

TEKNOLOGIFORSTÅELSE

MATEMATIK 8. KLASSE
FORÅR

OPDATER TERNINGER

Udarbejdet af Adrian Rau Bull i samarbejde med Camilla Finsterbach Kaup, 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.....	6
3. Forløbsnær del.....	8
3.1 Introfase: Forforståelse og kompetencer	8
3.2 Udfordrings- og konstruktionsfase.....	10
3.3 Outrofase: Ny forståelse og nye kompetencer	12
4. Perspektivering.....	12
4.1 Evaluering	12
4.2 Progression	13
4.3 Differentieringsmuligheder.....	13

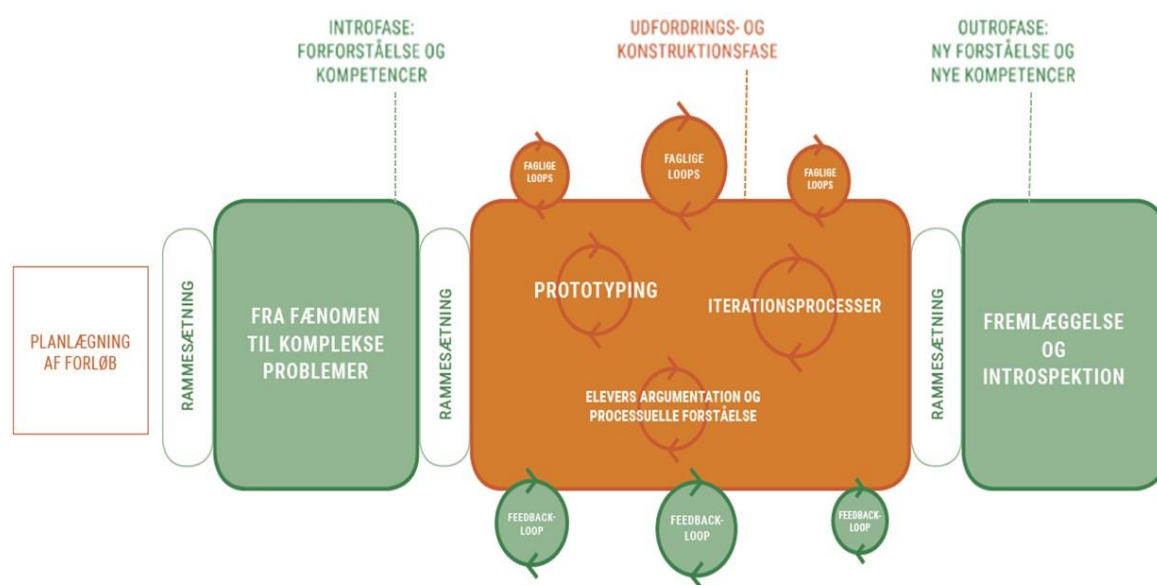
Version 2

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: Forløbsmodel for prototyperne



1.1 Beskrivelse

I dette forløb skal eleverne arbejde med at udvikle nye digitale terninger. Eleverne skal afprøve forskellige programmer med digitale terninger og selv redigere og udvikle disse programmer. Eleverne skal også konstruere nye programmer med digitale terninger i både Excel, GeoGebra og Micro:bit. Nogle af programmerne konstrueres gennem tilhørende vejledninger, og andre konstrueres gennem redigering og omstrukturering af eksisterende programmer.

Fysiske terninger kan bruges til at udvikle elevernes forståelse af sandsynlighed, hvor digitale terninger kan tage denne viden et skridt videre, fordi digitale terninger hurtigt kan undersøges gennem mange slag (10.000 eller flere), og fordi man lettere kan få fokus på ikke-regulære terningslag. Det kan også være terningslag, som ikke er muligt at lave fysisk, fx 3 sidede-terninger.

I udskolingen arbejdes der i matematik med både statistisk sandsynlighed, dvs. frekvens og teoretisk (beregnet) sandsynlighed. I dette forløb sker det ved, at eleverne skal simulere terningslag og undersøge disse terningslag ved at sammenligne antallet af udfald.

Eleverne skal udføre en undersøgelse af regulære 6-sidede terninger med de forskellige programmer og undersøge, hvordan programmerne er forskellige. Eleverne skal også reprogrammere disse programmer til at vise andre slags terninger, både regulære terninger, hvor der er samme sandsynlighed for alle udfald og ikke regulære terninger, hvor udfaldene ikke har samme sandsynlighed.

I dette forløb skal eleverne udvikle, undersøge og redigere programmeret simulering af terningslag gennem tre faser.

1. Udvikling og programmering af terningslag i Excel vha. af vejledning
2. Udvikling og programmering af terningslag i GeoGebra vha. af vejledning
3. Undersøgelse og reprogrammering af programmerne ” opdater 6-sidet···” . Tre forskelligt programmerede programmer i henholdsvis GeoGebra og Micro:bit

Disse tre faser vil samtidig fungere som progressionen af faglige loops, som eleverne skal igennem i forløbet. I hvert af disse faglige loops er der inspiration til forskellige temaer og aktiviteter, som eleverne kan arbejde med. Men det er læreren selv, der ud fra denne inspiration skal sammensætte et program for de tre faser, der er tilpasset eleverne i hans/hendes klasse.

Teknologiforståelsesdelen centrerer omkring forståelse og udvikling af simuleringer af digitale terningslag i matematik.

1.1.1 Produkt:

Udviklingen af digitale terninger, både regulære og ikke-regulære terninger; både i Excel, GeoGebra og Micro:Bit.

1.2 Rammer og praktiske forhold

1.2.1 Samlet varighed

Estimeret minimum til 11 lektioner. Kan dog variere meget afhængigt af elementerne i de faglige loops.

1.2.2 Materialer

Analoge teknologier/materialer

Ingen særlige krav

Digitale teknologier

Adgang til computere for alle elever eller grupper.

Brug af Micro:bit og adgang til programmerne Excel og GeoGebra.

Adgang til materialer: Programmerne:

Excel - og den tilhørende vejledning: "Om at konstruere og simulere terningslag i Excel" med de tilhørende filer: "Mange terninger", "Første mærkelige 6-sidet terning", "Anden mærkelige 6-sidet terning", "Mange mærkelige terninger".

GeoGebra - og den tilhørende vejledning: "Om at konstruere og simulere terningslag i GeoGebra" med den tilhørende fil: "En mærkelig terning GeoGebra", samt programmerne: "Opdater 6-sidet terning JavaScript", "Opdater 6-sidet terning objektprogrammeret".

Micro bit - og den tilhørende programmeringsmanual "Opdater 6-sidet terning Micro bit".

I introfasen: Excel-vejledningen "Om at konstruere og simulere terningslag i Excel", samt med dertil hørende Excel-filer: "Mange terninger", "Første mærkelige 6-sidet terning", "Anden mærkelige 6-sidet terning", "Mange mærkelige terninger".

I første del af konstruktionsfasen: GeoGebra-vejledningen "Om at konstruere og simulere terningslag i GeoGebra", samt med dertil hørende GeoGebra-filen "En mærkelig terning GeoGebra".

I Anden del af konstruktionsfasen: Konstruktionsbeskrivelse "Opdater 6-sidet terning Micro bit", samt GeoGebra-filerne: "Opdater 6-sidet terning objektprogrammering" og "Opdater 6-sidet terning JavaScript".

Elevhenvendte ressourcer (herunder evt. hjemmesideadresser, som ikke findes i ressourcebanken)

Af TEK-projektet udviklede vejledninger til Excel og GeoGebra, som eleverne skal bruge, når de udvikler programmer og de tilhørende programfiler.

Adgang til programmer i Excel, GeoGebra og Micro:bit, som eleverne skal redigere og omprogrammere.

Til forløbet kan de nævnte filer findes under forløbet på <https://tekforsøget.dk/>

GeoGebra-filerne kan desuden findes på:

<https://www.geogebra.org/m/ww4ntg6d>

Eksempel på ikke-regulær terning i MicroBit:

https://makecode.microbit.org/_EfyMsL5pkfHW

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 vidensområderne "programmering" og "data, algoritmer og strukturering".

KOMPETENCEOMRÅDER	STATISTIK OG SANDSYNLIGHED	TEKNOLOGIFORSTÅELSE
Kompetencemål (efter 9. klassetrin)	Eleven kan vurdere statistiske undersøgelser og anvende sandsynlighed	Eleven kan handle med dømmekraft med digitale teknologier i arbejdet med åbne problemstillinger fra omverdenen
Færdigheds- og vidensmål (efter 9. klassetrin)	<p>Sandsynlighed</p> <ul style="list-style-type: none"> ■ Eleven kan anvende udfaldsrum og tælle måder til at forbinde enkle sandsynligheder med tal ■ Eleven har viden om udfaldsrum og tælle måder ■ Eleven kan anvende sandsynlighedsregning ■ Eleven har viden om statistisk og teoretisk sandsynlighed <p>Statistik</p> <ul style="list-style-type: none"> ■ Eleven kan kritisk vurdere statistiske undersøgelser og præsentationer af data. ■ Eleven har viden om stikprøveundersøgelser og virkemidler i præsentation af data 	<p>Data, algoritmer og strukturering</p> <ul style="list-style-type: none"> ■ Eleven kan genkende og anvende mønstre i strukturering af data og algoritmer med udgangspunkt i konkrete problemstillinger. ■ Eleven har viden om mønstre i strukturering af data og algoritmer. <p>Programmering</p> <ul style="list-style-type: none"> ■ Eleven kan modificere og konstruere programmer til løsning af en given opgave ■ Eleven har viden om metoder til trinvis udvikling af programmer

Konkretiserede læringsmål:

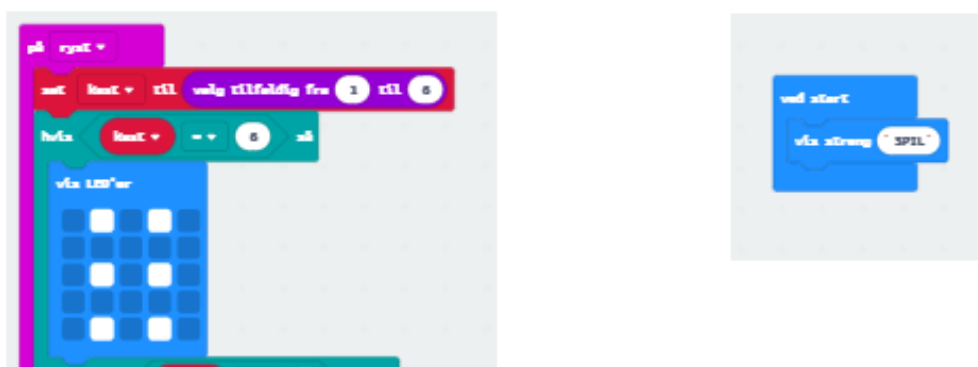
- Eleven kan genkende, (sammenligne) og anvende mønstre i strukturering af data i Excel og GeoGebra .
- Eleven får viden om metoder til trinvis udvikling af programmer Excel, GeoGebra og Micro:bit.
- Eleven kan modificere programmerne til løsning af en given opgave (undersøgelse af forskellige terninger, ved sammenligning af statistisk og teoretisk sandsynlighed).

Centrale begreber

Objekt-programmering (objektorienteret programmering): Er i dette forløb forstået som indkapsling, dvs. hvor objekter programmeres hver især, fx at man programmerer hvert objekt på tegnefladen i GeoGebra, dvs. hver tekst har en programmering, hver knap har en programmering osv. Eleverne skal ind i selve objekterne for at undersøge, hvordan de hver især er programmeret.

Block-programmering: Programmering som foregår gennem sammensætningen af forskellige blokke fx i Scratch og Micro:bit, hvor programmeringen af et terningslag foregår ved sammensætningen af blokke i forskellige farver (se illustrationen nedenfor).

Blocks



JavaScript: JavaScript er et avanceret programmeringssprog, der bl.a. bliver brugt til HTML og WWW, men det er også indlejret i GeoGebra, hvor det kan findes ved, at man i et objekt trykker på "Egenskaber" og derefter fanebladet "Scripting" og derunder fanebladet "Globalt JavaScript".

JavaScript er på engelsk og indeholder mange forskellige faste funktioner, som kun virker, hvis det er skrevet helt korrekt. fx giver $\text{Math.floor}((\text{Math.random}()*6)+1)$; en 6-sidet terning.

Elever skal i dette forløb programmere i JavaScript, men de skal prøve at indtaste en JavaScript i Micro:bit og redigere i eksisterende Globalt JavaScript i GeoGebra.

Teoretisk sandsynlighed:

Der er i dette forløb primært fokus på de kendte sandsynligheder ved kast med en regulær terning, fx at sandsynligheden for at slå "2" i et slag med en regulær 6-sidet terning er lig med $1/6$ eller 16,6 pct. Og sandsynligheden for at slå "9" i et slag med en 10-sidet terning er lig med $1/10$ eller 10 pct .

Statistik sandsynlighed:

Hvad er sandsynligheden, når man udfører en række eksperimenter, dvs. frekvensen for et udfald, når man statistisk set har udført eksperimentet et antal gange. Hvis man fx kaster en regulær 6-sidet terning 100.000 gange, så vil den statistiske sandsynlighed for slaget "3" være lig med $1/6$ eller 16,6 pct. Hvis vi kun kaster en 6-sidet terning tre gange og ikke får udfaldet "3", vil vi sige, at den statistiske sandsynlighed for "3" i dette tilfælde var nul; dog må man sige, at det ikke er et særligt godt eksperiment.

De store tals lov:

De store tals lov siger, at når man udfører et forsøg et tilpas højt antal gange, så vil den statistiske sandsynlighed gå mod den sandsynlighed, der reelt er. Dvs. hvis man slår nok slag med en regulær 6-sidet terning, vil den statistiske sandsynlig nærme sig den teoretiske sandsynlighed, som er $1/6$ eller 16,6 pct. for alle slagene. Hvis man kaster nok slag med en ikke regulær terning, kan man finde ud, hvad sandsynligheden er for de forskellige slag ved at se på frekvensen for slagene, men kun hvis antallet af slagene er stort nok.

Andre teknologiforståelsesbegreber kan findes på: <https://emu.dk/grundskole/teknologiforstaelse/ordliste>

3. Forløbsnær del

I dette forløb skal eleverne udvikle, undersøge og redigere programmeret simulering af terningslag gennem tre faser med henholdsvis Excel, GeoGebra og Micro:bit. Disse tre faser vil samtidig fungere som progressionen af faglige loops, som eleverne skal igennem i forløbet.

3.1 Introfase: Forforståelse og kompetencer

I Introdelen skal eleverne udvikle programmer, der simulerer terningslag i Excel. Udviklingen foregår ved, at eleverne følger vejledningen "Om at konstruere og simulere terningslag i Excel". I vejledningen bliver de hjulpet til at konstruere programmer, der kan simulere terningslag. En del af vejledningen går ud på, at eleverne skal undersøge og ændre i programmerne "Mange terninger", "første mærkelige 6-sidet terning" og "anden mærkelige 6-sidet terning". Disse programfiler præsenterer blandt andet nogle ikke-regulære 6-sidede terninger.

Til sidst skal de selv prøve at konstruere et program i Excel, der viser sammenhængen mellem statistisk sandsynlighed og teoretisk sandsynlighed for en ikke-regulær terning. Dvs. at de skal arbejde med de store tals lov, gennem statistisk simulering af mange terningslag med en skæv digital terning.

3.1.1 Varighed

Ca. 3 lektioner

3.1.2 Komplekst problemfelt

Over tid vil flere og flere terninger og spil blive digitaliseret. Således er mange kasinoer nu online. Alle har hørt om korttricks og andre gammeldags måder at snyde i spil, men hvordan fungerer spil, når de er programmeret, og hvilke muligheder er der for at påvirke sandsynligheden for bestemte udfald. Eleverne arbejder med at udvikle et konkret program, der kan simulere terningslag i GeoGebra, som kan give indblik i dette større problemfelt.

3.1.3 Problemstilling

Problemstilling: Hvordan undersøger man bedst den statistiske sandsynlighed for både almindelige terninger og ikke-regulære terninger, uden at slå terninger i dagevis hver gang?

3.1.4 Iscenesættelse/scenarie:

Læreren kan spørge eleverne *"Har I nogensinde spillet yatzy. I yatzy spilles der 15 runder i forsøg på at opnå 15 forskellige kombinationer, og spillet vindes af den spiller, der har flest point til slut. I hver runde har man tre kast til at opnå et ønsket resultat. Men hvordan ved man, hvilke terningslag, det er værd at satse på?"*

Læreren fortsætter: *"I har måske nogenlunde styr på, hvilke slag I vil satse på, når I spiller yatzy, men hvad hvis vi ændrer terningerne? Hvad sker der så?"*

3.1.5 Fagligt Loop

Eleverne arbejder med vejledningen "Om at konstruere og simulere terningslag i Excel". I denne vejledning bliver de hjulpet til at konstruere et program, der kan simulere terningslag. Vejledningen har forskellige undersøgelses- og programmeringsopgaver, der stiger i niveau. Vejledningen slutter med, at eleverne selv skal konstruere et program, der viser de store tals lov på en selvkonstrueret ikke-regulær terning.

Iscenesættelse til eleverne (introdelt):

Du kan konstruere dine egne terninger digitalt på en måde, hvor programmet viser sandsynligheden for de forskellige terningslag. Du kan selv vælge, hvilke digitale terninger du vil konstruere, og om de skal være retfærdige. I den del skal du lære at konstruere dine egne terningslag i Excel, så du opnår indsigt i, hvordan terningslag er konstrueret i online kasinoer, men også hvor let det er at ændre på terningslagene, så der ikke er lige stor sandsynlighed for de forskellige udfald.

3.1.6 Faglige loops

- Det kan evt. være nødvendigt at have et fagligt loop, hvor eleverne lærer om de tre typer sandsynligheder. Dette forløb har dog ikke særligt fokus på den intuitive sandsynlighed, men derimod er der et særligt fokus på de store tals lov, som er sammenhængen mellem den teoretiske sandsynlighed og den statistiske sandsynlighed.
- Der kan være et fagligt loop, der viser nogle basale ting i Excel, men det burde dog ikke være nødvendigt, da vejledningen går meget i detaljer med programmet.
- Hvis eleverne vil bruge Google Sheets i stedet for Excel, burde det være muligt. Det kan i så fald være nødvendigt med et fagligt loop, hvor de brugte funktioner i vejledningen bliver fundet og brugt i Google Sheets (dette indgår ikke i denne forløbsbeskrivelse).

Afrunding/feedbackloop

Hvert par/gruppe opsamler erfaringer fra introfasen, fx. i logbog.

- Hvad har vi undersøgt?
- Hvordan virker Excel til at udregne sandsynlighed?

3.2 Udfordrings- og konstruktionsfase

I udviklingen af programmet kommer eleverne til at arbejde med dynamiske elementer som "knap", "skyder" og "tekst". Dermed søges der at skabe forståelse for de mere avancerede programmer, som eleverne skal undersøge og redigere i anden del af konstruktionsfasen. Eleverne skal undersøge og ændre programmer med forskellige typer af programmering, både JavaScript, block-programmering og objekt-programmering. Eleverne får indsigt i regulære og ikke-regulære terninger fx til brug af online-kasinoer. Denne sammenhæng knyttes til retfærdighed i spil og valg i forbindelse med dette.

Faglige loops i konstruktionsfasen

1. Om at konstruere og simulere terningseslag i GeoGebra – eleverne skal ligesom i introen prøve at udvikle et program til simulering af terningseslag. Denne gang skal de udvikle programmet i GeoGebra.
2. Undersøge forskellige programmer med flere typer af programmering og redigere disse.

3.2.1 Varighed

Udforskningsfasen er estimeret til 6 lektioner (cirka 3 lektioner pr. del)

3.2.2 Konkret udfordring

1. I første del skal eleverne selv udvikle et terningeprogram i GeoGebra ved at bruge vejledningen "Om at konstruere og simulere terningseslag i GeoGebra".
2. Dernæst skal de undersøge forskellige programmer med flere typer af programmering og redigere disse (Micro:bit og GeoGebra).

3.2.3 Fagligt loop:

Eleverne følger i første del vejledningen "Om at konstruere og simulere terningseslag i GeoGebra". I denne bliver de hjulpet til at konstruere et program, der kan simulere terningseslag. Vejledning har forskellige undersøgelses- og programmeringsopgaver, der stiger i niveau. Vejledningen slutter med, at eleverne selv skal redigere programmet "En mærkelig terning GeoGebra" og undersøge programmeringen bag terningen i samarbejde med en klassekammerat.

Introduktion til eleverne:

I denne del skal lære at konstruere dine egne terningseslag i GeoGebra. Du kan konstruere dine egne terninger digitalt. Du kan selv vælge, hvilke digitale terninger du vil konstruere, og om de skal være retfærdige. På denne måde kan du få indsigt i, hvordan digitale terninger i GeoGebra bliver programmeret, men også på forskelle og ligheder mellem programmeringen af terninger i Excel og GeoGebra. Du kan også få indsigt i, hvor let det er at ændre på din terning i programmet.

3.2.4 Fagligt loop:

I andet loop skal eleverne gå på opdagelse i programmerne Micro:bit med to typer programmering og GeoGebra med to typer programmering.

Instruktion til eleverne:

I skal undersøge to programmer af digitale terninger.
Programmerne er konstrueret med to forskellige programmeringer.

I **MicroBit-delen** skal I først prøve at programmere jeres MicroBit med de to typer af programmering og undersøge, om det er det samme program, I ender ud med.

Herefter skal I omprogrammere, så I får en anden terning end den 6-sidede (fx en 7-sided terning).

Til sidst skal I konstruere en ikke-regulær terning (fx en 7-sided terning der oftere slår 2).

I **GeoGebra-delen** skal I undersøge de to programmer og finde ud af, hvad der er ens i programmerne, og hvad der er forskelligt; hvorfor er det ene program objektprogrammering, mens det andet er JavaScript? Herefter skal I ændre programmerne således, at I får en anden terning end den 6-sidede (fx en 5-sided terning i det ene program og en 4-sided terning i det andet program).

Til sidst skal I konstruere ikke regulære terninger i programmerne.

- Opdater 6-sided terning Micro:bit
 - Eleverne skal programmere en Micro:bit med to typer af kode.
 - Herefter skal eleverne lave den 6-terning om til en anden type terning (fx en 7-sided terning).
 - Endelig skal de lave en ikke-regulær terning (fx en 7-sided terning der oftere slår 2).
- Opdater 6-sided terning objektprogrammering
 - Eleverne skal redigere programmet til at vise en anden type terning (fx 5-sided terning).
 - Endelig skal de lave en ikke-regulær terning (fx en 5-sided terning der oftere slår 3).
- Opdater 6-sided terning JavaScript
 - Eleverne skal redigere programmet til at vise en anden type terning (fx 4-sided terning).
 - Endelig skal de lave en ikke-regulær terning (fx en 4-sided terning der oftere slår 4).

3.2.5 Feedbackloop

Eleverne præsenterer de redigerede programmer for hinanden og viser, hvordan programmerne kan bruges til at simulere terningslag og vise sammenhængen mellem teoretisk sandsynlighed for terningslag og statistisk sandsynlighed for terningslag. Denne sammenhæng kaldes for de store tals lov.

Drøft med eleverne:

- Hvordan virker programmerne?
- Er det et godt program?
- Hvordan kunne man gøre programmet bedre?

3.3 Outrofase: Ny forståelse og nye kompetencer

3.3.1 Varighed

Estimeret 1-2 lektioner a 45 minutter. Alt efter hvor meget og hvor mange elever, der ønsker at vise noget.

3.3.2 Fremlæggelse og introspektion

Gennem samtale af teknologifaglig og matematikfaglig karakter har eleverne mulighed for at udvikle bevidsthed om de erfaringer og den viden, de har genereret undervejs i forløbet - ligesom læreren kan se tegn på læring.

I denne fase skal eleverne forholde sig til:

- Hvordan er programmerne forskellige fra virkelige terningeslag?
- Hvad er forskelligt i programmerne?
- Hvad er ens i programmerne?
- Hvilke muligheder og begrænsninger er der, når man selv programmerer terningeslag?

I fasen kan eleverne også vise, hvordan man udvikler og redigerer programmerne: Excel, Micro:bit og GeoGebra. De må gerne vise andre elementer end lige terningeslag.

Fremlæggelsen sker ved, at eleverne i første omgang mundtligt præsenterer, hvordan de udviklede og redigerede programmerne. Gruppen viser deres programmer. Eleverne fortæller dernæst, hvilke problemer de er stødt på undervejs, og hvordan de har løst dem. Respons fra de afprøvende grupper og læreren.

Eleverne reflekterer således over det lærte gennem fremlæggelse af udfordringer undervejs samt de råd, de har modtaget af lærer og andre grupper. Hvordan førte arbejdet med udfordringerne frem til et andet og bedre resultat?

4. Perspektivering

I forløbet er der forskellige processer i spil. Det er derfor nødvendigt at sammenholde disse processer og forholde dem til den komplekse problemstilling. I dette afsnit kan man blive hjulpet til dette.

4.1 Evaluering

Feedback indbygges undervejs i alle faser, så hver fase afsluttes med et feedbackloop, hvor læreren og de andre elever giver feedback på fasens produkter.

Undervejs i forløbet vil det være nødvendigt at relatere til de komplekse problemstillinger. Derfor er det nødvendigt at fokusere eleverne opmærksomhed på de forskellige programmeringssprog – men også på sandsynlighederne ved forskellige terningeslag og brugen af disse i forskellige spil.

En måde systematisk at dokumentere elevernes progression kan være med lærerens logbog. Med den kan lærere notere systematiske optegnelser af iagttagelser i forhold til tegn på elevens læring undervejs i feedbackloops og ved præsentationen i sidste lektion; bl.a. ud fra nedenstående evalueringsspørgsmål i forhold til målene:

- Kan eleven formulere sig om gruppens design med anvendelse af fagbegreber?
- Er eleven tryk ved brugen af fagspecifikke begreber?
- Forstår eleven teoretisk sandsynlighed?
- Kan eleven forestille sig en rækkefølge af begivenheder, som kan lede frem mod et mål og omformulere forestillingen til en antagelse i lighed med: Hvis vi gør sådan, så får vi dette resultat?

4.2 Progression

Progression går hele tiden fra det rammesatte lærestyrede undervisningsmiljø mod elevcentrerede kreative designs. I starten foregår dette skift ved, at vejledningerne bliver mere og mere frie og åbne. Men i de senere aktiviteter (konstruktionsfasen 2.del) er der mindre vejledning og derved større frihedsgrader. Disse aktiviteter er rettet mod, at eleverne arbejder sammen i små grupper (evt. par) og bruger lærerne som guide, vejleder og facilitator. Forløbet tager desuden udgangspunkt i, at man har arbejdet med andre forløb med brug af GeoGebra-filer, fx statistiker med bias, hvor man lærer at downloade og afspille GeoGebra-filer.

4.3 Differentieringsmuligheder

En del af differentieringen er rettet mod, hvor langt eleverne når i de to vejledninger (introfase og konstruktionsfasen 1. del). Disse vejledninger bliver sværere og sværere samtidig med, at frihedsgraderne bliver større, og det er ikke nødvendigvis meningen, at alle eleverne når igennem det hele. Det vil være muligt at differentiere med forslag til minimumskrav og højdespringere i de forskellige faser. I den sene fase er det meningen, at eleverne arbejder sammen i grupper (konstruktionsfasen 2. del), og outro-delen kunne foregå i grupper eller i plenum men med individuel refleksion.

Introfase: Minimumskrav: Eleverne får konstrueret et program der kan slå med mange terninger i Excel.

Højdespringere: Eleverne får konstrueret et program, der kan undersøge sammenhængen mellem teoretisk og statistisk sandsynlighed for ikke-regulære terninger, der ikke er 6-sidede.

Konstruktionsfase 1. del: Minimumskrav: Eleverne får konstrueret et program, der kan slå med terninger i GeoGebra. **Højdespringere:** Eleverne får konstrueret et program, der kan undersøge sammenhængen mellem teoretisk og statistisk sandsynlighed for ikke-regulære terninger, der ikke er 6-sidede; og kan finde alle kombinationer af højest og mindst når skyderne går til 10.

Konstruktionsfase 2. del: Minimumskrav: Eleverne bliver opmærksomme på, at det samme program kan konstrueres i to forskellige programmeringssprog, og de har set eksempler på disse sprog.

Højdespringere: Eleverne kan konstruere et program med ikke-regulære terninger, der ikke er 6-sidede i MicroBit, og de har forståelse for både blokprogrammeringen og JavaScript bag dette program. Eleverne kan ligeledes redigere i begge GeoGebra programmerne således, at de viser korrekte statistikker af ikke-regulære terninger mindre end den 6-sidede.