VELKOMMEN!

Agentbaseret modellering i matematik, fysik og kemi

Webinar, torsdag d. 07. december, 2023

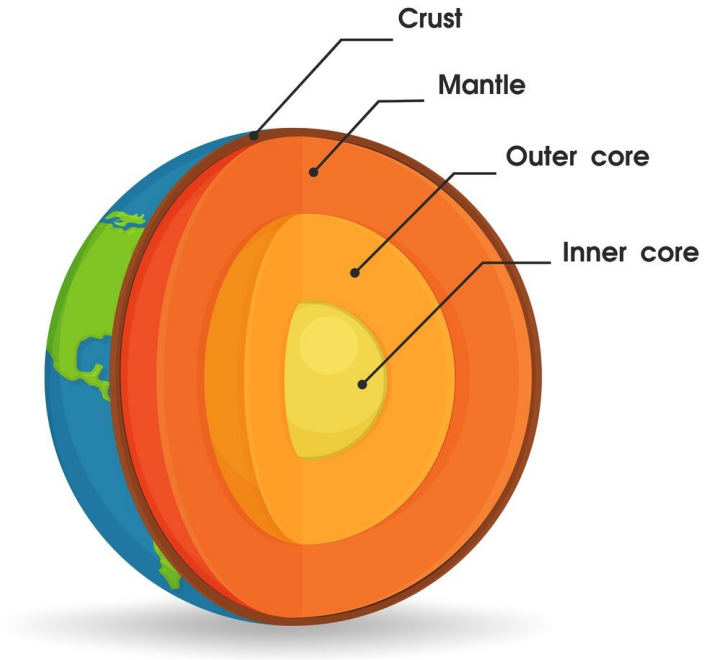
Jonas Ørbæk Hansen
Underviser i fysik og kemi ved Silkeborg Gymnasium
Tidligere: Center for Computational Thinking & Design, Aarhus Universitet
jha@sg.dk

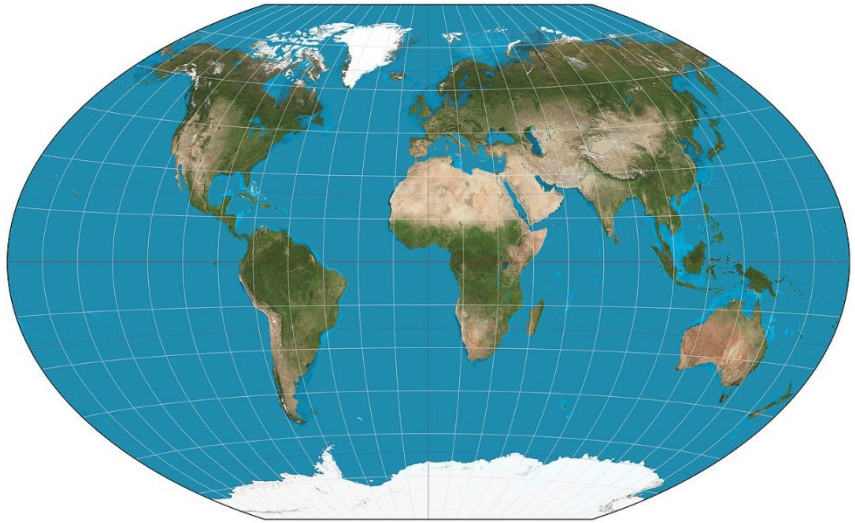
Programmet NetLogo og arbejdsark

- NetLogo kan installeres fra: https://ccl.northwestern.edu/netlogo/6.4.0/

- Modeller og arbejdsark jeg har vist i dag sendes til jer efterfølgende.

HVAD ER MODELLER?





"Essentially, all models are wrong, but some are useful"
(George Box, 1987)

SYSTEMNIVEAU (MAKRO) OG AGENTNIVEAU (MIKRO)

SYSTEM-NIVEAU	AGENT-NIVEAU
En klump radioaktivt materiale. Strålingen er givet ved henfaldsloven (makroskopisk model).	De enkelte radioaktive kerner. Hver kerne har en bestemt sandsynlighed for at henfalde hvert tidsenhed (mikroskopisk model).
Skattelettelser øger landets BNP (makroskopisk model).	Borgerne, der får forøget deres disponible indkomst. Dermed stiger deres forbrug og virksomhedernes omsætning (mikroskopisk model).
En opløsning med enzymer og substrater. Enzymaktiviteten afhænger af temperaturen (makroskopisk model).	De enkelte enzymer. Deres rummelige struktur afhænger af temperaturen (mikroskopisk model). Molekylernes hastighed afhænger af temperaturen (mikroskopisk model).
Fake news og postfaktuelle nyheder bliver mere udbredte (makroskopisk model).	Borgerne deler sig i højere grad efter værdier og holdninger, der ikke nødvendigvis hviler på fakta (mikroskopisk model).
Et kammer med en gas. Trykkets afhængighed af temperaturen er givet ved ædelgasligningen (makroskopisk model).	De enkelte atomer eller molekyler i gassen. De bevæger sig hurtigere, jo varmere det er, og rammer tryksensoren oftere og hårdere (mikroskopisk model).
En galakse. Rotationen afhænger af den samlede masse og størrelse af galaksen (makroskopisk model).	Stjernerne i en roterende galakse. De har en masse og hastighed og påvirker hinanden med massetiltrækningskraft (mikroskopisk model).

AGENTBASEREDE MODELLER (ABM)

Agenter er autonome individer (fx fugle)

Agenterne har nogle *egenskaber* (fx udseende, størrelse, position, retning , fart, osv.) og en *adfærd* (fx søg mod syd, styr uden om træer, ret ind efter dine "naboer ", osv.)

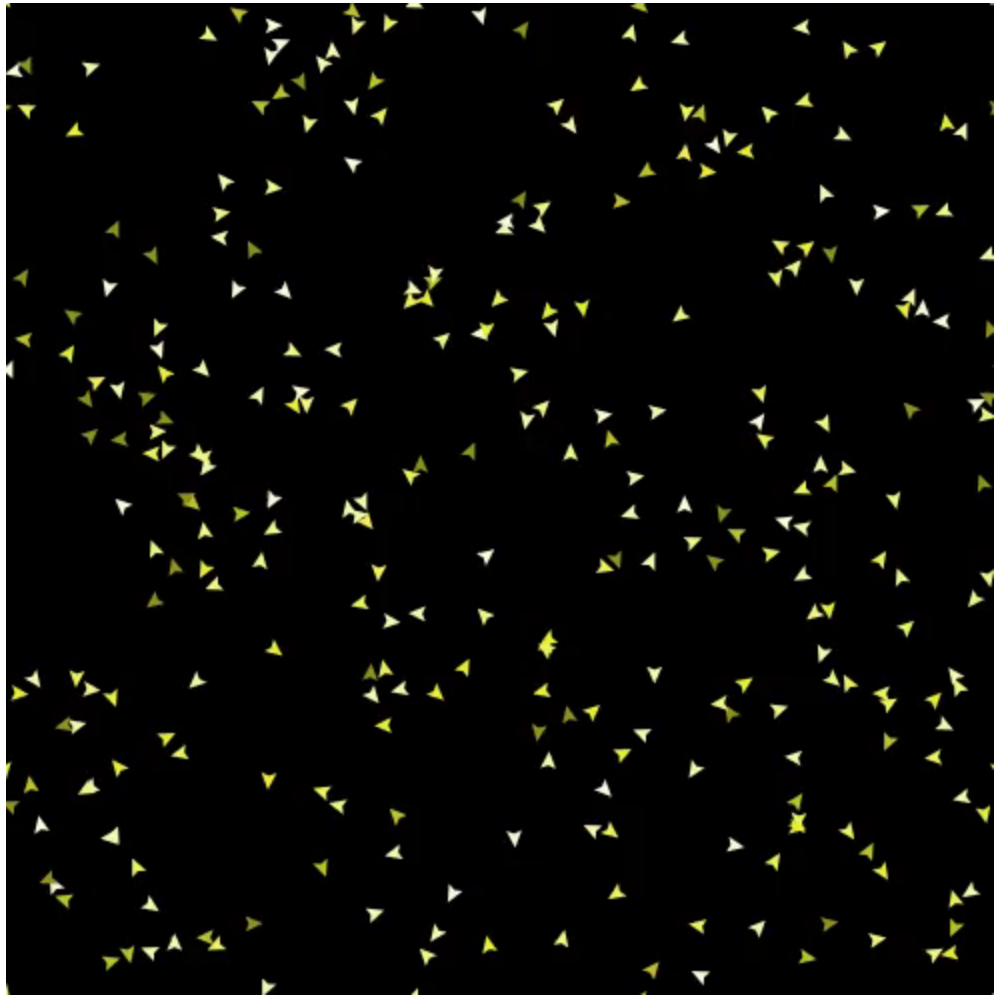
Agenternes *egenskaber* og *adfærd* definerer, hvordan de ser ud og opfører sig, og hvordan de interagerer med hinanden og deres omgivelser.

AGENTBASEREDE MODELLER (ABM)

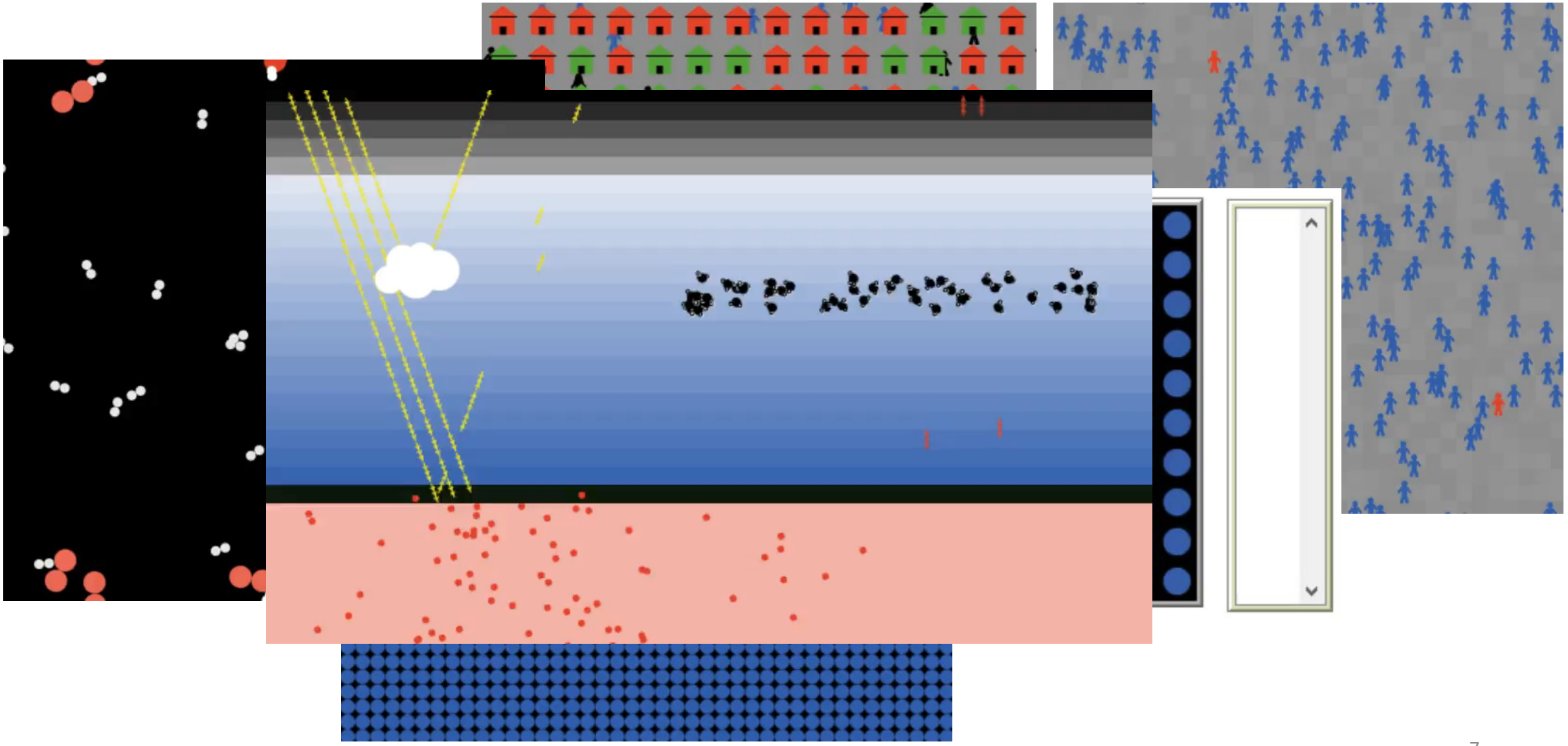
Vi programmerer en enkelt fugl (udstyrer den med egenskaber og adfærd) og fortæller modellen, at vi vil have 300 fugle. **Vi slipper dem løs og ser, hvad der sker…**

- Vi kan med ABM undersøge hvilken adfærd på agentniveau (mikroniveau) der fører til det fænomen, vi observerer på systemniveau (makroniveau).

- Vi kan undersøge hvordan ændringer på agentniveau påvirker systemniveau (lave forudsigelser/fremskrivninger)



AGENTBASEREDE MODELLER (ABM)



AGENTBASEREDE MODELLER (ABM)

En agentbaseret computermodel over et fagligt fænomen giver eleverne:

- egenskaber og adfærd som intuitiv tilgang til et fænomen

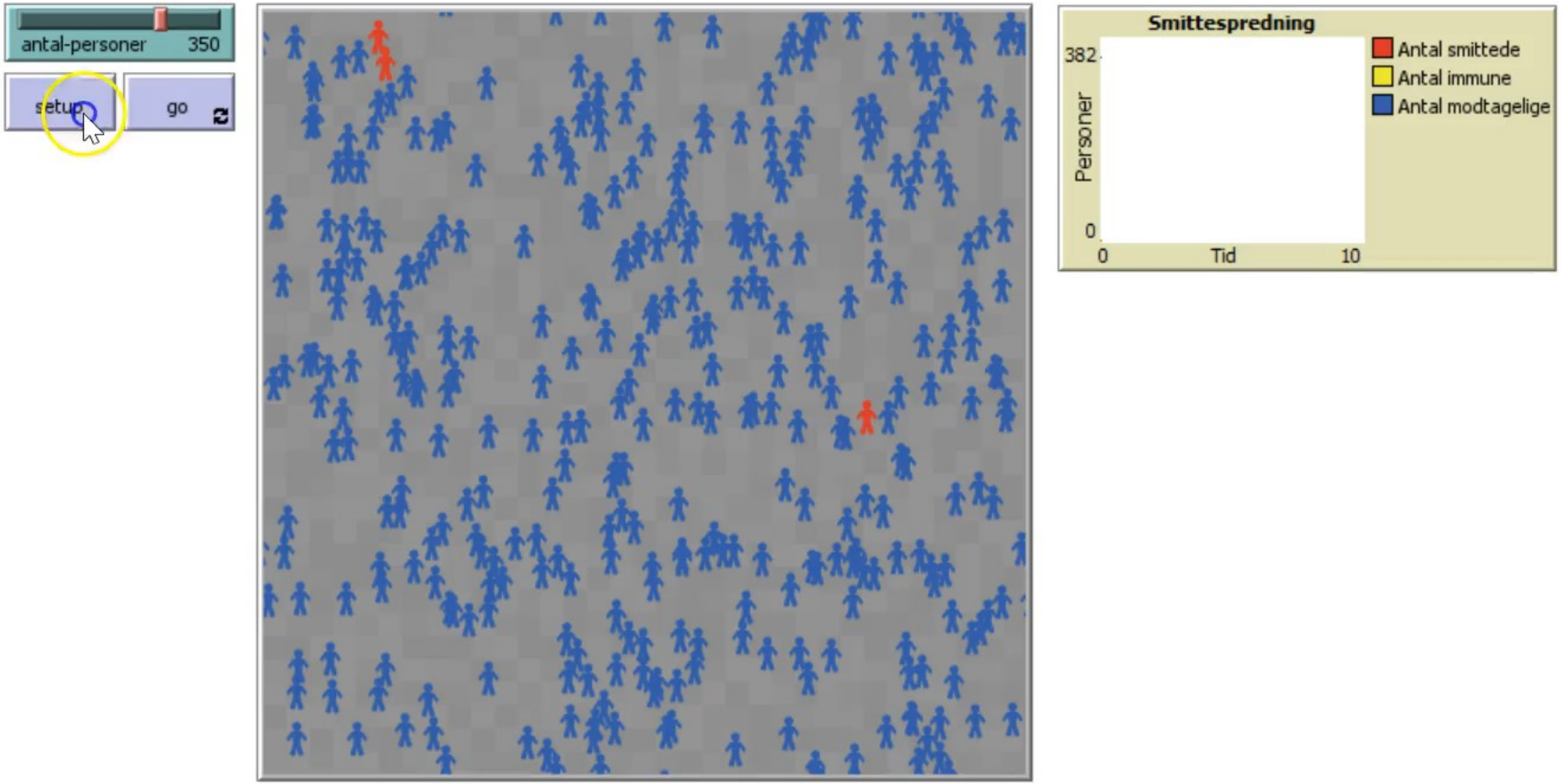
  selv for begyndere inden for modellering og programmering

- en tilgang til at beskrive og forstå sammenhængen mellem mikro- og makroniveau af et fagligt fænomen.

Store, komplekse fænomener reduceres til et spørgsmål om, hvordan de enkelte agenter agerer i bestemte situationer.

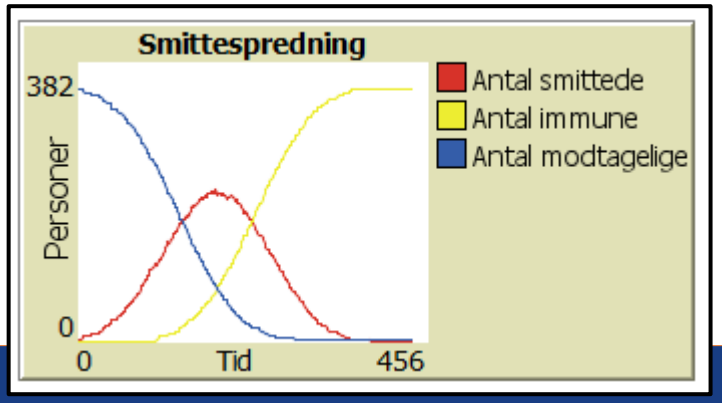
*ABM'er gør arbejdet med og programmeringen af modeller **konkret** for eleverne og giver dem mulighed for at arbejde med relativt komplekse fænomener og problemstillinger, som ellers ville være vanskelige at behandle i undervisningen (fx pga. svær matematik).*

SMITTESPREDNING



SMITTESPREDNING

Matematisk analyse vs. agentbaseret model



Koblede differentialligninger

Beskriver fænomenet på systemniveau (makroskopisk model)

$$\frac{dS}{dt} = -\frac{\beta}{N} I(t)S(t)$$

$$\frac{dI}{dt} = \frac{\beta}{N} I(t)S(t) - \gamma I(t)$$

$$\frac{dR}{dt} = \gamma I(t)$$

ABM model

Her beskrives reglerne for de enkelte agenters opførsel (mikroskopisk model)

```
ask persons with [ color = blue ] [
  if any? persons-here with [color = red] [ set color red ]
]
```

```
ask persons with [color = red] [
  if infection-time > 100 [ set color yellow ]
]
```

Her kan alle elever være med.

HVORFOR ABM?

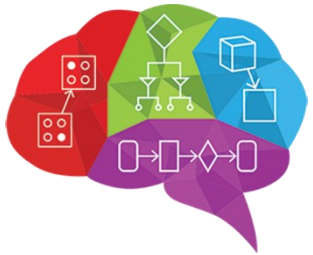
- Formålet med aktiviteterne er IKKE at forsyne eleverne med meget virkelighedstro computermodeller, som de kan bruge alene gennem modellens interface og som spytter meget præcise resultater ud.
- Formålet er at give eleverne mulighed for at ”åbne modellerne op” og se hvad der gemmer sig nede i dem. Se hvordan de er lavet, og hvordan de virker. Hvordan det faglige indhold er kodet ind i modellerne. Give dem mulighed for at modificere modellerne og arbejde kreativt med dem. Forbedre dem og udbygge dem, så de kan bruges til andre formål. Gøre dem til deres egne. Reflektere over modellens styrker og mangler. Og i sidste ende (for de dygtigste elever) blive i stand til at skabe deres egne faglige modeller (fx i en SRP).
- Det betyder (nogen gange) at eleverne ikke ser på alle aspekter af et fænomen, men zoomer ind på et enkelt delelement.
- => Innovationsprojekter, faglige samarbejder, SRP-opgaver (og videre studier/karriere)

HVAD FÅR ELEVERNE UD AF SÅDAN EN AKTIVITET?

- Afvekslende arbejdsform, hvor der arbejdes med en faglig problemstilling på en anderledes måde.
- Eleverne har lært noget fagligt, som de ikke kunne uden denne model:
  - Dybere indsigt i fænomenet og forståelse for sammenhænge mellem forskellige dele af fænomenet (og mellem mikroniveau og makroniveau , agent og system) - og forstår hvordan det er arbejdet/kodet ind i modellen.
  - Kunne arbejde med et komplekst fænomen (som måske ellers ville krævede svær matematik)
  - Større fagligt udbytte end med en lærerbog og PHET-simuleringer?
- Får snust til kode: Et programs opbygning, programstrukturer, kodesyntaks og algoritmer (fx *variabler* og *if-sætninger*). Oplever hvordan ændringer i koden fører til ændringer i modellen og dens opførsel/resultater.
- Oplever at computermodeller er noget man selv kan lave. Man behøver ikke været styret af, hvad andre har lavet. (=> SRP). Står tilbage med noget, de føler stort ejerskab overfor.
- Faglig og almen dannelse: Forståelse for hvad der sker i maskinrummet af en faglig computermodel/simulering (et kig i den sorte boks). Meget relevant for deres videre studier.

  Ser at der ligger nogle antagelser indbygget i computermodellen, som kan passe mere eller mindre godt med virkeligheden. => Gælder for ALLE computermodeller i faget og i samfundet.

Agentbaserede Modeller og Computational Thinking



Færdigheder der gør eleven I stand til at bruge computeren til at *tænke* med. Til at *udforske*, *analysere*, *vurdere* og *erkende* ting i faget med.

Eleverne skal være kreative og skabende. Ikke afhængige af hvad der allerede findes. Styrke elevernes handleevne og give nye muligheder.

Hvad er der i ”maskinrummet” af en computermodel? Hvor meget kan vi stole på resultater fra computermodeller (i faget/samfundet)? Hvad er modellens muligheder og begrænsninger?

Didaktiske principper

Simpel model med få elementer på Interface og *kort overskuelig kode*.

Eleverne kommer hurtigt frem til at forstå både fænomenet samt koden bag modellen (hvordan forskellige kodestykker styrer modellens opførsel). Det gør det muligt at designe aktiviteter, som både træner elevernes læring i faget og modellering og kodning.

Use-modify-create (UMC).

Eleverne går fra at være *brugere* af modellen (gennem dens interface) til at lave gradvist mere omfattende og komplekse ændringer i koden. Fra *udforskende* til *kodende* til *modellerende.*

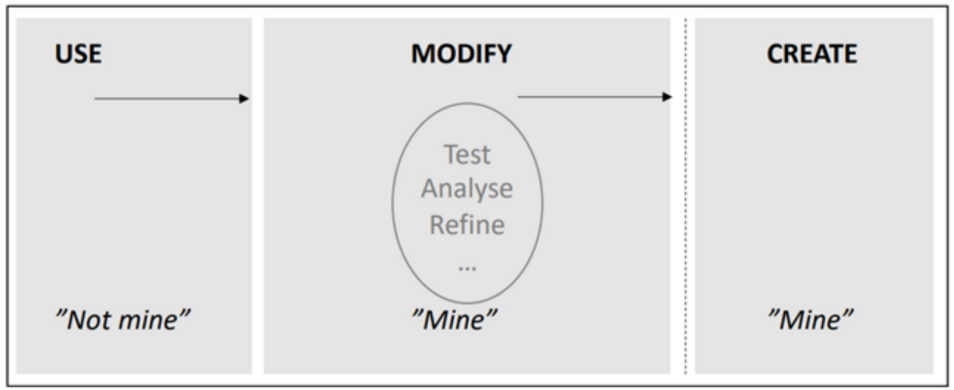
Eleverne går fra at *"lege" og prøve sig frem* (tinkering) til mere *velovervejede og fagligt begrundede ændringer*

Meningsfulde ændringer i forhold til en konkret faglig problemstilling (*CMC*).

Eleverne står med en bedre model, når de er færdige (kan mere eller give mere præcise resultater) => *MOTIVATION*

Agentbaseret modellering i NetLogo: Programmeringen tager udgangspunkt i de enkelte agenter.

Refleksion over modellens styrker og begrænsninger (mangler/forsimplinger/etc.) => Metoder i faget, betydning for samfundet



Princippet om UMC (efter Lee et al., 2011). I figuren er angivet, hvordan elevernes opfattelse af – og følelse af ejerskab for – computermodellen ændres undervejs i læringsaktiviteterne.

Hvordan kan der arbejdes med koden?

Eleverne får udleveret en kode, som de kan tage udgangspunkt i.

De starter med at foretage få og simple ændringer i den eksisterende kode. Det kan fx være:

- ændringer i en enkelt linje (et enkelt tal eller en betingelse i en if-sætning)
- agenternes form eller farve (modellen ser anderledes ud)
- en hastighed eller en sandsynlighed (modellen opfører sig anderledes => man når frem til et andet resultat)

Her oplever eleverne, at de faktisk kan forstå (dele af) en computerkode og lave ændringer deri, der virker (succesoplevelse). De ser, at der er en sammenhæng mellem koden og modellens udseende/opførsel.

Dernæst kan man bede dem om lidt mere omfattende ændringer, fx:

- flytte eller kopiere en kodestump fra ét sted til et andet og rette den til
- tilføje en linje kode
- udvælge og placere udleverede kodestykker i fornuftig rækkefølge, så det giver mening ift. en given opgave

Sådanne aktiviteter skal stilladseres.

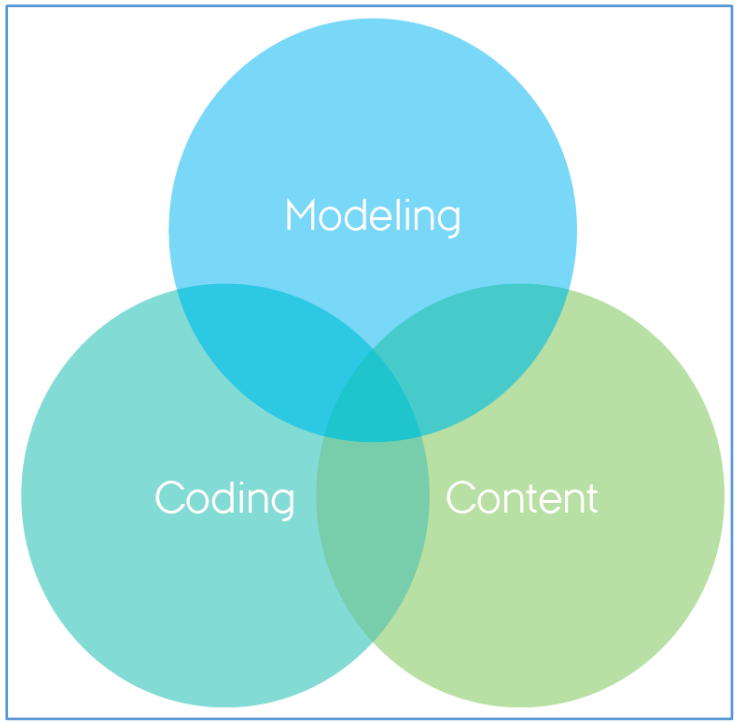
Nogle gange kan det lykkes at få enkelte elever til at skrive et større stykke kode fra bunden af. Men ikke noget man kan forvente af alle elever. Det er ikke realistisk, at en elev kan lave en computermodel fra bunden af selv.

OBS: Der skal være et mål / en faglig mening med ændringerne.

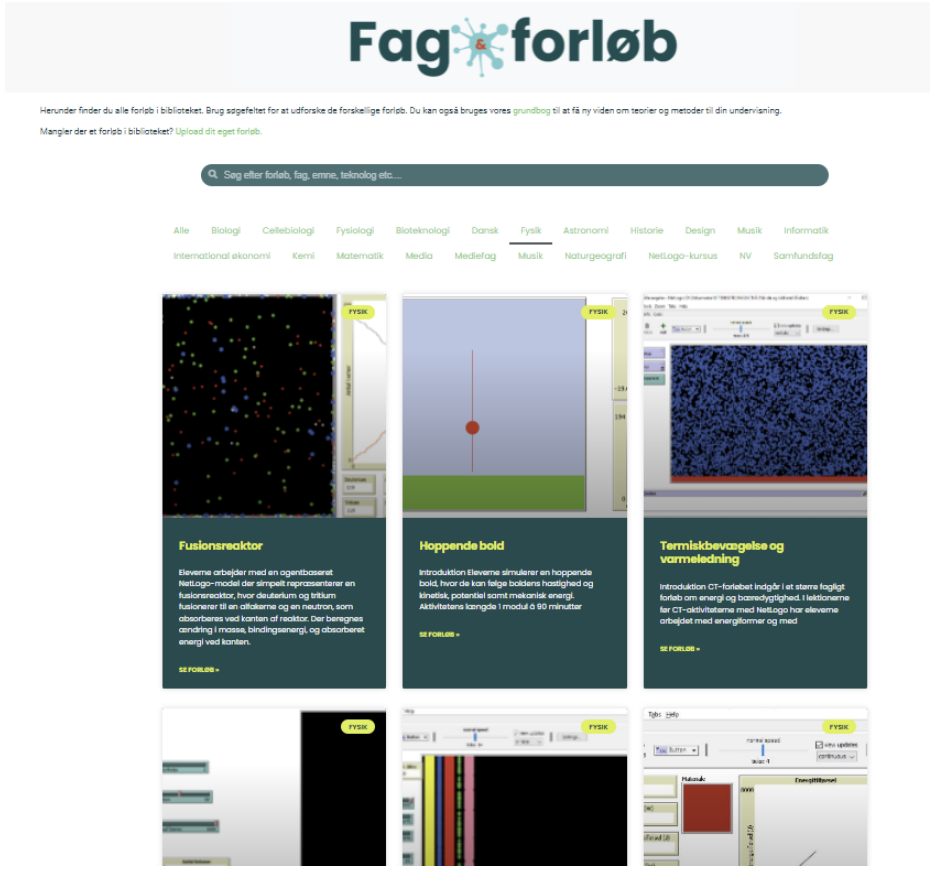
SKAL UNDERSTØTTE DEN FAGLIGE LÆRING

- Eleverne udvikler CT-kompetencer, mens de arbejder med faget
- Eleverne udnytter CT til at udforske faget på nye måder og i nye retninger

Udgangspunkt i faglige modeller og modellering

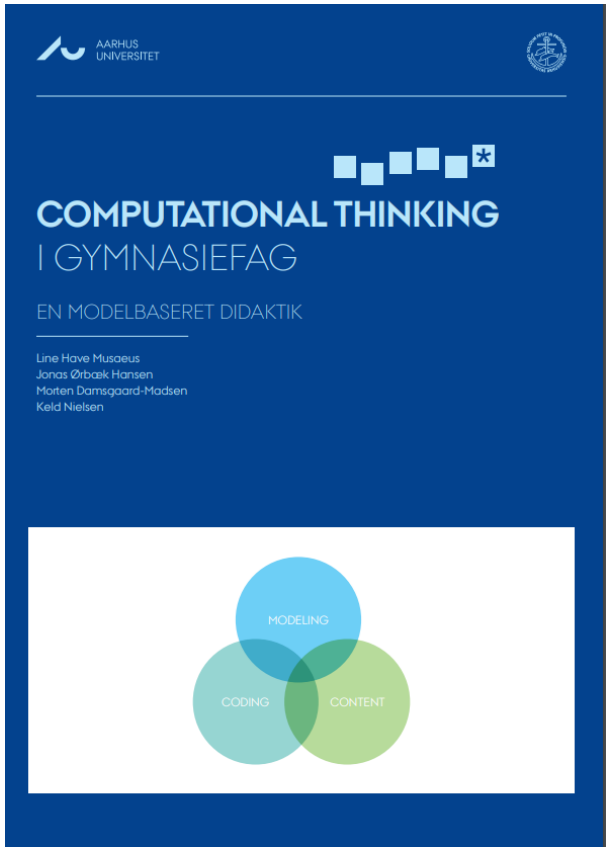


VIL DU VIDE MERE?



GRASP

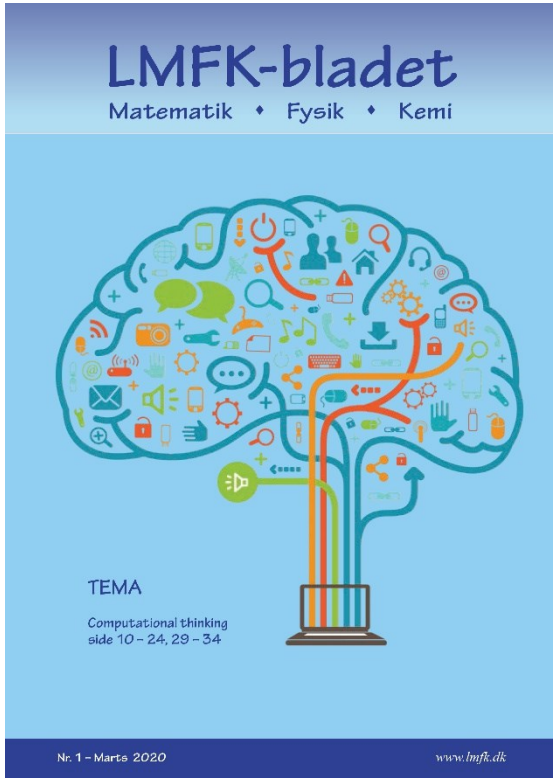
Modeller med tilhørende aktivitetsbeskrivelser og arbejdsark udviklet af danske gymnasielærere.
www.graspit.dk



Didaktikhæftet (udgivet af Center for Computational Thinking & Design, Aarhus Universitet, 2020) er blevet til på baggrund af erfaringerne fra fire udviklingsprojekter.

Didaktikhæfte:
https://pure.au.dk/portal/da/publications/computational-thinking-i-gymnasiefag(f7cf383e-18f9-439e-84a6-8fd264691678).html

VIL DU VIDE MERE?



Computational Thinking i gymnasiefag (fem artikler i LMFK-bladet, 2020). Skrevet af fire gymnasielærere på baggrund af deres erfaringer med at inddrage CT i deres undervisning.

Computational thinking i matematik, naturfag og samfundsfag

– hvorfor, hvad og hvordan?

Line Have Musaeus, Aarhus Universitet

Jonas Ørbæk Hansen, Silkeborg Gymnasium

Keld Nielsen, Aarhus Universitet

Abstract: *I et projekt der strakte sig over fire år, udviklede og afprøvede vi sammen med lærere fra gymnasiet undervisningsforløb der integrerer modelbaserede computationelle metoder i undervisningen i STEM-fagene. Der er publiceret og udviklet en didaktik for den integrerede undervisning, skrevet forskningsartikler herom samt udviklet ca. 100 nye undervisningsforløb der udnytter didaktikken. Samtidig er der udviklet et efteruddannelseskoncept som sætter lærerne i stand til selv at fortsætte arbejdet på egen skole. Projektet har vist at det er muligt at forny fagenes metoder og samtidig udvikle elevernes computationelle modelleringskompetencer ved at inddrage computationel tænkning og kodning i den faglige undervisning.*

Computational thinking i matematik, naturfag og samfundsfag - hvorfor, hvad og hvordan? (Artikel i MONA, 2023).

Vil du vide mere?

Du kan læse/høre/se mere om ABM og CT her:

- Computational Thinking og Modellering i STEM-fag i gymnasiet (kort YouTube-video fra Center for Computational Thinking & Design, Aarhus Universitet, 2020).
- Podcast-serie om computational thinking i forskning og uddannelse (Podcasts fra It-vest, 2020). Produceret af Anders Høeg Nissen, der er kendt fra DRs *Harddisken*.
- Computational Thinking — hvorfor, hvad og hvordan? (rapport fra It-vest, 2018).
- Caspersen, Michael E.: Computational thinking, kapitel 4.15 i Gymnasiepædagogik – En grundbog, 3. udg., side 470 – 478 (Hans Reitzels Forlag, 2017).

Vil du vide mere?

Syv særlige interessegrupper

- Grupperne er åbne for undervisere, forskere og andre interesserede fra alle uddannelsesniveauer
- En række eksperter fra vidt forskellige fagområder har påtaget sig at indgå i grupperne som ledelsesteams, så der er sikkerhed for, at gruppernes drives af dybt engagerede og kompetente folk.

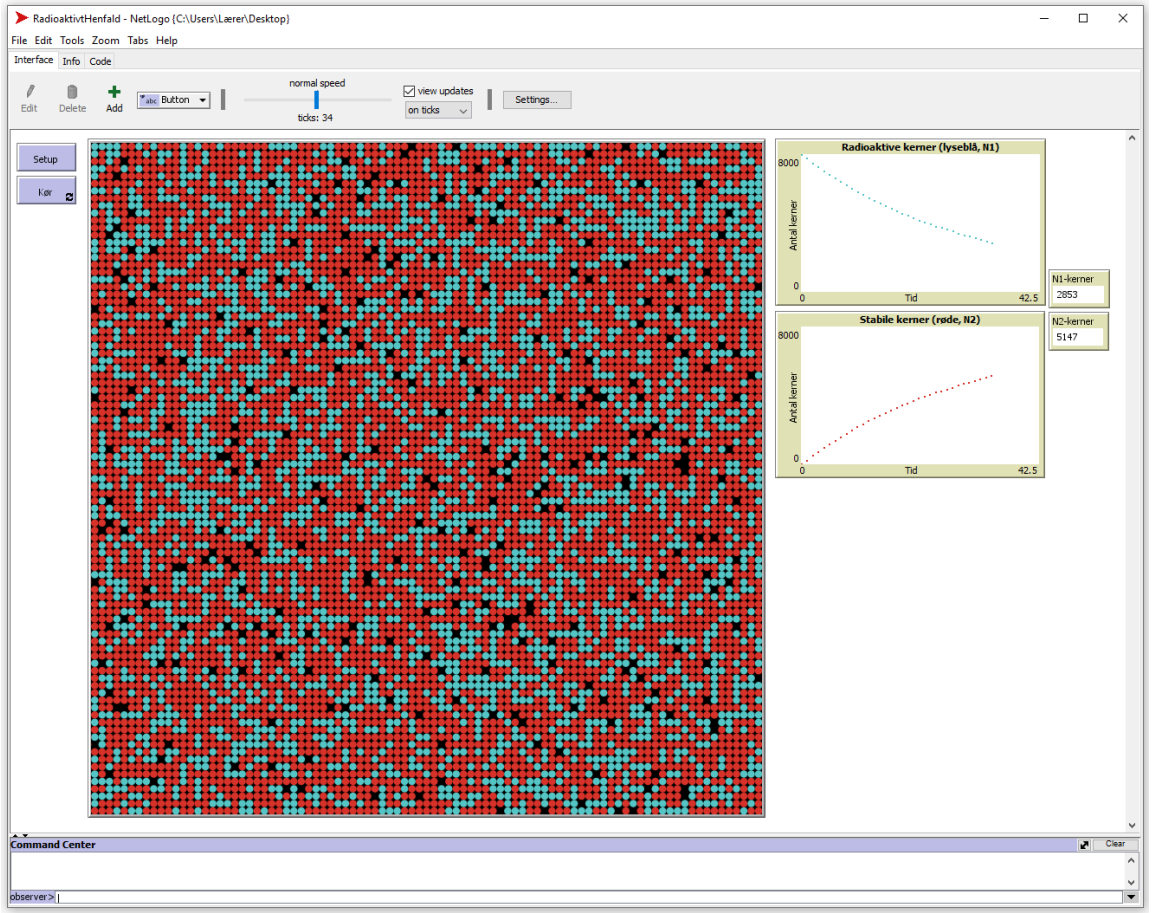
  - SIG: Agentbaseret modellering:
  - SIG: Almene humanistiske fag
  - SIG: Matematik
  - SIG: Musiske-æstetiske fag
  - SIG: Naturfag
  - SIG: Samfundsfaglige og merkantile fag
  - SIG: Sprog

https://www.it-vest.dk/aktiviteter/informatik-i-alle-uddannelser/saerlige-interessegrupper

EKSTRA SLIDES

Et eksempel: Radioaktivitet

Forløbet og materialer kan findes på:
https://library.ct-denmark.org/lmfk/



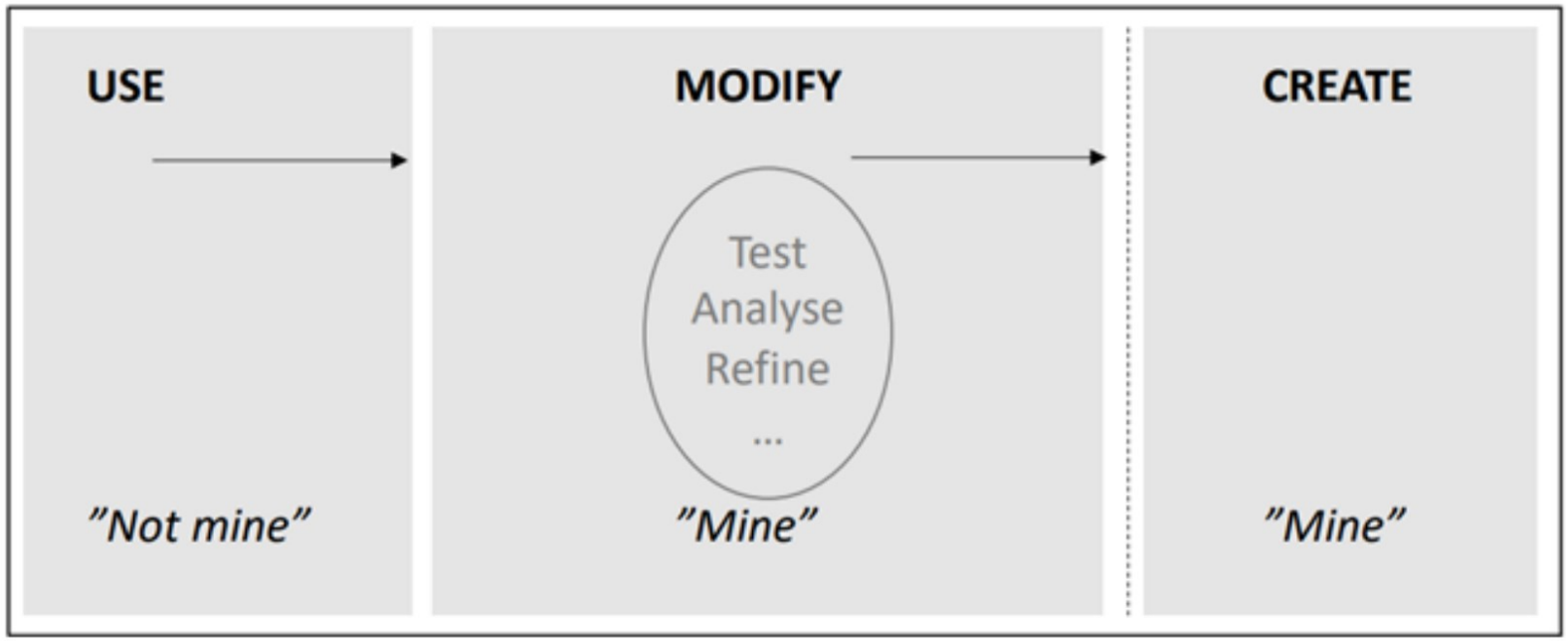

```
1  breed [N1s N1]
2  breed [N2s N2]
3
4  globals [
5    Sandsynlighed-for-henfald-N1
6    Startantal-N1
7    ]
8
9  to setup
10   clear-all
11
12   set Sandsynlighed-for-henfald-N1 3
13   set Startantal-N1 8000
14
15   create-N1s Startantal-N1 [
16     set shape "circle"
17     set color cyan
18     move-to one-of patches with [ not any? Turtles-here ]
19     ]
20
21   reset-ticks
22 end
23
24 to go
25   kør-henfald1
26   tick
27 end
28
29 to kør-henfald1
30   ask N1s [
31     if random-float 100.0 < Sandsynlighed-for-henfald-N1 [
32       set breed N2s
33       set shape "circle"
34       set color red
35       ]
36     ]
37 end
```

Loop når ”kør”-knappen er nede

Sværeste linje for eleverne at forstå

37 linjer kode

USE – Modify – create



Princippet om UMC (efter Lee et al., 2011). I figuren er angivet, hvordan elevernes opfattelse af – og følelse af ejerskab for – computermodellen ændres undervejs i læringsaktiviteterne.

Eleverne går fra at *"lege" og prøve sig frem* (tinkering) til mere *velovervejede og fagligt begrundede ændringer*

Progression i CT-aktiviteten

Eleverne får en fungerende model udleveret. De skal forbedre den og tilpasse den, så den kan anvendes til en anden problemstilling.

Progression (*use-modify-create*):

Use

1) ”Fri leg” – Leg med interface. Tryk på de forskellige knapper og se hvad der sker. 5 min.
2) Koble modellen til det faglige fænomen. Ud fra din faglige viden, hvilken farve har moderkernerne i modellen, og hvilken farve har datterkernerne? Fagligt (content).

Modify

3) Simple ændringer i computerkoden (form og farve af atomer + ændre på sandsynlighed for henfald og på startantal af kerner). Prøve sig frem, ok at fejle.

Use

4) Faglig undersøgelse. Køre model med forskellige startværdier. Hvordan ændrer graferne sig? Forklar… Lave regression og finde frem til henfaldsloven.
5) Lave eksperiment og sammenligne resultater med data fra computermodellen.

Modify

6) Mere omfattende ændringer i modellen (kode + interface), så den kan bruges til andre problemstillinger:
   a) Kulstof 14-datering
   b) Henfaldskæder (Te-131 => I-131 => Xe-131)

(Create)

Giver eleverne mulighed for at arbejde med et komplekst fænomen, som ellers ville kræve svær matematik.

algoritmer og evt. rutediagrammer

Motivation: De føler ejerskab over det færdige produkt. Føler det er noget de har skabt

Hvordan kan der arbejdes med koden?

Eleverne får udleveret en kode, som de kan tage udgangspunkt i.

De starter med at foretage få og simple ændringer i den eksisterende kode. Det kan fx være:

- ændringer i en enkelt linje (et enkelt tal eller en betingelse i en if-sætning)
- agenternes form eller farve (modellen ser anderledes ud)
- en hastighed eller en sandsynlighed (modellen opfører sig anderledes => man når frem til et andet resultat)

Her oplever eleverne, at de faktisk kan forstå (dele af) en computerkode og lave ændringer deri, der virker (succesoplevelse). De ser, at der er en sammenhæng mellem koden og modellens udseende/opførsel.

Dernæst kan man bede dem om lidt mere omfattende ændringer, fx:

- flytte eller kopiere en kodestump fra ét sted til et andet og rette den til
- tilføje en linje kode
- udvælge og placere udleverede kodestykker i fornuftig rækkefølge, så det giver mening ift. en given opgave

Sådanne aktiviteter skal stilladseres.

Nogle gange kan det lykkes at få enkelte elever til at skrive et større stykke kode fra bunden af. Men ikke noget man kan forvente af alle elever. Det er ikke realistisk, at en elev kan lave en computermodel fra bunden af selv.

OBS: Der skal være et mål / en faglig mening med ændringerne.

Hvorfor arbejde med koden?

Afvekslende arbejdsform, hvor der arbejdes med en faglig problemstilling på en anderledes måde.

Vi skal gøre kode ufarlig. Give dem muligheden for lege og eksperimentere med kode og algoritmer.

Eleverne gøres til *medskabere* og ikke blot *brugere*. En oplevelse af at modeller/dimser er noget man kan ændre på og forbedre/videreudvikle. Giver motivation og ejerskab.

At lave meningsfulde ændringer i koden af en computermodel kræver stor faglig indsigt.

- Tvinger eleverne til at tænke over det faglige fænomen i dybden (algoritmisk). Leder til gode faglige diskussioner.
- Et vindue til elevernes *mentale modeller* af fænomenet

Nye muligheder: Arbejde med komplekse faglige fænomener, udvikle nye ting.

Kunne vurdere en models styrker og begrænsninger, ser med egne øjne at en model afhænger af den kode, der er i den. Og at en computermodel har begræsninger.

Indsigt i hvad et program gør. Faglig og almen dannelse

HVORFOR NETLOGO?

- **Agentbaseret.** Programmeringen tager udgangspunkt i de enkelte agenter (egenskaber og adfærd) => mindre abstrakt, mere konkret og let at gå til for eleverne.
- **Udviklet til undervisningsbrug.** Lav tærskel for nybegyndere (programstrukturer og syntaks er relativ let at forstå) og "gratis" visualisering (grafer og animation).
- **Ligner andre programmeringssprog** tilpas meget til, at dét eleverne lærer i NetLogo (fx mht. et programs opbygning, programstrukturer, kodesyntaks og algoritmer) kan overføres til andre programmeringssprog.
- **Egner sig godt til CMC-tilgangen** (med fokus på modellering og arbejde med koden).
- **Stort bibliotek** med faglige modeller at tage udgangspunkt i og søge inspiration i.
- **Et fælles sprog**, så vi kan bruge hinandens materialer og udvikle et dansk praksisfællesskab.

NetLogo egner sig især godt til modellering af:

- Fænomener, der udvikler sig over tid.
- Komplekse systemer (mange agenter – og evt. agenttyper – der interagerer med hinanden)
- Emergerende fænomener: Muligt at undersøge sammenhænge mellem 'mikroskopisk' opførsel af individuelle agenter og 'makroskopiske' mønstre, som optræder på systemniveau pga. agenternes individuelle opførsel.