

## Programmering med matematik

Ola Helenius, NCM och Morten Misfeldt, Aalborg universitet

När man vill få datorprogram att fungera är matematik användbart, och att tänka matematiskt kan vara en hjälp när program ska konstrueras eller genomskådas. Därför kan arbete med programmering både användas till att lära sig matematik och till att använda matematik som man redan kan.

Att kunna programmera handlar inte bara om att kunna ett antal kommandon. Man programmerar alltid något som handlar om tal, rum, tid, logik eller språk. Matematik är ett kompakt och effektivt språk som kan beskriva sådana kvantitativa, rumsliga, logiska och semantiska fenomen på ett sätt som ofta relativt enkelt kan omvandlas till kod som kan tolkas av en dator. Som vi skrev i Del 2 är det dock inte säkert att den som programmerar alltid går vägen över en matematisk beskrivning av världen, eller uppfattar programkoden som en slags matematisk representation.

Utmaningen för undervisningen blir således hur vi ska bygga broar mellan matematikundervisningens matematik och den matematik som är ändamålsenlig i programmeringsarbetet. Syftet med detta material är att uppmärksamma detta, så att läraren kan organisera undervisning med programmering på ett sådant sätt att eleverna kan få använda den matematik de redan kan för att förbättra sina digitala produkter och lära sig mer matematik. Vi kommer att se på dels vilken matematik som kan användas i programmeringsuppgifter och dels på hur denna matematik tydliggörs i programmeringsarbetet.

### **Programmering som modellering**

Mycket av den matematik som eleverna lär sig i grundskolan och gymnasiet är användbar i programmeringsaktiviteter, eftersom den kan användas för att beskriva världen, och många programmeringsuppgifter bygger på en matematisk modell av världen. Det finns dessutom en del matematiskt tänkande i själva konstruktionen av ett datorprogram, med dess inbyggda logik och algoritmer.

När man exempelvis konstruerar ett spel i Scratch eller en robotstyrning i Lego mindstorms är det nödvändigt att bygga upp en god förståelse för den situation som ska programmeras. Denna förståelse ska vara explicit så att situationen kan programmeras. Om till exempel en figur ska röra sig behöver vi fundera ut hur. Ska den röra sig fram och tillbaka, upp och ned, i cirklar, i olika hastighet och så vidare? De aspekter av situationen som man vill ha med i sitt program måste så att säga göras operationaliserbara. Till exempel måste vi välja sätt att representera riktningar, lägen och hastigheter. På detta sätt påminner programmeringsaktiviteter om matematisk modellering, där man beskriver vissa utvalda aspekter av en situation med matematisk terminologi. När man programmerar uttrycker

man vissa utvalda aspekter av situationen i kod, eventuellt efter att först ha gått via en matematisk representation.

Som ett exempel på ovanstående kan vi se på att bygga – och i synnerhet att programmera – en robot i Lego mindstorms, som kan följa en på förhand given bana. Roboten är utrustad med 2 motorer, som gör att den kör rakt framåt och bakåt om båda motorerna är påslagna samtidigt och svänger om antingen endast den ena motorn är påslagen eller om de två motorerna drar åt varsitt håll. Robotens rörelser programmeras genom att man talar om hur motorerna ska röra sig, hur många motorvarv de ska göra. Eleverna kan exempelvis gå fram så här:

- 1: De kan undersöka hur långt roboten förflyttar sig på ett motorvarv. Detta kan de antingen göra genom att mäta hur långt roboten förflyttar sig på ett motorvarv (se Figur 1) eller genom att mäta hjulets diameter och beräkna vilken sträcka ett varv motsvarar.
- 2: På motsvarande sätt kan de bestämma hur mycket roboten svänger när de två motorerna snurrar åt varsitt håll.
- 3: Med denna information har eleverna tillräcklig information för att skapa en modell av hur roboten förflyttar sig framåt och svänger.
- 4: Varje bana som roboten ska köra kan nu modelleras med ett antal kör-framåt-rotationer och ett antal svängningar.



Figur 1. Elever mäter hur långt roboten kör på ett varv.

## **Matematik och programmering**

För att få en tydligare uppfattning om hur matematikundervisning och programmeringsaktiviteter ömsesidigt kan stärka varandra, kan vi jämföra de två områdena. Vilka aspekter av matematik och matematiskt arbete är viktiga i programmering och tvärt om? Vilka överlapp finns det mellan matematiskt tänkande och datalogiskt eller programmeringsrelaterat tänkande? I två forskningsartiklar (Ejsing-Duun & Misfeldt, 2016; Misfeldt & Ejsing-Duun, 2015) identifierades följande exempel på hur programmering och matematik samspelar.

*Algoritmer och matematik:* Vid programmering arbetar eleverna med algoritmer i betydelsen systematiska beskrivningar av problemlösningstrategier som bygger på samband mellan orsak och verkan samt på stegvisa beskrivningar av händelseförlopp. Algoritmiskt tänkande handlar om att få maskiner att utföra sådana algoritmer och är på många sätt detsamma som matematiskt tänkande. Den typ av matematik som används här är logik, ändliga följder och induktiva argument.

*Matematiken i elevernas spel:* De flesta program bygger på en modell av världen, och modellen är eller kan bli matematisk. Att programmera kan därför, i efterhand, på förhand eller parallellt med själva programmeringen, bli en slags matematisk modelleringsuppgift. Sådana modeller handlar ofta om geometriska förhållanden och rörelser, exempelvis beskrivna med koordinatsystem, eller förhållanden som kan beskrivas med algebra (uttryckas med en formel) och därmed ingå i ett datorprogram. I allmänhet kan det dock handla om matematik inom många olika områden.

*Matematiskt tänkande och kunnande kan utvecklas genom programmering:* Eleverna kan naturligtvis utveckla matematik och matematiskt tänkande när programmering används som ett pedagogiskt och matematiskt verktyg. Detta kommer att behandlas i Del 4.

För att utnyttja programmeringsaktiviteter i den vanliga matematikundervisningen är det bra att vara uppmärksam på de möjligheter till matematiklärande som uppstår i elevernas arbete med programmering. Det är inte riktigt samma sak som att på förhand själv fundera ut vilken matematik som är lämplig för att modellera det problem eller den situation som ska programmeras. I *elevernas* arbete kan nämligen annan matematik dyka upp, och kanske tolkar de inte heller själva situationen eller problemet likadant som läraren. Det här betyder att man inte alltid på förhand fullt ut kan planera vilka möjligheter till samspel mellan matematik och programmering som kan dyka upp. Det är dock ändå viktigt att vara beredd på att fånga dessa och veta vad man ska göra med dem. Om man på förhand själv noga tänker igenom programmeringsuppgiften och vilken potentiell matematik som man tror kan uppkomma, ökar möjligheterna att hitta och använda dessa möjligheter till samspel. Sådana förberedelser är en förutsättning för en genomtänkt orkestrering av klassrumsarbetet. Nedan presenterar vi några överväganden att göra samt ett exempel på hur möjligheter till matematiklärande kan utvecklas i en situation.

## **Att undervisa för att stödja samspelet mellan matematik och programmering**

Med utgångspunkt i det som är beskrivet ovan och i de orkestreringsformer som beskrivs i Del 2 kan vi nu se på hur undervisningen skulle kunna utformas för att matematik och programmering ska kunna stödja varandra.

- Hitta matematiken i elevernas aktiviteter och tydliggör den, uppmuntra eleverna att uttrycka sig om matematiken. Be eleverna att förklara sina idéer och vad de önskar göra så noggrant att det är möjligt att hjälpa dem att se de matematiska relationerna. Låt eleverna själva förklara de matematiska aspekterna av sina arbeten.
- Utveckla elevernas aktiviteter genom att erbjuda och tillföra matematiska formuleringar eller representationer. Använd ”spontana snabbgenomgångar” (se Del 2) som antingen skapas i stunden eller som planeras i förväg.
- Tänk igenom elevernas programmeringsaktiviteter med tanke på många möjliga relationer till matematik. Tänk igenom på vilka olika sätt eleverna kan förväntas angripa uppgifterna, försök att komma på så många olika sätt som möjligt. Försök att planera vad du ska göra om eleverna griper sig an uppgiften på det ena eller andra sättet.
- Använd orkestreringar som säkerställer att relationerna mellan matematik och programmering blir diskuterade i klassen. Om en elev har gjort något matematiskt intressant så samlar de närmaste eleverna eller hela klassen kring hans/hennes dator och låt eleven förklara, eller förklara själv, den matematiska relationen. Du kan också samlar ett antal elevarbeten för en gemensam genomgång i klassen.

## **Spelutvecklingsaktiviteter som exempel på samspel mellan matematik och programmering.**

### **Exempel 1 – Sänka skepp**

Det första exemplet visar hur matematiska möjligheter i spelutvecklingsaktiviteter kan utvecklas och hur viktigt det är att vara öppen och fånga dessa möjligheter. Exemplet är hämtat från artikeln *Läringsmålstyret undervisning och målförståelser – statiske og dynamiske mål* (Misfeldt & Tamborg 2016) och startar med att en lärare vid ett projektmöte om att arbeta med ”lärandemål” uttrycker osäkerhet för att han har varit tvungen att ändra ”målet” för ett undervisningsförlopp under tiden det pågick. Undervisningen var organiserad undersökande och projektorienterat och eleverna skulle utveckla spel på matematiklektionen. Läraren hade planerat förloppet och hade föreställt sig att eleverna skulle sätta upp regler för sina spel och att bra regler skulle vara ett tecken på att eleverna hade utvecklat sin förmåga att kommunicera matematik. Lärarens mål var alltså att eleverna skulle formulera bra regler. Som tecken på lärande delades målet upp i tre nivåer: 1) att spelet kan spelas med vägledning av en elev från den grupp som hade formulerat reglerna,

2) att spelet kan spelas med en kombination av stöd av en person från gruppen och av nedskrivna regler, samt 3) att spelet kan spelas av en annan grupp med hjälp endast av nedskrivna regler i form av en guide.

Under arbetets gång visade det sig emellertid att elevernas arbete med uppgiften tog en vändning som läraren inte hade förutsett. Eleverna blev mycket engagerade i att utveckla ”matematiska” innovationer i spelet, som en  $x$ -laser som rensar en hel rad, kvadrantbomber som rensar en kvadrant och andra exempel baserade på matematik.

Detta uppfinningsrika sätt att angripa spelet ”Sänka skepp” accepterade läraren, och han valde därför att förändra delar av det ursprungliga lärandemålet för lektionen. Det ändrades till att eleverna genom utveckling och utprövning av ett spel skulle lära sig att orientera sig i ett koordinatsystem. Tecken på lärande ändrades till att spelets regler ger möjligheter till att orientera sig i koordinatsystemet, eleverna själva har anpassat och utvecklat gemensamma idéer till sina egna idéer och eleverna kan genomsöka reglernas betydelse för spelets gång.

Det var denna ändring som läraren uttryckte osäkerhet om det rättmätiga i, trots att arbetet blev en succé och undervisningsupplägget senare framgångsrikt överlämnades till en kollega, som arbetade med det i parallellklassen. Lärarens osäkerhet gällde alltså huruvida han hade fuskat, eftersom han hade ändrat målen efterhand.

## Hur design och matematik möts, och didaktiska grepp i exemplet

Exemplet visar att matematik och spelutveckling kan ha betydelse för programmeringsaktiviteterna. Det gäller även om de spel som eleverna utvecklar inte görs på dator. Matematik kom uppenbart att spela roll för elevernas spelutveckling. Eleverna fick möjlighet att uppleva att den kunskap de hade om koordinatsystemet utvidgade ramarna för vad de kunde göra när det gäller att designa ett ”sänka skepp-spel”. Matematiken gav dem en tydlig struktur att utveckla spelet efter. På detta sätt skapades också motivation för eleverna att lära sig begrepp relaterade till koordinatsystem.

Arbetet med spelregler stödde dessutom deras tänkande i algoritmer och att tänka i termer av olika fall, när olika spelscenarier skulle undersökas och planläggas:

Även de grepp som läraren använder är det värt att uppmärksamma. Exemplet visar nämligen hur matematiska praktiker som utvecklas bland eleverna kan plockas upp av läraren och användas för att förbättra undervisningen, men det visar samtidigt att om man gör så kan det bli nödvändigt att justera planering och mål efterhand. Lektionen kan helt enkelt ta en annan matematisk bana än vad som först var tänkt.

Läraren såg att det var några elever som höll på att tolka uppgiften på ett sätt som var matematiskt meningsfullt och hjälpte eleverna i arbetet. Läraren riktade de andra elevernas uppmärksamhet till arbetet i denna grupp. Därefter spred sig metoden och läraren kunde diskutera koordinatsystemet i detalj med hela klassen samlad.

Konkret använde läraren en orkestrering som påminner om att visa en elevs skärm för resten av klassen och ”diskutera en elevs skärm” för att påverka de andra elevernas lösning av uppgiften. Detta visade sig vara synnerligen effektivt och omvälvande för denna matematiklektion.

Nästa exempel visar hur en lärare skapar en ”snabbgenomgång” i algebra för att eleverna behöver det i sina programmeringsaktiviteter.

## Exempel 2 – Programmera spel

Detta exempel är hämtat från artikeln *Programmering af robotenbeder i grundskolen* (Misfeldt & Ejsing-Duun 2015). I detta projekt var ett viktigt syfte att lärarna i sin undervisning skulle utnyttja elevernas arbeten. Elevernas uppgifter skulle inte bara resultera i ett svar eller att uppgiften blev mer eller mindre korrekt utförd, utan det som eleverna producerade, till exempel ett program, skulle användas som utgångspunkt för den fortsatta undervisningen.

Eleverna i exemplet fick i uppgift att programmera spel. De använder iPad och programmet Hopscotch. Med dessa elever blev det uppenbart att deras idéer om spelet och hur man vanligen interagerar med spel på lärplattor genererade många intressanta diskussioner och skapade inlärningsmöjligheter. Oliver försökte till exempel flytta sin figur genom att luta iPaden. Han bad de andra om hjälp och då frågade Ally honom ”Varför tabbar du inte bara?” Hon föreslog att Oliver skulle arbeta med en enklare typ av kommando för att få en liknande funktion. Oliver svarade: ”Därför att det är ett spel, Ally. Det ska vara roligt liksom”.

Här kan man notera att digitala spel är en naturlig del av elevernas vardag. Det sätt man vanligen interagerar med spel är så att säga lika internaliserat hos barnen som vilken annan del av deras vardagliga omgivning som helst. De förutsätter helt enkelt att sådant som nedräkning, poängsystem, att styra med pilar eller direkt med fingrarna på skärmen eller att luta själva lärplattan är något som ska ingå i ett spel. Det här genererar en speciell typ av motivation hos eleverna när de ska konstruera sina egna spel. I det här faller kunde läraren aktivt bygga på detta – hon insåg att många av elevernas önskemål beträffande spelen kunde fångas upp med begreppet variabler. Trots bristen på kunskap om variabler (ett matematiskt begrepp som anses ligga ”över deras nivå” i skolan) deltog hälften av klassen med stort intresse när läraren visade hur man använder variabler och koordinatsystem för att konstruera en styrfunktion med pilar.

## Möjligheter och didaktiska grepp

Det intressanta i detta exempel är dels hur stark motivation som finns i att designa ett spel – eleverna vill att det ska bli på ett visst sätt och de vill inte kompromissa om det. Detta innebär både en möjlighet att lära och en tänkbar svårighet, då eleverna lätt blir besvikna när de inte kan konstruera det de så gärna vill.

Dessutom visar exemplet att matematik och programmering möts i arbetet och att det är lärarens agerande som skapar detta möte. Läraren observerade först att några elever

arbetade med problem som alla kunde hanteras genom att de införde ett visst matematisk begrepp, nämligen variabler. Detta inkluderar sådan matematik som kan modellera positioner, rörelser och kommandon för att skapa rörelser. Därefter tog hon ett tydligt didaktiskt grepp när hon med utgångspunkt från just dessa elevers arbeten genomförde en mer eller mindre spontan ”snabbgenomgång”. Denna var främst riktad till de elever som hade problem med figurstyrningen, men hölls på ett sådant sätt att alla som var intresserade kunde lyssna. Några elevers produktioner och de frågeställningar som deras arbete hade genererat användes alltså för att skapa ett lärtillfälle för en större grupp elever. Det läraren alltså gör här är att fånga upp vissa fenomen som har med spelkonstruktionen att göra och rikta intresset för dessa fenomen till ett intresse för variabler och koordinatsystem. Sättet på vilket läraren gör detta får också effekten att antalet elever som är uppmärksamma på just detta problem vidgas från några få till nästan hela klassen.

## Referenser

Ejsing-Duun, S., & Misfeldt, M. (2016). Programmering af robotenheder i grundskolen. *Læring og Medier*, 8(14).

Misfeldt, M., & Ejsing-Duun, S. (2015). Learning Mathematics through Programming: An Instrumental Approach to Potentials and Pitfalls. In K. Krainer, & N. Vondrová (Eds.), *CERME9: Proceedings of the Ninth Congress of the European Society for Research in Mathematics Education* (pp. 2524-2530). Prague, Czech Republic: Charles University in Prague, Faculty of Education and ERME.

Misfeldt, M., & Tamborg, A. L. (2016). Læringsmålstyret undervisning og målforståelser - statiske og dynamiske mål. *Cursiv*, (19), 113-139.