

TEKNOLOGIFORSTÅELSE

MATEMATIK 1. KLASSE
EFTERÅR

Hvad kan en robot?

Udarbejdet af Camilla Finsterbach Kaup i samarbejde med Adrian Rau Bull, Bo Teglskov Kristensen, Charlotte Krog Skott og Peter Søgaard*

*Materialet er udviklet af Københavns Professionshøjskole, Professionshøjskolen UCN, VIA University College samt læremiddel.dk for Børne- og Undervisningsministeriet under rammerne for Forsøg med teknologiforståelse i folkeskolens obligatoriske undervisning. Læs mere om forsøget på www.tekforsøget.dk og www.emu.dk.



KØBENHAVNS
PROFESSIONS
HØJSKOLE



LÆRE
MIDDEL
DK



VIA University
College

ucn

RAMBOLL

INDHOLDSFORTEGNELSE

1. Forløbsbeskrivelse	3
1.1 Beskrivelse	3
1.2 Rammer og praktiske forhold	4
2. Mål og faglige begreber	5
3. Forløbsnær del	6
3.1 Introfase: Forforståelse og kompetencer	6
3.2 Udfordrings- og konstruktionsfase	9
3.3 Outrofase: Ny forståelse og nye kompetencer	12
4. Perspektivering	14
4.1 Evaluering	14
4.2 Progression	14
4.3 Differentieringsmuligheder	14
4.4 Særlige opmærksomhedspunkter	14
5. Lærer- og elevhenvendte ressourcer	15
5.1 Lærerhenvendte ressourcer	15
5.2 Elevhenvendte ressourcer	15

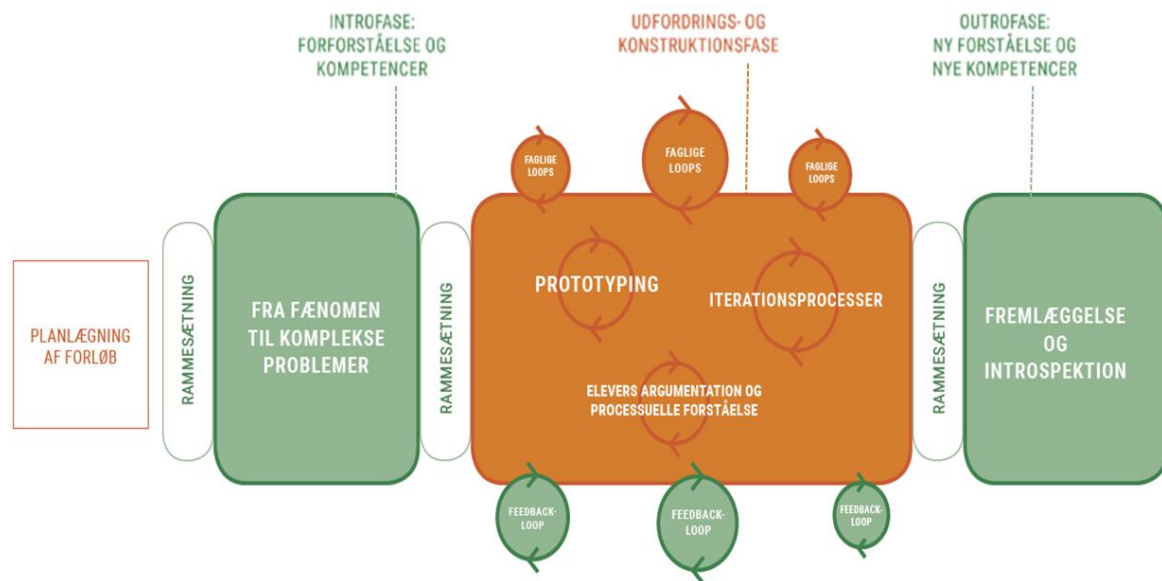
Version 2

Dette er version 2 af forløbet. I revisionen af forløbene har vi arbejdet med at præcisere mål, rammer og aktiviteter. Der er ikke ændret fundamentalt ved forløbet, så materialer, som er udviklet til den konkrete undervisning på skolerne på baggrund af den første version af forløbet, vil stadig kunne anvendes.

1. Forløbsbeskrivelse

Forløbet er bygget op over det didaktiske format for prototyperne med en introducerende del, en mere undersøgende/eksperimenterende del og en outro-del med opsamlinger og evalueringer, se figur 1.

Figur 1: Didaktisk prototypeformat



1.1 Beskrivelse

Forløbet sætter fokus på programmering gennem forskellige aktiviteter både unplugged og med robotter. Eleverne skal gennem forløbet udvikle en begyndende forståelse for, hvilken type informationer og instruktioner en robot kan modtage.

I introfasen arbejder eleverne med unplugged aktiviteter, hvor de arbejder med at opbygge en grundlæggende forståelse for programmering. Eleverne vil lære de basale principper, som det at give og modtage instruktioner. I udfordrings- og konstruktionsfasen arbejder eleverne med robotter og ikonprogrammering. De vil øve sig i at finde løsninger på et givent problem, som fx at programmere en robot gennem en forhindringsbane. Disse aktiviteter er med til at understøtte elevernes arbejde med at følge og tilrette simple programmer gennem systematiske beskrivelser af problematikker, instruktioner, løsningsstrategier og hændelsesforløb.

Gennem arbejdet med forløbet udvikler eleverne en forståelse for trin for trin instruktioner, som fx kan få en robot eller et program til at udføre det, man ønsker. Eleverne vil også arbejde med området logisk ræsonnement. Her vil de begynde at få en forståelse for, hvordan en robot reagerer på en given instruktion.

Det er her oplagt at inddrage programmering af bevægelser som en del af arbejdet med placeringer og flytninger fra kompetenceområdet "Geometri og måling".

Produkt:

Et kort rutediagram enten via ScratchJr eller ved hjælp af Beebotten.

1.2 Rammer og praktiske forhold

1.2.1 Samlet varighed

Forløbet er estimeret til 10-15 lektioner svarende til ca. 2 til 3 ugers undervisning – afhængigt af brugen af faglige loops.

1.2.2 Materialer

- Adgang til tablet, iPad eller Chromebook til alle elever eller grupper
- Adgang til robotter som fx Bee Bots (eller andre robotter)
- Forberedt materiale til unplugged aktiviteter (se fx under lærerhenvendte ressourcer)

NB! Kontakt evt. jeres lokale CFU Udlån, hvis skolen ikke har robotter. De har Bee Bots til udlån.

1.2.3 Lokaler

Ingen særlige krav; evt. en skolegård med fliser eller andet areal, hvor der kan optegnes felter.

2. Mål og faglige begreber

Af den samlede faglighed i teknologiforståelsesfaget integrerer matematik elementer fra alle fire kompetenceområder fra faget teknologiforståelse i det nye kompetenceområde "teknologiforståelse". I dette forløb er der fokus på færdigheds og vidensområdet "programmering".

KOMPETENCEOMRÅDER	MATEMATISKE KOMPETENCER	GEOMETRI OG MÅLING	TEKNOLOGIFORSTÅELSE
Kompetencemål (efter 3. klassetrin)	Eleven kan handle hensigtsmæssigt i situationer med matematik	Eleven kan anvende geometriske begreber og måle	Eleven kan handle hensigtsmæssigt med digitale teknologier i afgrænsede situationer fra deres hverdag.
Færdigheds- og vidensmål (efter 3. klassetrin)	Kommunikation: <ul style="list-style-type: none"> ▪ Eleven kan deltage i mundtlig og visuel kommunikation med og om matematik. ▪ Eleven har viden om enkle mundtlige og visuelle kommunikationsformer, herunder med digitale værktøjer. 	Placeringer og flytninger: <ul style="list-style-type: none"> ▪ Eleven kan beskrive objekters placering i forhold til hinanden. ▪ Eleven har viden om forholdsord, der kan beskrive placeringer. 	Programmering: <ul style="list-style-type: none"> ▪ Eleven kan følge og tilrette simple programmer ▪ Eleven har viden om grundlæggende konstruktioner i programmeringssprog baseret på ikoner.
	Hjælpemidler: <ul style="list-style-type: none"> ▪ Eleven kan anvende enkle hjælpemidler til tegning, beregning og undersøgelse. ▪ Eleven har viden om konkrete materialer og redskaber. 		

Vær opmærksom på, at målene i nedenstående tabel er mål for elevernes læring efter 3. klasse, og de konkrete læringsmål skal tilpasses til elevernes faglige niveau i 1. klasse på skolen. Det giver anledning til tilpasning af målene til elevernes faglige niveau, som er gjort i nedenstående læringsmål.

Konkretiserede læringsmål

- Eleven kan gennem unplugged aktiviteter sammensætte forskellige bevægelseshandlinger i sekvenser, der udgør en simpel algoritme.

- Eleven kan sammensætte en sekvens af ikoner til en algoritme, der får en digital figur (i Scratch kaldet en sprite) eller en robot til at bevæge sig på en bestemt måde.
- Eleven kan ved hjælp af forholdsord programmere sekvenser, der placerer deres kammerat/sprite/robot i forhold til tal og former.

Faglige begreber

Forholdsord og andre begreber, som kan beskrive objekters placering i forhold til hinanden, geometriske begreber såsom trekanter, kvadranter, og cirkler, talrække, plusstykker.

Teknologifaglige begreber

CS unplugged*, fejlfinding (debugging), ikon-programmering, logisk ræsonnement.

*CS unplugged dækker over undervisningsmateriale, der hjælper eleverne med at få en begyndende forståelse for Computer Science gennem lege, kort, spil og opgaver. De har det tilfælles, at det foregår uden brug af en computer eller anden teknologi.

3. Forløbsnær del

3.1 Introfase: Forforståelse og kompetencer

Eleverne bliver i denne fase introduceret til robotter, og de skal tage stilling til, hvordan de indgår i deres hverdag. Eleverne arbejder med unplugged aktiviteter, hvor de igennem forskellige opgaver får skabt en forståelse for, hvilke handlinger og inputs de kan bruge til at programmere en robot.

3.1.1 Varighed

2-4 lektioner a 45 minutter.

3.1.2 Problemfelt

Robotter bliver en større og større del af vores hverdag. Derfor er det centralt, at eleverne har forståelse for, hvordan robotter fungerer. Hermed begynder de også at få en forståelse for, hvilke problemer det kan skabe, når robotter overtager arbejde, som tidligere blev udført af mennesker. Dette forløb sætter fokus på, hvad en robot er, og hvordan eleverne kan programmere med en robot i form af simple instruktioner. I denne proces lærer eleverne at begå fejl og langsomt forbedre deres algoritme.

Eleverne skal tænke over, hvor i deres hverdag de møder robotter, og hvordan robotter kommer til at indgå i deres hverdag i fremtiden.

3.1.3 Problemstilling

- Hvad er en robot?

- Hvor er de i vores hverdag?
- Hvordan programmerer vi dem?

3.1.4 Iscenesættelse/scenarie:

Start forløbet op med en samtale, hvor eleverne kommer med deres bud på, hvad en robot er, og hvad vi bruger robotter til i vores hverdag. Spørg fx:

- Har I selv en robot?
- Hvordan indgår robotter i jeres hverdag? (Det kunne fx være noget legetøj, en robotplæneklipper, en robotstøvsuger osv).

Samtalen kan lede hen imod, hvilke instrukser en robot kan forstå.

3.1.5 Fagligt loop I (CS unplugged aktivitet): Hvilke inputs forstår en robot?

Formålet med aktiviteten er, at eleverne får en forståelse for vigtigheden af præcise retningsangivelser, når de skal instruere hinanden til at bevæge sig på bestemte måder.

Følgende aktiviteter kan være med til at starte forløbet op:

- Som lærer kan du bede eleverne om retningslinjer for at komme op til tavlen. Forklar eleverne, at en robot skal have helt enkle og klare instruktioner. Her er det væsentligt, at eleverne får en forståelse for, at en robot skal have præcise instrukser, fx 'gå tre skridt frem, drej til venstre, gå fem skridt frem'.
- Ved tavlen kan du fx bede eleverne om instruktioner til at tegne en trekant, firkant, cirkel osv. Igen trin for trin. Skriv gerne instruksen ned på tavlen, fx i form af pile. Forklar eleverne, at mange opgaver kan beskrives ved hjælp af en specifik liste af instruktioner, fx når de bager kager ved at følge en opskrift. Disse instruktioner udgør tilsammen det, man kalder en **algoritme**.
- En **algoritme** er altså en sekvens af instruktioner eller et sæt af regler til at udføre noget bestemt. Dette kan fx være en bageopskrift, der angiver i hvilken rækkefølge den, der bager, skal gøre nogle bestemte ting, fx veje den mængde mel, sukker, bagepulver og mælk der skal bruges, finde et antal æg, blande nogle ingredienser først og senere røre andre i, smøre formen, hælde dejen i og derefter bage kagen i 20 minutter ved 200 grader.
- Man kan også lave en algoritme for en hverdagssituation fra skolen, fx hvad eleverne skal gøre, når de møder:
 1. Tag overtøjet af.
 2. Hæng jakken op.
 3. Hæng skoletasken på stolen.
 4. Kom madkassen i køleskabet.
 5. Når klokken ringer, sætter du dig på stolen.Prøv derefter at bytte rundt på rækkefølgen af de enkelte trin og lad eleverne tænke over om rækkefølgen kan være anderledes. Denne øvelse kan også laves med andre rutiner fra elevernes hverdag, fx madpakkespisning.
- Lav nu et udsagn, som præcist fortæller, hvad en af eleverne skal gøre, fx 'rejs dig op, gå tre skridt frem'. Eleverne kan derefter gå sammen i mindre grupper, hvor de kan prøve at instruere hinanden på samme måde via 1-steps- (instrukser med kun en besked, fx gå tre skridt frem) og 2-steps (instrukser med to beskeder, fx gå to skridt frem og drej til højre). Dernæst kan de give hinanden nye beskeder. Dette kan enten være via arbejdsarkene eller foregå på gulvet.

3.1.6 Fagligt loop II (CS unplugged aktivitet): Lav en algoritme

Find evt. et sted med fliser eller optegn et sted i skolegården med et område med fx 8 gange 8 fliser/felter. Fordel eleverne langs felterne og tildel dem et udgangspunkt.

Elevernes opgave er at lave en beskrivelse af, hvordan de kommer ind til det felt, du har markeret. De skal altså lave en algoritme for, hvordan en person kan bevæge sig fra startfeltet til det markerede felt.

Eleverne må ikke gå ind i området, mens de laver algoritmen. Algoritmen kan fx lyde:

- Gå tre fliser frem
- Drej til venstre
- Gå fem fliser frem.

I fællesskab afprøves algoritmerne på skift. Giv tid til, at de grupper, der ikke nåede ind til det markerede felt, kan ændre deres algoritme. Der kan evt. sættes forhindringer op, fx kegler eleverne skal udenom for at komme ind til det markerede felt.

Der kan differentieres, og sværhedsgraden kan mindskes ved fx at lave forskellige markeringer til de enkelte grupper, der ikke er langt væk fra deres udgangspunkt. Dette kan fx markeres med kegler, der kan flyttes, så sværhedsgraden gradvist øges.

Eleverne kan også blive bedt om gå et givent sted hen, hvor de via deres skridt tegner bestemte figurer, fx en firkant.

Eleverne skal i gitteret placere sig i forhold til en kegle, fx foran, bagved ved siden af osv.

3.1.7 Fagligt loop III (CS Unplugged aktivitet): Robotkort

Under elevressourcer (se afsnit 5) ligger forskellige arbejdsark, som printes ud til eleverne.

Arbejdsarkene hjælper eleverne med at tænke forud og planlægge en kort rute fra figurens startposition til slutpositionen. Arbejdsarkene bygger videre på de unplugged aktiviteter, der blev arbejdet med under introduktionen.

- Lad eleverne være sammen i grupper af 2-4. Aktiviteten kan også laves samlet i klassen, hvis det giver mere ro.
- Klip pilene ud, så hver elev har sit eget arbejdsark og pile. Lad eleverne arbejde sig igennem arbejdsarkene.
- Nogle af arbejdsarkene vil have en algoritme, der skal afprøves af eleverne. Hvis denne er forkert, skal de selv ændre algoritmen til en korrekt angivelse.
- Der er mulighed for at eleverne kan lave deres egen algoritme, som derefter deles og afprøves/testes i gruppen.

3.1.8 Feedback-loop:

- Giv eleverne god tid til at justere deres beskrivelse, hvis de ikke rammer rigtigt.
- Lad eleverne rotere mellem grupperne, så de mødes i nye grupper og på den måde, får udfordret hinanden sprogligt og får øvet sig i at give hinanden instrukser.
- Brug lidt tid på øvelserne, så eleverne har et godt fundament til det videre arbejde.
- Eleverne giver hinanden feedback på opgaverne.

3.1.9 Afrunding

Afrund introfasen med en klassesamtale, hvor I taler om de enkelte aktiviteter. Prøv at sætte fokus på fejlfinding, og hvorfor det kan være relevant at have dette for øje.

Ofte når der arbejdes med programmering, er det væsentligt at kunne finde fejl i en algoritme, hvis fx robotten eller programmet ikke reagerer som tilsigtet.

- Hvorfor er det ikke alle algoritmer, der virker?
- Kunne I finde fejlen og rette den (jf. debugging)?
- Hvornår var det nemmest at få en klassekammerat til at udføre algoritmen/lave en opgave? Var det fx ved korte og præcise beskeder?
- Få klassen til at opsætte tommelfingerregler for god kommunikation med robotter.

3.2 Udfordrings- og konstruktionsfase

3.2.1 Varighed

6 lektioner a 45 minutter.

Denne fase er tænkt igennem to konkrete udfordringer. I den første udfordring arbejder eleverne med at programmere en Bee Bot, som er en robot, hvor eleverne ved hjælp af knapper nemt og enkelt kan programmere den. I den efterfølgende udfordring arbejder eleverne med Scratch Jr., hvor de via ikon-programmering skal programmere en sprite.

3.2.2 Konkret udfordring

Hvordan programmer vi bedst en robot til at komme igennem en forhindringsbane?

Igennem arbejdet i udfordrings- og konstruktionsfasen vil eleverne få en forståelse for sammenhængen mellem årsag og virkning.

3.2.3 Fagligt loop IV: Leg med Bee Bot – hjælp Bee Bot hjem til bikuben

Lad eleverne starte med at være en Bee Bot. Her kan de gå sammen tre og tre, hvor den ene er Bee Bot og den anden elev er programmør.

Den, der er Bee Bot, får bind for øjnene og skal derefter guides via kommandoer såsom "gå frem", "drej til venstre", "drej til højre", "gå tilbage" og "pause/stop". Den sidste elev kan tælle antal brugte skridt, så de

kan holde sig under fx 40 skridt (40 kommandoer er det man max kan programmere sin Bee Bot til at udføre).

Har I adgang til iPads, kan der i App Store downloades app'en "Bee Bot". Her skal bien programmeres igennem forskellige ruter. Dette kan evt. gøres på tavlen, så I starter ud fælles.

Hvad kan en Bee Bot? Få eleverne til at undersøge, hvad den kan. Hvordan den reagerer osv.

Opstil en forhindringsbane, som eleverne skal igennem fx via legeklodser, bøger eller de materialer, I har til rådighed (se eksempler under lærerressourcer, afsnit 5). Her kan det være godt at tilrettelægge banen, så der ikke kun er én rigtig løsning for at komme i mål. Herigennem kan eleverne begynde at sætte ord på, hvordan de løste opgaven.

Under elevressourcer (afsnit 5) findes der arbejdsark til inspiration. Til dette kan en transparent måtte inddelt i et gitter være godt. Alternativt kan en gennemsigtig voksdug eller en rulle papir inddeles i felter

- Løs regnestykker med Bee Bot (arbejdsark).
- Arbejd med forholdsord. Programmer fx Bee Bot til at placere sig foran den røde trekant, oven på den gule firkant, ved siden af cirklen osv.
- Lad eleverne selv formulere opgaver til hinanden.
- Eleverne kan også arbejde med at skrive algoritmer ned på arbejdsarket, inden de programmerer deres Bee Bot. Her kan gitteret fra elevhenvendte ressourcer under robot-kort bruges.
- Lad eleverne lave algoritmer til hinanden, som de senere skal teste.
- Giv også gerne eleverne opgaver, hvor der er indlagt fejl, og lad dem fejlsøge og se, om de kan finde, hvori algoritmefejlen ligger (se under elevhenvendte ressourcer "robot-kort 7").

3.2.4 Feedbackloop

Eleverne giver løbende hinanden feedback på de enkelte opgaver to og to. Fungerede programmerne? Kan der evt. findes alternative ruter? Er der noget, der skal forbedres eller ændres?

3.2.5 Fagligt loop V: Ikon-programmering

Formålet er her, at eleverne får en introduktion til programmering ved hjælp af ikoner.

Her vil eleverne gå fra den lidt mere håndgribelige robot til en sprite, der skal flyttes rundt på skærmen. Eleverne vil arbejde med forskellige sekvenser, som de kan sætte sammen til et samlet forløb.

Til dette tages der udgangspunkt i Scratch Jr., som er en forenkling af Scratch. Det er tilgængeligt i både App Store, Google Play og Chromebooks. Under elevressourcer (afsnit 5) findes der inspiration til, hvordan I kan arbejde med Scratch Jr. Da der er en del tekst til disse opgaver, må der forventes en del lærerstyring.

Til en start kan I sammen undersøge, hvad de enkelte ikoner betyder. Begynd evt. med de gule og blå. Lad evt. også eleverne udfører bevægelserne selv.

- Lad derefter eleverne undersøge spillet.
- Lad eleverne arbejde med at opstille en forhindringsbane af fx trekanter, firkanter og cirkler som deres sprite fx skal over, under, foran, ved siden af osv. Her kan der lægges op til de samme aktiviteter, som med Bee Bot blot ud fra et andet programmeringsprog.
- Lad eleverne designe en opgave, hvor de udfordrer deres sprite til at indtage forskellige placeringer (over, under, foran, bagved osv.).
- Eleverne kan her lave udfordringer til hinanden og afprøve hinandens opgaver, fx *du skal have din sprite til at gå rundt til alle fire hjørner, du skal have din sprite til at stå ovenpå firkanten osv.*

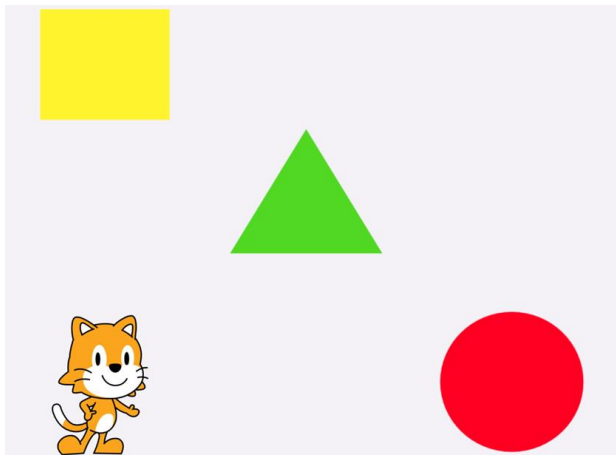
3.2.6 Feedback-loop

Eleverne afprøver og giver hinanden feedback på de enkelte forhindringsbaner og opgaver

- Hvordan kommer de enkelte placeringer til udtryk?
- Hvad kan ændres?

Eksempel:

En elev har lavet denne baggrund på sin iPad og beder nu en anden elev om at få denne sprite til at hoppe op under firkanten, gå over trekanten og stoppe foran cirklen. Eleverne kan fx give hinanden feedback på, om udfordringen var nem at udføre, og om den var forståelig. Ligeledes kan den anden elev give feedback på selve løsningen af udfordringen. Her kan eleverne også sammenligne deres algoritmer (denne kan gemmes inde i Scratch Jr.) for at se, om de er kommet frem til den samme løsning, eller om de er forskellige trods samme slutprodukt.



3.2.7 Afrunding

Afrund denne fase med en drøftelse om forskelle og ligheder mellem de to måder at programmere på.

- Hvad synes eleverne var udfordrende?
- Hvilke forskelle og ligheder kan eleverne få øje på?
- Hvilken type inputs og instruktioner forstod de to programmer (robotter)?

Ved at sammenligne de to måder at programmere på, får eleverne en forståelse for, at robotter kan programmeres igennem forskellige inputs, fx berøring og ikon-programmering. Når eleverne argumenterer for forskelle og ligheder, er dette samtidig med til at styrke deres kommunikationskompetence.

3.3 Outrofase: Ny forståelse og nye kompetencer

3.3.1 Varighed

Ca. 3 lektioner a 45 minutter.

I denne fase skal der ske en perspektivering af emnet, og den nye viden skal konsolideres.

Del 1

Her kan du eller eleverne opstille forskellige forhindringsbaner, som de skal programmere Bee Bot til at komme igennem. De skal her sætte ord og begreber på, hvad Bee Bot gør undervejs igennem forhindringsbanen, fx kører under broen, holder en pause foran trekanten, stopper ved siden af firkanten osv.

Det samme kan laves i Scratch Jr., hvor eleverne via fx smartboard kan vise og fortælle, hvordan de flytter deres sprite rundt på skærmen og hvilke funktioner de har benyttet.

Del 2

Eleverne laver i grupper tegninger, hvor de udtrykker et fremtidsscenario. Dette kunne fx være i form af en serie af tegninger, som de senere formidler for klassen.

- Hvordan tror du fremtiden bliver? Kommer der flere robotter? Prøv her at opstille mulige fremtidsscenarier sammen med eleverne.

Der sluttes af med en klassesamtale, hvor eleverne kan forholde sig til følgende spørgsmål:

- Hvad en robot er?
- Hvilken type algoritme vil en robot kunne forstå?
- Hvilke inputs forstår en robot? Lange eller korte beskeder?
- Hvilke eksempler har eleverne på robotter i deres hverdag?
 1. Hvordan man kan kommunikere med en robot i form af instruktioner? Og hvordan er dette anderledes end at kommunikere med mennesker?

3.3.2 Evaluering

Del 1

Eleverne slutter forløbet af med at formidle, deres tegninger. Prøv her at have fokus på elevernes proces:

- Hvad har de fået ud af at lave tegningen af fremtidssceneriet?
- Kan eleverne selv sætte ord på, hvad de har lært?
- Hvilke tegn på opfyldelse af læringsmålene kan ses?

Eleverne kan også inddrages ved at give respons på hinandens tegninger:

- Kommer der nogle af de former (trekanter, firkanter, cirkler) til udtryk, som I har arbejdet med?
- Hvilke former kommer til udtryk i tegningerne? Hvilke farver havde de?

- Hvilke forholdsord bliver brugt, når eleverne fortæller om deres tegninger?
- Hvad tænker I om gruppens fremtidsscenarie – tror I det bliver virkeligt?

Del 2

Eleverne skal her prøve at sætte ord på, hvilke erfaringer de tager med fra forløbet.

- Hvad vil de huske at gøre næste gang, de skal arbejde med programmering og robotter?
- Hertil kan der udformes en liste med fx tre punkter, der kan tages op næste gang eleverne møder de samme udfordringer.

4. Perspektivering

4.1 Evaluering

Rammen for evalueringen af elevernes udbytte er en vurdering af de tegn på læring, som under forløbet og præsentationer af tegninger er observeret og nedskrevet i lærerens logbog. Til at begrænse sig og fokusere og gøre deres tegn på læring til et anvendeligt redskab i evalueringen af elevernes udbytte er koblingen til forløbet læringsmål en hjælp.

Der samles op på de enkelte aktiviteter og elevernes opnåelse af læringsmålene for forløbet:

- Eleven kan gennem unplugged aktiviteter sammensætte forskellige sekvenser til simple programmer.
- Eleven kan sammensætte en sekvens af ikoner til et program, der får en kammerat/sprite/robot til at bevæge sig på en bestemt måde.
- Eleverne kan ved hjælp af forholdsord programmere sekvenser, der placerer deres kammerat/sprite/robot i forhold til tal og former.

Hvorvidt eleverne har nået de planlagte læringsmål kan observeres blandt andet gennem aktiviteterne i afsnit 3.3.2. Desuden lægges gruppernes tegninger på læringsplatformen, hvor de også indgår som data for lærerens samlede evaluering af elevernes udbytte.

4.2 Progression

Dette forløb danner et fundament for, at eleverne kan begynde at handle hensigtsmæssigt med digitale teknologier. Færdigheds- og vidensområdet programmering i kompetenceområdet teknologiforståelse trækker tråde ind i de fleste andre færdigheds- og vidensområder af teknologiforståelse i matematik. Forløbet danner grundlag for, at eleverne selv kan begynde at konstruere enkle programmer i fx Scratch Jr., og de får en forståelse for, hvordan teknologi og robotter er en del af deres hverdag.

4.3 Differentieringsmuligheder

Differentieringen er tænkt ind i forløbets faglige loops. Her er der mulighed for at sammensætte grupperne alt efter niveau, og alt efter hvor hurtigt eleverne løser opgaverne. Ligeledes lægger de faglige loops op til, at opgaverne kan løses på mange forskellige måder. Så her kan man også graduere sværhedsgraden for eleverne.

Under elevressourcer (afsnit 5) findes ligeledes forskellige arbejdsark, der kan differentieres alt efter elevernes niveau.

4.4 Særlige opmærksomhedspunkter

Det kan være relevant at tage kontakt til det lokale CFU med henblik på at låne fx Bee Bots eller andre robotter.

5. Lærer- og elevhenvendte ressourcer

5.1 Lærerhenvendte ressourcer

De lærerfaglige ressourcer er knyttet til loop-pakker: faglige loops og feedback-loops. De kan bl.a. indeholde lærebogstekster, beskrivelser af undersøgelser/studier, billeder, linksamlinger, it-vejledninger, designprocesmodel, m.m.

- Programmet Lightbot kan også bruges som introduktion til programmering. Her øger elevernes deres indsigt i programmering, som spillet skrider frem: <https://lightbot.com/>
- Scratch Jr.: Kirsten Dam Pedersen har udviklet denne vejledning på dansk til Scratch Jr på dansk: http://www.scratch-paa-dansk.nu/wp-content/uploads/2017/09/ScratchJr_undervisningsmateriale.pdf
- Scratch Jr. Interface: <https://www.scratchjr.org/learn/interface>
- Scratch Jr. guide YouTube: <https://www.youtube.com/watch?v=s6XvwEH0xpg>
- Bee Bot: MitCFU VIA - vejledning til brug af Bee Bot: <mitcfu.dk/pv/viacfu99872754.pdf>
- Bee Bot - aktiviteter:
 1. <http://klubkodesmart.dk/kodestafetten/bee-bot-hour-of-code/>
 2. <http://klubkodesmart.dk/kodestafetten/go-go-prints/>
- Kom i gang med Bee Bot: <https://www.youtube.com/watch?v=uULpYYE1Agc>
- Bee Bot - labyriant: <https://www.youtube.com/watch?v=9KNfgmD-PVM>
- Bee Bot App guide: <https://www.youtube.com/watch?v=aal6pF4cVUA>
- Unplugged aktiviteter: <https://code.org/curriculum/unplugged>

Link til artikler om fremtidsscenerier

- <https://www.dr.dk/nyheder/viden/teknologi/tyskland-vil-teste-flyvende-taxier>
- <https://www.dr.dk/nyheder/viden/teknologi/stjaeler-robotten-dit-arbejde-i-fremtiden>
- <https://www.dr.dk/nyheder/viden/teknologi/storbyer-tester-loes-her-er-selvkoerende-biler-paa-vejene-lige-nu>

5.2 Elevhenvendte ressourcer

De elevhenvendte ressourcer er faglige eller procesrelaterede materialer, som eleverne selvstændigt kan anvende i deres undersøgelses- og arbejdsproces. Det kan fx være vejledning til programmer eller teknologi, en lærebogstekst, ordlister, billeder, m.m.

Arbejdsarkene er tænkt som inspiration, som kan udvikles og tilpasses den enkelte skole. Arbejdsarkene kan tilgås via OneDrive og er langt ind som link herunder:

- [Arbejdsark: Robot-kort](#)
- [Arbejdsark: Bee Bots](#)
- [Arbejdsark: Scratch Jr.](#)