i Moen, 2015). For at elever skal få lære i den nærmeste utviklingssonen, må opplegget individuelt tilpasses og oppgavene differensieres. Særlig i programmering er det store nivåforskjeller og det er viktig å differensiere oppgavene for både å motivere elever med stort og lavere

læringspotensial (Haraldsrud et al., 2020). Elever med stort læringspotensial omfatter både elever som presterer høyt og elever som har potensiale for å gjøre det (Utdanningsdirektoratet, 2021), mens elever med lavere læringspotensialet definerer vi som elever som har utfordringer med læring. Til elevgruppen med stort læringspotensial er det en fordel å ha ekstra utfordrende oppgaver som enten er mer tidskrevende, har høyere nivå, er utforskende eller krever at eleven må søke på internett for å utvide programmeringsteknikkene (Haraldsrud et al., 2020). Til elevgruppen med lavere læringspotensial er det en fordel å ha hint knyttet til oppgavene for å motivere dem. Det kan være å starte med et halvferdig program, gi elevene deler av koden de skal bruke, forstå koden eller rette opp i koder (Gjøvik & Høyland, 2022). For øvrig for å differensiere oppgavene, kan læreren organisere elevene i par for at elevene skal lære i den nærmeste utviklingssonen. Dette pedagogiske virkemiddelet gjør at elevene kan hjelpe og støtte hverandre, samtidig som de også utfordrer og forklarer til hverandre.

Et annet pedagogisk virkemiddel er bruken av støttestrukturer. Støttestrukturer inkluderer alle redskapene elevene får med seg inn i et arbeid. Disse hjelpemidlene brukes som et stillas for å legge til rette for god læring hos elevene. Noen ganger er det nødvendig for eleven med tilrettelegging og støtte for å oppnå mestring, og eleven hadde ikke klart de faglige utfordringene uten denne støtten (Knain et al., 2019).

En potensiell utfordring med undervisning i programmering, er at læringsoppgaven kan føre til for stor kognitiv belastning for elevene. Mayer og Moreno (2003) beskriver at kognitiv overbelastning fremkalles av at læringsoppgaven overskrider prosesseringskapasiteten til det kognitive systemet hos elevene. Meningsfull læring krever at kognitiv prosessering har strenge grenser (Mayer & Moreno, 2003).

Teoretikeren Dewey er kjent for teorien om *Learning by doing*, som innebærer at man lærer gjennom å gjøre. Til tross for utsagnet mener Dewey (1996) at ingen erfaringer har betydning uten tanke og refleksjon. Dette begrunnes med at tanke og refleksjon er forbindelsen mellom det vi gjør og konsekvensen av handlingene våre. Når vi gjør noe, stimuleres vi til å tenke og reflektere for å bestemme og bearbeide betydningen av det som er utført og lært. Samtidig som erfaringer er avhengig av tanker, er tanker avhengige av erfaringer. Uten disse erfaringene hadde vi heller ikke hatt refleksjoner. Denne gjensidige prosessen er elementær for læring (Dewey, 1996). For at elever skal lære programmering må de ha mulighet til å erfare og reflektere over hva som fungerte og hva som ikke fungerte.

Selvbestemmelsesteorien til Ryan og Deci (2009) beskriver ulike typer motivasjon. Vi fokuserer på delen som beskriver at det beste læringsresultatet skjer som følge av at læringen er indre motivert. For indre motivasjon må læringsaktiviteten i seg selv være engasjerende og fascinerende. For at elever skal være indre motivert må arbeidet gi glede og tilfredsstillelse i seg selv og ikke skje som følge av ros eller belønning (Ryan & Deci, 2009, henvist i Skaalvik & Skaalvik, 2015).

#### 2.1.2.1 PRIMM

På bakgrunn av tidligere forskning om misoppfatninger med programmering som presenteres senere i kapitlet, har vi valgt å utvikle læringsressursen med utgangspunkt i *PRIMM*. PRIMM er en undervisningsmetode utviklet for å støtte lærere i gjennomføringen av programmeringsundervisning for nybegynnere (Sentance & Wait, 2017). Undervisningsmetoden baserer seg på å gradvis bygge kompetanse hos elevene,

og det fokuseres på at elever må kunne lese og forstå hvordan programmer fungerer, før de starter å lage og utvikle egne programmer (Sentance et al., 2019).

Den undervisningsmetoden PRIMM struktureres i undervisningen gjennom de fem stegene: *Predict, Run, Investigate, Modify* og *Make* (Sentance & Waite, 2017) som vi har oversatt til *Forutse, Kjør, Undersøk, Endre* og *Lag*. PRIMM-metoden starter med at elever jobber sammen to og to og skal *forutse* en kode i et programmeringsprogram. Gruppene skal diskutere hvilke output man får om koden kjøres, og etter diskusjonen får elevene *kjøre* koden og programmet. Elevene får dermed sjekket om de har forutsett riktig output til koden, og læreren diskuterer og forklarer koden med klassen. Videre får elevene jobbe med å *undersøke* koden gjennom veiledende aktiviteter, hvor man skal utforske strukturer, forklare prinsipper og øve på å identifisere og løse feil i programmet. Målet med aktivitetene og undersøkningen er at elevene skal utvikle en større forståelse for programmet. I det neste steget av undervisningen skal elevene *endre* og forbedre funksjonene til et ferdig program. Ettersom elevene arbeider i et eksisterende program, kan elevene raskt oppleve mestring ved å enkelt endre funksjoner og få eierskap til programmet. Ytterligere vil det å jobbe i et eksisterende program hjelpe elever med å utvikle forståelse for grunnprinsippene i programmering på et nivå som er tilpasset læringsnivået til den enkelte. I den siste delen av PRIMM-undervisningen skal elevene selv *lage* ett nytt program med en ny funksjon, men som likevel har samme strukturer som programmene elevene har jobbet med tidligere (Sentance et al., 2019).

### 2.1.3 Algoritmisk tenking

En definisjon av algoritmisk tenking er "*tankeprosessen som er involvert i å formulere problemer og deres løsninger slik at løsningene blir representert i en form som effektivt kan forstås av andre*" (Wing et al., 2010). Weintrop et al. (2016) har utformet et rammeverk som hovedsakelig tar for seg om arbeid som er relatert til naturfag- og matematikkundervisning, og som er dekkende å bruke ved utforskning av naturfenomener. Det som er mest sentralt for vår oppgave fra rammeverket er at å lage nye, utvide, vurdere, designe og konstruere allerede eksisterende modeller er en sentral del av algoritmisk tenking og naturfaglig praksis (Weintrop et al., 2016).

## 2.2 Tidligere forskning

I dette kapitlet presenteres tidligere forskning. I studien er målet å utvikle en læringsressurs som kombinerer programmering og naturlig seleksjon. Hensikten er å både gi elever kompetanse innen programmering og naturlig seleksjon, uten at disse temaene går på bekostning av hverandre. Av den grunn tar vi for oss tidligere forskning som beskriver studier innenfor programmering i naturfag, misoppfatninger i programmering, evolusjon og naturlig seleksjon, og modeller av naturfaglige fenomener.

### 2.2.1 Programmering i naturfag

En studie utført av Lai og Lai (2012) undersøkte effektiviteten av å bruke blokkprogrammeringsprogrammet Scratch som en del av naturfagundervisningen i 5.klasse. For å måle effektiviteten av Scratch ble det gjennomført en logisk test, en problemløsningstest og en spørreundersøkelse for å evaluere læringsutbytte. Funnene i studien viser at elevene fikk forbedret problemløsning- og logiske evner gjennom å

jobbe med programmet Scratch, og at omtrent 60% av elevene foretrakk å programmere i Scratch for å lære om andre emner i naturfag. Studien konkluderer med at naturfagundervisning i Scratch var effektivt for 5.klassingene (Lai & Lai, 2012).

I studien til Caballero et al. (2014) ble det undersøkt hvordan modellering gjennom programmering kan brukes for å lære elever i 9.klasse uten programmeringserfaring om naturfaglige fenomener i fysikkundervisningen. Mens hovedfunnene i Lai og Lai (2012) sin studie beskriver hvordan programmering i Scratch er effektivt, viser hovedfunnene i studien til Caballero et al. (2014) at elever har utfordringer med programmering. Elevene fant det utfordrende å forstå om årsaken til feilen i programmet skyldes programmeringen eller den faglige forståelsen. Samtidig hadde elevene utfordringer med løse feil i programmeringen. Funn i fra studien viser at det er utfordrende å koble programmeringen tydelig nok opp mot det naturfaglige innholdet, og konkluderer med at programmering må prioriteres i undervisningen for at elevene skal få et faglig utbytte av å jobbe med programmering (Caballero et al., 2014).

En studie gjennomført av Bungum et al. (2014) undersøkte hvordan det å jobbe med kontekstualisert kunnskap i *teknologi og design* kan overføres til å bli mer generalisert og avkontekstualisert kunnskap i naturfag og matematikk. Hovedfunnet i studien viser at det er utfordrende å inkludere fagkunnskap i arbeid med teknologi og design, tross av at lærerne og forskergruppen anerkjente fagkunnskapen som relevant. Imidlertid påpeker forskergruppen at teknologiprojekter kan gi relevant utbytte for elever i matematikk og naturfag, ved å gi elevene erfaringer som kan bidra til at konseptuell kunnskap gir mening. Det krever at læreren må presentere den teoretiske kunnskapen og forklare hvordan denne forholder seg til teknologien som elevene har bygget (Bungum et al., 2014).

### 2.2.2 Misoppfatninger i programmering

Qian og Lehman (2017) viser til at kunnskap om programmering kan kategoriseres i tre typer: syntaks, konseptuell og strategisk kunnskap. Syntaks kunnskap handler om språkfunksjoner i programmering, som det å ha kunnskap om bruk av tegnsetting i tekstprogrammering. Konseptuell kunnskap handler om hvordan viktige programmeringsfunksjoner fungerer. Strategisk kunnskap handler om hvordan planlegge, lage og feilsøke et program for å løse et problem ved å bruke syntakskunnskap og konseptuell kunnskap (Qian & Lehman, 2017).

Utfordringen med elevers konseptuelle kunnskap kan føre til betydelige misoppfatninger relatert til elevers mentale modeller om programmeringsutførelse. Viktige programmeringskomponenter er variabler, vilkår og løkker. Komponentene kan være krevende for elever å forstå og mange misoppfatninger er relatert til disse komponentene (Qian & Lehman, 2017). En studie av Grover og Basu (2017) rapporterte at elever særlig har vansker med å bruke og forstå variabler og løkker. Dette gjelder for både tekst- og blokkprogrammering. Misoppfatninger i konseptuell kunnskap påvirker også strategisk kunnskap og evnen til å lage et fungerende program.

Qian og Lehman (2017) trekker frem noen strategier og verktøy for å adressere misoppfatninger hos elever. En strategi er å bruke gode programmer som eksempler slik at elevene kan undersøke og forklare. Ved at elever prøver å lese, forutsi og forstå programmer, kan læreren avdekke elevenes misoppfatninger. Dersom man bruker ferdig utarbeidet programmer, reduserer man elevenes kognitive belastning (Qian & Lehman,

2017). En annen strategi er å benytte blokkprogrammeringsprogrammer som Scratch, for å forhindre mulighetene for syntaksfeil og reduserer elevenes kognitive belastning (Qian & Lehman, 2017; Resnick et al., 2009).

### 2.2.3 Modeller av naturlige fenomener

En studie av Wang og Tseng (2018) beskriver at virtuelle simuleringer har en avgjørende betydning for å tilrettelegge for elevers kunnskapsutvikling om komplekse naturlige fenomener. En forklaring er at virtuelle simuleringer kan vise klare observasjoner av usynlige fenomener, og dermed hjelpe elevene med å forstå underliggende abstrakte konsepter og mekanikker om et naturlig fenomen (Weng & Tseng, 2016). Denne studien stemmer overens med lignende studier (f.eks. Jaakkola & Nurmi, 2008; Zacharias et al., 2008).

I studien til Malone et al. (2019) har det blitt utviklet en interaktiv simulering om naturlig seleksjon av firfirsler og tilhørende oppgaver. Studien undersøkte virkningen av arbeid med den interaktive simuleringen, gjennom å sammenligne en gruppe som gjennomførte deres interaktive simulering og en gruppe som gjennomførte tradisjonell undervisning. En før- og ettertest dokumenterte at gruppen som gjennomførte den interaktive simuleringen viste større nøyaktighet for konseptuell vurdering av naturlig seleksjon. Videre dokumenterte studien at gruppen som gjennomførte tradisjonell undervisning fortsatte å opprettholde tidligere misoppfatninger, mens gruppen som benyttet simuleringen fikk større mulighet til å revidere og konfrontere tidligere misoppfatninger ved å analysere datasimuleringen. Funnene fra studien indikerer at naturlig seleksjon effektivt kan undervises ved hjelp av interaktive simuleringer med tilhørende oppgaver, hvor elever får utforske og vurdere datasimuleringen selv (Malone et al., 2019).

### 2.2.4 Misoppfatninger i evolusjon og naturlig seleksjon

Grunnet en stor mengde litteratur om misoppfatninger i evolusjon, har vi valgt å ta utgangspunkt i Harms og Reiss (2019) sin bok, som presenterer statusen for dagens evolusjonsundervisning. Boken presenterer en oversikt over flere studier som undersøker vanlige misoppfatninger om evolusjon, og peker på en rekke faktorer som kan føre til misoppfatninger. Funn ifra studiene viser blant annet at elever sliter med å forstå underliggende temaer som populasjon, adaptasjon og frekvenser. Det kan av den grunn være en fordel å være forsiktig med bruk av begreper og gi en god beskrivelse til disse når de nevnes. Videre trekker studier frem at organisering av nivåer er utfordrende for elever. Av den grunn er det viktig å forklare elevene hva som er forskjellen mellom gener, individer, populasjoner og arter, og videre forsikre seg at elevene forstår hvilket nivå som menes. Ellers viser funn fra studien at elever ofte har misoppfatning om tidsrammen for evolusjon, og er ikke bevisst på at evolusjon er en prosess som går over mange generasjoner (Gregory, 2009; Harms & Reiss, 2019).

Mange av misforståelsene elever har om naturlig seleksjon, er knyttet til begrepet *fitness*. Reydon (2021) har samlet disse misforståelsene og utarbeidet en liste med de fem vanligste misforståelsene om begrepet.

1. *Fitness handler om fysisk styrke.*
2. *Gjennom naturlig seleksjon er det kun de mest egnede og dermed fitteste som overlever.*
3. *Organismer kan endre seg for å bedre passe inn i miljøet.*
4. *Fitness er en egenskap hos arter og ikke hos organismer.*
5. *Fitness handler om gener, og dette betyr at alleler som er dominante er mer egnet enn recessive gener.*

Denne listen bekreftes av funn fra studien til Gregory (2009).



### 3. Metode og metodeteori

Metodekapittelet beskriver metodene som er brukt for å nå målene med utviklingsarbeidet i masterstudien. Kapittelet starter med å presentere studiens vitenskapelige ståsted, før vi definerer vår tolkning av pedagogisk designforskning og beskriver valg av forskningsdeltakere og forskningssted. Videre beskriver vi metodene som er brukt for å samle erfaringsgrunnlag og hvordan vi har brukt det i utviklingen av læringsressursen. Kapittelet avsluttes med å se på kvalitet i forskningen og forskningsetikken knyttet til masterprosjektet.

#### 3.1 Vitenskapsteoretisk ståsted

Masterprosjektet baserer seg på en hermeneutisk tilnærming, som innebærer at forskningen preges av fortolkninger. Fortolkningene i en hermeneutisk tilnærming gjøres på bakgrunn av deltakernes forståelse og handlinger i erfaringsgrunnlaget, sett i sammenheng med tidligere forskning og teoretisk grunnlag. Tolkningene må gjøres i en helhetsforståelse, som innebærer at forskeren må vurdere forskningsdeltakernes forståelse i lys av konteksten (Grønmo, 2016). I vårt masterprosjekt tolkes observasjoner og samtaler av gjennomføringen av opplegget. Dette gir forskeren en helhetforståelse og utgjør viktige erfaringer med opplegget for å analysere, evaluere og videreutvikle læringsressursen i masterprosjektet.

#### 3.2 Pedagogisk designforskning

Ifølge Bjørndal (2013) er pedagogisk designforskning en fellesbetegnelse for designstudier, utviklingsforskning og designeksperimenter. Denne typen forskning innebærer systematiske undersøkelser, som er knyttet til prosesser rundt utvikling, utprøving og evaluering av en egendesignet læringsressurs eller utdanningsprogram. Målsetningen for pedagogisk designforskning er å optimalisere undervisning og læring gjennom pedagogiske tiltak. Det er fire kjennetegn på pedagogisk designforskning, disse er (1) intervensjon, (2) at det er en syklisk prosess av design, evaluering og revisjon. Designforskning er (3) prosessorientert, som vil si at forskerblikket rettes mot forståelser, forbedring og intervensjon. Designforskning er (4) nytteorientert som innebærer at designet er basert på et teoriforslag og at testingen vil kunne bidra til å bygge denne teorien (Van den Akker et al., 2006; Bjørndal, 2013).

I pedagogisk designforskning forsøker forskeren å forbedre et design og teorien den bygger på, ved å se på detaljer i den praktiske situasjonen. På denne måten bidrar pedagogisk forskningsdesign i arbeidet med å utvikle nye teorier som kan testes ut i praksis. Målet med pedagogisk designforskning er dermed tosidig. Forskingen skal både utvikle pedagogisk praksis og være et vitenskapelig bidrag til pedagogisk teori (Bjørndal, 2013). Etersom masterprosjekt er begrenset til et semester er vårt fokus hovedsakelig på å utvikle en læringsressurs.

For å sikre at vår pedagogiske designforskning er nytteorientert, har vi gjennomført en pilotundersøkelse under vår siste praksisperiode på en skole i Trøndelag. Vi tar utgangspunkt i funnene fra piloten i forberedelsene av designeksperimentet. Samtidig vil læringsressursen testes ut i skolen og dette er med på å understreke at designforskningen er nytteorientert.



Designeksperimenter har tre faser; forberedelse, gjennomføring og retrospektive analyser. I et tradisjonelt og langvarig pedagogisk forskningsdesign vil alle fasene gjennomgå flere sykluser (Gravmeijer & Cobb, 2003; Bjørndal, 2013). Vårt pedagogiske forskningsdesign har et mindre omfang og derfor vil det bare være gjennomføringsfasen som gjennomgår to sykluser (se figur 1, s.12).

### 3.2.1 Forberedelse av designeksperimentet

I den første fasen av designeksperimentet utvikles læringsressursen med utgangspunkt i en hypotetisk lokal utviklingsteori (Bjørndal, 2013). Vår teori er at *programmering kan brukes for å utforske naturlige fenomener*. Bjørndal (2013) beskriver utviklingsfasen som å utvikle en læringsressurs med utgangspunkt i en lokal utviklingsteori. En teori er lokal hvis den refererer til et emne som bygger på antagelser om at en mulig prosess fører til læring og utvikling hos elever og hvilke læringsmidler som kan støtte og organisere elevenes læring. Forberedelse av design eksperimentet innebærer videre å gjøre seg kjent med tidligere relevant forskning (Bjørndal, 2013). Tidligere relevant forskning på disse emnene er presentert i teoridelen.

Vår læringsressurs ble kvalitetssikret av en fagperson innen blokkprogrammeringsundervisning for å sikre at læringsressursen er tilpasset skolen og dermed er nytteorientert. Forskningsdeltakeren fikk tilsendt simuleringen, oppgaveheftene og lærerveiledningen på forhånd og senere møttes vi til en samtale for å diskutere mulige endringer. Deltakerens innspill ble tatt i betraktning i forberedelsene av vår læringsressurs.

### 3.2.2 Gjennomføring av designeksperimentet

Den andre fasen i designeksperiment er gjennomføring. Målet med denne fasen er å teste ut den utviklede læringsressursen, for å kunne videreutvikle denne (Bjørndal, 2013). Læringsressursen ble gjennomført i to sykluser i to ulike klasser. Syklusene ble gjennomført to etterfølgende dager og derfor var det begrenset med tid til å analysere erfaringsgrunnlaget og gjøre endringer mellom syklusene. Erfaringsgrunnlaget som videreutviklingen baserer seg på er observasjon. Vi observerte gjennomføringen av den utviklede læringsressursen og skrev ned observasjoner om hvordan elevene jobbet med oppgavene. Vi har valgt å observere ved strukturert observasjon. En strukturert observasjon innebærer at observatøren har et bestemt fokus i observasjonen (Dalland et al., 2021). Fokuset i observasjonen var på gjennomføring, vanskelighetsnivået og aktivitet i oppgavejobbingen og engasjementet til elevene gjennom undervisningen. Vi fulgte anbefalingen til Bjørndal (2013) og var deltakende observatører i klasserommet under gjennomføringen av undervisningen. En deltakende observatør er tilstede under observasjonen og observerer deltakerne i deres miljø. Ved å være en deltakende observatør får man nærhet til deltakerne og datamaterialet (Ringdal, 2013). Dette ga mulighet til å hjelpe med utfordringer i gjennomføringen, notere observasjoner, lytte til gruppediskusjoner og avholde korte møter med læreren umiddelbart etter undervisningen for å utvikle felles fortolkninger av hva som måtte endres til neste gjennomføring. Det å ha korte samtaler mener Gravemeijer og Cobb (2006) er svært hensiktsmessig for pedagogisk designforskning. Disse samtalene ligger til grunn for endringene vi gjorde mellom første og andre syklus i designeksperimentet. Ettersom dette er en studie med begrenset tid, vil det dermed ikke være mulig å utføre flere

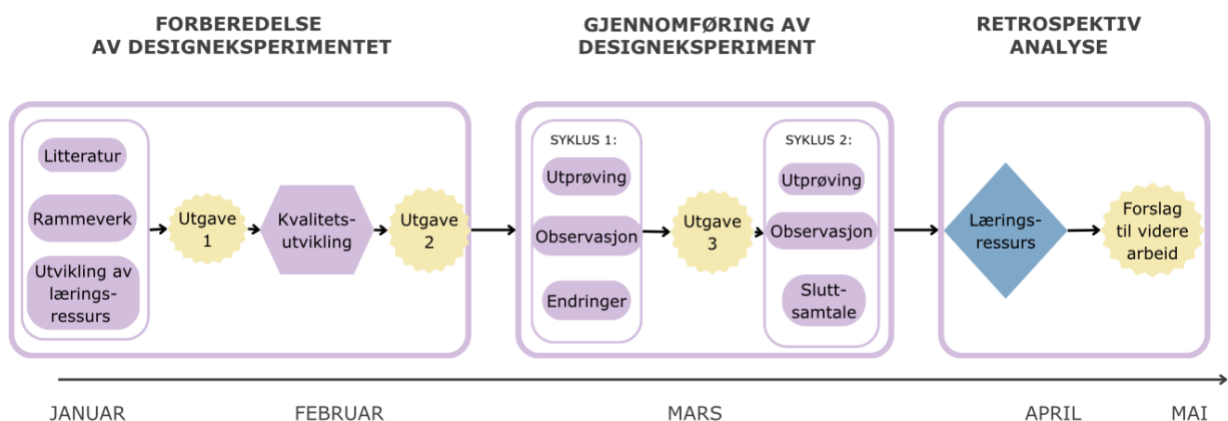
sykluser av designeksperimentet. En oversikt over syklusene i designeksperimentet finnes i figur 1.

I tillegg til observasjonene og korte samtaler med læreren mellom syklusene, hadde vi en sluttsamtale med læreren etter utprøvingene for å skape en felles fortolkning av hva som fungerte og hva som behøvdtes å endres. Samtalen med læreren baserer seg på observasjoner, erfaringer og lærerens forståelse fra gjennomføringen. Ved å diskutere alle parters verdier og forståelser fikk vi ny innsikt og nye tiltak for å utvikle læringsressursen.

### 3.2.3 Retrospektive analyser

I fasen retrospektiv analyse er formålet å videreutvikle læringsressursen (Bjørndal, 2013). Den retrospektive analysen baserer seg på observasjonene og sluttsamtalen med læreren. Under gjennomføringen og i samtalene med læreren fikk vi nye erfaringer med opplegget, og dette ga grunnlaget for videreutviklingen.

I Bjørndals (2013) doktorgradsprosjekt ble designeksperimentet gjennomført i tre sykluser som besto av utprøving, samtale, observasjon og drøfting. Vårt designeksperiment består av to sykluser. Endringene etter syklus 1 er basert på observasjoner og korte samtaler med lærer, mens endringene etter syklus 2 er basert på flere observasjoner og en mer omfattende sluttsamtale med lærer. I figur 1 visualiseres vår tilpassing til Bjørndals (2013) modell for pedagogisk forskningsdesign. Modellen viser vår utviklingsprosess og inkluderer fasene forberedelse, gjennomføring og den retrospektive analysen. I tillegg illustrerer modellen utviklingen av læringsressursen fra utgave 1 til læringsressursen. I modellen inkluderes det en tidslinje for å vise tidsbruken og rekkefølgen for de ulike fasene i designforskningen.



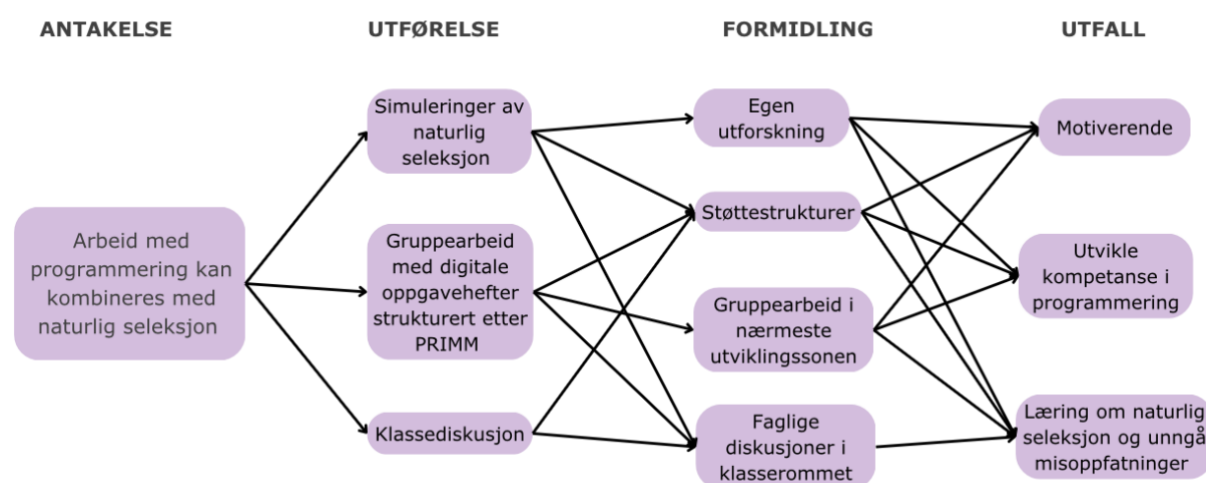
Figur 1: Tilpassing til Bjørndals (2013) modell av pedagogisk forskningsdesign, som viser oversikt over prosessene for utviklingen av vår læringsressurs. Utgavene er læringsressursen før vi gjør nye endringer grunnet nytt erfaringsgrunnlag.

### 3.3 Antagelseskart for pedagogisk designforskning

Som en støtte i vår pedagogiske designforskning har vi tatt utgangspunkt i Sandoval (2014) sitt generaliserte antagelseskart for pedagogisk designforskning. Kartleggingen av antagelser gir fokus på elementer i designet og deres funksjoner i å fremme læring. Denne prosessen hjelper forskere med å tilskrive feil til enten designet eller de

underliggende teoretiske begrunnelsene. Det er en teknikk som kan benyttes for å beskrive og utføre pedagogisk designforskning (Sandoval, 2014).

Vi har tatt utgangspunkt i Sandoval (2014) sitt antakelseskart for å kartlegge våre forventninger til studien (figur 2). Vi hadde en antakelse om at *arbeid med programmering kan kombineres med naturlig seleksjon*. Dette har vært utgangspunktet i utviklingen av læringsressursen. For å imøtekomme denne antagelsen, ønsket vi å designe en læringsressurs bestående av to simuleringer som elevene kan utforske om naturlig seleksjon, to oppgavehefter som er utviklet etter undervisningsmetoden PRIMM. Ytterligere ønsket vi at arbeidet med læringsressursen skal legge opp til gruppe- og klassediskusjoner. Vi antok at arbeid med læringsressursen tilrettelegger for utvikling av forståelse for naturlig seleksjon. I utviklingen av læringsressursen har vi hatt kvalitetsrådgivning med en ekspert innen blokkprogrammeringsundervisning og støtte av veileder med god kompetanse innen evolusjon. Dette har hjulpet oss i arbeidet med å utvikle læringsressursen som kombinerer arbeid med programmering og naturlig seleksjon. Gjennom arbeid med læringsressursen formidles kunnskap via egen utforsking av simuleringer, gruppearbeid i nærmeste utviklingssone, faglige diskusjoner og støttestrukturer for å veilede elevene. Etter arbeid med læringsressursen har vi forventet at utfallet er at læringsressursen er motiverende, utvikler kompetanse i programmering, gir læring om naturlig seleksjon, samt å forhindre og avklare tidligere misoppfatninger.



Figur 2: Vår tilpasning til Sandoval (2014) sitt antakelseskart for pedagogisk designforskning.

### 3.4 Forskningssted og forskningsdeltakere

Forskningen har funnet sted i to niendeklasser på en skole i Trøndelag. Valget av skole og lærer ble gjort tilfeldig ved at en naturfagslærer meldte interesse for prosjektet. Naturfagslæreren hadde lite tidligere erfaring med programmering, men var positivt innstilt til å arbeide med programmering i naturfagundervisning. Hensikten med lærerens deltakelse i masterprosjektet var å utvikle egen programmeringskompetanse. Samtidig ønsket læreren støtte for å nå kompetansemålet i naturfag etter 10.trinn om å *bruke programmering til å utforske naturfaglige fenomener* (Kunnskapsdepartementet, 2019).

Klassene besto av 26 elever med generelt lite erfaring med programmering. Temaet evolusjon var gjennomgått tidligere i skoleåret. Vår pilot og forskningsdeltakerne hadde

sammenlignbare tendenser når det kom til kompetanse og erfaringer hos lærerne. Vi antar dermed og opplevde at forskningsdeltakerne og forskningsstedet har et representativt nivå for koding i skolen.

### 3.5 Kvalitet i forskningen

I dette kapittelet diskuteres kvaliteten i masterstudien. Vi vurderer validiteten og reliabiliteten til studien. Som følge av at studien baserer seg på pedagogisk designforskning vil diskusjonen om kvaliteten i forskningen få et annet fokus enn i en empiribasert master.

Validitet viser i hvilken grad datagrunnlaget er relevant for målet for forskningen (Grønmo, 2016). For vårt masterprosjekt vil datagrunnlaget tilsvare erfaringsgrunnlaget. Det er flere faktorer som påvirker validiteten til masterprosjektet. En faktor som kan styrke validiteten i masterprosjektet er samtale med læreren. Gjennom samtaler har vi og læreren jobbet for å få en felles fortolkning og oppfatning av gjennomføringen som styrker validiteten. På en annen side kan samtale svekke validiteten ved at forskningsdeltakeren deler sine meninger om gjennomføringen og læringsressursen til forskerne som har utviklet læringsressursen. I tillegg til samtale, har vi samlet inn erfaringsgrunnlag til masterprosjektet gjennom observasjon. Selv om man ønsker å være objektiv vil observasjon alltid være preget av egen bakgrunn, erfaringer og kunnskap (Dalland et al., 2021). Subjektivitet gjelder også i masterprosjektet vårt og kan være med på å svekke validiteten. Det er vi som forskere som har utviklet læringsressursen, og dette kan medføre at våre observasjoner blir påvirket slik at vi potensielt overser eventuelle aspekter ved læringsressursen. Det er derfor viktig å være bevisst på at egne tanker og meninger vil prege observasjonene og påvirke tolkningene våre i prosjektet. For å forhindre at vi som forskere overser sider ved opplegget har vi i etterkant av observasjonen hatt en sluttsamtale med læreren som gjennomførte undervisningen. På denne måten vil lærerens erfaringer og refleksjoner kunne bidra til å motvirke dette.

Reliabilitet viser påliteligheten til datamaterialet. Reliabiliteten er graden av samsvar mellom ulike datainnsamlinger som er basert på samme innsamlingsmetode (Grønmo, 2016). I kvalitative studier blir datamaterialet preget av forskeren, ved at datainnsamlingen blir utviklet underveis i prosessen med utgangspunkt i forskerens egne fortolkninger av situasjonen (Grønmo, 2016). Dette er tilfellet i masteroppgaven vår. For å styrke reliabiliteten i masterprosjektet, har vi valgt å bruke både observasjon og samtale for å samle inn erfaringsgrunnlag. I forskningen brukes kombinasjonen av metoder for å få et mer utfyllende erfaringsgrunnlag, som inkluderer en beskrivelse av situasjonene og forskningsdeltakernes refleksjoner rundt dette (Dalland et al., 2021). Derfor har vi valgt å gjennomføre en sluttsamtale med læreren i kombinasjon med observasjonene.

### 3.6 Forskningsetikk

Forskningsetikk handler om verdier, normer og retningslinjer som påvirker den vitenskapelige praksisen. Ansvar for å sikre god etikk i forskningen har *Den nasjonale forskningsetiske komite for samfunnsvitenskap og humaniora, juss og teologi* (NESH) (Ringdal, 2013). For å sikre at det blir tatt gode etiske valg og ansvarlig forskning i masterprosjektet tar vi utgangspunkt i NESH-retningslinjene. I tillegg har vi søkt til

Norsk senter for forskningsdata AS (Sikt) og fått godkjent studien, for å sikre hensyn til forskningsdeltakere og etablere ansvarlig bruk av personopplysninger.

I vår masteroppgave har vi begrenset den personlige informasjonen som innsamles, men vi anser det som nødvendig å ta lydopptak av samtalen med læreren. Lydopptaket av samtalen vil regnes som personopplysninger, og som følge av dette har vi søkt til Sikt og fått samtykke til å gjennomføre forskningen.

Av hensyn til forskningsdeltakere, stilles det krav til at forskningen skal gjøres konfidensiell og at personopplysninger skal lagres forsvarlig (Ringdal, 2013). For å sikre dette i vårt forskningsprosjekt har vi brukt den nettbaserte skjema-løsningen nettskjema.no som er utviklet og driftet av Universitetet i Oslo ([nettskjema@usit.uio.no](mailto:nettskjema@usit.uio.no)), for å ta opp og lagre lydopptak.

I retningslinjene til NESH, står det beskrevet at forskningsdeltakere skal informeres om prosjektet og gi samtykke til deltakelse før datainnsamling. Det skal opplyses om at det er frivillig å delta, samt at deltakeren underveis i hele prosessen kan trekke sin deltakelse (NESH, 2021; Ringdal, 2013). For å sikre dette vil læreren som tilbys å delta i prosjektet, få tilsendt informasjon om dette via et informasjonsskriv og samtykkeskjema. Elevene og foresatte i klassen hvor undervisningen gjennomføres, vil bli informert på forhånd og har mulighet til å velge å ikke være en del av utprøvingen. Ved å ta disse type hensyn er vi med å på å sikre at vår forskning er etisk gjennomført.

## 4. Presentasjon av læringsressurs

I dette kapitlet presenteres læringsressursen som er utviklet i løpet av masterprosjektet. Læringsressursen tar for seg temaet naturlig seleksjon, og består av to ulike simuleringer som illustrerer seleksjonspress, to nivådelte oppgavehefter og en lærerveiledning. De ulike delene av læringsressursen ligger både tilgjengelig på lenke under delkapitlene og vedlagt som vedlegg.

### 4.1 Målet med læringsressursen

Læringsressursen kobles til følgende kompetansemål, hentet fra Læreplanen (LK20) i naturfag (Kunnskapsdepartementet, 2019):

Etter 7. trinn:

- *Bruke og vurdere modeller som representerer fenomener man ikke kan observere direkte, og gjøre rede for hvorfor det brukes modeller i naturfag.*
- *Utforske, lage og programmere teknologiske systemer som består av deler som virker sammen.*
- *Utforske og beskrive ulike næringsnett og bruke dette til å diskutere samspill i naturen.*

Etter 10. trinn:

- *Bruke og lage modeller for å forutsi eller beskrive naturlige prosesser og systemer og gjøre rede for modellenes styrker og begrensninger.*
- *Bruke programmering til å utforske naturfaglige fenomener.*
- *Beskrive hvordan forskere har kommet fram til evolusjonsteorien og bruke denne til å forklare utvikling av biologisk mangfold.*

For å tilpasse kompetansemålene til læringsressursen og for å tilgjengeliggjøre målene for elevene i undervisningen har vi utarbeidet læringsmål. Gjennom arbeid med læringsressursen er læringsmålene at elevene skal:

1. Bruke en simulering til å utforske samspill i naturen
2. Kunne bruke programmering til å gjøre endringer i en simulering.
3. Vurdere styrker og svakheter i en modell.

Læringsmål 1 er utviklet på bakgrunn av kompetansemål i naturfag etter 7.trinn om økologi og etter 10.trinn om evolusjon, og av den grunn har vi utviklet et felles læringsmål som omfatter begge kompetansemålene. Arbeid med modeller og programmering inkluderes både i kompetansemål etter 7. trinn og 10.trinn, og dermed kan alle læringsmålene for læringsressursen brukes både på mellomtrinnet og ungdomsskolen.

I tillegg til valgte kompetanse- og læringsmål ønsket vi at læringsressursen kan bidra med å avklare og forhindre misoppfatninger i naturlig seleksjon. Med dette som mål har vi designet simuleringen for å visualisere tydelig deler av naturlig seleksjon som forskning viser at elever ofte har misoppfatninger om. Misoppfatningene vi har fokusert på er at elever sliter med å forstå at evolusjon er en prosess over tid, elever tror at fitness handler om å være den sterkeste og at de tror at gener til et individ kan endre seg til å bli bedre tilpasset miljøet (Harms & Reiss, 2019).



## 4.2 Simuleringer

Vi har designet to simuleringer i programmeringsspråket Scratch som begge illustrerer naturlig seleksjon og seleksjonspress. For å se og bruke simuleringen må du trykke på lenken. Når du åpner lenken til simuleringen kommer du inn til prosjektsiden, hvor man kan lese om hvordan man bruker prosjektet. For å se simuleringen trykk: på *grønt flagg*, og for å gjøre endringer i kodene trykk: *se inni*.

Simuleringen *tilpasninger og overlevelse* viser hovedsakelig hva det vil si å være godt tilpasset et miljø. Her sammenligner vi en liten og en stor mus i flukt fra en katt. I arbeidet med simuleringen bør læreren legge opp til diskusjon om hvilken mus som er best tilpasset miljøet. Programmeringsnivået i simuleringen er enkelt slik at elevene kan utforske programmeringen og muligheter i Scratch. For å gjøre utforskningen av programmeringen enklere har vi hentet frem nyttige klosser for videreutvikling av programmet.

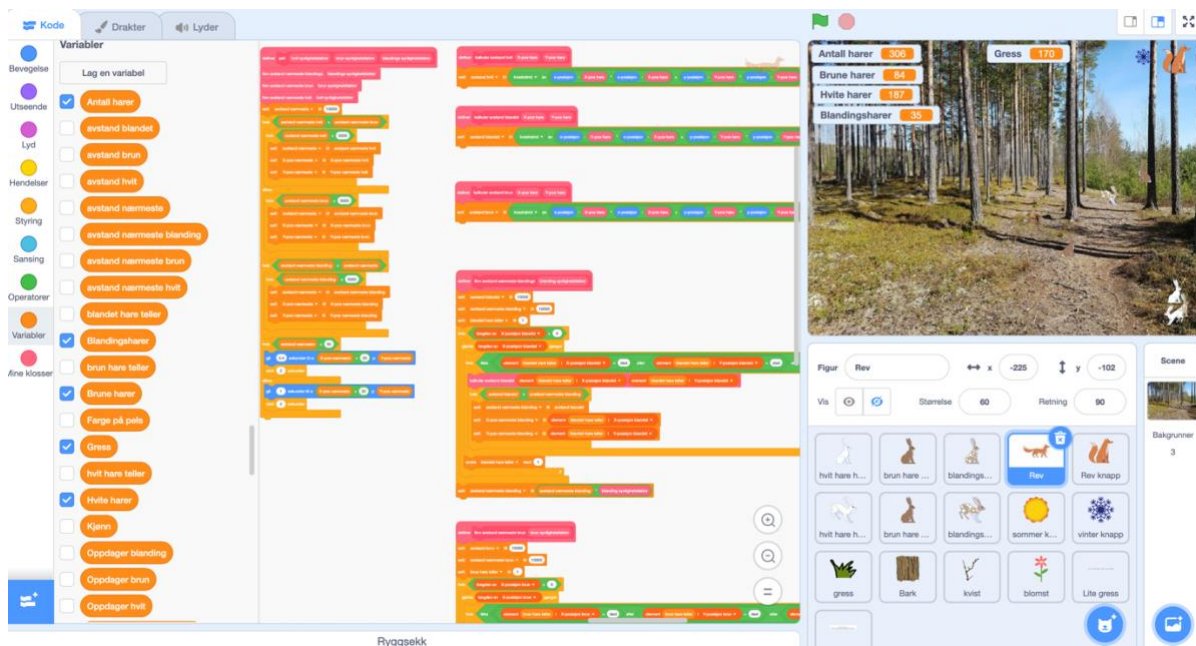
### [Simuleringen Tilpasninger og overlevelse](#)

Simuleringen *Naturlig seleksjon* er en komplett simulering av naturlig seleksjon hos harer. Denne simuleringen forenkler og illustrerer hvordan individer som er best tilpasset miljøet overlever og dermed reproducerer seg. Simuleringen inkluderer reproduksjon, arv og seleksjonspress som rovdyr og mattilgang ved ulike årstider. I simuleringen kan elevene gjøre små endringer på arv, reproduksjon og seleksjonspress som visualiseres i simuleringen. Nivået på programmeringen er krevende, men elevene skal hovedsakelig endre verdier i programmet.

### [Simuleringen naturlig seleksjon](#)



Figur 3a og b: Skjermbilder av simuleringen naturlig seleksjon ved ulike miljøer.



Figur 4: Utklipp fra programmeringen i simuleringen.

### 4.3 Oppgavehefter

Vi har utviklet to digitale oppgavehefter, et til hver simulering. Oppgaveheftet *Tilpasninger og overlevelse* har et enkelt nivå både programmeringsteknisk og faglig, mens oppgaveheftet *Naturlig seleksjon* har et høyere nivå.

[Oppgavehefte Tilpasninger og overlevelse](#)

[Oppgavehefte Naturlig seleksjon](#)

Oppgaveheftene er designet etter PRIMM tilnærmingen og følger de fem stegene *predict– run– investigate– modify– make* (Sentance et al., 2019). I tillegg inkluderer heftene en oppgave hvor elevene skal *vurdere* simuleringen, etter å ha utforsket den. Oppgavene kombinerer kunnskap om naturlig seleksjon med kunnskap og trening i programmering.

I arbeid med oppgaveheftene skal elevene jobbe i grupper på to personer. Som støtte og veiledning til elevenes arbeid inkluderer heftet en del med tips og veiledning til programmeringen. Tipsene inkluderer hvilke blokker elevene skal bruke, hvor i programmet elevene skal endre verdier og ferdige program. Dette er inkludert i oppgaveheftet for å hindre at selve programmeringen stopper elevene fra å utvikle kompetanse om naturlig seleksjon.



### **Overlevelse**

*For at individer skal ha mulighet til å reprodusere seg må de overleve til de blir voksne. Dette innebærer å skaffe og spise nok mat og beskytte seg mot farer slik som rovdyr. Harene i simuleringen må både bevege seg mye for å spise nok gress, samtidig som de må gjemme seg mot rovdyr. Derfor hjelper det at fargene på pelsen til harene ligner omgivelsene, dette kalles kamuflasje. Dette er en egenskap som er godt tilpasset miljøet og som er en fordel for individer å arve.*

#### **D. Hvilken pelsfarge tror du reven ser best på avstand når det er sommer?**

Gå inn i programmet og finn ut hvor reven jakter på de ulike harene og hva som er forskjellen for at reven oppdager hvite, brune og blandingsharer.

Hva vil skje om årstidene endres og bakken er dekket av snø?

Figur 5: Eksempel på oppgave fra oppgaveheftet *Naturlig seleksjon*.

#### **4.4 Lærerveiledning**

For å veilede og støtte læreren i arbeidet med simuleringene og oppgaveheftene har vi utviklet en tilhørende lærerveiledning. Lærerveiledningen inneholder mål for undervisningen, hvordan gjennomføre undervisningen, beskrivelse av oppgaveheftet og tips til hvordan tilpasse undervisningen.

##### Lærerveiledning

Lærerveiledningen inkluderer også en omfattende beskrivelse av simuleringen *Naturlig seleksjon*, ettersom programmet til simuleringen inneholder flere kompliserte koder. En viktig del av undervisningen er at elevene skal gjøre endringer i programmet, derfor er det essensielt at læreren forstår kodene i programmet. Videre har vi laget en forklaring til alle variabler som benyttes, hva ulike koder betyr og hvordan kodene skal illustrere fenomenet naturlig seleksjon. Vi har valgt bort å beskrive programmeringen i simuleringen *Tilpasninger og overlevelse*, ettersom dette programmet kun består av enkle koder.

<pre> definer reproduksjon hvit og brun si i 2 sekunder vent 5 sekunder Hoppe gjenta tilfeldig tall fra 0 til 3 ganger sett Farge på pels til tilfeldig tall fra 1 til 6 hvis Farge på pels = 1 eller Farge på pels = 2 eller Farge på pels = 3   Avkom er brun hare   vent 0.1 sekunder   Farge på pels = 4 eller Farge på pels = 5   Avkom er hvit hare   vent 0.1 sekunder   Farge på pels = 6   Avkom er blandet hare   vent 0.1 sekunder </pre>	<p><b>HARE:</b> Denne funksjonen viser sannsynligheten for hvilken pelsfarge avkommet til to harer med ulik pelsfarge vil få.</p> <p>Her møtes en hvit og en brun hare da vil sannsynligheten for at den blir brun være <math>3/6</math>, for at den blir hvit være <math>2/6</math> Og sannsynligheten for blandet vil være <math>1/6</math>.</p> <p>Det er variabelen tilfeldig tall som velger dette og den velger et tilfeldig tall fra 1-6.</p> <p>Dette viser arv av gener fra foreldrene. I dette tilfellet har foreldrene forskjellig pelsfarge, derfor vil avkommene arve ulike egenskaper fra foreldrene sine.</p>
--	--

Figur 6: Eksempel hentet fra lærerveiledningen. I venstre kolonne er et utklipp av en kodesekvens i Scratch. I høyre kolonne beskrives det hva kodene gjør og skal illustrere om naturlig seleksjon.

Ved å beskrive i detalj hva alle kodene betyr, er målet at lærerveiledningen skal være brukervennlig for naturfagslærere som har litt programmeringsbakgrunn fra tidligere. I tillegg inneholder lærerveiledningen en kort beskrivelse av evolusjon og en innføring i programmeringsspråket Scratch for å bidra til støtte og kompetanse for læreren.

## 5. Resultat og diskusjon

I kapittelet resultat og diskusjon drøftes det om vi har nådd målet med studien gjennom å drøfte hvert de to delmålene, før vi beskriver studiens begrensninger.

### 5.1 Utvikle en læringsressurs som kombinerer arbeid med programmering og naturlig seleksjon

I dette delkapittelet i masteroppgaven, vil vi fokusere på det første delmålet til studien. Fokuset vil være på prosessen og utviklingen av læringsressursen som kombinerer arbeid med programmering og naturlig seleksjon. Dette kapittelet vil beskrive valg og begrunnelser for valgene i utviklingen, erfaringer fra utprøving og endringer som er gjort på bakgrunn av erfaringer og teori. For å illustrere utviklingsprosessen har vi laget figur 1 (se kapittel 3, s.10). Figur 1 beskriver fasene i utviklingen av den ferdige læringsressursen, og vi tar utgangspunkt i fasene for å vurdere om vi har nådd målet om å utvikle en læringsressurs som kombinerer arbeid med programmering og naturlig seleksjon.

Delkapittelet er delt inn etter Bjørndal (2013) sine tre faser for et designeksperiment: utviklingsfasen, utprøvingsfasen og den retrospektive analysen. Hver av de tre delene i resultat- og diskusjonskapittelet er videre delt inn i læringsressursen og lærerveiledningen for å gjøre utviklingsprosessen mer oversiktlig. Avslutningsvis i kapittelet vurderer vi om delmålet er nådd.

#### 5.1.1 Utviklingsfasen

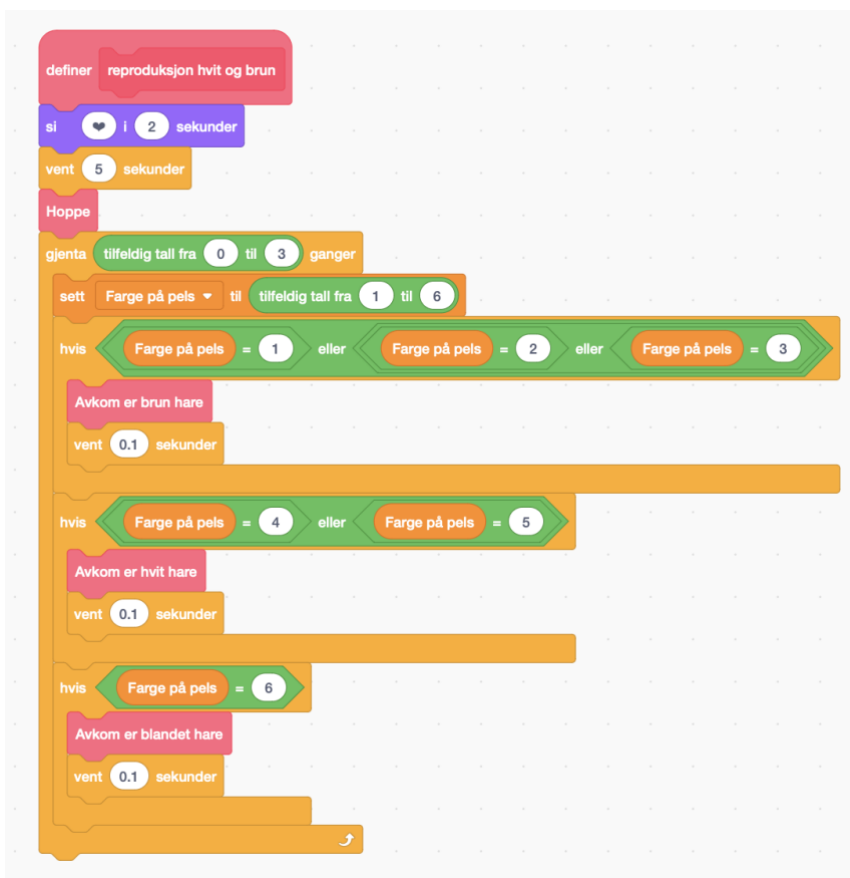
I utviklingsfasen beskrives prosessen, valgene og utviklingen av den første utgaven av læringsressursen. Utviklingen er gjort i lys av relevant forskning om undervisning i seleksjon og programmering. Det er tatt mange valg i utviklingsfasen og valgene samt begrunnelsene for valgene er beskrevet under.

##### 5.1.1.1 Simulering

Vi har valgt å utvikle to simuleringer som visualiserer og forenkler naturlig seleksjon for elever. På denne måten blir simuleringene en modell av naturlig seleksjon. Valget om å utvikle to simuleringer er gjort på bakgrunn av teori som sier at modeller kan forenkle og beskrive prosesser som er utfordrende å observere (Gilbert et al., 2000), og brukes som et verktøy for å forklare naturlige fenomener (Osborne, 2014). Samtidig viser en studie at virtuelle simuleringer kan ha avgjørende betydning for å tilrettelegge for elevers kunnskapsutvikling om komplekse naturlige fenomener (Wang & Tseng, 2018). Ytterligere beskriver tidligere forskning at naturlig seleksjon er en prosess som er vanskelig for elever å se for seg (Harms & Reiss, 2019). Dette støtter studien til Malone et al. (2019) som beskriver at naturlig seleksjon effektivt kan undervises ved hjelp av interaktive simuleringer, og at simuleringer kan gi elevene større muligheter til å revidere tidligere misoppfatninger om naturlig seleksjon (Malone et al., 2019). Av den grunn har vi ansett det å bruke en interaktiv simulering som en del av læringsressursen som hensiktsmessig for å forenkle og forklare naturlig seleksjon.

Vi startet med å utvikle simuleringen *Naturlig seleksjon*, før vi senere utviklet simuleringen *Tilpasninger og overlevelse*. I utviklingen av simuleringen *Naturlig seleksjon* har vi tatt mange valg. Vi har valgt å lage en simulering med hare som

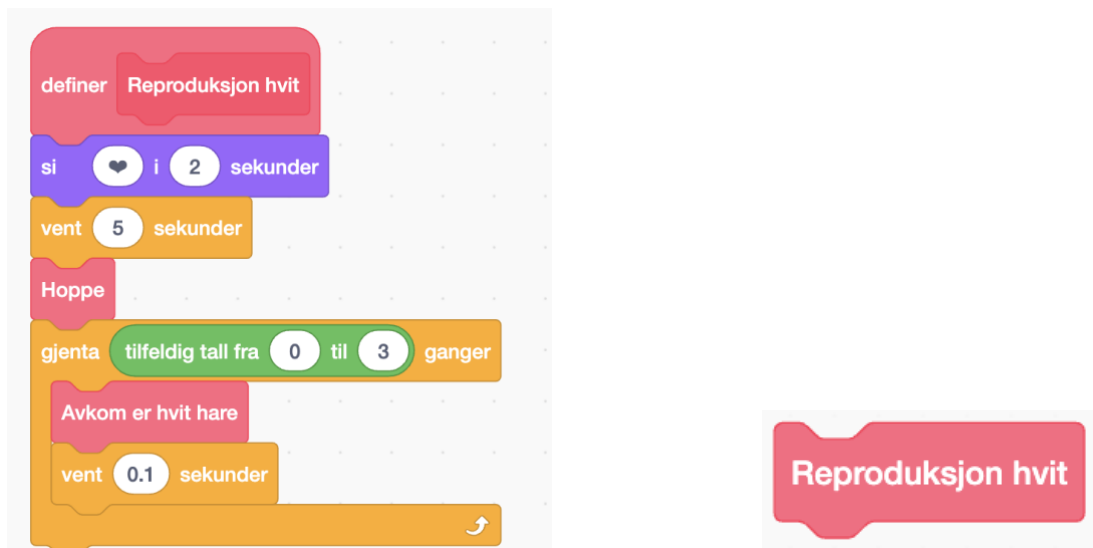
byttedyr og rev som rovdyr, ettersom dette er arter som er kjent for elevene. Simuleringen illustrerer blant annet hvordan harer formerer seg, og vi har tatt flere valg for å få formeringen i programmet til å bli realistisk. Vi har programmert inn at harene enten er hann- eller hunnkjønn. Det er programmert inn at harene må møte et individ av motsatt kjønn for å kunne reprodusere seg. Når to harer parrer seg vil haren klones et tilfeldig antall ganger mellom 0-3 for at harene i programmet skal få samme antall avkom som harer får i naturen (Frafjord, 2021). Det er også programmert inn hvilken pelsfarge avkommet får. Om en hunnhare og en hannhare med lik pelsfarge pares vil avkommets pelsfarge være lik foreldrenes. Men dersom to harer med ulik pelsfarge parres vil det være ulik sannsynlighet for de tre forskjellige pelsfargene. Dette skal illustrere grad av dominans i gener og at ulike genvarianter blandes ved reproduksjon og dermed gir ulik pelsfarge hos avkommene (se figur 7).



Figur 7: I denne kodesekvensen parres en hvit og en brun hare. Variabelen «farge på pels» velger tilfeldig et tall mellom 1-6. Om tallet er 1,2 eller 3 er avkommet brunt, ved 4 eller 5 er avkommet hvitt, og om tallet er 6 er pelsfargen til avkommet blandet.

I simuleringen har vi inkludert to seleksjonspress, disse er rovdyr og mattilgang. Reven er med på å illustrere hvordan individer som er bedre tilpasset oftere vil overleve og dermed reprodusere seg. Dersom bakken er brun, vil de brune harene ha god kamufasje ved å ligne omgivelsene, og det vil være mindre sjanse for at reven oppdager disse harene enn de hvite- og blandingsharene som i større grad vil stikke seg frem (Stenseth, 2020). For å illustrere dette er reven programmert til å oppdage de ulike harene på ulik avstand i forhold til miljøet. Gresset er et annet seleksjonspress, og skal illustrere mattilgang for harene. Dersom harene ikke har tilgang til mat vil den sulte, og med dette ønsker vi at elevene skal utvikle forståelse for at det er flere faktorer som skaper seleksjonspress.

Programmeringsteknisk har vi valgt å lage funksjoner for blant annet hopping, mattilgang, reproduksjon og hvordan reven skal jakte. Figur 8a er et eksempel på en slik funksjon. Ved å bruke flere funksjoner i programmet vil selve programmeringen være ryddigere og enklere for eleven å forstå. Istedenfor at det står en lang kode hver gang et hvitt hunnkjønn møter et hvitt hannkjønn brukes funksjonen *reproduksjon hvit* (se figur 8b). Ved å dele opp programmet i funksjoner kan hver funksjon kjøres alene, og dette gir elevene muligheter til å utforske programmeringskomponenter og se hvordan akkurat denne funksjonen fungerer. Dette er et valg tatt på bakgrunn av teori som sier at elever synes det er krevende å forstå løkker og variabler i blokkprogrammering (Grover & Basu, 2017).



Figur 8a og b: Funksjonen for reproduksjon for to hvite harer. Inne i denne funksjonen benyttes det en løkke. Ved å undersøke denne funksjonen kan det være mer forståelig hvordan og hvorfor en løkke brukes. I dette eksempelet brukes løkker for å illustrere hvor mange avkom som fødes. Figuren viser også at vi bruker begrepet «avkom er hvit hare» istedenfor «lag kone av meg».

En utfordring med kodingen i simuleringen er at for å reprodusere harene, må man i programmet lage en klon av haren man ønsker å reprodusere. Dette innebærer at det står *opprett klon av meg* i programmet. Klonebegrepet kan skape misoppfatninger for elevene, da de kan tro at avkommet til harene er en direkte kopi av mor. For å hindre at det oppstår en misoppfatning om kopiering av gener, har vi valgt å lage egne funksjoner for reproduksjon. I programmet som elevene jobber med vil det brukes funksjonen *reproduksjon* og ikke begrepet *kloning* (se figur 8b). Ved å lage en funksjon som heter *reproduksjon*, ønsker vi å illustrere for elevene at formering hos harer ikke er en kloningsprosess, men at klonen kun er en del av det programmeringstekniske for å kunne lage simuleringen.

Under kvalitetsutviklingen erfarte vi at nivået på programmeringen var vanskelig for elevene. Det var behov for en nivådeling i både kunnskap om programmering og kunnskap om naturlig seleksjon. Av den grunn utviklet vi enda en simulering som heter *Tilpasninger og overlevelse*. Denne simuleringen skal hovedsakelig illustrere betydningen av begrepet *survival of the fittest* og belyse misoppfatningen om at det alltid er den største og sterkeste som er best tilpasset et miljø (Reydon, 2021). Denne simuleringen inneholder i likhet med simuleringen *Naturlig seleksjon* et byttedyr og et rovdyr, men her i form av en katt og en mus. Forskjellen mellom simuleringene er at simuleringen *Tilpasninger og overlevelse* visualiserer at størrelse påvirker muligheten for overlevelse, mens *Naturlig seleksjon* visualiserer hvordan utseende som en tilpasning. En annen

forskjell er at simuleringen *Tilpasninger og overlevelse* er ikke en ferdig simulering, slik som *Naturlig seleksjon*. Vi har utviklet denne simuleringen for at elevene kan utforske programmering i Scratch og lage ferdig simuleringen selv. Ved å skulle designe deler av simuleringen selv får elevene interagert med det naturlige fenomenet (German, 2019). Simuleringen inneholder enklere programmering og ser hovedsakelig kun på hvilke egenskaper og tilpasninger som er godt tilpasset til et miljø. Denne simuleringen er en svært forenklet versjon av virkeligheten, men forklarer likevel et viktig poeng med naturlig seleksjon (Hannisdal & Ringnes, 2015). Ved å ha utviklet to simuleringer er det større mulighet for å tilpasse læringsressursen innad i klassene og mellom klassetrinn (Vygotsky, 1978, henvist i Moen, 2015).

#### 5.1.1.2 Oppgavehefte

I arbeid med læringsressursen og oppgaveheftene, skal elevene jobbe i grupper på tre. Dette valget er tatt på bakgrunn av litteratur som presenterer at grupper på tre personer er mest effektive og presterer best (Laughlin et al., 2006). Gjennom arbeidet med oppgaveheftene får elevene jobbet med samarbeid, seleksjon og programmering.

Vi har valgt å utvikle to oppgavehefter med ulikt nivå, slik at læringsressursen kan tilpasses ulike forutsetninger og alderstrinn. Valget er gjort for å forhindre at oppgavens nivå er utenfor grensen av hva elevene klarer (Vygotsky, 1978, henvist i Moen, 2015). Oppgaveheftet *Tilpasninger og overlevelse* er tilpasset et enklere faglig og programmeringsteknisk nivå. Ved å ha et enklere nivå i programmeringen og kunnskapen om seleksjon vil oppgaveheftet *Tilpasninger og overlevelse* enten kunne brukes som en egen undervisning eller som en tilpasning, forberedelse og programmeringsøving til oppgaveheftet *Naturlig seleksjon*. For å tilpasse for ulike nivåer i oppgaveheftene har vi lagt inn en tipsknapp slik at elever som finner det programmeringstekniske utfordrende, kan få hjelp og støtte for å oppnå mestring (Knain, 2019; Vygotsky 1978, henvist i Moen, 2015). Tipsene og hjelpen kommer i form av å få utdelt klosser, et halvferdig program og hint om fremgangsmåte (Gjøvik & Høyland, 2022). Både valget om å ha to programmeringsnivåer i oppgaveheftene og å ha en tipsknapp er gjort for å forhindre at programmering tar fokuset bort fra naturlig seleksjon. Disse valgene er gjort med bakgrunn i studien til Caballero et al. (2014) som beskriver at programmering må prioriteres og øves på i undervisningen for at elever skal få et faglig utbytte av andre naturfaglige temaer i kombinasjonsundervisning.

I utviklingen av oppgaveheftene har vi tatt utgangspunkt i undervisningsmetoden PRIMM (Sentance & Wait, 2017). For å gi elevene øving og kompetanse i programmering har vi valgt å legge opp oppgaveheftene etter PR-PRIMM. Dette innebærer at det vil være to oppgaver i heftene hvor elevene skal *forutse* og *kjøre* koden, ved å få et utklipp av en kode fra programmet. På denne måten får elevene øving i Scratch, og ved å starte med enkle oppgaver som går på programmering er målet at elevene forberedes for videre og mer avanserte programmeringsoppgaver.

I tillegg til at oppgaveheftet er laget ut ifra stegene: *forutse, kjør, undersøk, endre og lage* har vi valgt å inkludere en *vurderingsoppgave*. Ved å inkludere en vurderingsoppgave i heftet, får elevene øvd seg på å vurdere simuleringen av naturlig seleksjon. Det å kunne vurdere modeller er beskrevet i kompetansemålene i naturfag etter 10.trinn: *bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensninger* (Kunnskapsdepartementet, 2019). *Vurderingsoppgaven* vil i tillegg til å dekke enda et kompetansemål, være viktig å inkludere i



undervisningen for å forhindre at elevene lager seg nye misoppfatninger knyttet til simuleringen (Harms & Reiss, 2019). Læreren som gjennomfører undervisningen bør understreke modellens styrker og svakheter, og legge vekt på at simuleringen kun er en forenklet versjon av virkeligheten (Hannisdal & Ringnes, 2015)

Til tross for at PRIMM er en undervisningsmetode for å jobbe med programmering har vi valgt å ha fokus på naturlig seleksjon i oppgavene. I oppgaveheftene finnes det informasjonsbokser som handler om temaer i naturlig seleksjon før oppgaver om det samme tema (se figur 5, s.19). Her inkluderes en informasjonsboks som beskriver overlevelse og kamuflasje før en oppgave hvor elevene skal undersøke harenes kamuflasje i simuleringen. Et annet eksempel på naturfaglig fokus er fra oppgaveheftet *Tilpasninger og overlevelse*. I *endreoppgaven* skal elevene både endre miljøet og utseende til musen slik at den ville vært bedre tilpasset miljøet. Gjennom disse oppgavene fokuserer elevene på hvordan miljø og egenskaper påvirker overlevelse.

I oppgaveheftet har vi valgt å forklare viktige begreper om evolusjon og naturlig seleksjon. Funn fra studiene til Harms & Reiss (2019) viser at elever strever med å forstå underliggende temaer som populasjon, adaptasjon og frekvenser. Videre trekker studier frem at organisering av nivåer er utfordrende for elever. Av den grunn er det viktig å forklare elevene hva som er forskjellen mellom gener, individer, populasjoner og arter og forsikre seg at elevene forstår hvilket nivå det refereres til. I tillegg har vi valgt å inkludere en forklaring av begreper for at lærere og elever skal benytte riktige begreper for å unngå misoppfatninger (Harms & Reiss, 2019).

Noen viktige begreper:

- **Individ:** Er en enkelt hare. Den har unike genvarianter og kun eneggede tvillinger har like gener.
- **Populasjon:** Er en samling individer av en bestemt art som lever innenfor et område.
- **Gener:** er en oppskrift for egenskaper til et individ som ligger inne i cellekjernen til alle cellene i kroppen.
- **Egenskaper:** Gener, som er arvet fra mor og far, bestemmer egenskapene til et individ.
- **Avkom:** Er harenes unger. De arver egenskaper fra mor og far.
- **Kull:** Er hvor mange harer som fødes hver gang en hare er drektig.

Figur 9: Liste og forklaring av begreper fra økologi som brukes i oppgaveheftet *Naturlig seleksjon*.

Å implementere seleksjon i informasjonsbokser i oppgaveheftene er et valg gjort på bakgrunn av tidligere studier som viser at når man jobber med teknologi blir det andre faglige fokuset betydelig redusert (Bungum et al., 2014; Caballero et al., 2014).

### 5.1.1.3 Lærerveiledning

Vi har valgt å utvikle en lærerveiledning for å gi lærerene veiledning i deres arbeid med læringsressursen.

I lærerveiledningen har vi valgt å beskrive hvordan læreren kan bruke læringsressursen i egen undervisning, ved å blant annet anbefale hvilke oppgaver som bør tas i plenum. I tillegg har vi valgt å beskrive hvordan læreren kan tilpasse undervisningen til elevene i klassen. En slik differensiering kan være at elevene får utdelt noen av klossene de

trenger fra start for å gjøre oppgavene enklere (Gjøvik & Høyland, 2022) eller at elevene kan utforske programmet selv eller lage et eget program slik at de elevene med høyest læringspotensial får utfordret seg (Haraldsrud et, al., 2020).

Ytterligere har vi valgt å beskrive programmeringen i simuleringen *Naturlig seleksjon*. Dette er gjort for at læreren skal få forståelse for hva som skjer både faglig og programmeringsteknisk. Vi har også inkludert en beskrivelse av evolusjon og Scratch slik at lærere med lav kompetanse i programmering og evolusjon også kan ta i bruk læringsressursen.

### 5.1.2 Utprøvningsfasen

I dette kapittelet presenteres erfaringene fra utprøvingen. Dette inkluderer både observasjoner som vi gjorde i utprøvingen og erfaringer vi i felleskap med læreren har opparbeidet oss.

En uke før gjennomføringen fikk læreren som vi samarbeidet med utdelt simuleringen, oppgaveheftene og lærerveiledningen. Dermed var det rom for å stille oppklarende spørsmål og gjøre endringer, om dette skulle være nødvendig, før gjennomføringen av opplegget. Læreren hadde ingen spørsmål før gjennomføringen, og hadde lest gjennom hele lærerveiledningen utenom innføringen i Scratch og evolusjon.

Undervisningen ble gjennomført i to klasser med to undervisningstimer i hver klasse. På grunn av lite programmeringskompetanse hos klassene og læreren valgte læreren å starte med det enkleste oppgaveheftet *Tilpasninger og overlevelse*. Elevene fikk utdelt papirutgaven av oppgaveheftet og ble delt i grupper på tre. Læreren gikk så igjennom oppgave 1 i felleskap, før elevene fikk jobbe i gruppene med oppgaveheftet. Utover i arbeidet stoppet læreren så opp arbeidet for å ta *endreoppgaven* i felleskap. Dette begrunnet han med at han ønsket å sikre at elevene forsto hvordan de kunne endre på variabler i programmet. Det fungerte godt å begynne med oppgaveheftet *Tilpasninger og overlevelse*, hvor elevene fikk øvd seg på enklere programmering før de videre jobbet med oppgaveheftet *Naturlig seleksjon*.

I time nummer to startet elevene med å få utdelt papirutgaven av oppgaveheftet *Naturlig seleksjon* og læreren viste elevene simuleringen av naturlig seleksjon. Læreren forklarte prinsipper om simuleringen og viste elevene kodene i programmet. Dette skapte en diskusjon i klassen om hvilke av harene som ville overleve og hvorfor. Før gruppene startet å jobbe med heftet på egenhånd, ble oppgavene om å forutse og kjøre koden gjort i plenum for å sikre at elevene forsto det programmeringstekniske.

Etter en kort samtale med læreren og observasjoner fra første gjennomføring erfarte vi at arbeid med læringsressursen fungerte godt, men at oppgaveheftet hadde for mange oppgaver og at noen av oppgavene hadde for høy vanskelighetsgrad. Videre erfarte vi at det er utfordrende for elevene å arbeide lenge med oppgaver og at det derfor i neste gjennomføring bør tas flere refleksjoner og diskusjoner om oppgavene i plenum. Med utgangspunkt i dette reduserte vi antallet oppgaver i oppgaveheftene, og forenklet noen av oppgavene imellom gjennomføringene. I tillegg ble vi enige med læreren om å legge opp til mer diskusjoner i klassen og mindre selvstendig oppgavejobbing. Vi endret også noen feil i koden til programmet. En annen viktig endring var at vi også delte ut heftet digitalt, for å tydeliggjøre tips til programmeringen.



Under andre gjennomføring gjennomførte vi opplegget mye på den samme måten. Endringene fra første gjennomføringene var å i større grad samle klassen og ta en gjennomgang av utvalgte oppgaver i plenum. Under andre gjennomføring var det spesielt fokus på å ta *diskuter- og vurderoppgavene* i fellesskap, ved at elevene først fikk diskutere i grupper. En annen endring i andre gjennomføring var at læreren brukte tipsene i oppgaveheftet når oppgavene ble tatt i fellesskap. Vi erfarte dermed at elevene i større grad benyttet seg av tips for oppgavene når oppgavene ble gitt som et digitalt hefte og at oppgavene da ble gjennomført mer effektivt av elevene.

Etter andre gjennomføring erfarte vi i samråd med lærer at det var mer effektivt å ha et muntlig fokus på oppgavene. Dette sikret i større grad at elevene fokuserte på seleksjon og ikke bare programmering. Generelt fra begge gjennomføringene erfarte vi at programmering kan motivere elevene til å jobbe med oppgaver, og at elevene satt pris på å ha programmering som et alternativ til deres ordinære undervisning. Angående gruppestørrelse erfarte vi at tre elever på en gruppe var lite effektivt, da kun en elev får jobbet i programmet av gangen.

### 5.1.3 Retrospektiv analysen

Etter å ha observert en lærer gjennomføre undervisningen i to ulike klasser, avholdt vi en reflekterende sluttsamtale med læreren om gjennomføringen av læringsressursen. Erfaringene fra de to gjennomføringene og sluttsamtalen med læreren ga grunnlag for videreutvikling av læringsressursen og er beskrevet i den retrospektive analysen. Endringene og utviklingen av læringsressursen etter utprøvingen er begrunnet i relevant litteratur. Videreutviklingen startet allerede mellom første og andre gjennomføring. Etersom vi har tilpasset Bjørndals (2013) modell om pedagogisk forskningsdesign for vår oppgave ved å benytte to sykluser av utviklingsfasen (se figur 1, s.12) har vi valgt å begrunne alle endringer etter syklus 1 og syklus 2 under dette delkapittelet.

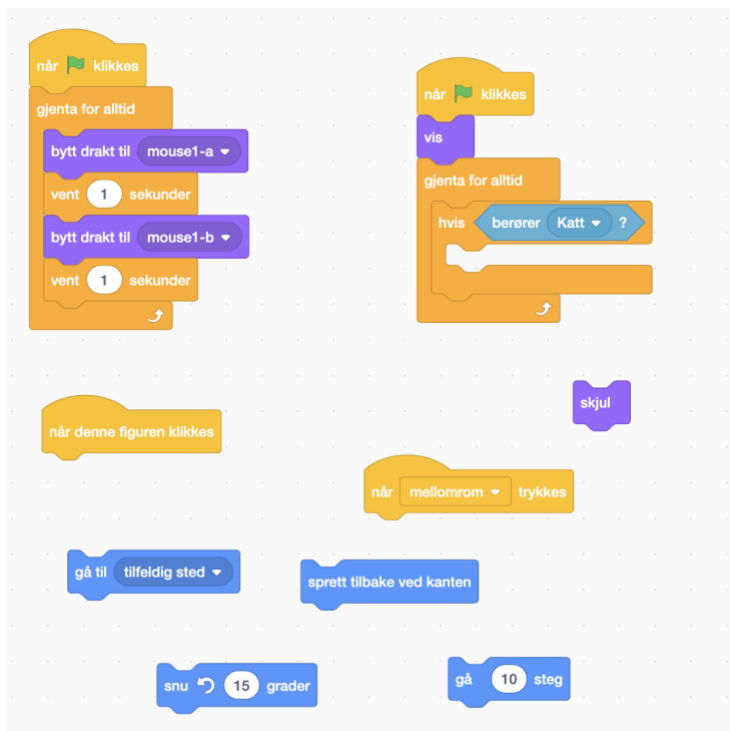
Fra gjennomføringen erfarte vi at det å ha tre elever i en gruppe, er lite effektivt når man skal jobbe med programmering. Vi har derfor valgt å endre anbefalingen i lærerveiledningen til å være to elever på hver gruppe. Etersom kun en elev kan programmere og endre på faktorer i programmet av gangen, er det mest effektivt at færrest mulig som samarbeider, for å sørge for at alle på gruppen er delaktige i arbeidet. På denne måten kan elevene samarbeide, uten at noen melder seg ut av samarbeidet. Det å ha to elever på en gruppe i arbeid med programmering, støttes av undervisningsmetoden PRIMM (Sentance et al., 2019).

Vi erfarte at det var effektivt å ha mer muntlig fokus i undervisningen. Vi anbefaler dermed å samle elevene til en felles refleksjon etter oppgavejobbing i grupper. Under andre gjennomføring erfarte vi at det ble et mer faglig fokus på naturlig seleksjon, når undervisningen ble endret til å ha mer felles refleksjoner og diskusjoner. Ved å ha felles samtaler kan læreren flytte fokuset fra kun programmering til å inkludere naturlig seleksjon. Vi har derfor justert oppgaveheftet og lagt inn anbefaling i lærerveiledningen om å samle klassen og i fellesskap diskutere flere av oppgavene som elevene har jobbet med. Dermed gir læreren elevene veiledning som kan være med på å utvikle elevers kompetanse i den nærmeste utviklingssonen (Vygotsky 1978, henvist i Moen, 2015). I tillegg begrunner vi denne endringen i undervisningen på bakgrunn av at det tidvis er nødvendig med tilrettelegging og støtte for å oppnå mestring, spesielt ved oppgaver som elever ikke hadde mestret uten læreren. Læreren og deres arbeid med å reflektere rundt oppgavene i heftet blir dermed en viktig støttestruktur for elevene (Knain et al., 2019).

Vi erfarte at programmering var motiverende for mange av elevene og at dette ga elevene mulighet til å kombinere læring om programmering og naturlig seleksjon. Læringsaktiviteten antar vi at ble utført i stor grad av indre motivasjon, ettersom elevene syntes det var interessant og at læringsaktivitetene ga glede og tilfredsstillelse (Deci & Ryan, 2009; Skaalvik & Skaalvik, 2015). Ved å kombinere programmering og arbeid med seleksjon kan man engasjere og motivere elever med et annet fokus enn en tradisjonell undervisningsmetode. Studien til Lai og Lai (2012) viser at 60% av elevene foretrakk å ha naturfagundervisning i Scratch. De erfaringene vi gjorde oss under utprøvingen støtter denne studien ved at vårt utvalg av elever synes arbeid med programmering i Scratch i stor grad er en motiverende aktivitet som en variasjon til deres ordinære naturfagundervisning. Våre erfaringer om elevenes indre motivasjon bygger på lærerens utsagn, elevutsagn i gruppearbeidet og observasjon.

### 5.1.3.1 Simulering

Erfaringene fra utprøvingen var at simuleringen *Tilpasninger og overlevelse* fungerte godt som en oppvarming til arbeid med simuleringen *Naturlig seleksjon*. Likevel erfarte vi at enkelte elever fant det utfordrende å ferdigstille simuleringen. For å sikre at alle elevene får ferdigstilt simuleringen la vi inn de klossene som elevene trengte for å ferdigstille simuleringen løst i programmet (se figur 10).



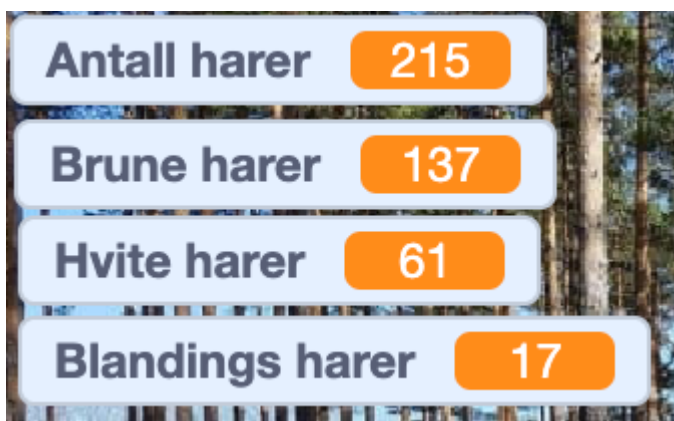
Figur 10: Klossene som elevene skal bruke for å ferdigstille programmet er hentet frem.

Dette er en tilpassing hvor elevene har et mindre utvalg klosser de vurderer i byggingen av programmet (Gjøvik & Høyland, 2022). På denne måten er programmeringen lettere tilgjengelig for alle, og dette gjør at det kreves mindre modellering av elevene for å ferdigstille simuleringen. Dette valget har vi tatt basert på at alle elevene skal ha mulighet til å lage ferdig simuleringen, ettersom teori sier at evnen til å lage modeller er essensiell for å kunne tilegne seg dybdekunnskap (Hallstøm & Schönborn, 2019).

Etter sluttsamtalen med læreren var det enighet om at det er viktig at også simuleringen *Tilpasninger og overlevelse* vurderes for styrker og svakheter. Dermed har vi inkludert en oppgave i oppgaveheftet *Tilpasninger og overlevelse* hvor elevene skal vurdere simuleringen. Dette valget er gjort på bakgrunn av at det er det å være kritisk til modeller og kunne vurdere modellenes svakheter som gjør naturfagundervisningen mer autentisk (Hallstøm & Schønborn, 2019).

Etter utprøvingen og sluttsamtalen med læreren, ble vi enige om å gjøre noen endringer med simuleringen *Naturlig seleksjon*. Vi erfarte at det var en stor mengde koder i flere figurer i programmet og derfor vanskelig å finne frem til hvilken kode som gjør hva i programmet. På bakgrunn av denne erfaringen har vi ryddet, forenklet og gitt kodene i Scratch enklere navn. For eksempel har vi endret variabelen *GlobalID* til *oversikt over hvite/brune/blandingsharer*. På denne måten blir det enklere og tydeligere for elevene når de skal endre deler av programmet. Vi har samlet de kodesekvensene som hører sammen. Videre har vi plassert de kodesekvensene som er for vanskelige og som elevene ikke skal bruke unna de kodesekvensene som elevene skal bruke. Ryddingen i programmet kan være med å forhindre at programmet overskrider prosesseringskapasiteten til elever som kan føre til for stor kognitiv belastning (Mayer & Moreno, 2003). Ved å bruke gode, ferdig utarbeidede programmer kan man redusere kognitiv belastning hos elevene (Qian & Lehman, 2017).

I tillegg til å rydde i programmet *Naturlig seleksjon*, har vi laget en del i lærerveiledningen som beskriver alle variabler i programmet. En variabel lagrer verdier i programmet og kan være et tall, en posisjon eller en tekst. Et eksempel på dette er variabelen som teller antall harer (se figur 11). I lærerveiledningen er det beskrevet hvilke variabler som kan endres for å illustrere ulike deler av naturlig seleksjon, disse er det anbefalt at læreren viser til elevene. På denne måten kan læreren vise elevene hvordan man endrer programmet og dette kan gjøre oppgavejobbingen lettere for elevene. Det er viktig at læreren støtter elevene i deres arbeid for å mestre programmeringen (Knain et al., 2019). Dette valget er gjort på bakgrunn av at elever synes det er krevende med variabler (Qian & Lehman, 2017).



Figur 11: Variabel som viser antall harer i simuleringen.

Under gjennomføringen erfarte vi at det kan være utfordrende å forstå hva som skjer i simuleringen, når harer dør av sult eller når reven har spist opp alle harene. Derfor har vi valgt å legge inn flere meldinger som forteller hva som har hendt i programmet. På denne måten kommer naturlig seleksjon tydeligere frem i simuleringen.

## Det er bare en hare igjen

Denne har ingen å reprodusere seg med og derfor vil harene dø ut  
Reven må finne seg et nytt område å jakte på hvis ikke vil den sulte

Figur 12: Eksempel på en melding som forteller hva som har hendt i programmet.

### 5.1.3.2 Oppgavehefte

Erfaringer fra utprøvingen viste at flere av kodene som elevene skulle jobbe med i oppgaveheftene var kompliserte. På bakgrunn av denne erfaringen har vi valgt å endre og forenkle kodene i *forutse-* og *kjør-oppgavene* til å bare inneholde den begrensede delen av koden som vi ønsker at elevene skal fokusere på (se figur 13b). På denne måten slipper elevene å bli forvirret av avanserte koder som kun er med for å få simuleringen til å fungere. Denne forenklingen gjøres for å hindre for stor kognitiv belastning for elevene (Mayer & Moreno, 2003).

**Oppgave 3: FORUTSE OG KJØR**

Hva tror dere denne koden gjør?  
**Tips:** Hva tror dere vil skje hvis en hare og en rev møtes

The image shows two parts of a Scratch code editor. On the left (Figur 13a), a full sequence of code blocks is visible, including a 'when clicked' event, a 'limit rotation' block, a 'pick random direction' block, a 'go to random place' block, a 'repeat until' loop with a 'y-position < 0' condition, an 'if' block with a 'y-position > 0' condition, a 'show' block, a 'repeat forever' loop with a 'wait random time from 0 to 1 seconds' block, a 'hop' block, a 'meet partner' block, an 'if' block with a 'touches Rev?' condition, 'change white rabbits by -1' and 'change number of rabbits by -1' blocks, a 'hide' block, a 'go to x: 15000 y: -15000' block, a 'set original 2 dead to 1' block, and another 'if' block with a 'y-position > -10' condition and a 'turn 180 degrees' block. On the right (Figur 13b), a simplified version of the code is shown, focusing on the 'if' block with the 'touches Rev?' condition, the 'change white rabbits by -1' and 'change number of rabbits by -1' blocks, and the 'hide' block.

Figur 13a og b: Viser endringer som ble gjort i oppgaveheftet. I stedet for å bruke hele kodesekvensen brukes bare en begrenset del som elevene skal fokusere på i oppgaven.

Gjennom utprøvingen erfarte vi at det var utydelig hvilke oppgaver som skulle tas i plenum og hvilke oppgaver som skulle jobbes med i grupper. Av den grunn har vi tydeliggjort i oppgaveheftet og i lærerveiledningen hvilke oppgaver som skal tas i plenum.

I samråd med læreren ble vi også enige om å flytte *vrderoppgaven* først i oppgaveheftet *naturlig seleksjon*. Ved å starte undervisningen med å vurdere simuleringen, forhindrer man at elevene utvikler misoppfatninger av svakhetene til modellen. I tillegg valgte vi å inkludere en *vrderoppgave* i oppgaveheftet *Tilpasninger og overlevelse*. Disse endringene er gjort på bakgrunn av teori som sier det er viktig å poengtere at modeller er forenklinger av virkeligheten, og at alle deler av modellen ikke vil samsvare med virkeligheten (Hannisdal & Ringnes, 2015). Videre beskriver teori hvordan det er viktig å være kritisk til modeller for å få en mer autentisk naturfagundervisning (Hallstøm & Schönborn, 2019). Derfor har vi utarbeidet en liste over svakheter og styrker som er viktig å trekke frem med simuleringen. Dette vil inkluderes i lærerveiledningen som en støtte i lærerens arbeid med å vurdere simuleringen.

### 5.1.3.3 Lærerveiledningen

Totalintrykket til læreren var at lærerveiledningens innhold var dekkende for at lærere med noe programmeringskompetanse skal kunne ta læringsressursen i bruk. Likevel erfarte læreren at lærerveiledningen var misvisende når det kom til nivåinndeling etter klassetrinn. Etter utprøvingen erfarte vi at både undervisning om *Tilpasninger og overlevelse* og *Naturlig seleksjon* var lærerikt og hensiktsmessig i niende klasse. Derfor har vi endret beskrivelsene av de to oppgaveheftene i lærerveiledningen til å være beskrevet som nivå og ikke etter klassetrinn. På denne måten er det lettere for læreren å bruke de ulike oppgaveheftene på en måte som er tilpasset nivået i klassene. I tillegg til at læreren kan tilpasse nivået etter hvilket oppgavehefte undervisningen baserer seg på, vil det være flere muligheter for å velge ulik vanskelighetsgrad på oppgavene. Elevene kan dermed velge ut og utfordre seg på oppgaver i deres nærmeste utviklingssone (Vygotsky 1978, henvist i Moen, 2015). Det er viktig at læreren differensierer undervisningen for å motivere elever både med stort og lavt læringspotensialet (Haraldsrud et al., 2020). For å tilpasse undervisningen til elever med stort læringspotensial kan læreren gi elevene oppgaver som er mer tidkrevende, har høyere nivå, benytter tekstprogrammering eller er utforskende (Haraldsrud et al., 2020). Som et vedlegg i lærerveiledningen er en krevende oppgave hvor elevene skal lage sin egen simulering i Scratch som illustrerer sykdommen harepest. Til elever med lavere læringspotensial inkluderer oppgaveheftet hint som består av halvferdige programmer, klosser som trengs for å gjøre programmet ferdig og veiledning til hvor i programmet man skal lete (Gjøvik & Høyland, 2022). Valget om å inkludere både vedlegg med ekstra oppgaver og hint, er basert på at det er store nivåforskjeller i programmering (Haraldsrud et al., 2020).

Læreren erfarte at det var utfordrende å gi hjelp til de elevene med stort læringspotensial i programmering. Vi erfarte også at enkeltelever var kreative og løste oppgavene på en annen måten enn vi hadde tatt utgangspunkt i tipsene. For at læringsressursen skal kunne tas i bruk av alle naturfagslærere uavhengig av ferdighetsnivå i programmering mener læreren at tips for denne elevgruppen også bør inkluderes i lærerveiledningen. Derfor har vi fylt på lærerveiledningen og gitt forslag til hvordan oppgavene kan løses.

#### 5.1.4 Har vi nådd målet i utviklingsprosessen?

For å vurdere om vi har nådd delmålet om å utvikle en læringsressurs hvor programmering brukes i kombinasjon med naturlig seleksjon, tar vi utgangspunkt i vår versjon av Sandovals (2014) antakelseskart for pedagogisk designforskning (se figur 2, s.13).

Under kartleggingen av forventningene til studien, planla vi at læringsressursen skulle inkludere arbeid med simuleringer av naturlig seleksjon, gruppearbeid med oppgaveheftet og klassesdiskusjoner. På denne måten blir kunnskapen formidlet til elevene gjennom egen utforskning, veiledning og støttestrukturer, interaksjoner i gruppearbeid med medelever og faglige diskusjoner i klasserommet. Dette stemmer overens med erfaringen fra utprøvingen.

I utviklingen av læringsressursen forventet vi at utfallet av arbeid med læringsressursen er motiverende, utvikler kompetanse i programmering, gir elevene læring om naturlig seleksjon og forhindrer misoppfatninger hos elevene. Etter å ha prøvd ut læringsressursen har vi indikasjoner på at våre tiltenkte utfall stemmer. Derimot så består ikke vårt erfaringsgrunnlag av undersøkelser som kan konkludere om dette stemmer. Det kreves en egen studie som måler motivasjonen, og læringsutbyttet for å bekrefte om læringsressursen oppnår disse ønskede resultatene.

På tross av at vi ikke har målt motivasjonen til elevene erfarte vi under gjennomføring at elevene virket motiverte under arbeidet med læringsressursen. Vi erfarte at kombinasjonen av programmering og naturlig seleksjon i undervisningen var motiverende så lenge det ble gitt tilstrekkelig med støtte til elevene når de møtte utfordringer.

Resultatet fra utprøving tyder også på at elevene utviklet egen kompetanse i programmering ved å bruke læringsressursen. I arbeid med oppgaveheftene økte gradvis vanskelighetsgraden på oppgavene, og vi observerte at mange elever mestret økningen av nivå.

Om undervisningsressursen fører til læring om naturlig seleksjon og forhindrer misoppfatninger hos elevene, har vi heller ikke gjennomført tilstrekkelig med undersøkelser for å konkludere. Likevel erfarte vi at det ble gjennomført gode diskusjoner om naturlig seleksjon både som klassesdiskusjon og i elevgrupper, noe som tyder på at elever enten hadde tidligere kompetanse eller har utviklet kompetanse om naturlig seleksjon.

Forskning viser også at simuleringen kan tilrettelegge for kunnskapsutvikling og hjelpe elever med å forstå underliggende abstrakte konsepter og mekanikker om et naturlig fenomen (Weng & Tseng, 2016). Ved å benytte simuleringer i læringsressursen legges det til rette for at undervisningen beskriver og forenkler naturlig seleksjon (Gilbert et al., 2000), simuleringen kan brukes som et verktøy for å oppklare misoppfatninger i naturlig seleksjon (Osborne, 2014). Imidlertid har ikke vår studie tilstrekkelige erfaringer for å konkludere om dette er tilfellet med vår læringsressurs.

For å evaluere utviklingsarbeidet tar vi utgangspunkt i erfaringene fra observasjon og samtalen med læreren, som tyder på at elevene har utviklet kompetanse i programmering og naturlig seleksjon. Dermed mener vi at dette er en læringsressurs som støtter kompetansemålet om å *bruke programmering til å undersøke naturfaglige fenomener* og at denne læringsressursen kan være en støtte for lærere i deres naturfagundervisning. Vi kan dermed påstå at vi har nådd målet om å utvikle en



læringsressurs som kombinerer arbeid med programmering og naturlig seleksjon. Følgelig kan vi påstå å ha bekreftet antagelsen for studien. Likevel kan vi ikke konkludere om utfallet av studien ble som forventet, da vi ikke har gjennomført undersøkelser eller brukt metoder som gir innsikt i læringsutbytte, misoppfatninger og motivasjon.

## 5.2 Delmål 2: Utvikle en læringsressurs som imøtekommer utfordringene i skolen knyttet til kombinasjonen av programmering og naturlig seleksjon

Tidligere forskning viser at elever ofte har flere misoppfatninger innenfor programmering og naturlig seleksjon (Gregory, 2009; Harms & Reiss, 2019; Qian & Lehman, 2017). For å unngå at elever utvikler misoppfatninger som forstyrrer deres læring, har vi hatt fokus på å imøtekomme utfordringer knyttet til programmering og naturlig seleksjon i utviklingen av læringsressursen. I dette delkapittelet drøftes det om læringsressursen imøtekommer utfordringer som beskrives i tidligere forskning.

### 5.2.1 Utfordringer med arbeid i programmering

I utviklingen av læringsressursen har vi tatt utgangspunkt i tidligere forskning om misoppfatninger i programmering for å unngå at elevenes programmeringskompetanse skal forhindre elevenes arbeid med naturlig seleksjon. Ved å benytte blokkprogrammering i læringsressursen begrenser vi behovet for syntakskunnskap hos elevene. Blokkprogrammering gjør at elevene får større kognitiv kapasitet til å konsentrere seg om konseptuell kunnskap og strategisk kunnskap (se kap. 2.2.2, s.7) som er nødvendig for å kunne programmere et fungerende program (Qian & Lehman, 2017). I følge Wing et al (2010) sin definisjon av algoritmisk tenking skal løsningen på et problem presenteres på en form som effektivt kan forstås av andre. Uten forståelse for konseptuell og strategisk kunnskap er det ikke mulig for elever å forstå hvordan de skal løse et problem med programmering eller å presentere en løsning som andre også kan forstå (Wing et al., 2010). Ved å benytte blokkprogrammeringsprogrammet Scratch, får elevene ferdige klosser og dette vil dermed forhindre utfordringer som feil i kodene i programmeringen (Resnick et al., 2009). Ved å velge Scratch gir man elevene muligheten til å visuelt se hva klossene og kodene gjør i programmet. Scratch gjør det enklere for elever å utforske simuleringen og dermed undersøke oppbygningen av programmet. Dette gir også en mulighet for at elever kan endre faktorer i simuleringen som er knyttet til naturlig seleksjon og observere hvordan endringene påvirker simuleringen.

Innenfor konseptuell kunnskap er det særlig variabler, løkker og vilkår som elever har misoppfatninger om (Grover & Basu, 2017). Slike misoppfatninger påvirker elevenes evne å forstå programmer og selv lage fungerende programmering. Derfor har vi særlig fokus på å forklare disse programmeringskomponentene i lærerveiledningen for at læreren skal kunne veilede elevene. Ved å benytte rammeverket PRIMM økes vanskelighetsgraden for programmeringen gradvis (Sentance et al., 2019). Særlig de to første oppgavene *forutse-kjør* brukes som en inngang til programmering og særlig variabler, løkker og vilkår. Vi valgte å inkludere to *forutse-kjør-oppgaver*, slik at formen på oppgaveheftet var PR-PRIMM. Dermed får elevene erfare og diskutere viktige programmeringskomponenter i starten av undervisningsøkten, slik at elevene har større forståelse for dem og kan ta dem i bruk i senere oppgaver. Vi erfarte i utprøvingen at

elevene diskuterte løkker, variabler og vilkår under *forutse* og *kjør* oppgavene, og at elevene senere brukte disse programmeringskomponentene videre i arbeidet.

### 5.2.2 Utfordringer med naturlig seleksjon

Utviklingen av læringsressursen er basert på tre betydelige misoppfatninger om naturlig seleksjon. Vi har utviklet to simuleringer om naturlig seleksjon, ettersom simuleringer kan brukes som et verktøy for å hjelpe elever med å forstå underliggende abstrakte konsepter og mekanikker om naturlige fenomener (Wang & Tseng, 2018). I utviklingsprosessen har vi hatt fokus på å forhindre misoppfatninger om naturlig seleksjon. Forskningen til Malone et al. (2019) viser at utforskning av deres interaktive simulering gjorde at elever, i større grad enn i ordinær undervisning, reviderte sine tidligere misoppfatninger. Om misoppfatninger hos elever revideres med bruk av vår simulering er utfordrende å si ettersom vi ikke har målt læringsutbytte hos elevene. Videre i delkapittelet beskrives de valgene vi har tatt for å oppklare misoppfatninger om naturlig seleksjon i simuleringen.

Det er mange misoppfatninger knyttet til begrepet *Survival of the fittest*. Mange elever oppfatter at fitness handler om fysisk styrke (Reydon, 2021). For å illustrere denne misoppfatningen har vi utviklet simuleringen *Tilpasninger og overlevelse*. Denne simuleringen er svært enkel og skal illustrere at det å være størst og sterkest ikke nødvendigvis betyr at et individ er godt tilpasset et miljø. Simuleringen vil illustrere at den minste musen er mer skjult for katten. For å ikke lage nye misoppfatninger er det viktig som lærer understreker at ulike egenskaper er fordelaktige i ulike miljø og for ulike arter. I oppgavene knyttet til simuleringen skal elevene endre musen til å bedre være tilpasset miljøet.

Elever har ofte misoppfatninger rundt tidsrammen for evolusjon (Gregory, 2009; Harms & Reiss, 2019). Dermed har vi utviklet simuleringen *Naturlig seleksjon* i læringsressursen, som kan synliggjøre at prosessen med naturlig seleksjon foregår over flere generasjoner. I simuleringen illustreres dette gjennom reproduksjonen til harene og ved å observere antall av harer med ulik pelsdrakt over tid. At organismer kan endre seg for å bedre passe inn i miljøet er en annen misoppfatning som er vanlig hos elever (Reydon, 2021). Denne misoppfatningen imøtekommes ved at elevene kan utforske simuleringen *Naturlig Seleksjon*, hvor de gjennom simuleringen og oppgaver kan erfare at harene ikke kan endre pelsdrakt, men at de med best kamuflasje overlever lenger og dermed kan reproducere seg og føre sine gener videre.

Imidlertid oppstår det et problem når vi skal oppklare denne misoppfatningen ettersom harer faktisk endrer pelsdrakt for å være bedre tilpasset miljøet. De har brun pelsfarge om sommeren og hvit pelsfarge om vinteren. I simuleringen illustreres det hvilken pelsfarge som er best tilpasset ulike miljøer, men ikke at haren faktisk endrer pelsfarge. Det er viktig å påpeke at selv om haren endrer pelsfarge, så endres ikke genene til haren. Diskusjon om pelsendring er viktig å inkludere i *vurderoppgaven* og derfor er det nevnt i lærerveiledningen at dette problemet skal inkluderes i klassediskusjonen. En modell er bare en forenkling av virkeligheten og alle deler av modellen samsvarer ikke med virkeligheten (Hannisdal & Ringnes, 2015), dette gjelder også for vår simulering. Av den grunn fokuserer vi i oppgaveheftene på at elevene skal vurdere modellens sterke og svake sider, for å unngå og avkrefte misoppfatninger om naturlig seleksjon. Å være kritisk til modellens formål og svakheter leder til en autentisk naturfagundervisning



(Hallstrøm & Schönborn, 2019), derfor tenker vi det er en fordel å bruke oppgaver hvor elevene skal vurdere og revidere simuleringene om naturlig seleksjon.

### 5.2.3 Utfordringer med å kombinere programmering og naturlig seleksjon

Et av kompetansemålene etter 10.trinn i læreplanen i naturfag sier at elever skal *bruke programmering til å utforske naturfaglige fenomener*. Læreplanen legger dermed opp til at undervisningen skal kombinere programmering med et annet naturfaglig tema. På tross av dette viser funn fra studien til Caballero et al. (2014) at det er utfordrende å koble programmeringen tydelig nok opp mot det naturfaglige innholdet. Med dette i bakgrunn har vi utviklet læringsressursen med fokus på kombinasjonen av programmering og naturlig seleksjon. Derfor har vi tatt utgangspunkt i at programmeringen benyttes i en simulering som visualiserer naturlig seleksjon. Etterpå skal elevene selv måtte gå inn i programmet og endre verdier for å utforske effektene i naturlig seleksjon. På denne måten er utforskning og jobbing med programmering en viktig del av undervisningen for å arbeide med naturlig seleksjon.

Ytterligere beskriver studien til Caballero et al. (2014) at elever har utfordringer med å skjønne om årsaken til feilkoden ligger i programmeringen eller i den faglige forståelsen. Ved å kombinere programmering og naturlig seleksjon kan elevene på denne måten få mindre læringsbytte av naturlig seleksjon om de strever med det programtekniske. For å unngå dette i vår læringsressurs har vi derfor inkludert tips og veiledning for programmering i oppgaveheftene. På denne måten ønsker vi å forhindre at elevs utfordringer med programmering skal forhindre læringsutbytte i naturlig seleksjon.

Videre viser studien til Bungum et al. (2014) at det er utfordrende å koble fagkunnskap i arbeid med teknologi. Studiens funn viser at arbeid med teknologi kombinert med andre temaer kan gi nyttige erfaringer for elever, men at dette krever at kunnskapen om det andre naturfaglige temaet må adresseres eksplisitt i undervisningen. Derfor inkluderer vi i oppgaveheftet fagbøker med informasjon om naturlig seleksjon. I løpet av arbeidet med læringsressursen vil læreren samle elevene fra oppgavejobbingen for å ha en felles klasseromsdiskusjon. På denne måten kan læreren legge vekt på også det faglige ved naturlig seleksjon, og ikke kun programmeringen. Ytterligere gir dette mulighet til å oppklare og forhindre misoppfatninger som kan komme av at simuleringen kun er en modell av virkeligheten.

Malone et al. (2019) trekker frem gode erfaringer med bruk av en interaktiv simulering om naturlig seleksjon fra sin studie. Deres studie viste at naturlig seleksjon effektivt kan undervises ved hjelp av en interaktiv simulering som elevene kan utforske og vurdere selv (Malone et al., 2019). Hva som skiller Malone et al. (2019) sine simuleringer fra vår simulering er at i deres simulering er programmeringen skjult og ikke noe elevene skal lære om. I vår simulering kan elevene endre årstid for å undersøke kamuflasje, undersøke et miljø med og uten rovdyr, endre mattilgang, endre i hvilken grad reven oppdager ulike pelsdrakter, endre pelsdrakt for avkom og endre hvor mange harer som fødes. Alle disse endringene gjøres ved å forstå programmet og endre på relevante koder. Totalt sett gir datasimuleringen og oppgavene gode muligheter for å interagere med det naturlige fenomenet naturlig seleksjon samtidig som elevene lærer om programmering.

Modeller er et fundamentalt aspekt for undervisning i disiplinene naturfag, teknologi, ingeniørarbeid og matematikk (Hallstrøm & Schönborn, 2019). Dermed kan modeller brukes som et naturlig bindeledd mellom de ulike disiplinene. Vår læringsressurs bruker

simuleringer av naturlig seleksjon som et bindeledd mellom teknologi og kunnskap om naturlig seleksjon. Modeller som datasimuleringer er hensiktsmessig å bruke til naturlig seleksjon som er et naturfaglig fenomen som er vanskelig å observere. I en simulering kan elever utforske "hva-hvis"- scenarioer på en effektiv måte, som ikke er mulig å visualisere i virkeligheten (German, 2019). Ettersom simuleringene er laget ved å programmere i Scratch og for å endre betingelser om naturlig seleksjon må man gjøre endringer med programmering, kombinerer man flere fagdisipliner i arbeidet med simuleringen. Her brukes modellen av naturlig seleksjon som et verktøy for å kombinere kunnskap og ferdigheter i programmering og naturlig seleksjon.

Weintrop et al. (2016) sitt rammeverk for algoritmisk tenking relatert til arbeid i naturfag og matematikkundervisning er dekkende å bruke ved utforsking av naturfenomener. I dette rammeverket kommer det frem at å lage nye, utvide, vurdere, designe og konstruere allerede eksisterende modeller er en sentral del av algoritmisk tenking og naturfaglig praksis (Weintrop et al., 2016). I vår læringsressurs inngår alle disse stegene fullstendig eller delvis. Av den grunn er arbeid med modeller en god måte å arbeide med algoritmisk tenking i skolen. I likhet med modeller er også programmering arbeid med algoritmisk tenking fordi det handler om å presentere løsninger i en form som effektivt kan forstås av andre (Wing et al., 2010). Derfor knytter vi arbeid med en modell av naturlig seleksjon sammen med arbeid med programmering fordi de begge er forståelse av algoritmisk tenking.

PRIMM er en undervisningsmetode utviklet for å støtte lærere i gjennomføringen av programmeringsundervisning for nybegynnere (Sentance & Wait, 2017). Selv om PRIMM er en undervisningsmetode utviklet utelukkende for programmeringsundervisning har vi valgt å ta utgangspunkt i denne metoden i oppgaveheftene om naturlig seleksjon. Etter utprøving erfarte vi at det fungerer godt å kombinere programmering og naturlig seleksjon med utgangspunkt i PRIMM. Undervisningsmetoden er designet for å gradvis utvikle elevers kompetanse i programmering. Nivået på oppgaver om naturlig seleksjon utvikles også gradvis fra å starte med å forutse hva som vil skje hvis en hare og en rev berører hverandre til å endre miljøet og næringskilder i simuleringen. Vi erfarte at denne gradvise oppbyggingen av vanskelighetsgrad av typer oppgaver fungerte bra også i kombinasjon med et annet naturfaglig tema. I tillegg til at oppgaveheftene baserer seg på PRIMM-metoden, anbefales læreren i lærerveiledningen å kombinere oppgavejobbing med diskusjoner i fellesskap. Ved å kombinere oppgavejobbing og felles diskusjoner får elevene reflektert rundt erfaringene med oppgavene. Dette støtter Dewey (1996) sin tanke om at erfaringer har begrenset betydning uten tanke og refleksjon. Læreren bør derfor få elevene til å tenke over hva de har gjort i programmering og hvorfor de gjør dette med tanke på naturlig seleksjon. Dette er essensielt for at elevene skal lære om programmering og naturlig seleksjon (Dewey, 1916, referert i Dale, 1996). Gjennom arbeid med læringsressursen legges det vekt på at elever skal observere, endre og utforske simuleringene. Det er derfor essensielt å få elevene til å reflektere rundt hva de observerer, og hva endringene utgjør for å bevisstgjøre elevene om hva de har erfart om naturlig seleksjon gjennom programmeringen.

En potensiell utfordring med undervisning i programmering er at det kan føre til stor kognitiv belastning for elever (Mayer & Moreno, 2003). Ved å kombinere undervisning om programmering med undervisning om et annet naturfaglig tema er vi forsiktig med at ikke læringsoppgavene skal føre til en for stor kognitiv belastning som en helhet. For å unngå stor kognitiv belastning har vi gjennomført noen grep med læringsressursen. Først og fremst benytter vi blokkprogrammering for å unngå syntaks kunnskap slik at

selve programmeringen blir mindre krevende (Qian & Lehman, 2017). Videre benytter vi PRIMM rammeverket (Sentance et al., 2019) for at nivået på oppgavene øker gradvis. Ved å hovedsakelig bruke et ferdig utarbeidet program reduseres elevenes kognitive belastning (Resnick et al., 2009). Oppgavene i elevheftene handler i størst grad om å gjøre endringer i programmet og ikke å lage ny kode. Ved at læreren samler elevene i diskusjon om naturlig seleksjon, vil det også være enklere for elever å lære om naturlig seleksjon. Ved å begrense elevenes kognitive prosessering gjennom undervisningen er målet meningsfull læring om både programmering og naturlig seleksjon (Mayer & Moreno, 2003).

#### 5.2.4 Har vi nådd målet om å utvikle en læringsressurs som imøtekommer utfordringer i skolen knyttet til kombinasjonen av programmering og naturlig seleksjon?

Balansen mellom å kombinere programmering og naturlig seleksjon er utfordrende, og derfor har vi ønsket å utvikle en læringsressurs som har fokus på denne kombinasjonen. Både programmering og naturlig seleksjon er temaer som elever har flere misoppfatninger knyttet til. Vi har derfor fokusert på hvordan vi skal kombinere disse faktorene, uten at det ene går utover den andre. Vi har utviklet læringsressursen i forsøk på å imøtekomme de utfordringene som vi har identifisert i tidligere forskning om programmering i naturfag, misoppfatninger i programmering og naturlig seleksjon. Fra gjennomføringen av læringsressursen har vi positive erfaringer angående bruk i skolen. Likevel har vi ikke andre undersøkelser enn egne observasjoner og utsagn fra læreren, som tilsier om de grepene vi har gjort har hatt effekt. Derfor må det gjennomføres videre undersøkelser for å undersøke om læringsressursen imøtekommer utfordringer i skolen knyttet til kombinasjonen av programmering og naturlig seleksjon.

### 5.3 Studiens begrensninger

I denne delen vil vi beskrive et utvalg av begrensningene i vår studie. Først vil vi beskrive vår rolle som forskere og hvordan dette kan påvirke våre resultater. Videre beskrives begrensninger som har oppstått grunnet tidsrammen til studien, og til slutt vil vi reflektere rundt utvalget i studien.

#### 5.3.1 Vår rolle som forskere og utviklere

Pedagogisk designforskning som metode har blitt kritisert for å være forskerstyrt (Bjørndal, 2013). I utviklingen av vår læringsressurs har vår rolle som forskere preget både valg av tema, teori og utvikling. Dette kan føre til en begrensning i form av at forskeren får lite innspill fra andre. For å sikre troverdighet i forskningen har vi i utviklingen av læringsressursen tatt utgangspunkt i litteratur som bygger opp om valgene som er gjort. Ved å gjennomføre en pilot og en kvalitetsrådgivning sikrer vi mer troverdighet. Ytterligere har vi fått utprøvd læringsressursen gjennom to sykluser. På denne måten har vi fått erfaringer fra to ulike klasser, som kan være med på å styrke troverdigheten. Likevel hadde det vært ønskelig å få gjennomført en større pilot, flere kvalitetsrådgivninger og flere sykluser for videreutvikling av læringsressursen. Det ville vært interessant å se gjennomføringen av undervisningen i andre klasser og med andre lærere.

### 5.3.2 Begrensninger som følge av tid

Bjørndal (2013) beskriver at målsettingen til pedagogisk designforskning både består av utvikling av pedagogisk praksis og et pedagogisk bidrag ved å utvikle en lokal læringsteori. Grunnet tidsrammen har vi måtte begrense designforskningen til å hovedsakelig utvikle en læringsressurs og ikke det å utvikle en lokal læringsteori. Tidsbegrensningen vil også begrense utviklingsprosessen av læringsressursen til å kun bestå av to sykluser med utprøving. Dette har begrenset vårt erfaringsgrunnlag og dermed begrenset muligheten for ytterligere videreutvikling. Dersom vi hadde hatt mer tid til rådighet, ville vi gjennomført en tredjesyklus på mellomtrinnet. Det hadde også vært en fordel og innhentet innspill, særlig om lærerveiledningen, fra flere lærere med ulik kompetanse innenfor programmering. Men grunnet tidsmessige årsaker har vi ikke hatt mulighet til å gjennomføre dette.

### 5.3.3 Refleksjoner rundt utvalg

Pedagogisk designforskning kan være begrenset med tanke på generalisering, på grunn av utvalgets størrelse ofte er lite (Bjørndal, 2013). I vår studie er utvalget lite og bestående av en lærer og to klasser, og dermed er det begrensede muligheter for å generalisere om læringsressursen fungerer på samme måte i andre klasser. Ytterligere er det utfordrende å vite om kunnskapsnivået innen programmering i vårt utvalg stemmer overens med dagens kunnskapsnivå i skolen. Utvalget kan dermed ha hatt betydning for hvilken retning læringsressursen har utviklet seg i.

## 6. Forslag til videreutvikling av læringsressursen

Et kjennetegn på designeksperiment er at utviklingen av et nytt design skjer i en gjentakende prosess som består av sykluser av design, testing, analyser og refleksjon (Wæge, 2007; Bjørndal, 2013). Ettersom studien vår har en begrensning som følge av tid har vi valgt å inkludere et kapittel i masteroppgaven som beskriver noen muligheter for videreutvikling av læringsressursen.

I masterprosjektet har læringsressursen blitt utprøvd av en lærer i to niende klasser. Dette har gitt oss erfaringer rundt nivået til en niendeklasse, men dette har også begrenset våre erfaringer for hvordan undervisningen fungerer for andre klassetrinn. En videreutvikling kan være å teste opplegget i flere klassetrinn med ulike lærere. Dette ville både gitt oss flere erfaringer rundt opplegget og ny erfaring på hvordan ulike klassetrinn responderer på opplegget, slik at læringsressursen kan tilpasses både mellomtrinnet og for naturfag på videregående.

Dersom vi skulle gjennomført flere utprøvinger hadde en mulighet vært og fokusert på å måle læringsutbyttet av undervisningen både innen kunnskap om naturlig seleksjon og programmering. Læringsutbytte kan måles ved å intervjuere elever eller ved å teste før- og etterkunnskap. Da kan forskeren få en større pekepinn på om læringsressursen mestrer kunsten å kombinere programmering og naturlig seleksjon, uten at det faglige fokuset reduseres. Samtidig som vi kunne hatt mulighet til å kartlegge elevenes misoppfatninger før og etter bruk av læringsressursen, og dermed vurdere om programmering i denne situasjonen kan brukes for å unngå og oppklare misforståelser om naturlig seleksjon.

Det vil også være hensiktsmessig å videreutvikle simuleringen *Naturlig seleksjon*. For å illustrere naturlig seleksjon bedre i simuleringen vil det være en fordel å inkludere flere seleksjonspress. Slike seleksjonspress kan være flere næringskilder, gjemteplasser, sykdommer og tydeligere konkurranse innad og mellom artene. Slik simuleringen er designet nå er næringstilgang faktoren mangelfull. Nå fungerer den slik at om det er mindre enn 20 gress totalt i programmet vil simuleringen stoppe.

Vår læringsressurs tar utgangspunkt i blokkprogrammeringsprogrammet Scratch, men det finnes også muligheter for å videreutvikle læringsressursen til å også inkludere andre programmeringsspråk. En mulighet er å inkludere tekstprogrammering for noen eller alle oppgavene. Et prosjekt som er planlagt av forelesere ved naturfagsseksjonen på institutt for lærerutdanning ved NTNU skal videreutvikle vår didaktiske ressurs med å ta i bruk programmet Micro:bit. Dette innebærer at elevene er aktive som harene eller reven og benytter avstandsdetektorer på mikrobitene for å illustrere kamuflasje, reproduksjon og overlevelse.

## 7. Konklusjon

Til tross for innføringen av programmering i LK20 (Statped, 2021) viser undersøkelser at lærere mangler kompetanse i programmering og at dette preger naturfagundervisningen (Berggren & Jom, 2019). I tillegg beskriver studier at det er utfordrende å koble programmering til andre faglige temaer (Bungum et al., 2014; Caballero et al, 2014). Det er da også få eksempler på læringsaktiviteter som kombinerer programmering og arbeid med andre temaer i naturfag.

Med dette i bakgrunn har det overordnede målet med studien vært å *utvikle en læringsressurs som bruker programmering til å utforske naturlige fenomener*. Vi erfarte at det var flere potensielle gevinster med å kombinere arbeidet med programmering med et annet tema i naturfag. Det å bruke programmering i form av en interaktiv simulering ga en god inngang til diskusjon både om et naturlig fenomen og programmering. Ved å ta utgangspunkt i en simulering fikk elevene også visualisert et naturlig fenomen. I tillegg erfarte vi at kombinasjonen av programmering og et annet naturfaglig tema er en variasjon til den tradisjonelle undervisningen, som dermed gir elevene motivasjon. En annen gevinst med å kombinere temaene er at denne undervisningsmåten kan bidra til mindre tidspress, ettersom man jobber samtidig med to temaer og flere kompetansemål samtidig. I tillegg er en potensiell gevinst at elevene får erfaring med programmering og kan se hvordan programmering kan gripe inn i det meste i samfunnet.

Vi ønsket å utvikle en læringsressurs som kombinerer arbeid med programmering og naturlig seleksjon. Derfor har vi utviklet og utprøvd en læringsressurs som består av to simuleringer av naturlig seleksjon, to nivådelte oppgavehefter og en lærerveiledning (se kapittel 4, s.16). Gjennom utprøvingene har vi indikasjoner på at arbeid med læringsressursen, og dermed kombinasjonen av programmering og naturlig seleksjon, er motiverende og kan bidra til at elevene får større forståelse for evolusjon. Vi har også indikasjoner på at læringsressursen er tilpasset skolen og imøtekommer utfordringer som er knyttet til en kombinasjon av programmering og naturlig seleksjon.

På tross av indikasjonene om at arbeid med læringsressursen gir økt motivasjon, læringsutbytte og imøtekommer utfordringene med de aktuelle temaene i skolen, må det likevel gjennomføres egne studier for å konkludere dette. Vi anbefaler i tillegg at det utvikles tilsvarende opplegg og forskes på kombinasjonen av programmering med andre temaer enn seleksjon. Vi anbefaler dermed forskning om hvordan programmering kan brukes for å oppklare og forhindre misoppfatninger i andre temaer.

## 8. Litteraturliste

- Berggren, S & Jom, P. (2019). Fagartikkel: Lærerne er positive til programmering - men mangler kunnskap. *Utdanningsnytt*. <https://www.utdanningsnytt.no/fagartikkel/fagartikkel-laererne-er-positive-til-programmering---men-mangler-kunnskap/220753>
- Bungum, B., Esjeholm, B.-T., & Lysne, D. A. (2014). Science and Mathematics as part of practical projects in technology and design: An analysis of challenges in realising the curriculum in Norwegian schools. *Nordina : Nordic Studies in Science Education*, 10(1), 3–15. <https://doi.org/10.5617/nordina.547>
- 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*. En innføring i forskningsarbeid i skolen, 245-257. 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>
- Dalland, C. P., Bjørnstad, E. & Andersson-Bakken, E. (2021). Observasjon som metode i barnehage- og klasseromsforskning. I E. Andersson-Bakken & C. P. Dalland (Red.), *Metoder i klasseromsforskning- forskningsdesign, datainnsamling og analyse*. Universitetsforlaget.
- Dewey, J. (1996). Erfaring og tenkning. I E. L. Dale (Red.), *Skolens undervisning og barnets utvikling: klassiske tekster*, 53-66. Gyldendal.
- Frafjord, K. (2021, 21.desember). Harer. I *Store norske leksikon*. <https://snl.no/harer>
- German, S. (2019). Computational thinking. *Science Scope*, 42(9), 36-39. <https://www.jstor.org/stable/26899029>
- Gilbert, J. K., Boulter, C. J., & Elmer, R. (2000). Positioning models in science education and in design and technology education. *Developing models in science education*, 3-17. [https://doi.org/10.1007/978-94-010-0876-1\\_1](https://doi.org/10.1007/978-94-010-0876-1_1)
- Gjøvik, Ø., & Høyland, J. (2022). *Kloss for kloss: blokkprogrammering for lærere*. Universitetsforlaget.
- Gravemeijer, K. & Cobb, P. (2006). Design research from a learning design perspective. I van den Akker, K. Gravemeijer, S. McKenney & N. Niveen (Red.), *Educational design research*, 17-51. Routledge. <https://doi.org/10.4324/9780203088364>
- Gregory, T. R. (2009). Understanding natural selection: essential concepts and common misconceptions. *Evolution: Education and outreach*, 2(2), 156-175. <https://doi.org/10.1007/s12052-009-0128-1>
- Grover, S. & Basu, S. (2017). Measuring Student Learning in Introductory Block-Based Programming: Examining Misconceptions of Loops, Variables, and Boolean Logic. I *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 267–272. <https://doi.org/10.1145/3017680.3017723>
- Grønmo, S. (2016). *Samfunnsvitenskapelige metoder* (2. utg.). Fagbokforl.



- Hannisdal, M., & Ringnes, V. (2015). *Kjemi for lærere: naturfag i grunnskolelærerutdanningen 5-10.trinn*. Gyldendal.
- Haraldsrud, A. D., Sveinsson, H. A., & Løvold, H. H. (2020). *Programmering i skolen*. Universitetsforlaget.
- Harms, U., & Reiss, M. J. (2019). The present status of evolution education. *Evolution education re-considered: Understanding what works*, 1-19. [https://doi.org/10.1007/978-3-030-14698-6\\_1](https://doi.org/10.1007/978-3-030-14698-6_1)
- Jaakkola, T. & Nurmi, S. (2008). Fostering elementary school students' understanding of simple electricity by combining simulation and laboratory activities. *Journal of Computer Assisted Learning*, 24(4), 271-283. <https://doi.org/10.1111/j.1365-2729.2007.00259.x>
- Lai, C. S., & Lai, M. H. (2012). Using computer programming to enhance science learning for 5th graders in Taipei. I *2012 International Symposium on Computer, Consumer and Control*, 146-148. <https://doi.org/10.1109/IS3C.2012.45>
- Laughlin, P. R., Hatch, E. C., Silver, J. S., & Boh, L. (2006). Groups perform better than the best individuals on letters-to-numbers problems: Effects of group size. *Journal of Personality and Social Psychology*, 90(4), 644-651. <https://doi.org/10.1037/0022-3514.90.4.644>
- Knain, E., Bjønes, B. & Kolstø, S. D. (2019). Rammer og støttestrukturer i utforskende arbeidsmåter. I Knain, E., & Kolstø, S. D. (Red.), *Elever som forskere i naturfag* (2. Utgave), 70-102. Universitetsforlaget.
- Kunnskapsdepartementet. (2019). *Læreplan i naturfag (NAT01-04)*. Fastsett som forskrift. Læreplanverket for Kunnskapsløftet 2020. <https://www.udir.no/lk20/nat01-04>
- Mayer, R. E., & Moreno, R. (2003). Nine ways to reduce cognitive load in multimedia learning. *Educational psychologist*, 38(1), 43-52. [https://doi.org/10.1207/S15326985EP3801\\_6](https://doi.org/10.1207/S15326985EP3801_6)
- Moen, T. (2015). Vygotsky i teori og praksis. I R. Karlsdottir & I. D. Hybertsen (Red.), *Læring, utvikling, læringsmiljø. En innføring i pedagogisk psykologi*, 251-265. Fagbokforlaget.
- NESH. (2021). *Forskningsetiske retningslinjer for samfunnsvitenskap og humaniora, juss og teologi*. Hentet 14.november 2022 fra <https://www.forskningsetikk.no/retningslinjer/hum-sam/forskningsetiske-retningslinjer-for-samfunnsvitenskap-og-humaniora/>
- Osborne, J. (2014). Teaching scientific practices: Meeting the challenge of change. *Journal of Science Teacher Education*, 25(2), 177-196. <https://doi.org/10.1007/s10972-014-9384-1>
- PhET Interactive Simulations (u.å.). *Natural Selection*. Hentet 24. Januar 2023 fra <https://phet.colorado.edu/en/simulations/natural-selection/about>
- Qian, Y., & Lehman, J. (2017). Students' misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education (TOCE)*, 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.  
<http://dx.doi.org/10.1145/1592761.1592779>
- Reydon, T. A. (2021). Misconceptions, conceptual pluralism, and conceptual toolkits: bringing the philosophy of science to the teaching of evolution. *European Journal for Philosophy of Science*, 11(2), 48.  
<https://doi.org/10.1007/s13194-021-00363-8>
- Ringdal. (2013). *Enheter og mangfold: samfunnsvitenskapelig forskning og kvantitativ metode* (3. Utg.). Fagbokforlaget.
- Skaalvik, E. M., & Skaalvik, Sidsel. (2015). *Motivasjon for læring: teori og praksis*. Universitetsforl.
- Sandoval, W. (2014). Conjecture mapping: An approach to systematic educational design research. *Journal of the learning sciences*, 23(1), 18-36.  
<https://doi.org/10.1080/10508406.2013.778204>
- Sentance, S., Waite, J., & Kallia, M. (2019). Teaching computer programming with PRIMM: a sociocultural perspective. *Computer Science Education*, 29(2-3), 136-176. <https://doi.org/10.1080/08993408.2019.1608781>
- Sentance, S., & Waite, J. (2017). PRIMM: Exploring pedagogical approaches for teaching text-based programming in school. I *Proceedings of the 12th Workshop on Primary and Secondary Computing Education: WIPSC 17*, 113-114.  
<https://doi.org/10.1145/3137065.3137084>
- Statped. (2021). *Programmering*. Statelig spesialpedagogisk tjeneste. Hentet 10.oktober 2022 fra <https://www.statped.no/laringsressurser/teknologitema/programmering-for-barn-med-saerskilte-behov/programmering/programmering-i->
- Stenseth, N.C. (2020, 28.desember). Kamouflasje (biologi). I *Store norske leksikon*.  
[https://snl.no/kamuflasje\\_-\\_biologi](https://snl.no/kamuflasje_-_biologi)
- Utdanningsdirektoratet. (2021). *Elever med stort læringspotensial*. Hentet 1.mai 2023 fra <https://www.udir.no/laring-og-trivsel/tilpasset-opplaring/elever-med-stort-laringspotensial/>
- Van den Akker, J., Gravmeijer, K., McKennedy, S. & Nieveen, N. (2006). Introducing educational design research. I J. Van den Akker, K. Gravmeijer, S. McKennedy & N. Niveen (Red.), *Educational design research*, 3-7. Routledge.
- Wang, T.-L., & Tseng, Y.-K. (2018). The Comparative Effectiveness of Physical, Virtual, and Virtual-Physical Manipulatives on Third-Grade Students' Science Achievement and Conceptual Understanding of Evaporation and Condensation. *International Journal of Science and Mathematics Education*, 16(2), 203-219.  
<https://doi.org/10.1007/s10763-016-9774-2>
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of science education and technology*, 25, 127-147.  
<https://doi.org/10.1007/s10956-015-9581-5>

Wæge, K. (2007). *Elevenes motivasjon for å lære matematikk og undersøkende matematikkundervisning* (Doktoravhandling, Norges teknisk-naturvitenskapelige universitet, Nasjonalt senter for matematikk i opplæringen, Fakultet for informasjonsteknologi, matematikk og elektronikk, Institutt for matematiske fag). Norges teknisk-naturvitenskapelige universitet.

Zacharias, Z. C., Olympiou, G. & Papaevripidou, M. (2008). Effects of experimenting with physical and virtual manipulatives on students' conceptual understanding in heat and temperature. *Journal of Research in Science Teaching*, 45(9), 1021–1035.  
<https://doi.org/10.1002/tea.20260>

## Vedlegg

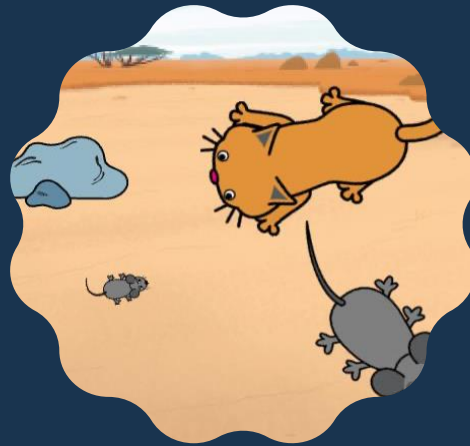
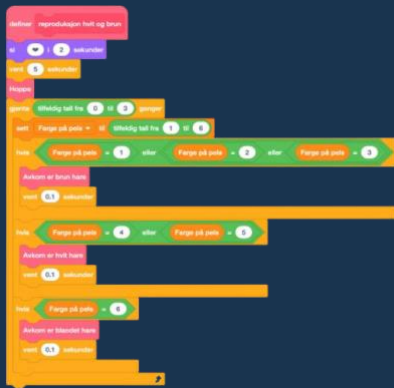
Vedlegg 1: Oppgaveheftet *Tilpasninger og overlevelse*

Vedlegg 2: Oppgaveheftet *Naturlig seleksjon*

Vedlegg 3: Lærerveiledning

Vedlegg 4: Prosessdokument

# TILPASNINGER OG OVERLEVELSE OPPGAVEHEFTE



INNEHOLDER OPPGAVER  
MED TIPS TIL  
PROGRAMMERINGEN



Katarina Sørhøy og  
Rikke-Helen Aakernes Nielsen



Simuleringen er laget i blokkprogrammeringsspråket Scratch. I løpet av denne undervisningen skal du lære å forstå, endre og lage koder i Scratch. Oppgavene handler om naturlig seleksjon og hvilke tilpasninger som hjelper dyr for å overleve i naturen.

**Læringsmålet er å:**

- Bruke en simulering for å utforske samspill i naturen.
- Kunne bruke programmering til å gjøre endringer i en simulering.
- Vurderer styrke og svakheter i en modell.

Til de oppgavene som kan være vanskelige har vi laget noen tips til hjelp. Da trykker du på linken hvor det står [trykk her for tips](#).

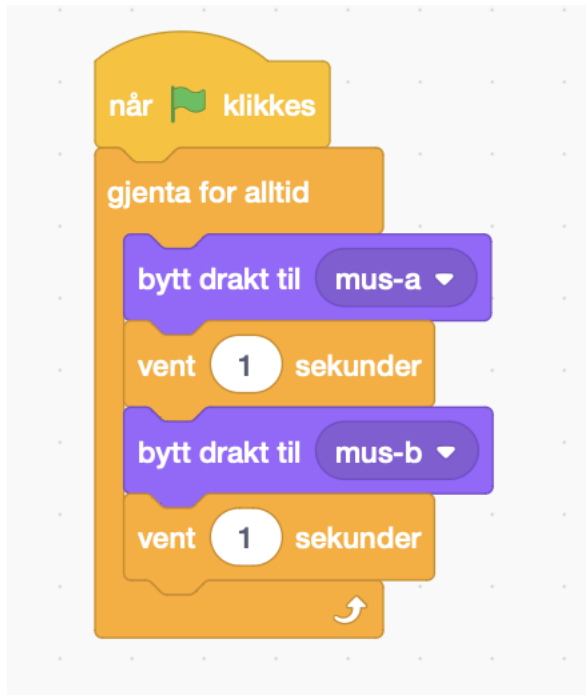
Link til prosjektet: <https://scratch.mit.edu/projects/815613357>

For å se og bruke simuleringen må du trykke på lenkene. Når du åpner lenken til simuleringen kommer du inn til prosjektsiden, hvor man kan lese om hvordan man bruker prosjektet. For å se simuleringen trykk: på *grønt flagg*, og for å gjøre endringer i kodene trykk: *se inni*.

### Oppgave 1: FORUTSE OG KJØR

Hva tror du denne koden gjør?

**Tips:** se på draktene på bildet under.



Mus-a



Mus-b



## Oppgave 2: FORUTSE OG KJØR

Hva tror du denne koden gjør?

The image shows two Scratch code snippets and a cartoon illustration. The left snippet starts with a 'when clicked' event, followed by a 'go to random place' block, and then a 'repeat forever' loop containing 'move 2 steps' and 'bounce off edges' blocks. The right snippet starts with a 'when clicked' event, followed by a 'repeat forever' loop containing an 'if touches mouse?' block and a 'say Mat for 2 seconds' block. Below the code is a cartoon illustration of a savanna landscape with acacia trees, a large grey mouse, a small grey mouse, and a brown cat.

### Oppgave 3: UNDERSØK

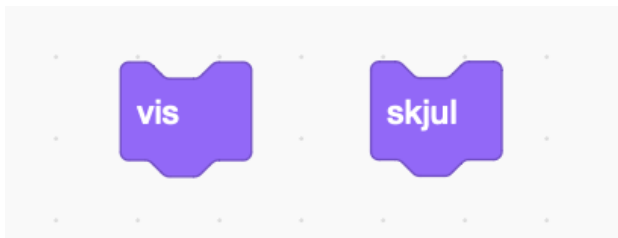
Test disse klossene i Scratch og finn ut hva de gjør.

Link til prosjektet: <https://scratch.mit.edu/projects/815613357>

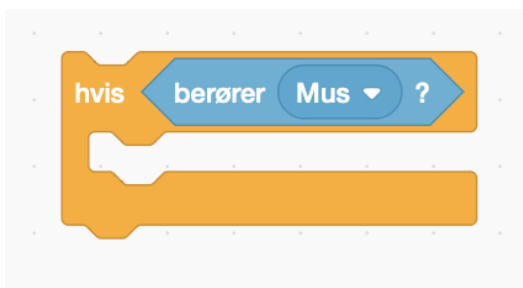
- A) Hvorfor bruker vi *gjenta for alltid*-klossen rundt gå-sekvensen. Hva ville skjedd uten?



- B) Hva kan vi bruke vis- og skjul-klossene til?



- C) Undersøk hva denne klossen gjør. Hva vil skje med det som er inne i denne klossen?



### Oppgave 4: Endre

Link til prosjektet: <https://scratch.mit.edu/projects/815613357>

- A) Endre miljøet slik at musen blir vanskeligere å oppdage for katten.  
B) Endre utseende til musen slik at det ville overlevd lengre i naturen.

For tips til oppgavene [trykk her](#)



### **Oppgave 5: LAG**

Start et nytt prosjekt i Scratch og lag et spill hvor du kan styre musen vekk fra katten som prøver å spise den. Tenk på hvilke egenskaper som musen bør å ha, for å unngå å bli spist.

For deg som synes programmering er litt vanskelig og ønsker starthjelp, [trykk her](#)

### **Oppgave 6: VURDER**

**Vurder om programmet du har jobbet med er en god modell på tilpasninger og overlevelse i naturen.**

Diskuter hvilke deler av programmet som er en modell av virkeligheten og hvilke deler av programmet som ikke er en god modell av virkeligheten.

Gi programmet en karakter fra 1 til 6. Hva ville du gjort annerledes som du skulle lage en modell av tilpasninger og overlevelse?

### **Oppgave 7: DISKUTER**

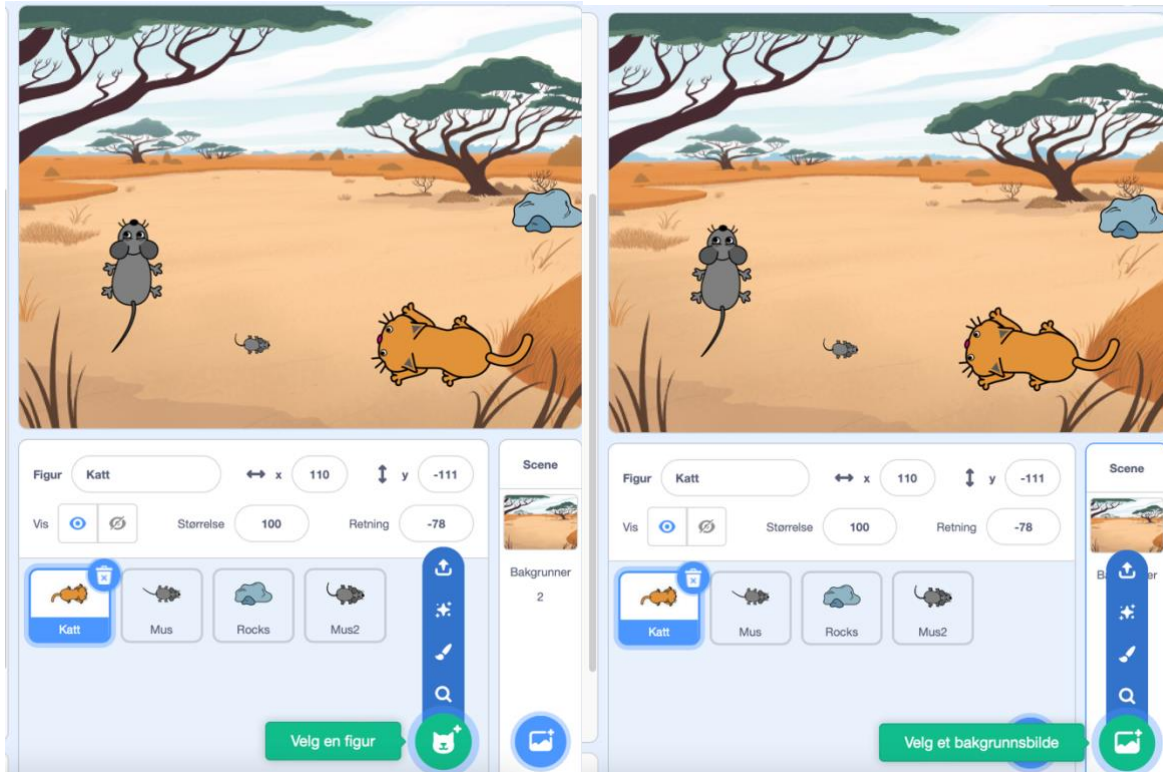
Snakk med læringspartner og diskuter i fellesskap:

- A) Hvilke egenskaper hjelper katten for å jakte byttedyr?
- B) Hvilke egenskaper hjelper musene til å unnsnippe rovdyr? Om 100 år hvilke egenskaper tror du at musene har arvet?
- C) Hva betyr *survival of the fittest*?

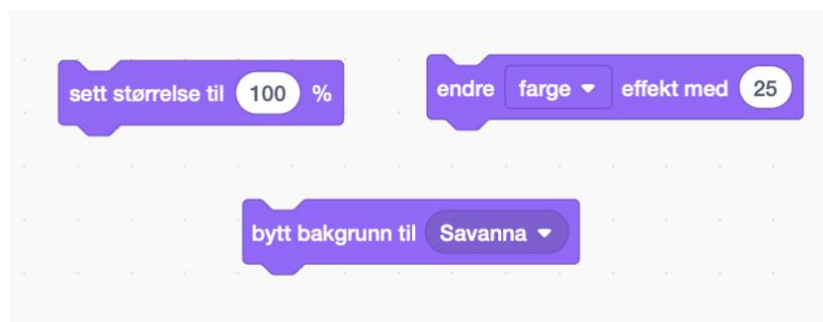
## Tips til å komme i gang med programmeringen

### Oppgave 4

- A) For å endre miljøet slik at musen blir vanskeligere å oppdage, kan man legge til nye figurer som musen kan gjemme seg bak eller endre bakgrunnsbildet sånn at musen blir vanskeligere å se for katten.



- B) Når du skal endre utseende til musen slik at den overlever lengre i naturen kan du enten endre størrelsen, fargen eller miljøet. Kanskje man også kan endre alle disse faktorene for at musen skal bli mindre synlig for katten?



### Oppgave 5

For deg som synes programmering er litt utfordrende og ønsker starthjelp, her har vi allerede startet på et spill som du kan gjøre ferdig. Trykk på linken under og lag ferdig spillet. I programmet ligger mange av klossene som du trenger for å kunne bruke pilene til å styre musen unna katten.

<https://scratch.mit.edu/projects/803269298>

# NATURLIG SELEKSJON OPPGAVEHEFTE



INNEHOLDER OPPGAVER  
MED TIPS TIL  
PROGRAMMERINGEN



KATARINA SØRHØY OG  
RIKKE-HELEN AAKERNES NIELSEN



Dette er et arbeidshefte som er knyttet til en simulering om temaet naturlig seleksjon. Simuleringen er laget i blokkprogrammeringsspråket Scratch. I løpet av denne undervisningen skal du lære å forstå, endre og lage koder i Scratch. Oppgavene handler om naturlig seleksjon og hvordan samspillet er mellom dyrene rev og hare i den norske naturen.

**Læringsmålet er å:**

- Bruke en simulering for å utforske samspill i naturen.
- Kunne bruke programmering til å gjøre endringer i en simulering.
- Vurderer styrke og svakheter i en modell.

Til de oppgavene som kan være vanskelige har vi laget noen tips til hjelp. Da trykker du på linken hvor det står [trykk her for tips](#).

Undersøk i dette programmet: <https://scratch.mit.edu/projects/815547970>

For å se og bruke simuleringen må du trykke på lenkene. Når du åpner lenken til simuleringen kommer du inn til prosjektsiden, hvor man kan lese om hvordan man bruker prosjektet. For å se simuleringen trykk: på *grønt flagg*, og for å gjøre endringer i kodene trykk: *se inni*.

### Oppgave 1: VURDER

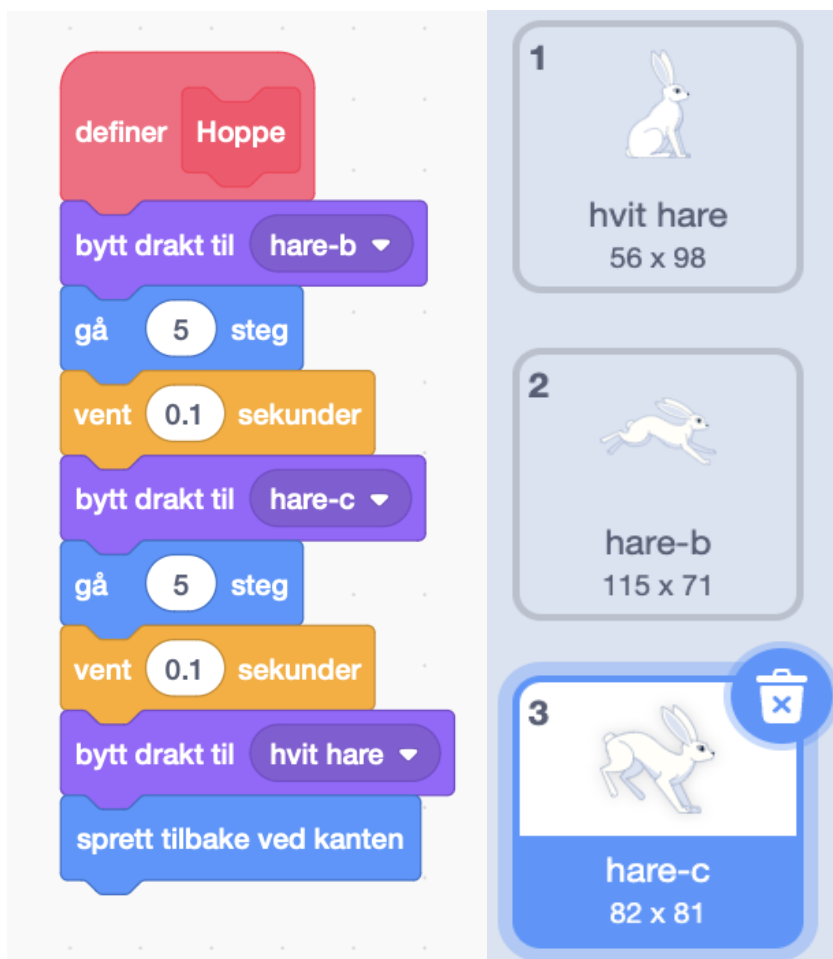
#### Vurder om simuleringen er en god modell på naturlig seleksjon.

Diskuter hvilke deler av simuleringen som er en modell av virkeligheten og hvilke deler av simuleringen som ikke er en god modell av virkeligheten.

Gi simuleringen en karakter fra 1 til 6. Hva ville du gjort annerledes som du skulle lage en simulering av naturlig seleksjon?

### Oppgave 2: FORUTSE OG KJØR

Hva tror dere denne koden gjør?



The image shows a Scratch script on the left and a sprite list on the right. The script consists of the following blocks:

- definer Hoppe (red block)
- bytt drakt til hare-b (purple block)
- gå 5 steg (blue block)
- vent 0.1 sekunder (orange block)
- bytt drakt til hare-c (purple block)
- gå 5 steg (blue block)
- vent 0.1 sekunder (orange block)
- bytt drakt til hvit hare (purple block)
- sprett tilbake ved kanten (blue block)

The sprite list on the right contains three items:

- 1. hvit hare (56 x 98) with a white rabbit icon.
- 2. hare-b (115 x 71) with a white rabbit icon.
- 3. hare-c (82 x 81) with a white rabbit icon and a trash can icon.

### Oppgave 3: FORUTSE OG KJØR

Hva tror dere denne koden gjør?

**Tips:** Hva tror dere vil skje hvis en hare og en rev møtes



### Oppgave 4: UNDERSØK

#### Naturlig utvalg

*I naturen vil de individene som er best tilpasset miljøforholdene de lever i ha større sannsynlighet for å overleve enn de som er mindre tilpasset. Det er dermed de best tilpassede individene som vil få barn og føre genene og dermed egenskapene sine videre til den neste generasjon. Dette kalles naturlig utvalg.*

*For at naturlig utvalg skal skje må egenskaper videreføres gjennom reproduksjon. Når to individer av ulikt kjønn møtes og reproducerer seg, vil de få avkom som arver noen av egenskapene til dem begge. Avkommet vil ligne mer på foreldrene enn andre individer i en populasjon. Individene i en populasjon er ikke identiske, men har varierende egenskaper grunnet genetiske forskjeller. To søsken vil heller ikke være identiske fordi de arver forskjellige egenskaper fra mor og far, unntaket er eneggede tvillinger.*



Noen viktige begreper:

- **Individ:** Er en enkelt hare. Den har unike genvarianter og kun eneggede tvillinger har like gener.
- **Populasjon:** Er en samling individer av en bestemt art som lever innenfor et område.
- **Gener:** er en oppskrift for egenskaper til et individ som ligger inne i cellekjernen til alle cellene i kroppen.
- **Egenskaper:** Gener, som er arvet fra mor og far, bestemmer egenskapene til et individ.
- **Avkom:** Er harenes unger. De arver egenskaper fra mor og far.
- **Kull:** Er hvor mange harer som fødes hver gang en hare er drektig.

Undersøk dette i programmet: <https://scratch.mit.edu/projects/815547970>

Dersom du står fast på noen av oppgavene, [trykk her](#) for å få tips.

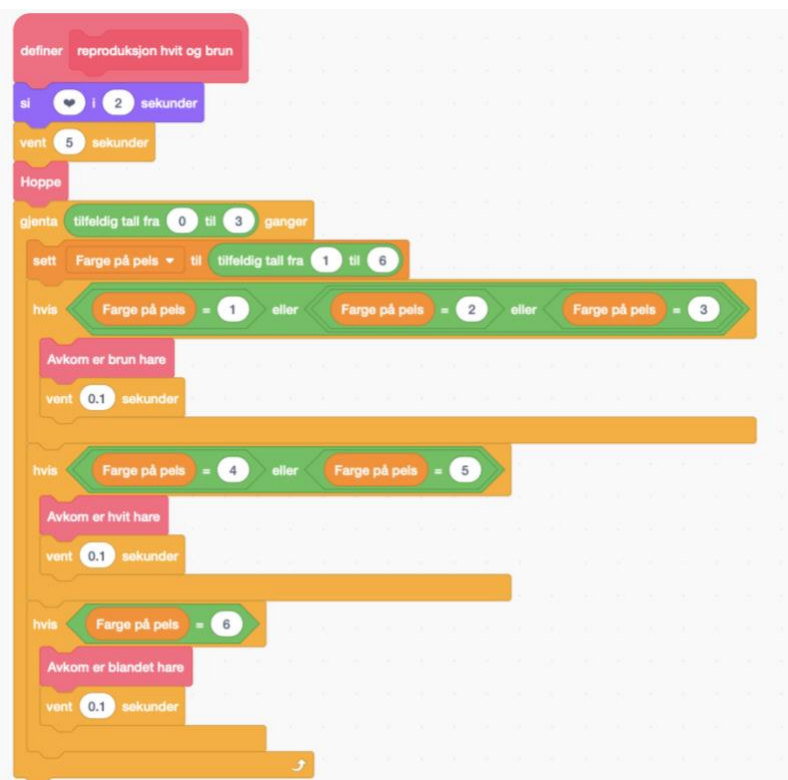
### A. Undersøk hva som vil skje om en brun og en hvit hare møtes?

Hvordan vil deres avkom se ut? Hvilken farge er det mest sannsynlig at avkommet får?

### B. Hvor mange avkom får harene i et kull?

*Gjenta: tilfeldig tall fra 0 til 3 ganger* er programmet for at harene får 0-3 avkom i hvert kull.

Stemmer det at haren får mellom 0-3 avkom i hvert kull? Undersøk på internett og endre slik det blir riktig.





## Overlevelse

For at individer skal ha mulighet til å reprodusere seg må de overleve til de blir voksne. Dette innebærer å skaffe mat og beskytte seg mot farer slik som rovdyr. Dette er harene i simuleringen også avhengig av. Derfor hjelper det at fargen på pelsen til harene ligner omgivelsene, dette kalles kamuflasje. Dette er en egenskap som er godt tilpasset miljøet og en fordel for individer å arve.

### C. Hvilken pelsfarge tror du reven ser best på avstand når det er sommer?

Gå inn i programmet og finn koden for at reven jakter og oppdager de ulike harene. Hva er forskjellen for at reven oppdager hvite, brune og blandingsharer?

Hva vil skje om årstidene endres og bakken er dekket av snø?

## Oppgave 5: ENDRE

Velg minst en av oppgavene under.

For tips til oppgavene [trykk her](#)

Gjør endringene i dette programmet: <https://scratch.mit.edu/projects/815547970>

**A) Det har vært tørke i Norge og derfor er det mindre gress i fjellet. Endre slik at det vokser mindre og sjeldnere gress i simuleringen. Hvordan påvirker dette harene og reven?**

**B) Det er en feil i programmet. Det vokser ikke gress om vinteren på områder med snø og is. Derfor må harene finne seg annen næring.**

Undersøk på internett hvilken næringskilde harene benytter seg av om vinteren. Velg riktig næringskilde blant figurene og lag et fungerende program med vinternæring inkludert.





**C) Det er januar i høyfjellet og harene har vinterpels. Men på grunn av klimaendringer er miljøet forandret, slik at bakken som tidligere var dekket av snø og is nå er smeltet. Hva vil skje med kamuflasjen til de hvite harene?**

Endre miljøet i simuleringen og finn ut av hva som skjer med harene i hvit pels når landskapet ikke lenger består av is og snø. Bytt bakgrunn til den som heter klimaendringer og observer hvilke harer som er lettest å observere?

Videre kan du endre hvordan reven jakter på det nye miljøet, ved å lage egne *oppdager* til det nye miljøet.

**D) Endre draktene til blandingssharen i programmet slik du tror den skal se ut. Blandingssharen er avkommet av en brun og en hvit hare.**

### **Oppgave 6: LAG**

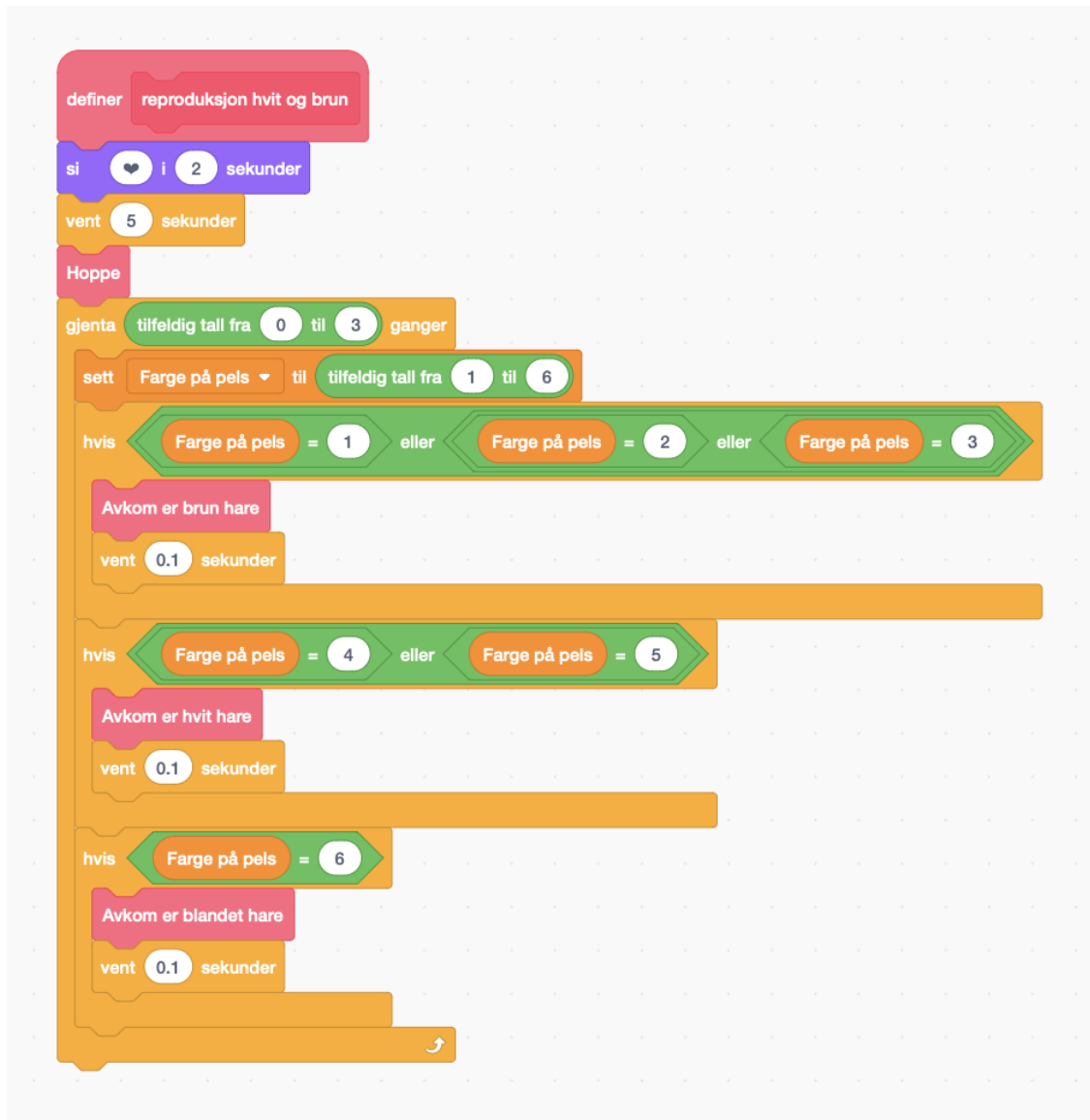
Om du er ferdig med oppgavene over og ønsker en utfordring ta kontakt med læreren din så får du en oppgave hvor du kan lage et program helt selv.

## Tips til å komme i gang med programmeringen

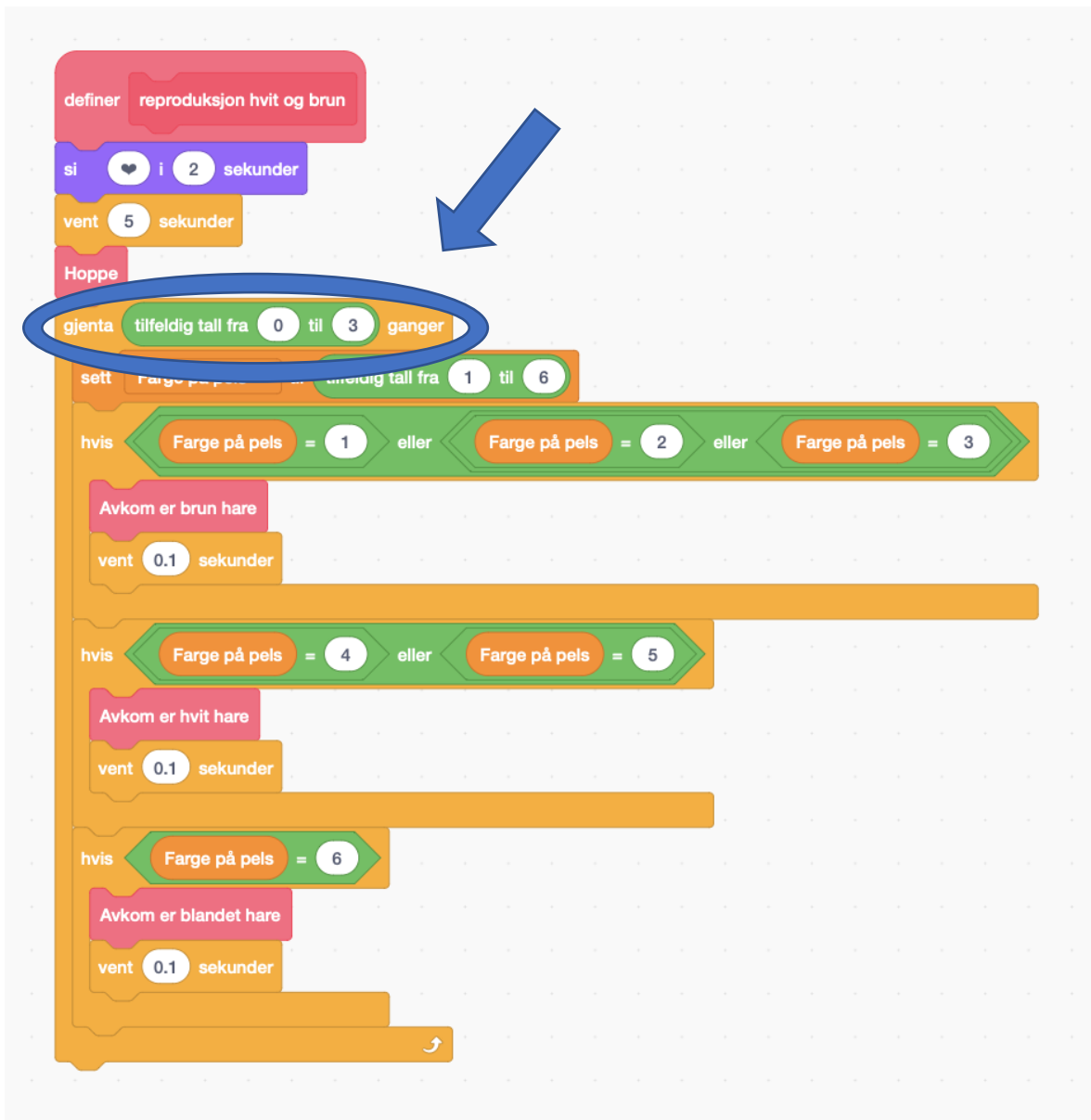
### Oppgave 4

A) Denne koden finner du både hos hvit og brun hunn harer. Først finnes det er kloss som sjekker om den hvite hunnharen møter en brun hannhare.

Hvilken farge avkommet får avgjøres av variabelen **Tilfeldig tall**. Tilfeldig tall kan være mellom 1 og 6. Hvis tallet blir 1, 2 eller 3 vil avkommet få hvit farge. Om tallet bli 4 eller 5 blir fargen brun og hvis tallet er 6 blir fargen blandet. Sannsynligheten for ulike farger kan du endre selv.

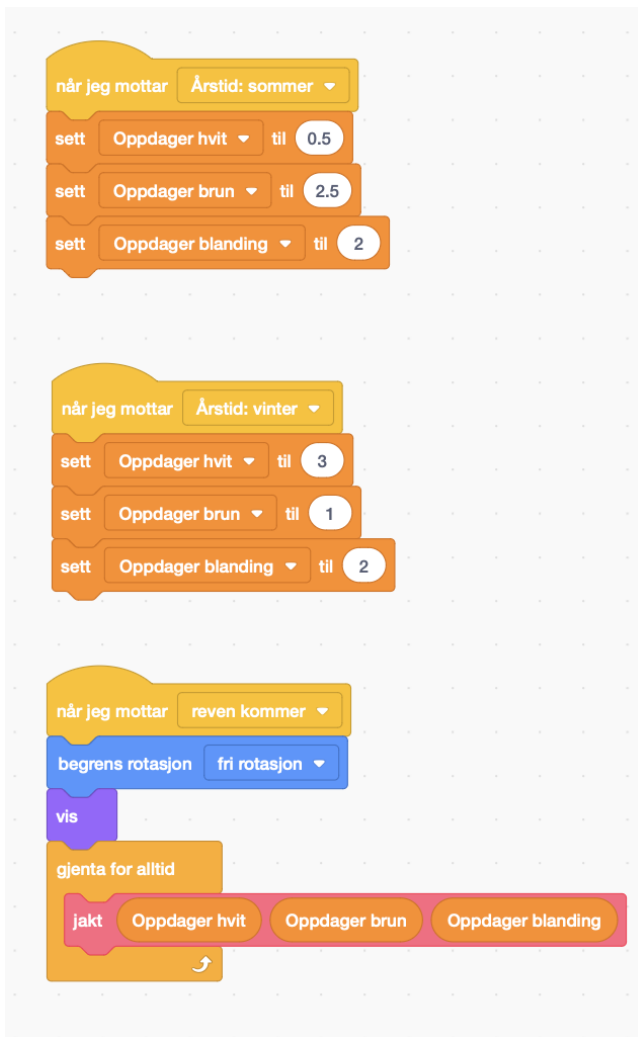


- B)** Det er blokken som er markert under som bestemmer hvor mange barn harene får i hvert kull.



- C)** Når årstidene endres, vil miljøet også endres. På sommeren er bakken bar og brunlig, mens på vinteren er bakken dekket av is og snø. Tenk hvilke farger som gir best kamuflasje i miljøet.

*Oppdager* avgjør hvor lett reven kan oppdage de ulike harene. Den er forskjellig til ulike årstider. Et lavt tall betyr at denne fargen på haren er lett å oppdage, mens et høyre tall betyr at fargen på haren er vanskeligere å oppdage.



## Oppgave 5

- A) Inne i koden til gresset kan man endre hvor raskt gresset vokser. Om man venter lengre vil gresset vokse sjeldnere. Man kan også endre størrelsen og fargen til gresset ved å bruke en av utseendeklossene.

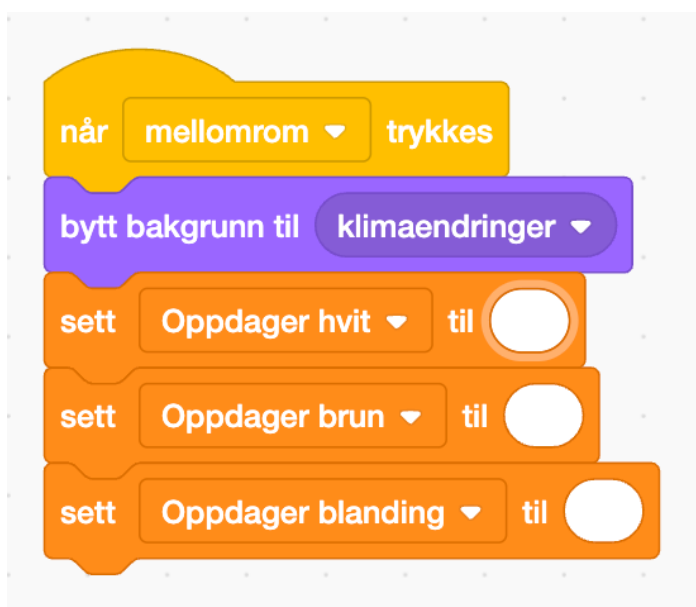


- B)** Ved å kopiere all koden som er lagt inn i gresset over til den riktige næringskilden kan man lett lage et program som fungerer. Husk også å endre at det er den nye næringskilden som berører harene, og ikke gresset.





- C)** Ved å dobbeltrykke på bakgrunnen kan man se alle bakgrunnene som er lagt inn. Her har vi lagt inn en ferdig en som heter "klimaendringer", bruk denne eller tegn/finn din egen bakgrunn.


Nedenfor er en mulig løsning til hvordan man kan endre miljø og hvordan reven oppdager harene med det nye miljøet.




- D)** Velg blandingssharen og gå inn på dens drakter. Her kan du utforske med de ulike tegneverktøyene og designe blandingssharen slik du ønsker. Husk å endre alle draktene nokså lik slik at det ser naturlig ut når haren hopper.



Kode **Drakter** Lyder

1    
hare-a2  
56 x 98




2   
hare-b2  
114 x 70



3   
hare-c2  
81 x 81


Drakt


Fyll  Kant 


---

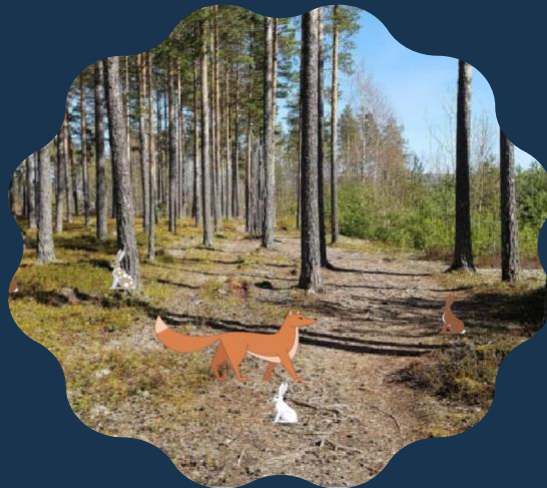
 

 **T**

 **O**



# NATURLIG SELEKSJON LÆRERVEILEDNING



INNEHOLDER  
UNDERSVINGSSOPPLEGG,  
SIMULERING, OPPGAVEHEFTE MED  
FORKLARINGER, TILPASSNINGER TIL  
ULIKE NIVÅER OG TIPS TIL  
PROGRAMMERINGEN



Katarina Sørhøy og  
Rikke-Helen Aakernes Nielsen



## Innholdsfortegnelse

Beskrivelse av lærerveiledning .....	67
Hvordan bruke læringsressursen? .....	67
To nivåer på simuleringene og oppgavene .....	68
Hvordan jobbe med simuleringene og oppgavene? .....	68
Utstyrliste .....	68
Forklaring til oppgaveheftene .....	68
Hvordan tilpasse nivået i undervisningen? .....	70
Simuleringen.....	72
Forklaring av konstanter, variabler og lister .....	72
Forklaring til programmeringen.....	79
PRIMM .....	86
Evolusjon .....	87
Vanskelige begreper i evolusjon.....	87
Survival of the fittest .....	87
Scratch .....	88
Vedlegg.....	92
Lag ditt eget program .....	92
Litteraturliste .....	93



## Beskrivelse av lærerveiledning

Lærerveiledningen er en del av en digital læringsressurs som tilrettelegger for undervisning som kombinerer arbeid med programmering og naturlig seleksjon. Den digitale læringsressursen består av to simuleringer, to oppgavehefter og denne lærerveiledningen.

[Simuleringen Naturlig seleksjon](#)

[Oppgavehefte Naturlig seleksjon](#)

[Simuleringen Tilpasninger og overlevelse](#)

[Oppgavehefte Tilpasninger og overlevelse](#)

### Læringsmålet er å:

- Bruke en simulering for å utforske samspill i naturen.
- Kunne bruke programmering til å gjøre endringer i en simulering.
- Vurderer styrke og svakheter i en modell.

Lærerveiledningen har et omfattende innhold, for å være bruker vennlig for naturfagslærere som har litt programmeringsbakgrunn fra tidligere. Som lærer velger du derfor ut hvilke deler av lærerveiledningen du leser før du bruker læringsressursen.

### Hvordan bruke læringsressursen?

Læringsressursen baserer seg på relevante kompetansemål som er hentet fra læreplanen i naturfag fra 2020 (Kunnskapsdepartementet, 2019).

#### Etter 7. trinn:

- Bruke og vurdere modeller som representerer fenomener man ikke kan observere direkte, og gjøre rede for hvorfor det brukes modeller i naturfag.
- Utforske, lage og programmere teknologiske systemer som består av deler som virker sammen.
- Utforske og beskrive ulike næringsnett og bruke dette til å diskutere samspill i naturen

#### Etter 10 trinn:

- Bruke og lage modeller for å forutsi eller beskrive naturlige prosesser og systemer og gjøre rede for modellenes styrker og begrensninger
- Bruke programmering til å utforske naturfaglige fenomener.
- Beskrive hvordan forskere har kommet fram til evolusjonsteorien og bruke denne til å forklare utvikling av biologisk mangfold.

### To nivåer på simuleringene og oppgavene

Ved at det er laget to simuleringer og to elevhefter som begge bruker programmering til å utforske naturlig seleksjon, kan undervisningen både gjennomføres på mellomtrinnet og på ungdomsskolen. Dette krever at læreren bruker de ulike oppgaveheftene for å justere nivå og tidsbruk. Det anbefales å minst sette av en dobbeltime til hver av oppgaveheftene, og om begge heftene tas i bruk anbefales det å bruke flere undervisningstimer.

Den ene simuleringen og oppgaveheftet *Tilpasninger og overlevelse* har en enkel vanskelighetsgrad både faglig og programmeringsteknisk, mens den andre simuleringen og oppgaveheftet *Naturlig utvalg* og har en mer krevende vanskelighetsgrad både faglig og programmeringsteknisk. *Tilpasninger og overlevelse* gjør det lett for elevene å utforske i programmeringsspråket Scratch på egenhånd. Temaet for elevheftet og simuleringen er tilpasninger og overlevelse i naturen. Simuleringen og oppgaveheftet kan benyttes som en egen undervisning eller være en oppvarming til neste hefte som er mer krevende programmeringsteknisk. *Naturlig utvalg* er et elevhefte og en simulering som består av noe mer krevende programmering i Scratch. Simuleringen belyser arv, fordelaktige egenskaper og overlevelse over tid. Denne simuleringen og oppgaveheftet kan brukes som en egen undervisningstime eller gjennomføres etter *Tilpasninger og overlevelse*.

Om undervisningen ikke gjennomføres som en del av tema om evolusjon anbefales det en kort gjennomgang om evolusjon, naturlig seleksjon og begrepet *survival of the fittest* før de begynner med oppgavene. Den gjennomgangen kan gjerne gjennomføres ved hjelp av simuleringene uten å vise noe om koding.

### Hvordan jobbe med simuleringene og oppgavene?

Det anbefales at elevene arbeider i grupper på to med elevheftet. Dette gruppeantallet er anbefalt ettersom programmering ikke er en tilskuersport, samtidig ønsker vi at elevene skal samarbeide og diskutere spørsmål knyttet til naturlig seleksjon og programmering. Det en fordel om alle på gruppen har tilgang til hver sin datamaskin hvor de kan prøve og feile på programmeringen.

### Utstyrliste

- PC (en til hver elev er ideelt)
- Tilgang til Elevhefte *Tilpasninger og overlevelse*
- Tilgang til Elevhefte *Naturlig seleksjon*
  - o Ekstraoppgave til Elevhefte naturlig seleksjon
- Lærerveiledning

## Forklaring til oppgaveheftene

Læringsressursen inkluderer to oppgavehefter som elevene skal jobbe med.

Opgaveheftet er designet etter undervisningsmetoden PRIMM, og følger de fem stegene *forutse- kjør - undersøk - endre - lag*. I heftet er det markert hvilken oppgave som hører til hvilken kategori. I tillegg til å ta utgangspunkt PRIMM sine fem steg, har vi valgt å inkludere stegene *diskuter* og *vurder*.

### [Oppgavehefte Naturlig seleksjon](#)

### [Oppgavehefte Tilpasninger og overlevelse](#)

*Forutse- og kjør* oppgavene anbefales å diskuteres i fellesskap i klassen. Oppgavene kan gjennomføres ved *tenk-par-del*, hvor elevene først skal tenke individuelt, så diskuteres det i par, før de deler sine tanker med klassen. På denne måten får læreren forklart viktige programmeringsfunksjoner som løkker (gjenta for alltid) og vilkår (hvis) for elevene.

*Undersøk* og *endre* er oppgaver hvor elevene skal undersøke og endre faktorer i programmet. Dette kan være krevende for noen av elevene. Derfor inneholder oppgaveheftene som kan veilede elevene. I undervisningen anbefales det å vise elevene hvor de finne tipsene og oppfordre elevene til å ta dem i bruk. Tips er en måte å tilpasse undervisningen på.

*Lag* er oppgaver hvor elevene selv skal programmere et program. I *Tilpasninger og overlevelse* skal elevene programmere et spill. Denne oppgaven er tenkt å være motiverende og en utfordring for elever med høyt læringspotensial. Denne oppgaven er naturlig tilpasset nivå og et halvferdig program kan brukes for de som ikke er så dyktige i programmering. I *Naturlig utvalg* er ikke *lag* oppgaven inkludert i elevheftet fordi arbeidsmengden allerede er stor og vi ønsker ikke at elevene skal bli motløse av mengden arbeid. For elever med stort læringspotensialet kan læreren gi egne oppgaver når de er ferdige med elevheftet. En beskrivelse av en *lag* oppgave er lagt ved i lærerveiledningen, og til denne følger et forslag til en løsning. Lignede oppgaver kan gis til elever som ønsker mer krevende arbeid. En annen mulighet er at eleven kan skrive deler av koden i Scratch over til programmeringsspråket Python eller andre skriftprogrammeringsspråk. Disse oppgavene har vi ikke et forslag på løsning for og elevene må da finne ut av dette på egenhånd.

Et annet forslag er at oppgavene *endre* og *lag* kan leveres som et arbeid til læreren. Det å levere et arbeid fra undervisningen kan være med på å motivere elevene eller brukes som vurderingsgrunnlag for læreren. Det kan enten være noe de har jobbet med i denne undervisningstimen, eller noe de kan fortsette med hjemme. Da er det viktig at lærer lager seg en lærerkonto på Scratch. Med lærerkonto kan du har du mulighet til å nullstille og endre elevenes passord og kan gå inn på hver profil for å se prosjektene de har lagt ut. Elevene lager seg en elevkonto, hvor de kan remixe simuleringene og levere sin remix til læreren.

Opgaven *vurder* bør gjøres i plenum. *Vurder* er inkludert for å sikre et fokus på naturlig seleksjon i arbeidet med simuleringen. Det er viktig å poengtere for elevene at en modell ikke kan illustrere hele virkeligheten og at i disse simuleringene er det fokus på noen aspekter ved tilpasninger og overlevelser og naturlig seleksjon, men ikke alle. I oppgaven *vurder* skal elevene diskuterer styrker og svakheter ved simuleringen. Under er det listet opp noen aspekter ved naturlig seleksjon som bør diskuteres for å ikke

skape misoppfatninger. I *Tilpasninger og overlevelse* kan *vrderoppgaven* gjøres som en oppsummering av timen sammen med *diskuter* oppgaven. I *Naturlig seleksjon* kan *vrderoppgaven* tas etter å ha vist simuleringen i plenum. Ved å gjøre disse oppgavene i plenum kan læreren sikre en felles forståelse og forhindre eventuelle misoppfatninger knyttet til simuleringen og evolusjon.

*Simuleringen Tilpasninger og overlevelse:*

<b>Styrker:</b>	<b>Svakheter:</b>
Kamuflasje Tilpasninger til miljøet Jakt og overlevelse Viser at å være stor og sterk, nødvendigvis ikke betyr at man er godt tilpasset	Viser ikke reproduksjon Viser ikke arv Viser ikke mattilgang for musene Bare to arter i dette miljøet

*Simuleringen Naturlig seleksjon:*

<b>Styrker:</b>	<b>Svakheter:</b>
Kamuflasje Fordelaktige egenskaper Naturen som velger ut hvem som overlever og reproducerer seg Seleksjonspress som rovdyr og mat.	<i>Det viktigste å poengtere for elevene er at dette er en prosess som skjer over lang tid.</i> Misvisende at avkom fødes så hyppig. Andre seleksjonspress som sykdommer og boplass/gjemmesteder Mangler andre arter i miljøet Det er begrenset med plass i simuleringen Hvordan pelsfarge er hos harene i virkeligheten (byter til hver årstid) Konkurransen innad i en art Reven jakter unaturlig og spiser unaturlig mye Harene kan bevege seg rett mot reven.

## Hvordan tilpasse nivået i undervisningen?

*Dette kapitlet er basert på råd for tilpassing av blokkprogrammerings oppgaver i boken Kloss for kloss (2022) av Øistein Gjøvik og Joakim Høyland.*

### Gi elevene noen av klossene

Du kan dele ut klosser til elevene som de skal bruke i programmeringen. Elevene må da velge i hvilken rekkefølge de skal bygge klossene og får eksperimentere med et mindre utvalg muligheter. Man kan da velge om elevene skal "bruke opp alle klossene" eller at noen av klossene er lagt inn som en feil og som elevene skal skjønne at de ikke kan bruke (distraktorer). Dette kalles Parsons problemer.

## Gjøre endringer i ferdig program

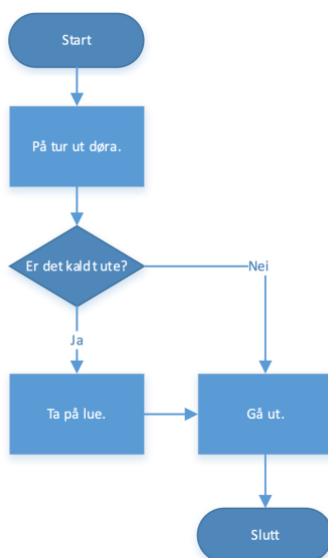
En mulighet er å gi elevene et ferdig program hvor de skal lage et lignende program selv. En oppgave kan være å lage et program med en vri på det originale programmet. Det kan være mye læring i å ta inspirasjon og gjøre endringer i ferdige programmer.

## Start med et halvferdig program

Elevene får et halvferdig program utdelt hvor det gjenstår mer programmering og justeringer for at det skal fungere. Dette kan brukes om man ikke ønsker at elevene skal bruke mye tid på deler av programmeringen som ikke er relevant i den sammenhengen. Denne tilpasningen kan også kombineres med å gi elevene klosser.

## La elevene planlegge med et flytdiagram

Før elevene starter programmering kan de planlegge hvordan programmet skal fungere. Å lage et flytskjema er en skjematisk oversikt over hvordan programmeringen skal være. Et flytdiagram består av en start og slutt, rektangler hvor hva som skal gjøres, romber når noe blir spurt etter og piler som viser gangen i programmeringen.



Figur 1: Eksempel på et flytdiagram som planlegger hvordan noe skal fungere.

## Rett opp i feil

Det å rette opp i feil kalles debugging og er viktige ferdigheter i programmering. Gi elevene en kode med feil i som de skal finne og rette opp i.

## For elever med høyt læringspotensial

Mange lærere vil erfare å ha elever som kan mer enn dem i programmering og trenger hjelp på et nivå høyere enn det læreren kan tilby. Til denne gruppen anbefaler vi å lære dem å feilsøke i programmet og søke etter tips på internett. Da er det lurt å søke «Scratch how to make a...» eller benytte seg av <https://en.scratch-wiki.info/>. Til den vanskelige *lag*-oppgaven i naturlig seleksjonsheftet har vi laget et forslag til løsning, men oppfordrer elever til finne egne løsninger.

## Simuleringen

For å se og bruke simuleringen må du trykke på lenken. Når du åpner lenken til simuleringen kommer du inn til prosjektsiden, hvor man kan lese om hvordan man bruker prosjektet. For å se simuleringen trykk: på *grønt flagg*, og for å gjøre endringer i kodene trykk: *se inni*.

### [Simuleringen Tilpasninger og overlevelse](#)

Simuleringen *Tilpasninger og overlevelse* er designet for å kunne diskutere begrepet *survival of the fittest* og unngå misoppfatninger rundt begrepet. Simuleringen er laget for å være så enkel som mulig, slik at elevene enkelt kan forstå og jobbe i programmet selv. Simuleringen og oppgaveheftet som hører til benytter mange ulike tilpasninger til undervisningen som er nevnt ovenfor. Ettersom programmeringen er såpass enkel, vil ikke programmeringen blir forklart i detalj.

### [Simuleringen Naturlig seleksjon](#)

Simuleringen *Naturlig seleksjon* skal vise elevene hvordan individer som er best tilpasset miljøet overlever. Elever har ofte en misoppfatning når det gjelder tidsrammen for evolusjon. Elever har utfordringer med at evolusjon er en prosess som går over lang tid og som ikke er observerbar. Derfor kan en simulering være til hjelp for å forstå at arter over tid kan endre egenskaper for å være bedre tilpasset, men at individer ikke kan endre sine gener gjennom et liv.

Vi anbefaler ikke at elevene skal lære eller prøve å forstå alt av koder i denne simuleringen, men elevene skal for det meste bare gjøre små justeringer i programmet. Programmet er sortert slik at de kodene elevene ikke skal bruke eller forstå er plassert unna de kodene elevene skal endre. Simuleringen forklares i lærerveiledningen og det pekes på hvilke deler av kodene som det kan være hensiktsmessig å bruke og vise til elever. Vi har valgt å lage flere funksjoner som for å hoppe, reproduksjon og for å sjekke om man møter hunn- eller hannkjønn. Totalinntrykket av simuleringen vil da være lettere å forstå.

### [Forklaring av konstanter, variabler og lister](#)

Denne delen av lærerveiledningen inkludere en forklaring av alle konstanter, variabler og lister for at simuleringen *Naturlig seleksjon*, for at simuleringen skal være mest mulig forståelig. Det forklares også hva hver konstant, variabel og liste skal illustrere om naturlig seleksjon.

I tekstprogrammering brukes kommentarer som en måte å gi instruksjoner eller dokumentasjon til leseren av programmet. Det er god praksis i programmering å inkludere kommentarer i et program som beskriver hva som skjer, og hva ulike variabler inneholder. Det gjøres for at andre skal forstå programmet og hva som menes med ulike deler av programmet. I Scratch er det ikke mulighet til å legge inn kommentarer utenom på startsidene av programmet. Derfor inkluderer vi denne delen i lærerveiledningen. En forklaring av variabler i Scratch forklares i kapittelet om Scratch.

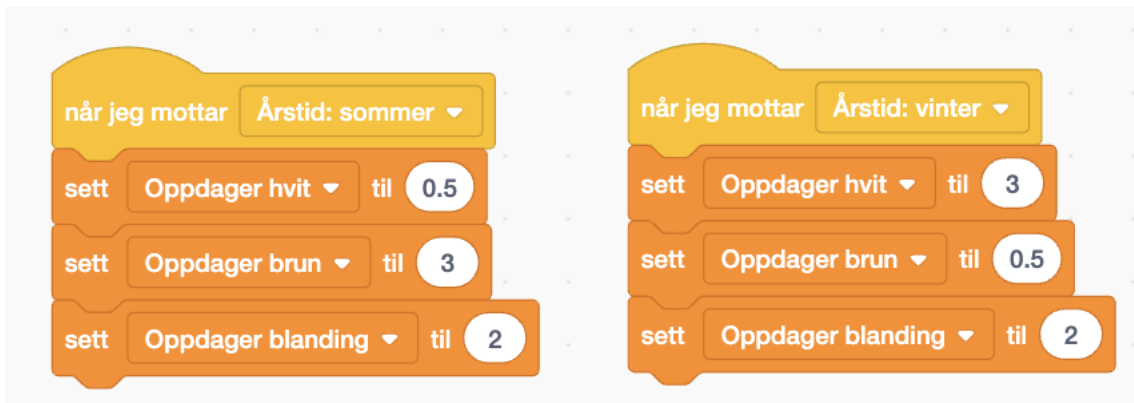
### Konstanter

En konstant er en variabel som er bestemt og som ikke forandrer seg gjennom programmet. I programmet brukes flere konstanter som illustrerer viktige poenger ved naturlig seleksjon. Konstantene kan brukeren av programmet endre mellom hver gjennomkjøring for å teste ut ulike hypoteser og sammenligne resultatene. Det anbefales at læreren kan vise hvilke konstanter som elevene kan endre mellom hver gjennomføring i plenum.

### I hvilken grad reven oppdager harer med ulik pelsfarge:

Ønsket datatype: desimaltall

Hvilke konstanter som brukes avhenger av hvilken årstid det er. Et lavt tall betyr at den pelsfargen blir oppdaget på lengre avstand. Et større tall betyr at denne pelsfargen er vanskeligere å oppdage. I hvilken grad reven oppdager harer med ulik pelsfarge skal illustrere kamuflasje og at dette er en tilpasning som påvirker harenes overlevelse.



Figur 2: Variabel som bestemmer i hvilken grad reven oppdager harene. Variabelen avhenger av valgt årstid.

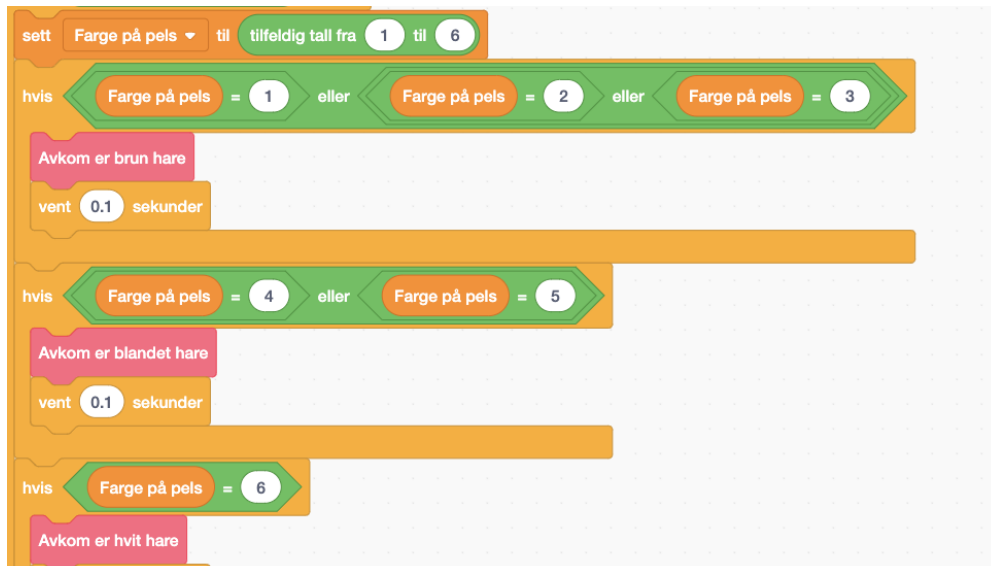
### Pelsfarge:

Ønsket datatype: Heltall, mellom 1 og 6

Denne konstanten velger ikke brukeren selv, men velges tilfeldig av programmet. Dette heltallet viser sannsynligheten for å få ulik pelsfarge. Programmet velger tilfeldig et tall mellom 1 og 6. Om dette tallet blir 1, 2 eller 3 vil dette avkommet få pelsfargen brun, om tallet blir 4 eller 5 vil avkommet bli en blandingshare og hvis tallet blir 6 vil avkommet få hvit pelsfarge. Brukeren kan endre på sannsynlighet for ulik pelsfarge ved å endre på

hvilke tall som tilhører hvilken farge eller endre hvilke tall programmet kan velge mellom når den skal trekke et tilfeldig tall.

Denne konstanten skal illustrere hva som skjer når harer med ulik pelsfarge møtes. Hvilken pelsfarge avkommet deres får er tilfeldig, men noen pelsfarger er mer sannsynlig enn andre.



Figur 3: Variabelen «farge på pels» som bestemmer pelsfargen til harene som blir født.

### Variabler

Ved å bruke variabler lagres verdier i programmet. En variabel kan være et tall, en posisjon eller en tekst. I Scratch kan alle de runde klossene brukes som en variabel. I simuleringen benyttes variabler som endres underveis i programmet. Variablene brukes for å telle antall, for å velge mellom ulike valg, for å få noe til å skje eller for å lagre posisjoner. Variablene er bestemt og det er ikke noe poeng at brukere av programmet skal endre disse.

### Antall harer:

Ønsket datatype: Heltall.

Antallet kan øke eller minke, men det totale antallet harer kan ikke være et negativt tall, fordi det gir ikke mening å ha negativt antall harer. Denne variabelen teller antall harer i programmet. Når programmet starter er det to harer av hver farge i programmet (se figur 4a), hvis de parrer seg vil antallet øke med antall avkom. Om en hare blir spist av reven vil antallet minke med 1. Ved å sammenligne antallet kan vi se hvilken pelsfarge som er best tilpasset miljøet når reven begynner å jakte (se bilde 4b).





Figur 4a: Verdien for antall harer som starter i simuleringen.

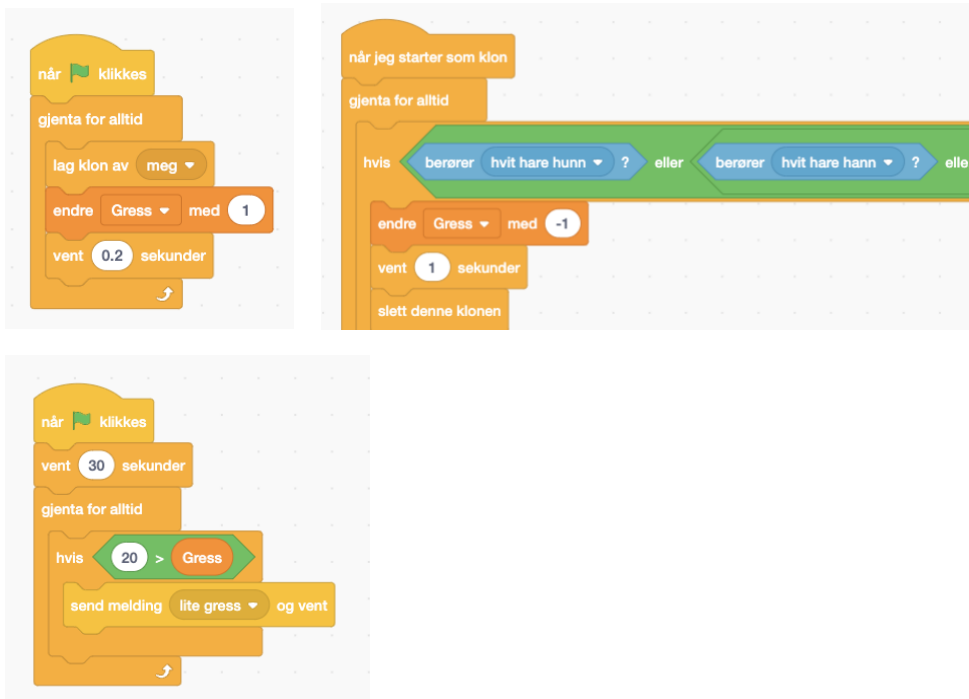
Figur 4b: Viser verdiene antall harer har underveis i simuleringen

### Antall gress:

Ønsket datatype: Heltall.

Antallet kan øke eller minke, men det totale antallet gress kan ikke være et negativt tall, fordi det gir ikke mening å ha negativt antall gress. Gresset vil «vokse» (lage klon av seg selv) kontinuerlig, da vil telleren øke med 1. Om en hare berører gresset vil gresset forsvinne og telleren vil endres med -1. Om det totale antallet gress er mindre enn 20, vil programmet stoppe for å illustrere næringsmangel blant harene. Hvilket antall gress som er grensen for at harene sulter kan elevene endre ved å endre «hvis (>)Gress».

Variabelen «gress» illustrer mattilgang, som er en type seleksjonspress.



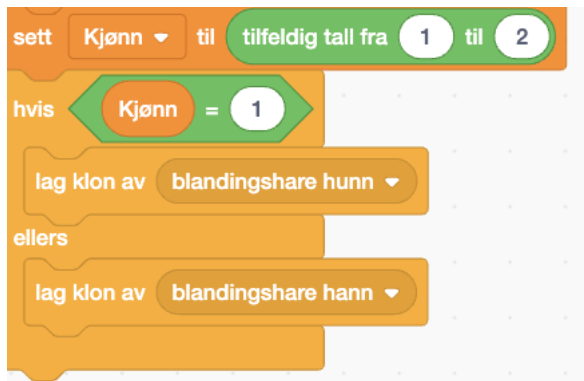
Figur 5a: Bestemmer hvor fort gresset vokser. Figur 5b: antallet gress minker ved berøring av en hare. Figur 5c: Simuleringen stopper ved dette antallet gress.

## Kjønn:

Ønsket datatype: Heltall, 1 eller 2

Denne variabelen velges tilfeldig av programmet. Det er 50% sannsynlighet for hvert kjønn. Om det tilfeldige tallet er 1 vil avkommet være en hunnhare, mens hvis det tilfeldige tallet ikke blir 1, altså 2, vil avkommet være en hannhare.

Denne variabelen viser at det er tilfeldig om avkommet blir en hann eller en hunn. Ved å ha harer av ulikt kjønn kan det illustrere parring og reproduksjon.



Figur 6: Verdien til variabelen «kjønn» bestemmer hvilket kjønn som fødes.

## En av de originale harene er spist:

Ønsket datatype: Heltall, 0 eller 1

Variabelen fungerer som en av eller på knapp. Når programmet starter, har denne variabelen verdien 0. Om noen av de harene som startet programmet berører reven blir de spist og verdien endres til 1. Da vil deres posisjon i programmet endres til «død» slik at de ikke forstyrrer programmet videre.

Denne variabelen illustrer ikke noe om naturlig seleksjon, men benyttes for listene med posisjonene til harene skal være riktig. Listene med posisjoner brukes slik at reven kan jakte på harer ut ifra deres posisjon, som illustrer tilpasninger i miljøet.



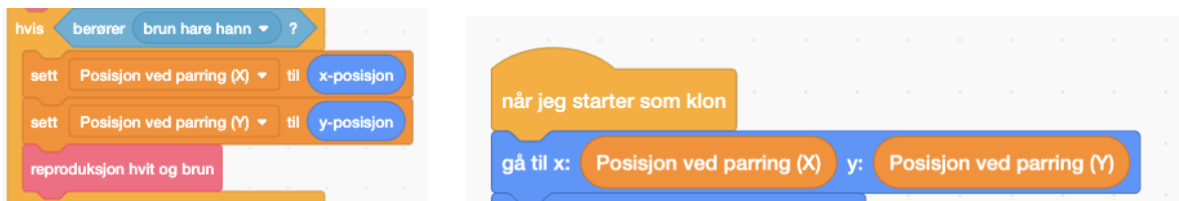
Figur 7a: Variabelen «original 1 død» bestemmer om de originale harene lever eller er spist. Figur 7b: Variabelen «original 1 død» endres fra 0 til 1 ved berøring av reven.

### Posisjon ved parring:

Ønsket datatype: desimaltall (posisjoner)

Vi lagrer x- og y-posisjonen hvor en hann og hunnhare møtes i variabelen «Posisjon ved parring (X) og (Y)». Slik at når det lages avkom (kloner) vil de starte fra denne posisjonen.

Dette illustrere at hunnharen og hannharen parrer seg, at hunnharen har vært drektig og at avkommet fødes som et resultat av dette.



Bilde 8a og b: Variabelen lagrer x- og y-verdien en hare har ved paring.

### Posisjon i liste hvit/brun/blanding:

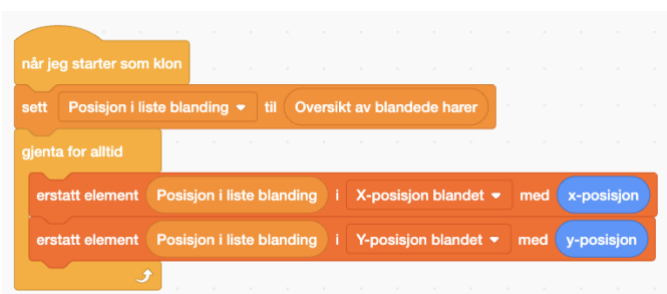
Ønsket datatype: Heltall

Dette er en variabel som lagrer en individuell verdi som hver enkelt hare har. Den forteller hvilken posisjon haren har i listene med x- og y-posisjoner og blir brukt til å oppdatere posisjonen harene har når de beveger seg og til å erklære harer for død.

### Oversikt over hvite/brune/blandede harer:

Ønsket posisjon: Heltall.

Er en oversikt over hvor mange harer som finnes i listene over harenes x- og y-posisjon. Denne variabelen blir brukt til å tildele nye avkom en unik og korrekt *posisjon i liste hvit/brun/blanding*. Denne variabelen brukes for å legge til nye avkom nederst i posisjonslistene.



X-posisjon brun	Y-posisjon brun	X-posisjon brun	Y-posisjon brun
1 231.7025...	1 9.177154...	1 231.7025...	1 9.177154...
2 47.22831...	2 -79.6970...	2 47.22831...	2 -79.6970...
3 død	3 død	3 død	3 død
4 -77.0525...	4 -147.134...	4 -77.0525...	4 -147.134...
5 222.1855...	5 -145.060...	5 222.1855...	5 -145.060...
6 162.6904...	6 -33.6338...	6 162.6904...	6 -33.6338...
7 106.1210...	7 82.5662...	7 106.1210...	7 82.5662...
+ lengde 150 =	+ lengde 150 =	+ lengde 150 =	+ lengde 150 =

Figur 9a: Viser hvordan variablene «posisjon i liste» og «oversikt» blir brukt i programmet. Figur 9b: verdiene for posisjon i liste er markert med gul. Figur 9c: Oversikt variabelen er markert med grønt.

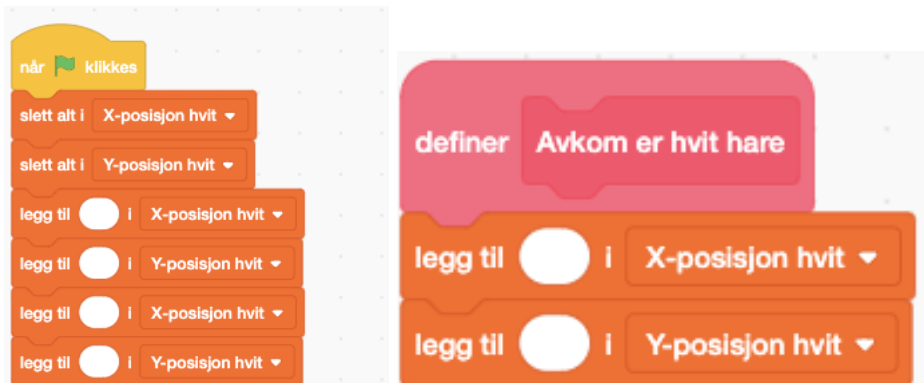
### Lister:

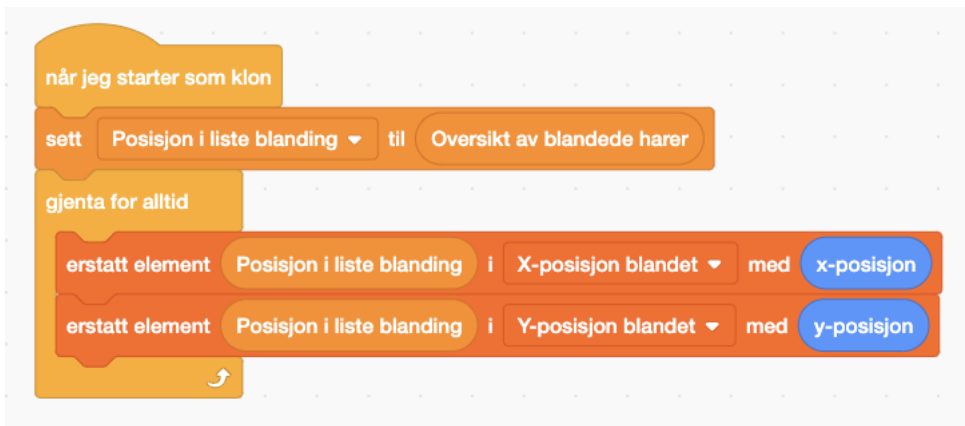
En liste er en samling av verdier som vi refererer til med samme navn. Det finnes mange funksjoner knyttet til lister som å legge til og fjerne verdier, endre på rekkefølge, bruke lengden av listen og regne med alle verdiene i listen.

### Lister over x- og y-posisjoner for harene:

I programmet benytter vi oss av en liste for alle x- og y-verdier som harene har. Verdiene lagres i listene «X-posisjon hvit» og «Y-posisjon hvit», dette gjelder for alle pelsfargene.

Listene illustrer ikke noe om naturlig seleksjon, men benyttes for at posisjonene til harene skal være riktig. Listene med posisjoner brukes slik at reven kan jakte på harer ut ifra deres posisjon, som illustrer tilpasninger i miljøet.





Figur 10a: Viser hvordan programmet starter med de originale harene. Figur 10b: Viser at avkom legges inn i listene. Figur 10c: de nye avkommenes posisjon legges inn korrekt i listene.

### Forklaring til programmeringen

I forklaringen markeres det først og i tykk skrift i hvilken figur de ulike kodene hører til. For hver kode forklares det både hva det gjør med tanke på programmering og hva den viser innenfor seleksjon. Alle de rosa klossene er funksjoner som er laget for å illustrere deler av naturlig seleksjon.

Disse delene kan være hensiktsmessig å forklare til elever

	<p><b>HARE:</b></p> <p>Dette er en funksjon for en bevegelsessekvens. Ved å bruke ulike drakter vil det være mer naturlig bevegelse enn om det ikke hadde byttet drakter. Det kan være en fordel å vise elevene de ulike draktene</p> <div style="text-align: center;"> </div> <p>Denne klossen kan brukes senere i programmet istedenfor å repetere hele dette scriptet for hvert hopp.</p> <p>Harene beveger seg rundt for å spise mat, møte partnere og unngå å bli spist.</p>
--	---

```

definer Reproduksjon hvit
si ♥ i 2 sekunder
vent 5 sekunder
Hoppe
gjenta tilfeldig tall fra 0 til 3 ganger
  Avkom er hvit hare
  vent 0.1 sekunder

```

**HARE:**

Reproduksjon er en funksjon. Gjenta tilfeldig tall fra 0-3 ganger betyr hvor mange avkom harene får ved hver parring.

Denne funksjonen definerer hva som skjer når en hvit hunnhare møter en hvit hannhare. Da vil avkommet få hvit pelsfarge.

Reproduksjon i virkeligheten handler om å møte en partner og føre gener videre. Når mor og far begge har samme pelsfarge er sannsynligheten for at avkommet vil arve samme pelsdrakt stor.

```

definer reproduksjon hvit og brun
si ♥ i 2 sekunder
vent 5 sekunder
Hoppe
gjenta tilfeldig tall fra 0 til 3 ganger
  sett Farge på pels til tilfeldig tall fra 1 til 6
  hvis Farge på pels = 1 eller Farge på pels = 2 eller Farge på pels = 3
    Avkom er brun hare
    vent 0.1 sekunder
  hvis Farge på pels = 4 eller Farge på pels = 5
    Avkom er hvit hare
    vent 0.1 sekunder
  hvis Farge på pels = 6
    Avkom er blandet hare
    vent 0.1 sekunder

```

**HARE:**

Denne funksjonen viser sannsynligheten for hvilken pelsfarge avkommet til to harer med ulik pelsfarge vil få.

Her møtes en hvit og en brun hare da vil sannsynligheten for at den blir brun være 3/6, for at den blir hvit være 2/6 Og sannsynligheten for blandet vil være 1/6.

Det er variabelen tilfeldig tall som velger dette og den velger et tilfeldig tall fra 1-6.

Dette viser arv av gener fra foreldrene. I dette tilfellet har foreldrene forskjellig pelsfarge, derfor vil avkommene arve ulike egenskaper fra foreldrene sine.

```

når jeg starter som klon
  gå til x: Posisjon ved parring (X) y: Posisjon ved parring (Y)
  begrens rotasjon vend sideveis
  pek i retning tilfeldig tall fra -180 til 180
  vis
  endre Blandings harer med 1
  endre antall harer med 1
  vent 0.6 sekunder
  gjenta for alltid
    vent tilfeldig tall fra 0 til 1 sekunder
    Hoppe
    Hvit hare møter partner sjekk
    hvis berører Rev ?
      endre Blandings harer med -1
      endre antall harer med -1
      erstatt element PersonligID i X-posisjon blandet med død
      erstatt element PersonligID i Y-posisjon blandet med død
      slett denne klonen
    hvis y-posisjon > -10
      snu 180 grader
  
```

**HARE:**

Avkommet vil bevege seg i tilfeldig posisjon og ikke følge etter de andre harene.

Når klonene starter vil de hoppe rundt og møte andre harer, møter de en av det motsatte kjønn kan det skje reproduksjon.

Møter de en rev, dør haren. Tellingene av hvite harer vil gå ned og posisjonene til denne klonen vil endres til død i listene.

y-posisjonen kan ikke være høyere enn -10 for da beveger harene seg i himmelen. Når de denne grensen vil de snu.

```

når klikkes
  skjul
  gå til x: -325 y: -102
  sett Brune harer til 2
  sett Hvite harer til 2
  sett Blandings harer til 2
  sett antall harer til 6

når jeg mottar Årstid: sommer
  sett Oppdager hvit til 0.5
  sett Oppdager brun til 2.5
  sett Oppdager blanding til 2

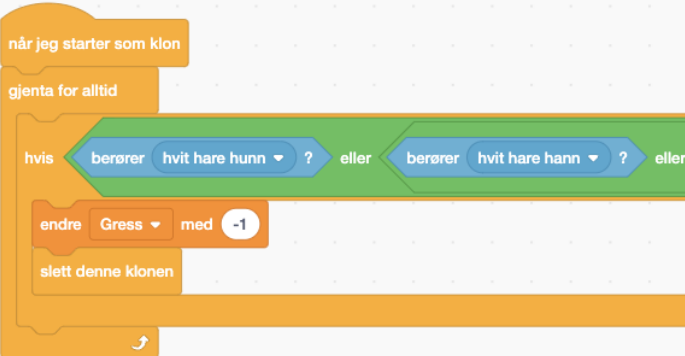
når jeg mottar Årstid: vinter
  sett Oppdager hvit til 3
  sett Oppdager brun til 0.5
  sett Oppdager blanding til 2
  
```

**REV:** Den første koden bestemmer ulike variabler når flagget trykkes. De første er antallet på de ulike harene.

«Oppdage» bestemmer hvor godt reven ser de ulike pelsfargene ved ulike årstider.

Meldingene årstid sommer og årstid vinter forandrer hvordan reven oppdager de ulike pelsfargene.

Byttedyr og rovdyr har pelsfarge som gir dem kamuflasje i miljøet. Når miljøet endrer seg er det en fordel at også pelsfargen endrer seg.

	<p><b>REV:</b>  Reven kommer meldingen sendes når reveknappen trykkes.  Revens jakt bestemmes av hvor godt reven ser de ulike pelsdraktene til de ulike årstidene.</p> <p>Reven er et rovdyr og i hvilken grad reven ser harene vil avgjøre hvilken pelsfarge den vil jakte på.</p>
	<p><b>GRESS:</b>  Gresset legger seg bakerst slik at de andre figurene kan bevege seg over.</p> <p>Gresset vokser ikke i himmelen, og derfor går ikke y-posisjon høyere enn 4.</p>
	<p><b>GRESS:</b>  Gjelder også om gresset berører de andre harene</p> <p>Om harene berører gresset, spiser de det og derfor forsvinner gresset.</p>



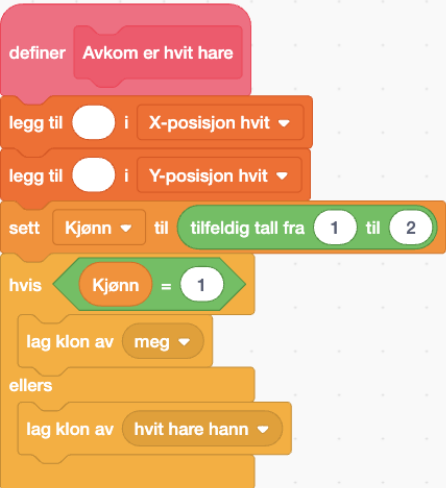
 	<p><b>GRESS:</b> Om det er mindre enn 20 gress i simuleringen vil programmet stoppe. Vent er lagt inn slik at simuleringen har litt tid på seg til å starte og gresset kan vokse i begynnelsen.</p> <p>Bruker melding og mottaker for å kommunisere mellom figurer</p> <p>Om mattilgangen er dårlig vil ikke harene kunne leve i dette miljøet. I virkeligheten vil ikke alle harene sulte samtidig, men det vil fødes færre harer og færre vil overleve. En tekst som forklarer dette vil komme på skjermen når dette skjer.</p>
--	---

Disse delene er ikke hensiktsmessig å forklare til elever

Disse scriptene er ikke hensiktsmessig å forklare til elever. Ingenting av det som er i disse scriptene handler om evolusjon, men er der for at simuleringen skal fungere.

Vi har plassert disse scriptene unna de scriptene som elevene skal se, forstå og endre. Dette for at det skal være lettere å bruke programmet og for at elevene skal kunne finne ting i programmet. Elever som er svært gode i programmering kan gjerne se på disse scriptene om de ønsker.

Disse scriptene er ikke nødvendig for lærer å forstå, men er beskrevet om det skulle være nødvendig.

	<p><b>HARE:</b> En variabel som velger tall 1 eller 2 brukes til å bestemme hvilket kjønn avkommet får.</p> <p>Her brukes klossen <i>hvis-ellers</i>. Hvis tallet er 1 vil det bli en hunnhare, mens hvis det ikke blir 1 blir det en hannhare.</p> <p>I virkeligheten er det også ca. 50% sannsynlighet for hann eller hunnkjønn ved avkom.</p>
---	--

```

definer Hvit hare møter partner
hvis berører brun hare hann ?
  sett Posisjon ved parring (X) til x-posisjon
  sett Posisjon ved parring (Y) til y-posisjon
  reproduksjon hvit og brun
hvis berører hvit hare hann ?
  sett Posisjon ved parring (X) til x-posisjon
  sett Posisjon ved parring (Y) til y-posisjon
  Reproduksjon hvit
hvis berører blandingshare hann ?
  sett Posisjon ved parring (X) til x-posisjon
  sett Posisjon ved parring (Y) til y-posisjon
  Reproduksjon hvit og blanding

```

**HARE:**

Om hunnharen berører en hannhare vil denne posisjonen lagres, og avkommet vil fødes på dette stedet.

I virkeligheten vil ikke harene føde sekunder etter at de har møtt en hare av motsatt kjønn. I simuleringen er det slik for å gjøre det tydelig hvem foreldrene til avkommet er.

```

gjenta for alltid
  hvis Original 1 død = 0
    erstatt element 1 i X-posisjon hvit med x-posisjon
    erstatt element 1 i Y-posisjon hvit med y-posisjon
  ellers
    erstatt element 1 i X-posisjon hvit med død
    erstatt element 1 i Y-posisjon hvit med død
    vent 100 sekunder

```

**HARE:**

«Original hare 1» død betyr at den originale hunnharen er død. «Original hare 2» død betyr at den originale hannharen er død. Hvis denne variabelen er 0 er haren i live og hvis variabelen er 1 er den død (som en av og på bryter). X-posisjonene og y-posisjonene i listene byttes ut med død.

```

når jeg starter som klon
  sett Posisjon i liste hvit til Oversikt av hvite harer
  gjenta for alltid
    erstatt element Posisjon i liste hvit i X-posisjon hvit med x-posisjon
    erstatt element Posisjon i liste hvit i Y-posisjon hvit med y-posisjon

```

**HARE:** Brukes for å lagre x- og y-posisjonene til klonene i en liste.

```

definer jakt hvit synlighetsfaktor brun synlighetsfaktor blandings synlighetsfaktor

finn avstand nærmeste blandings blandings synlighetsfaktor
finn avstand nærmeste brun brun synlighetsfaktor
finn avstand nærmeste hvit hvit synlighetsfaktor

hvis avstand nærmeste hvit < avstand nærmeste brun
sett avstand nærmeste til avstand nærmeste hvit
sett X-pos nærmeste til X-pos nærmeste hvit
sett Y-pos nærmeste til Y-pos nærmeste hvit
ellers
sett avstand nærmeste til avstand nærmeste brun
sett X-pos nærmeste til X-pos nærmeste brun
sett Y-pos nærmeste til Y-pos nærmeste brun

hvis avstand nærmeste blanding < avstand nærmeste
sett avstand nærmeste til avstand nærmeste blanding
sett X-pos nærmeste til X-pos nærmeste blanding
sett Y-pos nærmeste til Y-pos nærmeste blanding

hvis avstand nærmeste < 50
gli 0.7 sekunder til x: X-pos nærmeste + 20 y: Y-pos nærmeste
vent 5 sekunder
ellers
gli 0.7 sekunder til x: X-pos nærmeste + 20 y: Y-pos nærmeste
vent 5 sekunder

```

**REV:** Dette scriptet viser hvordan reven jakter ut ifra hvilken hare som er nærmest. Det er også lagt inn at den observerer noen farger av harer bedre enn andre og derfor vil observere en synlig farge fra lengre avstand enn en farge som er bedre kamuflert.

Den brukes posisjonen til alle klonene til harene for å oppdage hvilken som er nærmest.

```

definer finn avstand nærmeste brun brun synlighetsfaktor

sett avstand brun til 10000
sett avstand nærmeste brun til 10000
sett brun hare teller til 1

hvis lengden av X-posisjon brun > 0
gjenta lengden av X-posisjon brun ganger
hvis ikke element hvit hare teller i X-posisjon brun = død eller element hvit hare teller i Y-posisjon brun = død
kalkuler avstand brun element brun hare teller i X-posisjon brun element brun hare teller i Y-posisjon brun
hvis avstand brun < avstand nærmeste brun
sett avstand nærmeste brun til avstand brun
sett X-pos nærmeste brun til element brun hare teller i X-posisjon brun
sett Y-pos nærmeste brun til element brun hare teller i Y-posisjon brun
endre brun hare teller med 1
sett avstand nærmeste brun til avstand nærmeste brun * brun synlighetsfaktor

```


Setter først en veldig lang avstand som referanse. Alle harer lengre unna en 10 000 ser ikke reven i det hele tatt.

Sjekker deretter om listen inneholder minst en hare, for sikkerhets skyld. Avbryter søket dersom dette ikke er tilfellet.

Går så gjennom alle elementene i listen, og regner ut avstanden mellom reven og harene. Harer som er døde blir ikke sjekket.

Posisjonen til den nærmeste haren blir lagret fortløpende, og byttet ut når den finner et nærmere alternativ. Avstanden til reven blir også tatt vare på, sann at man har et enkelt sammenligningsgrunnlag til neste sjekk.

Når alle harer har blitt sjekket, blir den nærmeste haren fra lista regnet ut basert på farge/kamuflasje

	<p><b>REV:</b> Pythagoras brukes for å finne avstanden til hver enkelt hare. Bruker da hypotenusen for å finne nærmeste hare.</p>
---	---

## PRIMM

PRIMM er en undervisningsmetode utviklet for å støtte lærere i gjennomføringen av programmeringsundervisning for nybegynnere (Sentance et al., 2019). Denne tilnærmingen er brukt for å planlegge undervisning med fokus på naturfagligkunnskap og blokkprogrammeringsferdigheter.

<b>Forutse (predict)</b>	Elevene skal prøve å forstå hva en kode vil gjøre
<b>Kjøre (run)</b>	Elevene får kjøre koden og sjekke om de har forutsett riktig
<b>Undersøk (investigate)</b>	Elevene får jobbe med aktiviteter og spørsmål til koden som er med på å utvikle deres forståelse. På denne måten får elevene utforsket strukturer, forklart prinsipper og øvd på debugging med støttestrukturer.
<b>Endre (modify)</b>	Elevene forbedrer og endrer funksjonene til et utdelt program. Ved at elevene skal jobbe med å forbedre et eksisterende program vil dette tilpasse undervisningen slik at elevene raskt opplever mestring og ved å endre funksjoner vil de få et eierskap til programmet. Det å jobbe i et eksisterende program hjelper elever med å utvikle forståelse for grunnprinsippene i programmering på et nivå hvor alle kan jobbe.
<b>Lag (make)</b>	Elevene får selv lage ett nytt program med et nytt problem som imidlertid har samme strukturer som programmene elevene har jobbet med tidligere i undervisningen

## Evolusjon

(Bøhle & Ervik, 2020).

Evolusjon handler om hvordan livet har utviklet seg på jorda, fra de første encellede organismene til et stort mangfold av arter med ulike tilpasninger. Evolusjonen handler både om hvordan den genetiske sammensetningen innen en populasjon endrer seg, og om hvordan nye slekter og arter oppstår.

Naturlig utvalg er den viktigste mekanismen i evolusjonen. Gener går i arv og det er genetisk variasjon i populasjonen. Populasjonen blir stadig bedre tilpasset miljøet den lever i, når de individene som er best tilpasset, fører genene sine videre. Selv om alle individene i en populasjon ser like ut, har de likevel små genetiske variasjoner. Det gjør at det innenfor en populasjon er et mangfold av genetisk bestemte egenskaper.

Naturlig seleksjon gir individer tilpasninger som gir dem en fordel i konkurranse med andre individer i egen art, bedre sjanse til å formere seg, mulighet til å leve i ulike miljøer og kamuflasje for beskyttelse og jakt.

### Vanskelige begreper i evolusjon

Gregory (2009); Harms & Reiss (2019).

- Elever sliter med å forstå *underliggende temaer som populasjon, adaptasjon og frekvenser*. Det kan derfor være en fordel å være forsiktig med bruk av begreper og gi en god beskrivelse til disse når de nevnes.
- *Nivåer av organisering* er også vanskelig for elever å forstå. Forklar elevene hva som er forskjellen mellom gener, individer, populasjoner og arter og når disse begrepene brukes forsikre seg at elevene forstår hvilket nivå det er snakk om.
- Elever har ofte en misoppfatning når det gjelder *tidsrammen for evolusjon*. Dette er en prosess som går over lang tid og som regel ikke er observerbar. Derfor kan en simulering være til hjelp for å forstå at ikke individer kan endre egenskaper gjennom et liv, men at arter over tid kan endre egenskaper for å være bedre tilpasset.

### Survival of the fittest

Reydon (2019).

Survival of the fittest er et engelsk uttrykk som beskriver at de best tilpassede individene overlever i naturen og dermed får reprodusert seg til neste generasjon. Til dette engelske uttrykket er det knyttet mange misoppfatninger.

1. *Fitness handler om fysisk styrke.*
2. *Gjennom naturlig seleksjon er det kun de mest egnede og dermed fitteste som overlever.*
3. *Organismer kan endre seg for å bedre passe inn i miljøet.*
4. *Fitness er en egenskap hos arter og ikke hos organismer.*
5. *Fitness handler om gener, og dette betyr at alleler som er dominante er mer egnet enn recessive gener.*

## Scratch

*Dette kapittelet er basert på innføringen av Scratch i boken Kloss for kloss (2022) av Øistein Gjøvik og Joakim Høyland.*

Scratch er et blokkprogrammeringsspråk som er utviklet som et visuelt programmeringsspråk for barn av MIT. I Scratch kan man bygge sammen klosser med kommandoer som kjøres etter hverandre ovenfra og ned. Fordelen med blokkprogrammering er at det er visuelt og dermed mer forståelig enn tekstprogrammering. Ved å trene på blokkprogrammering lærer elevene om programmeringsprinsipp de får bruk for i tekstprogrammering senere.

### Klosser

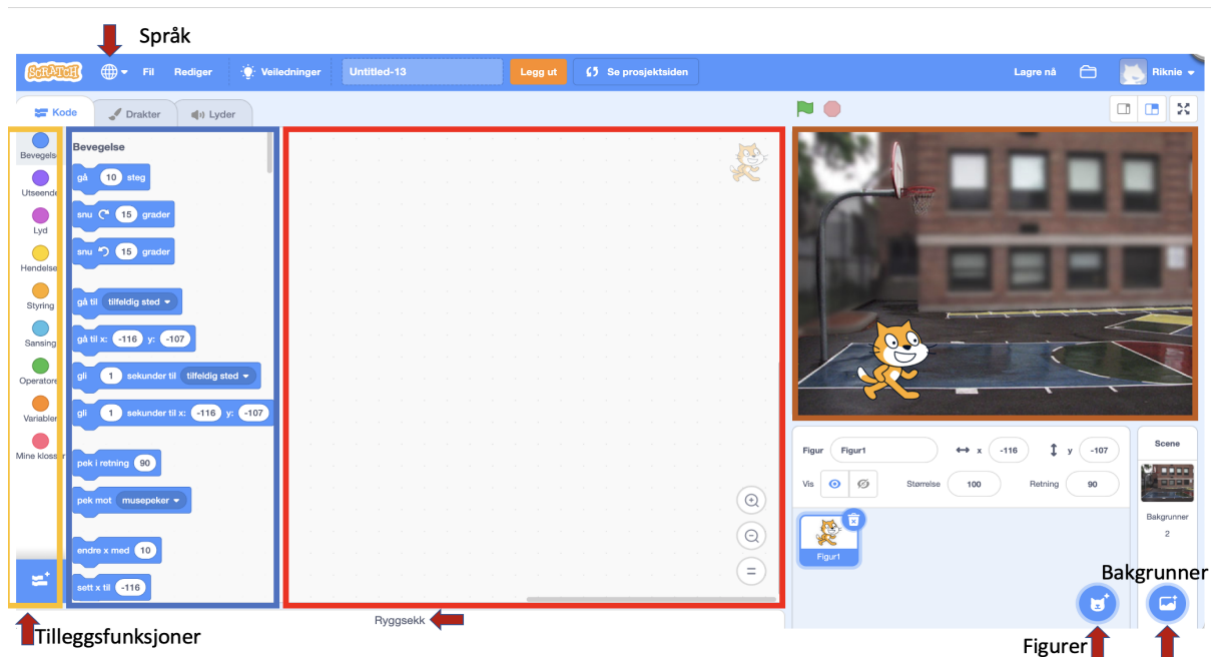
Klossene er kommandoer. Helt til venstre er klossene fargekodet og fordelt i kategorier (se område markert i gult og blått). For eksempel er alle kommandoene som handler om bevegelse mørk blå. Midt på skjermen er det et stort blankt område der du slipper klossene og pusler dem sammen. Klosser som er puslet sammen kalles et script og dette området kalles skriptområdet (se område markert i rødt). En figur kan ha en eller flere skript og kan også ha interaksjon med andre figurer. Trykker du på koden vil du se den i aksjon på scenen.

### Figurer

Til å starte med ser man bare en figur, en katt som kalles Scratchie, som står på scenen (se område markert i oransje). Scenen viser resultatet av hele programmet når det kjøres. Klossene som dras i feltet, forteller hva Scratchie skal gjøre. Om klossen gå 10 steg er valgt vil katten gå 10 steg. Flere figurer kan velges ved å trykke på figurtegnet, da kan man velge mellom noen figurer på Scratch, last opp en figur selv eller tegne en egen. Figurene kan ha flere ulike drakter, disse kan ha ulik farge eller bevegelsesmønster. Ved å bytte drakter kan man for eksempel illustrere bevegelse og endring.

### Bakgrunner

Det er mulig å endre bakgrunn i programmet. Også her kan man velge mellom noen ferdiglagde eller lage bakgrunnen selv. Om man dobbeltrykker på bakgrunnen kan man lage en kode i selve bakgrunnen.



Figur 11: Oversikt over Scratch.

## Oppstartklossen

Disse klossene har en buet overside som viser det ikke er mulig å stable noe over disse klossene. Denne typer klosser finner du under kategorien **Hendelser**. Klossene i seg selv gjør ingenting, men er der for å vise hvor man skal starte de forskjellige skriptene når programmene kjøres. Klossene som henger sammen med disse klossene vil bli kjørt når den hendelsen skjer og vil bli kjørt fra toppen og nedover. Klosser som ikke er koblet til en oppstartskloss vil ikke påvirke programmet.

## Løkker

I kategorien **styring** finnes det flere løkker som for eksempel klossen **"Gjenta \_ ganger"**, hvor man angir hvor mange ganger handlinger skal gjentas. Løkkeklossene er formet som en fleksibel klamme der klossene som skal gjentas plasseres inni klammen. Løkker brukes for å gjøre programmer så effektivt som mulig og for å lage en oversiktlig kode.





Figur 12a, b og c: Viser 3 ulike løkker i Scratch.

## Tester og vilkår

Gjør at programmet kan ta et valg basert på vilkår vi definerer. En test sjekker om et vilkår er sant eller usant. I Scratch finnes det to typer tester, **"hvis"** og **"hvis \_ ellers"**.

Disse finnes i kategorien **Styring**. "Hvis" -klossene har klammer slik at man kan plassere klosser inni og omfavner en løkke.

	<p>Hvis vilkåret er sant, vil algoritmen inne i klammen utføres. Hvis vilkåret er usant vil algoritmen hoppe over algoritmen i klammen og gå videre i skriptet.</p>
	<p>Hvis vilkåret er sant vil algoritmen inne i klammen utføres. Om vilkåret er usant vil algoritmen i "ellers" klammen kjøres. På denne måten vil en kode kjøres uansett.</p>

## Operatorer

Operatorer trengs når vi skal jobbe med tall og variabler. Det finnes en egen kategori for operatorer og som er grønne. Klosser som brukes til utregninger har runde ender, mens klosser som brukes for å sjekke noe har spisse ender.

## Variabler

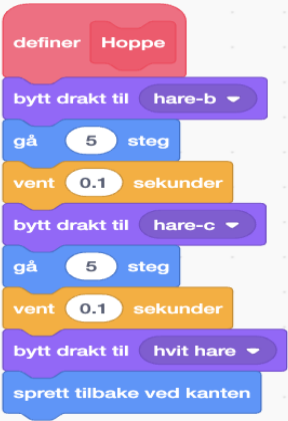
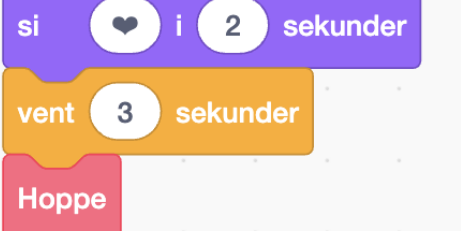
En variabel er en tom boks som kan inneholde forskjellige verdier. Vi kan endre og manipulere variabelverdiene basert på hva som skjer i programmet. Variabler kan høre til en figur, høre til alle figurene. Det er greit å gi variabelen et navn som forteller noe om rollen variabelen har i programmet. Alle variablene ligger på en liste under kategorien **variabler**. Om man krysser av en bestemt variabel kan man se variabelens verdi på scenen, dette kan være nyttig for å se om programmer fungerer som det skal.

*Tips: det er lurt å "sett \_ til \_" i starten av programmet slik at programmet nullstiller seg før det kjøres på nytt.*

## Funksjoner

Dette finner vi under **Mine klosser** og **lag en kloss**. Her kan man legge til et felt, tall eller tekst. Ordet for funksjonen blir nå gjort om til en lokal variabel som bare gjelder for funksjonen og som kan brukes for å gjøre beregninger. Hensikten med å lage en funksjon er å generalisere og effektivisere koden.



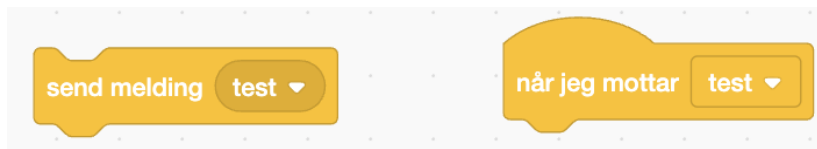
	
<p>Her er en funksjon som brukes i programmet. "hoppe"</p>	<p>Her brukes funksjonen i programmet</p>

## Lister

En liste kan brukes for å lagre mange verdier i en og samme variabel. På samme måte som en handleliste. Lister har funksjoner som kan fortelle oss om lengden på listen, sette inn eller endre en verdi, endre en verdi på en viss plass, legge til eller slette et element. *Tips: slett alt i listen til å begynne med, ellers vil den beholde alle verdier fra forrige gjennomkjøring.*

## Meldinger

Om man har flere figurer i et program kan de kommunisere ved hjelp av meldinger. "Send melding \_" finner man under kategorien **Hendelser**.



Figur 13: Viser sender og mottaker i Scratch.

## Tips:

1. Velg språk. Velg språk til det språket eleven er mest komfortabel med. Det finnes også begge målformer på norsk. Trykk på globusen øverst til venstre.
2. Det går an å bruke desimaltall. Men husk å bruke punktum som desimaltegn.
3. Hvordan slette klosser:
  - Høyre trykk på hver enkelt kloss og velg **slett**
  - Høyreklipp på skriptområdet og velg **slett xx klosser**
  - Dra en kloss eller et skript over til venstre side av skjermen
  - Du kan starte et nytt prosjekt ved å trykke **Fil -> Ny**
4. Det finnes ikke en angre funksjon i menyene. Bruk Ctrl + Z for å angre.

5. Ryggsekken: Om du vil spare på noe i Scratch og for eksempel bruke dette på en annen figur eller en annen fil kan det legges i ryggsekken. Dobbeltklikk på ryggsekk området nederst på skjermen og dra hva du ønsker å lagre ned hit.
6. Bevegelse:
  - 10 skritt er lite i Scratch, hvis du vil at en figur skal bevege seg over skjermen bruk nærmere 100.
  - Om du ønsker synlig bevegelse legg inn vent i programmet (finnes under **styring**)
7. Bruk koordinater for å bestemme hvor figurer begynner på skjermen. Koordinatene ser man på figurvinduet.
  - Bruk "**Pek i retning \_**" for å bestemme hvilken retning figuren skal bevege seg. 90 grader er til høyre.
8. Figurer kan være både synlige og usynlige. Trykk på øyet i figurvinduet. Figurer som er skjult kan fremdeles utføre handlinger.
9. Dobbeltklikk på skjermen og velg **Rydd opp klossene** for å rydde opp i kodene på skjermen. Hvert skript vil da legges under hverandre, de kan vurderes om en slik organisering er hensiktsmessig.

## Vedlegg

### Lag ditt eget program

**LAG: Lag en simulering som viser harepest hos harene. Du kan enten endre programmet du allerede har jobbet i eller lag ditt helt eget program.**

#### *Harepest:*

*Smitte spres mellom både dyr og mennesker, men er mest vanlig blant hare dyr og smågnagere og skyldes bakterien *Franciscella tularensis*. En hare som er smittet av harepest blir svært syk og dør vanligvis av blodforgiftning etter et par dager. Harene som blir smittet beveger seg saktere enn vanlig noen dager før de dør.*



Tips: her er et mulig løsningsforslag, finn gjerne din egen måte å løse problemet på <https://scratch.mit.edu/projects/814850495>

## Litteraturliste

- Gjøvik, Ø., & Høyland, J. (2022). *Kloss for kloss: blokkprogrammering for lærere*. Universitetsforlaget.
- Gregory, T. R. (2009). Understanding natural selection: essential concepts and common misconceptions. *Evolution: Education and outreach*, 2(2), 156-175.  
<https://doi.org/10.1007/s12052-009-0128-1>
- Harms, U., & Reiss, M. J. (2019). The present status of evolution education. *Evolution education re-considered: Understanding what works*, 1-19.  
[https://doi.org/10.1007/978-3-030-14698-6\\_1](https://doi.org/10.1007/978-3-030-14698-6_1)
- Kunnskapsdepartementet. (2019). *Læreplan i naturfag (NAT01-04)*. Fastsatt som forskrift. Læreplanverket for Kunnskapsløftet 2020.  
<https://www.udir.no/lk20/nat01-04>
- Reydon, T. A. (2021). Misconceptions, conceptual pluralism, and conceptual toolkits: bringing the philosophy of science to the teaching of evolution. *European Journal for Philosophy of Science*, 11(2), 48.  
<https://doi.org/10.1007/s13194-021-00363-8>
- Sentance, S., Waite, J., & Kallia, M. (2019). Teaching computer programming with PRIMM: a sociocultural perspective. *Computer Science Education*, 29(2-3), 136-176.  
<https://doi.org/10.1080/08993408.2019.1608781>
- Bøhle, K. & Ervik, H. (2020). *Naturlig utvalg*. NDLA. Hentet 12. mars 2023 fra  
<https://ndla.no/nb/subject:1:f18b0daa-6507-4025-8998-b8a11c8ccc70/topic:1:efd20b4a-3160-468c-ad68-92f92970faf4/topic:1:35134346-5eae-4115-85fc-558493942f2f/resource:ea274d43-3217-430f-be49-ad9c13be86c7>

## Vedlegg 4:

I dette masterprosjektet har vi jobbet sammen. Vi har hatt et godt samarbeid i hele prosessen, hvor vi har hatt åpen kommunikasjon og vært ærlige med hverandre. Vi har stort sett jobbet sammen på lesesal, og ellers har vi hatt en tett dialog igjennom hele arbeidet.

Vi har sett på det som en styrke og være to stykker som jobber sammen om masterprosjektet. Spesielt ettersom vi har skrevet en utviklingsmaster hvor vi har utviklet en læringsressurs. Det har vært fint å være to i prosessen masteroppgaven. Vi har dermed vært to stykker som har lest og forstått teori før vi har diskutert, argumentert og vurdert hvilke valg vi skal ta i utviklingsprosessen med hverandre. Det å være to som arbeider sammen har gitt oss flere perspektiver og sider å vurdere når vi tar valg i prosessen, som trolig har gitt oss dypere innsikt og bedre forståelse for utviklingsprosessen. I tillegg har det at vi har jobbet i par gitt oss bedre problemløsningsevne som har vært spesielt nyttig i vår master og programmeringen av simuleringen. Vi har dermed fått jobbet på en mer effektiv måte i utviklingen. Vi vil også påstå at ved å være to som utvikler en læringsressurs har vi hatt ulike ideer som gjør at vi har fått mer og bedre løsninger på utviklingen.

Det å være to som samskriver masteroppgaven, har også vært nyttig i datainnsamlingen. Ved å være to som observerer får man flere øyne som ser, og man kan fokusere på ulike fokus. Dette har vært en styrke for masteroppgaven vår.

Når det kommer til selve skrivingen av masteroppgaven, har vi fordelt avsnittene og de ulike delene mellom oss. I fordelingen av avsnittene har vi blitt enige om hva innholdet skal være, men vi har skrevet ulike deler. Likevel har vi alltid lest over, diskutert, kommentert og revidert det den andre har skrevet. På denne måten har vi fått god kjennskap til alle delene av masteroppgaven. Vi vil derfor påstå av begge har vært delaktige i hele masteroppgaven og at vi har hatt en jevn arbeidsfordeling hvor begge har bidratt i like stor grad i masteroppgaven.

Når det kommer til ansvarsfordeling av arbeid, har vi brukt strategien om å sette opp en liste over gjøremål som må gjøres med masteren. Videre har vi i felleskap blitt enige om hvem som skal gjøre hvilket arbeid. I tillegg har vi satt opp gjøremål, som vi begge kan starte på om vi blir ferdige med våre egne deler. Vi har dermed begge hatt likt ansvar for hele prosessen. I tillegg har vi også hjulpet hverandre med deler som vi synes er utfordrende.

Vi konkluderer derfor med at vi har hatt et godt samarbeid igjennom hele prosessen med masteroppgaven. Vi er enige om at begge har vært like delaktige i masterprosjektet og vi har begge like stort eierskap til masteroppgaven.

Trondheim, Mai 2023

Katarina Sørhøy og Rikke-Helen Aakernes Nielsen

