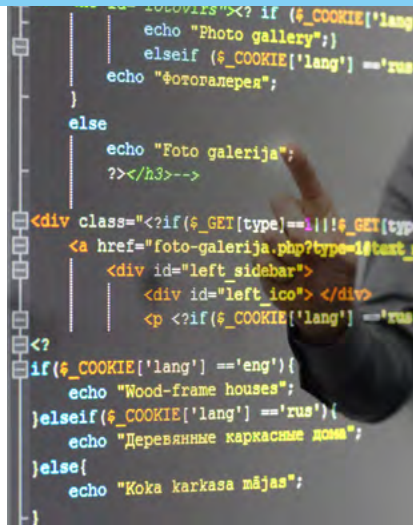
DIGITAL LÆRING

PÅ UNGDOMSUDDANNELSERNE



Programmering
i undervisningen

I dette hæfte gives eksempler på, hvordan programmering kan inddrages i undervisningen.

Indholdsfortegnelse

Dette inspirationshæfte er det fjerde i serien Digital læring på ungdomsuddannelserne. Serien har fokus på digitale indsatser på ungdomsuddannelserne i Region Midt.

Forfattere: Tobias Kidde Skov, eVidenCenter
Redaktør: Daniella Tasic Hansen, eVidenCenter
Layout: Anne Sofie Holm Sandgaard, eVidenCenter
Sproglig korrektur: Daniella Tasic Hansen, eVidenCenter
Faglig korrektur: Daniella Tasic Hansen, Lars Nielsen og Simon Kruse Barslev, eVidenCenter
ISBN: 978-87-998209-3-1

Denne udgivelse er en del af et videndelingsprojekt under eSkoler Midt, der er Region Midtjyllands strategi for digital læring. eSkoler Midt er er koordineret af eVidenCenter, Det Nationale Videncenter for e-læring.

Læs mere på www.eskolermidt.dk og www.evidencenter.dk

Forord

eVidenCenter
Det Nationale Videncenter for e-læring

Programmering er ikke længere forbeholdt programmører og udvikling af store softwareprojekter som platforme, online betalingsmuligheder eller spil. Fra flere steder peges der på, at programmering bliver en moderne kompetence, en del af 21st Century Skills[1], som bliver afgørende for nye generationer at forstå og kunne.

Det betyder ikke, at alle kommende generationer skal være programmører eller lære at programmere. De skal i stedet forstå, hvad programmering er; hvordan det er en del af deres teknologiske hverdag, og hvordan selv enkel programmering kan gøre dem i stand til fx at udvikle og skabe nye teknologiske muligheder. Det handler om teknologiforståelse og digital dannelse.

Derfor mener flere også, at programmering skal fylde mere i uddannelsessystemet (fx Apple, Dansk Erhverv, IT-branchen m.fl.). Dette inspirationshæftes argument er, at implementeringen af programmering i uddannelsesverdenen med fordel kan ske gennem inddragelsen i en fagfaglig kontekst, så programmering bliver et værktøj i de fagfaglige fag.

Derfor vil dette inspirationshæfte dreje sig om, hvad programmering er, og hvor programmering kan være relevant - som en integreret del af unges uddannelse. Derudover præsenteres der to konkrete eksempler på, hvordan visuel programmering kan integreres i undervisningssammenhænge. Udgangspunktet for eksemplerne er inddragelsen af visuel programmering i en fagfaglig kontekst for at illustrere, hvordan programmering kan være med til at styrke arbejdet med, i denne sammenhæng, temaer fra matematik og dansk, og ikke kun den kreative, abstrakte tænkning og udvikling.

Det Nationale Videncenter for e-læring,
Aarhus, april 2016

I uddannelsessammenhæng er det programmeringssprog, der er nemmest at komme igang med, det visuelle programmeringssprog

Hvad er programmering?

Som skrevet i introduktionen kan programmering foregå i mange forskellige sprog. Generelt kan man dog beskrive programmering som de instruktioner, man giver et objekt, fx en computer, når der skal ske en handling eller udvikling. Man forsøger at skabe denne udvikling gennem sine programmeringer. Udviklingerne kan deles op i mange forskellige kategorier. Fx kan man snakke om "hændelser", der betegner den aktivitet, der sker, når du eksempelvis trykker på en knap på en hjemmeside. Du kan også arbejde med variabler og skabe en udvikling ved at ændre værdierne i variablerne. Det handler om at have nogle regler for, hvad der skal ske, når man arbejder med objektet.

Der ligger altså en masse beskrevne hændelser, metoder og variabler bag, når du fx søger rundt på en hjemmeside. Alt fra de knapper du trykker på, hvordan knapperne ser ud, til hvilke ændringer der sker, når du trykker på de forskellige knapper.

Microsofts skaber Bill Gates beskriver kort programmering på denne måde:

"Så programmering er i virkeligheden en lille smule matematik, og nogle hvis-sætninger, hvor beslutninger bliver taget" – Bill Gates
www.youtube.com/watch?v=m2Ux2PnJe6E

Gates beskriver med dette udsagn både kompleksiteten og simpliciteten i programmering.

Det programmeringssprog, som i uddannelses- og undervisningssammenhænge giver den nemmeste mulighed for at komme i gang med programmering, er det visuelle programmeringssprog. Visuel programmeringssprog er et sprog, som bruger grafiske elementer til at skabe den før omtalte udvikling. En fordel ved brugen af visuel programmering er især det grafiske udtryk, men også at brugen af visuelle blokke sikrer, at syntaksten altid er i orden. Brugen af syntaks, hvor brugeren laver sætningsdannelser for at skabe en udvikling, er ofte der, hvor fejlene opstår i en tekstbåret programmering.

I dette eksempel fra programmet Scratch, ses syntaksten tydeligt. I Scratch bygger man blokke med sætninger oven på hinanden, som så skaber en sammenhæng. Sætningen i Scratch bliver her: "Når du klikker på flaget starter du en aktivitet, der (for evigt) hvis du trykker på mellemrum, (får dit objekt til at sige) Hello!"

I denne kode er der tale om, at hver gang du starter spillet ved at trykke på det grønne flag og trykker på mellemrumstasten, "siger" den figur, du har valgt at programmere: "Hello!". Den sproglige syntakst skabes gennem de forskelligfarvede blokke, som kan sammensættes på mange forskellige måder.

Det visuelle programmeringssprog kan med sit enkle og intuitive udtryk give en god introduktion til programmering, også selvom det visuelle sprog ikke ligner det skrevne professionelle programmeringssprog.

Hvorfor programmering?

Elevernes arbejde med programmering kan udvikle deres innovative tænkning og samtidig understøtte deres kreativitet, samarbejdsevner og nytænkning

På baggrund af den korte introduktion til visuel programmering er det relevant at spørge, hvorfor elever skal lære om at programmere som en del af deres skolegang, og hvorfor det overhovedet er relevant at lære, hvad programmering er.

En af de væsentligste årsager til, at programmering er blevet interessant i uddannelses- og undervisningssammenhænge, er for det første den teknologiske udvikling i samfundet, og for det andet demokratiseringen af digitale teknologier, der nu gør det muligt for alle at komme i gang med bl.a. at programmere, 3D-printe og producere film og musik. Programmering er samtidig grundlaget for hver eneste interaktion mellem mennesker og computere. Det er interaktioner med alt fra mobiltelefonen, Apps, spil, GPS, paskontrol eller onlinebetalinger.

Eleverne bør derfor lære at forstå programmering, fordi det er blevet så stor en del af deres hverdag, og fordi de den måde kan lære, hvad der ligger bag alle de interaktioner med computere, de indgår i hver dag.

EU Kommissionen mener at programmering, eller kodning, er en kompetence, en 21st Century Skill, som eleverne skal udvikle for at matche fremtidens krav (EU Kommissionen, 2015). Kommissionen vurderer, at 90 % af fremtidige job kommer til at indeholde krav til 21st Century Skills, og dermed også forståelse af grundlæggende programmering. Dansk Erhverv ønsker også at programmering får mere plads i skolen, og bliver et sprog, der skal læres på samme vis som eksempelvis tysk (b.dk, 2016)

Men ud over at kunne skabe kompetencer til fremtiden og gøre elever mere opmærksomme på, hvilken teknologi de omgiver sig med, kan programmering også styrke eleverne innovative og kreative tænkning.

Professor Mitchel Resnick, som er medskaber af det visuelle programmeringsværktøj Scratch (Scratch.mit.edu) mener, at elevernes arbejde med programmering, kan udvikle deres innovative tænkning, og samtidig understøtte deres kreativitet, samarbejdsevner og nytænkning. Som det også ville blive vist i de kommende eksempler på inddragelsen af programmering i undervisningen, så kan programmering som værktøj være med til at understøtte opgaver og øvelser, som skaber rum og frihed til at udforske og tænke abstrakt. Den legende tilgang til programmering, som Resnick argumenterer for, er en tilgang, der kan være med til at skabe motivation omkring et emne, som ellers kan være svært at få taget hul på.

I det følgende vil vi se på to konkrete eksempler på, hvordan man kan inddrage programmering som et værktøj i undervisningen.

To konkrete eksempler på programmering i undervisningen

I dette afsnit gives to konkrete eksempler på, hvordan programmering kan inddrages i undervisningen som et understøttende værktøj til arbejdet med fagfagligt stof. Programmering bliver i denne sammenhæng ikke set som et fag i sig selv men som et værktøj, der kan styrke elevernes læring, nysgerrighed, kreative tænkning og fagfaglighed.

Eleverne skal ikke blive eksperter i avanceret programmering men skal via enkelte visuelle programmeringsværktøjer bl.a. skabe og udvikle forudsætninger for at forstå mulighederne med programmering.

Begge eksempler bruger det gratis programmeringsværktøjer Scratch, som er udviklet på en afdeling hos Massachusetts Institute of Technology (MIT). Der findes mange andre visuelle programmeringsværktøjer (fx Hopscotch, Touch Develop mfl.), så det er bare om at undersøge hvilke muligheder, der passer bedst til egen inddragelse.

Scratch er visuel programmering, som er opbygget af blokke inddelt i temaer, som fortæller om det indhold blokkene har. Det kan fx være bevægelse, lyd, hændelser eller styring. Den overordnede programmering sker gennem "Sprites", som er den eller de ting og figurer, som skal skabe en udvikling.

Scratch er bygget op i et kartesis koordinatsystem, og det er derved koordinater, der bruges, når en Sprite eksempelvis skal lave en bevægelse.

Det gratis udgangspunkt for Scratch betyder at oversættelse fra engelsk til dansk stadig mangler nogle steder, og samtidig bliver der ikke udviklet på værtøjet med samme hastighed som ved betalingsværktøjer. Til gengæld er der gennem den fri adgang til Scratch skabt et stort delingsfællesskab, hvor elever over hele verden deler deres programmeringsprojekter og de programmeringssyntakser, de har skabt.

På youtube.com findes der et hav af tutorials, som kan være til hjælp, når man skal i gang med at bruge Scratch. Derudover findes der ligeledes introduktionsopgaver til Scratch samt konkrete opgaver til eksempelvis matematik på forskellige sites.
Se fx www.opfinderklubben.dk, og www.4code.dk.

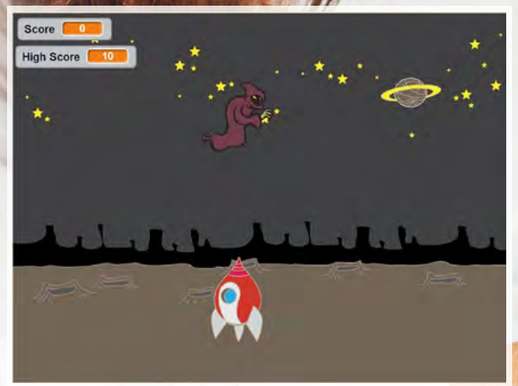
1. Produktion af computerspil, med et matematisk udtryk

Det første eksempel på inddragelse af programmering i undervisningen tager udgangspunkt i den åbenlyse matematiske tilgang, der ligger i at programmere. I Scratch kommer eleverne til at stifte bekendtskab med matematiske emner som koordinatsystemet, grader, algoritmer, algebra m.fl. Men det er ikke ensbetydende med, at de skal arbejde med alle emnerne på en gang.

Den konkrete opgave går ud på at eleverne skal producere og udvikle deres eget computerspil, via Scratch. Spillet skal indeholde en progression, og det skal være muligt at styre en figur gennem spillet. Målene er, at eleverne skal producere et computerspil og på baggrund af deres produktion, skal eleverne kunne redegøre, demonstrere og argumentere for, hvordan de har arbejdet med eksempelvis algoritmer eller algebra.

Et eksempel på et spil kunne være en et klassisk 2D arkadespil, hvor hovedpersonen er fikseret på y-aksen, og hvor fjenden kommer ned fra x-aksen. Billedet til højre er et eksempel på netop denne type spil.[2]

Matematisk kan eleverne arbejde med, hvordan raketten eller spøgelserne ved hjælp af koordinater flytter sig rundt. De kan også arbejde med udregninger af et scoresystem, hvor der tildeles point, hver gang en fjende skydes ned. Det er op til underviseren (og potentielt også eleverne), hvilket tema der kunne være relevant at arbejde med gennem Scratch.



Billedet viser et eksempel på et klassisk 2D arkadespil produceret i Scratch.

I dette eksempel arbejder eleverne med matematik samtidig med, at de laver digitale produktioner og tænker kreativt og abstrakt. Scratch faciliterer på denne måde inddragelsen af programmering som et værktøj til at understøtte arbejdet med forskellige matematiske temaer.

De færdige produktioner kan præsenteres og fremlægges på flere forskellige måder - med afprøvning af spillet som den mest åbenlyse præsentation. Vi vil senere vende tilbage til, hvordan præsentation og feedback på produktionerne kan se ud.

2. Produktion af computerspil, med et fortællende udtryk

Computerspil er i dag, ifl. lektor Bo Kampmann Walther, et dominerende medie og en moderne og central fortællerform (Kampmann 2012), og computerspils dramaturgi er blevet omdrejningspunktet for den måde flere film og serie i dag, fortæller deres historier på (Videnskab.dk 2012).

En oplagt måde at inddrage programmering i undervisningen på er derfor ved at bygge historier op omkring de computerspil, som eleverne producerer. Her er det ikke nok bare at skabe og programmere en raket, som skal skyde nogle spøgelser. Når man spiller spillet, skal man kende historien om raketten: Hvorfor den er ude på at skyde spøgelserne, hvorfor spøgelserne er fjenden, og hvad der kan ske, hvis ikke spøgelserne bliver skudt? Det kan være det klassiske græske drama om protagonisten og antagonisten, men det kan også være alternative fortællinger, som ikke spiller på et helt/fjende forhold.

Eleverne skal gennem en forståelse af både klassisk og moderne fortællerteknik og dramaturgi, skabe et computerspil med en fortælling, som danner rammen om den produktion, som computerspillet er.

Med dette eksempel er der ikke fokus på den matematiske baggrund for udviklingen af spillet men derimod fortællingen, der skaber spillet. Det er dog stadig vigtigt at programmeringerne i spillet, hænger sammen med historien.

Eksemplet leder naturligt op til at andre skal prøve elevernes produktioner, og at de skal dele dem med hinanden.

Evaluering og feedback af produktioner

I elevernes forskellige produktioner af computerspil ligger der en naturlig evalueringsform, hvor eleverne kan præsentere spillene for hinanden og afprøve dem. Samtidig er der i det matematiske perspektiv også en mulighed for at præsentere og dele de måder, hvorpå eleverne på forskellige vis har skabt deres programmeringer. Her vil der også være et stort potentiale i, at eleverne videndeler deres programmeringer. Hvordan var det man fik raketten til at skyde så præcist, eller hvordan var det man programmerede en high score?

Ud over at prøve spillene kan eleverne også lave peer feedback, hvor eleverne løbende giver hinanden feedback på eksempelvis dramaturgien i spillene, eller spillets funktioner gennem programmeringerne.

Peer feedback går ud på, at man deler sine skrevne tekster med andre elever allerede tidligt i arbejdet. Formålet er at forbedre den enkelte tekst samtidig med, at eleverne oparbejder generelle tekstkompetencer og bliver bevidste om deres skriveprocesser.[3]

Denne feedbackform kan med fordel projekteres over i arbejdet med programmering i Scratch, hvor eleverne løbende sparrer med hinanden og kommer med feedback. Det kan handle om at forstå, hvordan man konkret spiller spillet, hvordan programmeringerne er lavet, og om de virker, eller hvordan dramaturgien er bygget op.

I elevernes forskellige produktioner af computerspil ligger der en naturlig evalueringsform, hvor eleverne kan præsentere spillene for hinanden og afprøve dem.

Afslutning

Dette inspirationshæfte har forsøgt at redegøre for hvad programmering er, og hvorfor programmering er relevant i undervisningssammenhænge gennem bl.a. eksempler på, hvordan programmering kan inddrages i undervisningen som et værktøj i en fagfaglig kontekst.

Programmering er helt sikkert en kompetence, som vil få mere opmærksomhed de kommende år, men selvom dette inspirationshæfte argumenterer for, at programmering kan blive en vigtig fremtidig kompetence, betyder det ikke, at det nødvendigvis skal indgå som et nyt obligatorisk fag i skolen. Fremtidens elever skal kunne forstå moderne teknologi, herunder den programmering som ligger bag, men det kan med fordel indgå i undervisningen der, hvor det giver mening. For eksempel som værktøj til at understøtte arbejde med matematiske temaer, historiefortælling og dramaturgi.

Mulighederne er mange, så det handler bare om at komme i gang og udforske.

Kilder:

Litteraturliste:

b.dk (29.18.16)
www.b.dk/nationalt/erhvervslivets-vaekst-bremses-it-programmering-skal-paa-skemaet-i-folkeskolen

EU Kommissionen
www.ec.europa.eu/digital-single-market/en/news/eu-code-week-10-18-october-bringing-ideas-life

Jensen, Tine Wirenfeldt (m.fl.) (2012) Peer feedback i gymnasiet – fokus på elevernes skriveprocesser, Gymnasieskolen.dk,
www.gymnasieskolen.dk/peer-feedback-i-gymnasiet-%E2%80%93-fokus-p%C3%A5-elevernes-skriveprocesser

Kampmann Walther, Bo (2012) Computerspil og de nye mediefortællinger, Samfundslitteratur

Mitchel Resnick om programmering – Ted Talk (offentliggjort 29 jan. 2013)
www.youtube.com/watch?v=Ok6LbV6bqaE

Videnskab.dk (2012)
www.videnskab.dk/kultur-samfund/film-og-serier-bruger-computerspils-dramaturgi

Noter:

1. Læs om 21st Century Skills her: www.p21.org/our-work/p21-framework
2. Prøv spillet her www.scratch.mit.edu/projects/34836534/
3. Gymnasieskolen.dk, 2012

Egne noter

Det Nationale Videncenter for e-læring

Dette inspirationshæfte indeholder inspiration til arbejdet med programmering i uddannelsesmæssig sammenhæng.

Målet er at inspirere undervisere til at inddrage programmering i deres undervisning til bl.a. at understøtte arbejdet med fagfagligt indhold.

Inspirationshæftet er det fjerde i serien Digital læring på ungdomsuddannelserne. Serien har fokus på digitale indsatser på ungdomsuddannelserne i Region Midt.

Se mere på
www.eskolermidt.dk og www.evidencenter.dk

ungdomsuddannelserne regionmidtjylland

978-87-996209-3-1