Computational Thinking i undervisningen

Anders B. Larsen, pædagogisk konsulent hos VidenCenter

Hvad er Computational Thinking (CT)?

- Cuny-Snyder-Wing definition af **Computational Thinking (CT)**:

> "Computational Thinking is the thought processes involved in formulating **problems** and their **solutions** so that the solutions are represented in a **form** that can be effectively carried out by an **information-processing agent**." [Wing2010]

- Definitionen er ret standard, men ikke operationel og stadig under udvikling.

Kilder: [Wing2010] og [Shute2017]

CT kompetencer

Computational Thinking (CT) omfatter **kompetencerne**:

Svarer lidt til **Analyse** og **fortolkning** i f.eks. dansk

Dekomposition	• Nedbryd problemet i mindre, overskuelige delelementer.
Abstraktion	• Find mønstrer blandt delelementer, udelad irrelevant information, uddrag generelle sammenhænge og lav en model.
Algoritme	• Design algoritme, som kan løse problemet vha. automation.
Evaluering	• Evaluer algoritmen iterativt ved afprøvning, fejlsøgning og tilretning (debugging og trial and error).
Generalisation	• Generaliser løsningen til lignende problemer.

Algoritmedesign er central i CT og er med til at adskille CT fra f.eks. *Mathematical Thinking*

Kilder: [Shute2017]

CT er en problemløsningsmetode

- CT er en **problemløsningsmetode**, hvor løsningen på et problem er en **algoritme** *i bred forstand* (program, bevis, software system, trinvis proces …), som i princippet skal kunne automatiseres.

> Cuny-Snyder-Wing definition af CT:
>
> "Computational Thinking is the thought processes involved in formulating **problems** and their **solutions** so that the solutions are represented in a **form** that can be effectively carried out by an information-processing agent."
> [Wing2010]

- CT er derfor *ikke* en generel problemløsningsmetode.

Hvad er en algoritme?

- En **algoritme** er egentligt bare en **kogebogsopskrift** (i snæver forstand):

Input		**Ingredienser**
Algoritme	⟺	**Opskrift**
Output		**Mad**

Hvad er en algoritme?

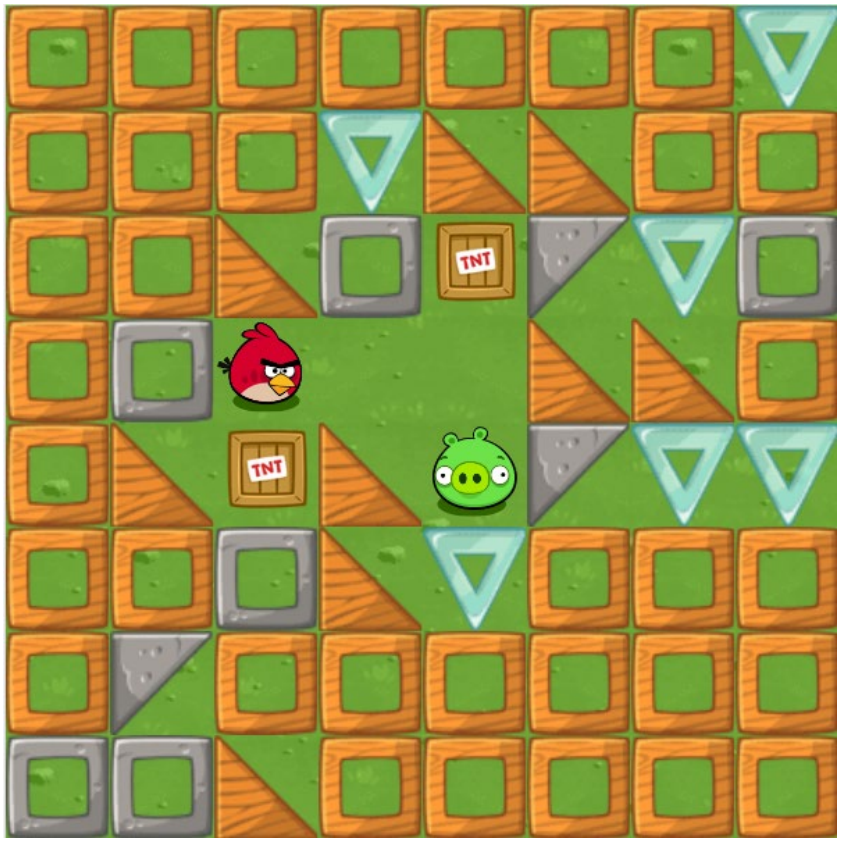
- En algoritmes **delelementer** er:

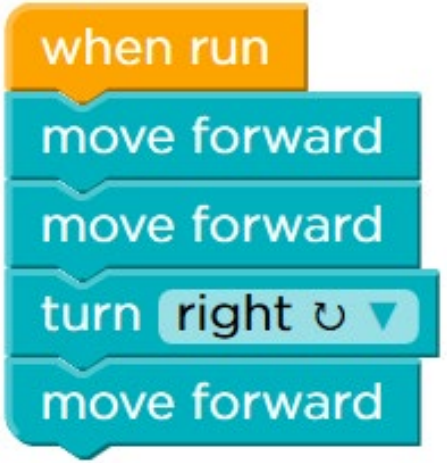
Sekvens	**Først** blandes vand, gær og salt, **derefter** tilsættes mel, **derefter**…
Forgrening	**Hvis** brødet koldhæves, **så** tilsættes 8 g gær. **Ellers** tilsættes 20 g gær.
Løkke	Brødet bages, **indtil** det er sprødt og gyldent (while-løkke).

- Øv dig i algoritmer med Classic Maze.

Hvad er en algoritme?

En **sekvens**:





Kilder: Classic Maze (trin 3)

Hvad er en algoritme?

En while-**løkke** (udenom en sekvens):

Kilder: Classic Maze (trin 13)

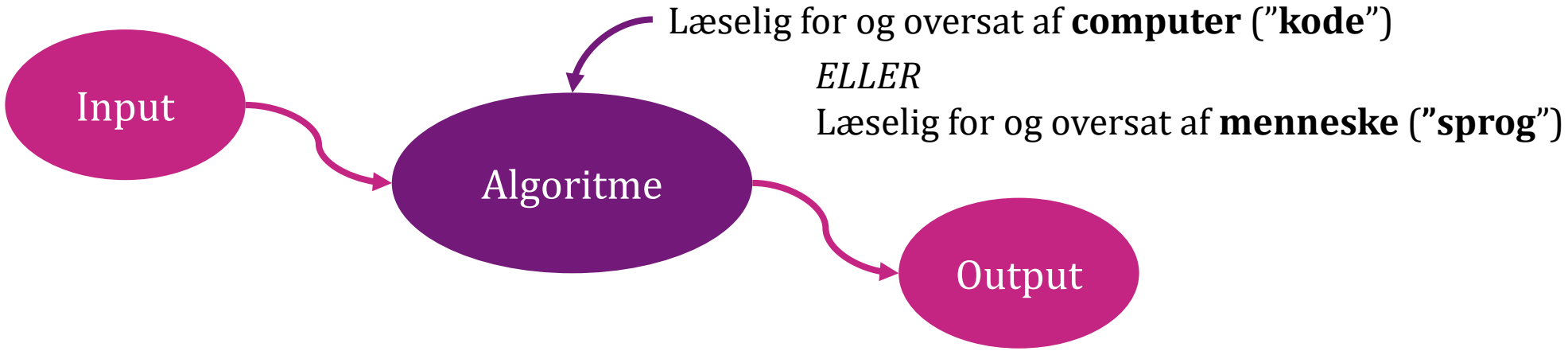
Hvad er en algoritme?

En **forgrening** (inde i en while-løkke):

Kilder: Classic Maze (trin 16)

CT er *ikke* programmering

- CT forudsætter *ikke* programmering eller brug af en computer.



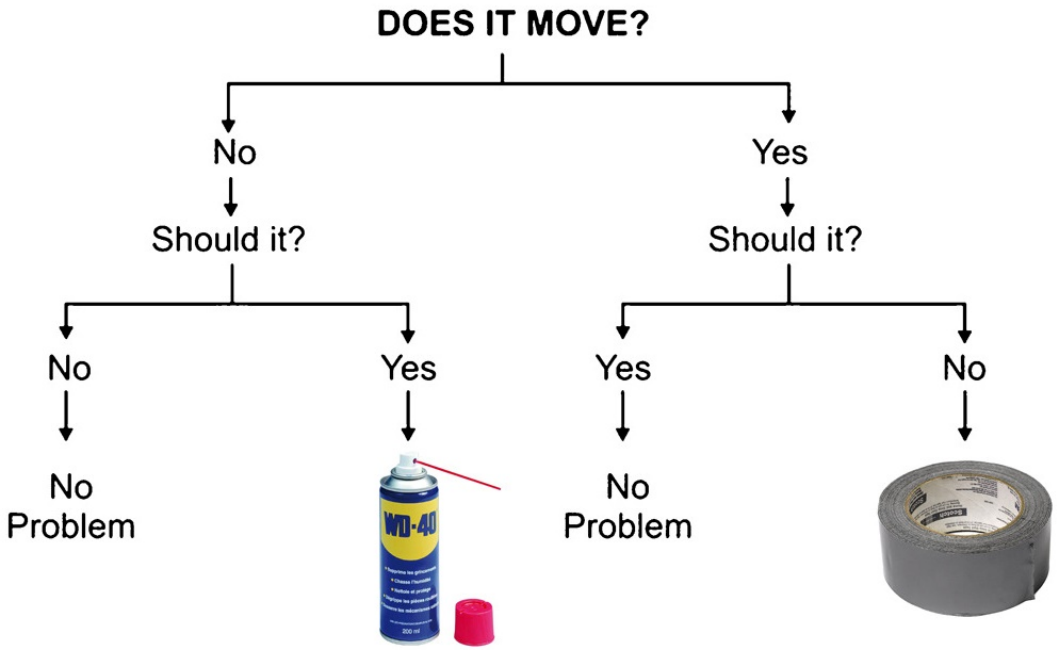
Cuny-Snyder-Wing definition af CT:

“Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an **information-processing agent**.” [Wing2010]

Kilder: [Wing2010] og [Lu2009]

Algoritme uden kode… men hvordan?

Flowchart (rutediagram)



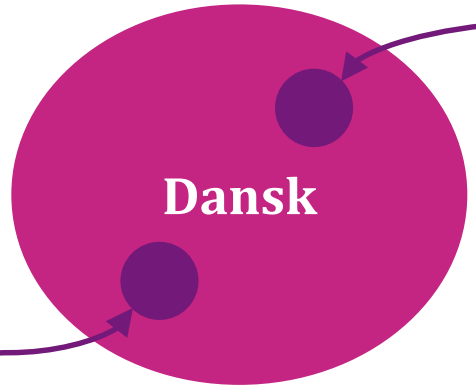
Pseudokode

- Tag postkassenøglen.
- Åbn hoveddøren.
- Gå ud til postkassen.
- Lås postkassen op.
- **Hvis** der er post i postkassen, **så** tages dette op, og postkassen låses. **Ellers** låses postkassen.
- Gå ind ad hoveddøren.
- Luk hoveddøren.
- Læg nøglen på plads.

Kilder: [Lu2009]

CT tilgang I: Identifikation af CT kompetencer i fag

- En gængs tilgang til udvikling af CT forløb i og på tværs af fag synes at være at identificere CT kompetencer i faget:



Dekomposition
(f.eks. analyse)

Abstraktion
(f.eks. fortolkning)

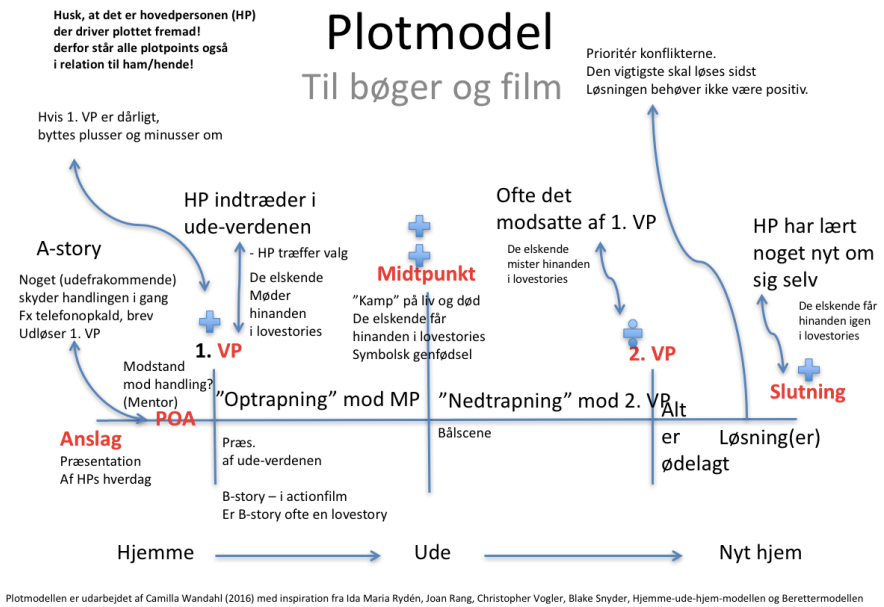
Algoritme
(f.eks. nulpunkts-beregning for parabel)

- CT kompetencer overføres ikke automatisk imellem fag/forløb.
- Et CT forløb kan derfor med fordel indeholde *samtlige* CT kompetencer.
- Ikke alle faglige forløb egner sig til CT.

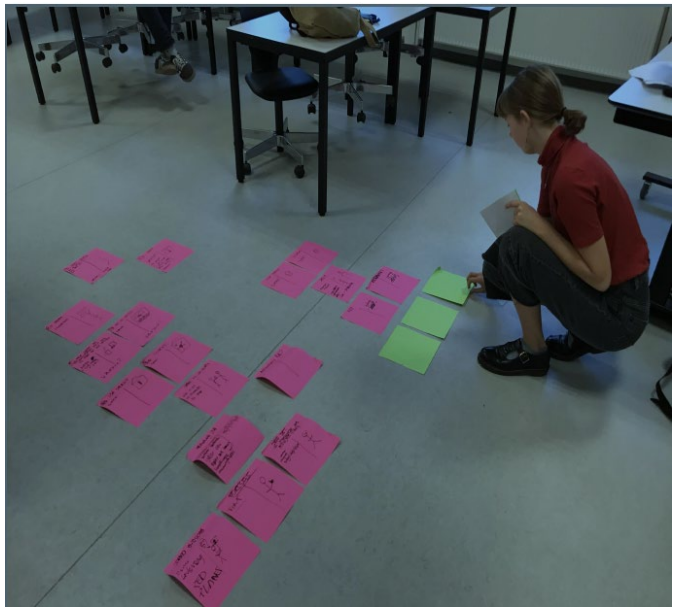
Kilder: [Barr2011], [Perkovic2010] og [Tedre2016]

Inspiration: Interaktive dilemmaspil i mediefag

- Michael Møller har udarbejdet et CT forløb om dilemmaspil i mediefag:



Plotmodellen er udarbejdet af Camilla Wandahl (2016) med inspiration fra Ida Maria Rydén, Joan Rang, Christopher Vogler, Blake Snyder, Hjemme-ude-hjem-modellen og Berettermodellen



Dekomposition og **abstraktion**

Identifikation og nedbrydning af fortællingens betydningsbærende dramaturgiske elementer vha. plotmodel og udarbejdelse af egne dilemmaspil på bagrund af plotmodel.

Algoritme

Flowcharts for "dramaturgiske veje" (forgreninger) og implementering med grafik og video i WordPress (automatisering).

Kilder: PowerPoint show ved Michael Møller, underviser i dansk og mediefag på Mulernes Legatskole
Se dilemmaspillene på Guldastronaut

Inspiration: Algoritmer i kemi og dansk

- Anne Boie Johannesson har udarbejdet et tværfagligt CT forløb i kemi og dansk:
  - Start: Simple øvelser i **algoritmisk tankegang** med **pseudokode**, f.eks. tømning af postkasse.
  - Kemi: Design af algoritmer for **mængdeberegningsopgaver** i grupper og afprøvning af andre gruppers algoritmer.
  - Dansk: Design af algoritme for udarbejdelse af **personkarakteristikker** i forbindelse med novelleanalyse.
- Algoritmisk systematisering af arbejdsgange synes at bidrage positivt til elevernes læring.

Kilder: [Johannesson2018]

Hvis jeg nu gerne vil kode…

Scratch

- Visuelt "blok"-programmeringssprog
- Intuitivt, lettilgængeligt og brugt verden over
- Ny HTML5 version med LEGO Mindstorms EV3/WeDo 2.0 og micro:bit integration er på trapperne

LEGO WeDo 2.0

- Robot med motorer, sensorer og Bluetooth.
- Visuel "blok"-programmering

LEGO Mindstorms Education EV3

- Robot med motorer, sensorer, USB, WiFi og Bluetooth
- Visuel "blok"-programmering
- Konkret og håndgribelig virkeliggørelse af CT
- Allerede brugt af en del IT undervisere

micro:bit

- "Lille computer" med sensorer, LED lys, knapper, USB og Bluetooth brugt verden over
- Visuelt "blok"-programmering (Blocks) eller tekstbaseret programmering (JavaScript/Python)

Kilder: [Lockwood2017], [Lockwood2018] og [Shute2017]

Hvis jeg nu gerne vil kode…

GeoGebra

- Matematik software med mulighed for tekstbaseret programmering (GGBScript og JavaScript)
- Allerede brugt af mange matematikundervisere/elever

Og alle de andre…

NetLogo

MIT APP INVENTOR

ARDUINO

Snap!

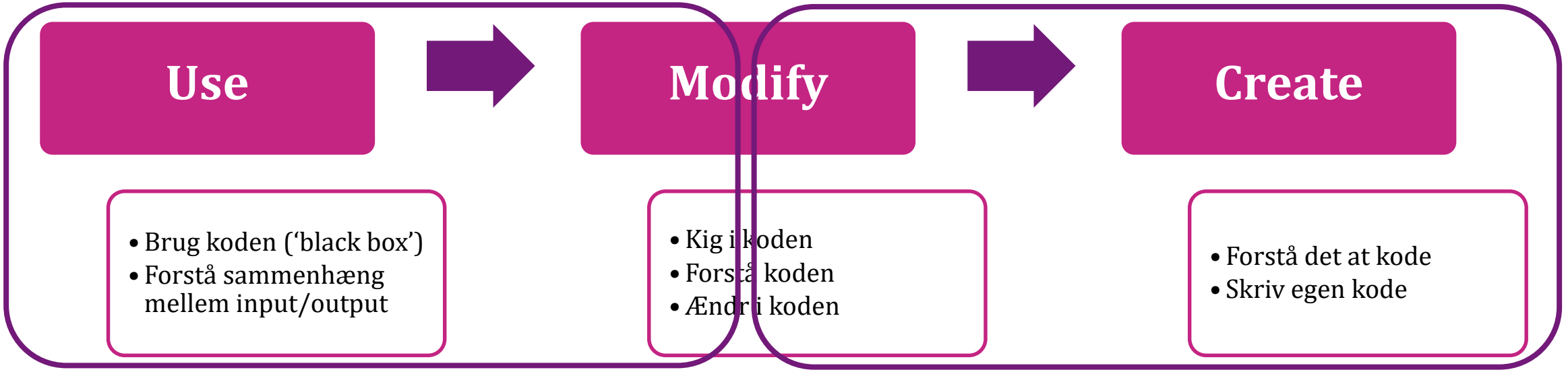
Kilder: [Lockwood2017], [Lockwood2018] og [Shute2017]

Hvis jeg nu gerne vil kode...

CT tilgang II: Use-Modify-Create

- En gængs tilgang til undervisning i CT synes være *Use-Modify-Create*:

"Ikke mit" (ingen ejerskabsfornemmelse) **"Mit" (ejerskabsfornemmelse)**

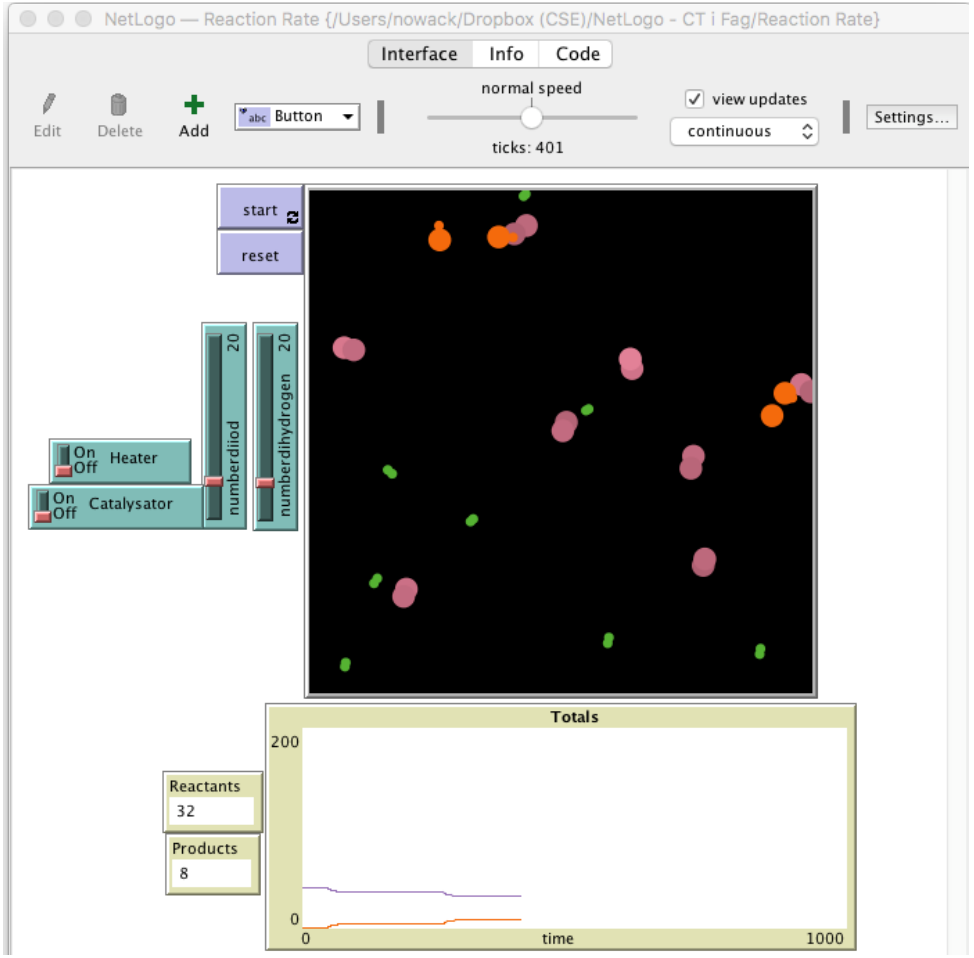


- Brug eksempler på algoritmer til Use/Modify.
- Modify/Create kan være svært at nå med tekstbaseret kode, så brug pseudokode, flowcharts eller et visuelt programmeringssprog.

Kilder: [Lee2011]

Inspiration: CT i gymnasiefag

- Center for Computational Thinking and Design (CCTD) har samarbejdet med 9 gymnasier i Region Midtjylland om at udvikle, afprøve og evaluere 6 korte CT-undervisningsforløb (samfundsfag, biologi/bioteknologi og kemi).
- Undervisningsforløbene er bygget over Use-Modify-Create.
- Eleverne fik et fagligt udbytte og de fleste var i stand til at ændre og tilføje i koden (Modify/Create).



Kilder: [CCTD2018] og Forløbsbeskrivelser

VidenCenter

Etik, dannelse og CT?

- CT leder *ikke* nødvendigvis til etiske/kritiske (selv)refleksioner eller (digital) dannelse.
- Etik og dannelse i CT kræver rammesætning.
  - Michael Møllers dilemmaspilforløb tog udgangspunkt i FN's verdensmål om bæredygtig udvikling.
  - Anne Bois har tænkt at afslutte sit CT forløb med algoritmer i dansk og kemi med en diskussion af algoritmernes betydning og konsekvenser (med Facebook og Google som eksempler).

Advarsel: Facemash

Mark Zuckerbergs **Facemash** (rekonstruktion!):

"I'm a programmer and I'm interested in the algorithms and math behind it"
- Zuckerberg

Kreativitet, innovation og CT?

- Et CT-problem kan være et **lukket problem** (med én løsning), som kan løses vha. **standardalgoritmer**.
  - Eksempel: Find det største tal i en given talrække.
- CT er derfor *ikke* nødvendigvis en kreativ eller innovativ proces.
  - Kreativitet: At skabe en ny idé.
  - Innovation: At føre en ny idé ud i livet.
- Kreativitet/innovation i CT kræver rammesætning.
  - Micheal Møller startede CT forløbet om dilemmaspil med idégenerering (kreativitet), som ledte til et nyt multimedieprodukt offentliggjort på internettet (innovation).

CT tilgang III: Papert, Resnick og konstruktionisme

- Seymour Paperts ”inversionsprincipper” (nogle af dem):

”What comes first, using it or 'getting it'?” (Power principle)	• Anvendelse leder til forståelse.
”Project before Problem”	• Personligt engagerende, længerevarende projekter fremfor kedelige problemløsningsopgaver bestemt af underviseren. • Problemerne skal opstå i projektet.
“Object before Operation” (Thingness principle)	• Gør operationerne (“algoritmerne”) fysikalske eller tingslige i form af f.eks. blokprogrammering, post-it’s, robotter, mm.

Kilder: [Papert1996]

CT tilgang III: Papert, Resnick og konstruktionisme

- Resnicks fire P'er:

Project | **Peers** | **Passion** | **Play**

- Legende skabelse af personligt meningsfulde projekter i samarbejde med andre.
- CT kan være en måde at udtrykke sig på, at eksperimentere på, at skabe på, og at samarbejde med andre på.

Kilder: [Resnick2014] og [Brennan2012]

Idéer til udvikling af CT forløb

- Identificer CT kompetencer i et fagligt forløb eller aktivitet (dekomposition, abstraktion, algoritme, evaluering, generalisation)
  - Dekomposition og abstraktion: Arbejder faget med ”nedbrydning til enkeltdele og opbygning til helhed” (f.eks. analyse/syntese, mønstergenkendelse/generelle sammenhænge)?
  - Algoritme: Kan arbejdsgange, metoder osv. i faget automatiseres vha. en algoritme?

- Overvej, hvordan eleven undervises i CT kompetencer, især algoritmisk tænkning og algoritmedesign.
  - Use-Modify-Create, pseudokode/flowcharts, programmering, project before problem, samarbejde, etik/dannelse, kreativitet/innovation,…

Tak for opmærksomheden!

Kilder

- [Barr2011] *Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community?*, Valerie Barr og Chris Stephenson, ACM Inroads, 2011.
- [Brennan2012] *New frameworks for studying and assessing the development of computational thinking*, Karen Brennan og Mitchel Resnick, online artikel, 2012.
- [Caspersen2013] *Computational Thinking and Practice — A Generic Approach to Computing in Danish High Schools*, Michael E. Caspersen og Palle Nowack, Proceedings of the Fifteenth Australasian Computing Education Conference, vol. 136, s. 137-143. , 2013.
- [CCTD2018] *CT i Gymnasiefag*, online rapport offentliggjort på cctd.au.dk, 2018.
- [Johannesson2018] *Algoritmer og faglige metoder*, Anne Boie Johannesson, online artikel fra gymnasieskolen.dk, 23. jan 2018.
- [Jones2011] *The Trouble with Computational Thinking*, Elizabeth Jones, online artikel, 2011.
- [Lee2011] *Computational Thinking for Youth in Practice*, Irene Lee et al., ACM Inroads, 2011.
- [Lockwood2017] *Computational Thinking in Education: Where does it fit? A systematic literary review*, James Lockwood og Aidan Mooney, fra arxiv.org, 2017.

Kilder (fortsat)

- [Lockwood2018] *Computational Thinking in Secondary Education: Where does it fit? A systematic literary review*, James Lockwood og Aidan Mooney, International Journal of Computer Science Education in Schools, vol. 2, num. 1, januar 2018.
- [Lu2009] *Thinking about computational thinking*, James J. Lu og George H.L. Fletcher, SIGCSE Bull., vol. 41, num. 1, s. 260-264, marts 2009.
- [Papert1996] *An Exploration in the Space of Mathematics Educations*, Seymour Papert, online artikel, 1996.
- [Perkovic2010] *A Framework for Computational Thinking across the Curriculum*, Ljubomir Perkovic et al., ACM, 2010.
- [Resnick2014] *Give P's a chance: Projects, peers, passion, play*, Mitchel Resnick, online artikel, 2014.
- [Shute2017] *Demystifying computational thinking*, Valerie J. Shute, Chen Sun og Jodi Asbell-Clarke, Educational Research Review, vol. 22, 2017.
- [Tedre2016] *The Long Quest for Computational Thinking*, Matti Tedre og Peter J. Denning, Proceedings of the 16th Koli Calling Conference on Computing Education Research, s. 120-129, 2016.

Kilder (fortsat)

- [Wing2006] *Computational thinking*, Jeannette M. Wing, Communications of the ACM, vol. 49, num. 3, s. 33–35, marts 2006.
- [Wing2010] *Computational Thinking: What and Why?*, Jeannette M. Wing, Link Magazine, 2010.