

Spelprogrammering, ett sätt att lära sig matematik.

Sofia Bäckström är lärare i matematik och programmering vid Angeredsgymnasiet i Göteborg

Marie Rudenstam är lärare i matematik och idrott vid Angeredsgymnasiet i Göteborg.

Varför sjunker nivån på svenska elevers matematikkunskaper?

Enligt den senaste PISA undersökningen så sjunker nivån på matematikkunskaperna hos svenska elever (www.skolverket.se). Vi tror att en av orsakerna till detta är att eleverna inte fångas av matematiken på grund av att de inte förstår vad de ska ha matematikkunskaperna till. De upplever inte att matematik är något de behöver utan något man endast använder för att lösa uppgifter i matteboken. Känner eleverna inget behov av kunskaperna är risken stor att matematikundervisningen upplevs som meningslös vilket medför att intresset och kunskapsnivån sjunker.

Vad kan vi göra för att sätta fart på den matematiska utvecklingen?

En av de starkaste drivkrafterna bakom matematikens utveckling är behovet att lösa olika typer av problem. På samma sätt tror vi att för att det ska bli fart på den matematiska utvecklingen hos eleverna krävs att de känner ett behov av matematiken för att lösa problem som är meningsfulla och intressanta för dem.

Hur kan vi göra så att eleverna känner ett behov av matematiken?

Knyta an till vardagen.

En metod som används är att utforma uppgifterna i böckerna så att eleverna ska få idéer om hur de kan använda begreppen för att lösa problem i praktiken. Men då innehållet i uppgifterna oftast är irrelevant för eleverna och mer fungerar som utgångspunkt för färdighetsträning (se t ex Jan Nilsson 2001) räcker inte detta för att eleverna ska uppleva att de har ett behov av matematiken.

Konkretisera begreppen

Övningar för att hjälpa eleverna att konkretisera begreppen är en förutsättning för förståelsen och därmed intresset av matematiken. Men vår uppfattning är att flörtkulor, pärlor och pappersfigurer inte räcker för att få eleverna att känna att de behöver begreppen.

Utmanande problem

Själva processen att arbeta med utmanande matematiska problem, t ex hitta mönster och samband, och tillfredställelsen att lösa dessa kan vara en drivkraft till att utvecklas matematiskt (se t ex Emanuelsson m fl, 1996). Vi tror att detta är en viktig faktor men när det gäller att hitta utmaningar som fångar dagens elever har vi väldigt mycket att konkurrera med.

Hur får man eleverna att känna ett behov av en andragradsekvationen?

Vi har förslag på en aktivitet där eleverna upplever att de gör något kreativt och meningsfullt med sina matematikkunskaper samtidigt som dessa tränas och utvecklas. Dessutom ger den upphov till nya situationer där eleverna behöver erövra nya kunskaper i matematik för att lösa de uppkomna problemen. Vidare är aktiviteten kopplad till det som idag är flera ungdomars största intresse, dataspel.

Virtuella världar modelleras med matematik.

Dataspel är program som simulerar en virtuell värld på datorskärmen. Hela denna virtuella värld är uppbyggd av matematik. En av matematikens viktigaste uppgifter är att beskriva hur saker i den verkliga världen uppför sig och detta använder programmeraren för att bygga upp dataspelens virtuella världar. Matematiken är alltså ett nödvändigt verktyg för spelprogrammeraren och ju mer matematiska metoder han behärskar desto mer avancerade spel. Spelprogrammeraren tillämpar sina matematikkunskaper på ett kreativt och produktivt sätt och vi är utanför matematikboken.

Låt oss se på ett exempel.

Saker i spelet rör sig och en andragradskurva kan användas för att modellera saker som kastas/skjuts upp i spelet. Genom att experimentera och variera koefficienterna får man kurvan att uppföra sig som man vill. Begreppen, i det här fallet andragradskurvan, konkretiseras i samband med att de används för att skapa något som eleven har intresse av. Eleven vill att något som skjuts eller kastas ska uppföra sig på ett visst sätt och behöver använda matematiska begrepp för att modellera detta.

I spel är det ofta intressant när saker vänder. Problemet kan lösas genom prövning men det blir mycket enklare om man använder en matematisk metod för att hitta maximipunkten. Eleven har ett behov av metoden och det blir en vinst när man tillägnar sig kunskapen.

Vidare är det intressant var saker landar vilket innebär eleven behöver/efterfrågar en metod för att hitta nollställan.

Pedagogiska vinster med spelprogrammering

I kombination med att spelprogrammering i sig är väldigt underhållande och därför oerhört motiverande och driver kunskapsutvecklingen framåt så finns det många viktiga pedagogiska vinster med programmeringen.

Syntaxen, hur man skriver.

Då kompilatorn endast accepterar uttryck som är korrekt formulerade lär sig eleverna snabbt att formulera sådana. En bonus är att allt de skriver rättas automatisk med felmeddelanden av kompilatorn.

Semantiken, betydelsen av det man skriver.

Elever har ofta ganska diffusa uppfattningar om vad de uttryck de formulerar betyder och det är inte ovanligt att de har en annan tolkning av uttrycken än den matematiskt korrekta. För en kompilator har varje uttryck bara en tolkning och den översätter precis det eleven formulerar. Det finns ingen möjlighet att komplettera uttryck med förklaringar i ord. Kommentarer som "Du förstår ändå vad jag menar" är helt överflödiga till en kompilator. Detta resulterar i att eleven lär sig vikten av att kunna uttrycka sig och kommunicera på ett formellt språk.

Att prata matematik, i den meningen att man resonerar i ord runt ett matematiskt problem är bra. Men att förklara hur ett problem ska lösas, steg för steg, på ett formellt språk med någon som bara förstår detta är en suverän övning. Feedbacken eleverna får i form av att de får se betydelsen av sina uttryck spelas upp i en dataspelsvärld är svår att slå.

Variabler och Algebra

Flera undersökningar (bl.a Kucheman 1981) visar att elever har svårt att förstå bokstavssymboler och att de inte vet hur de ska användas. Vi tror att förklaringen är att eleverna inte förstår varför de behöver räkna med bokstavssymboler då deras gamla metoder fungerar lika bra. Dessa uppfattningar hos eleverna stärks antagligen av det faktum att de introduceras till variabler i

början av mellanstadiet men klarar sig bra med sina gamla metoder ända upp i gymnasiet. Har man för avsikt att programmera måste man använda variabler. När man programmerar får begreppet variabel en konkret förklaring, det är en plats i minnet där datorn kan spara olika saker som till exempel hur antalet liv ändras under en spelomgång.

Funktionsbegreppet

Förståelsen för funktionsbegreppet tränas på ett utmärkt sätt eftersom allt i princip kodas i form av funktioner. Funktionerna konkretiseras i och med att man skriver funktionsdefinitioner och förståelsen utvecklas när man ser hur funktionerna uppför sig på skärmen.

Spelprogrammering kan ha en naturlig plats i matematikundervisningen.

Vi anser att spelprogrammeringen kan ha en naturlig plats i matematikundervisningen som komplement till traditionella metoder. Programmeringen kan kopplas till de matematiska moment som tas upp i kursen så att eleverna får använda och utveckla betydelsen av dessa.

Tittar man närmare på *målen att sträva* mot i gymnasieskolans kursplaner i matematik låter det som om dessa är formulerade för att beskriva spelutvecklingsprocessen.

Att utveckla sin förmåga att tolka en problemsituation och att formulera den med matematiska begrepp och symboler, metoder och uttrycksformer.

Det här är en bra beskrivning av spelprogrammerarens arbetssätt. Han vill att något ska hända i spelet och han måste översätta händelsen och formulera den med hjälp av matematiska begrepp.

Att utveckla sin förmåga att utforma, förfina och använda matematiska modeller samt att kritiskt bedöma modellernas förutsättningar, möjligheter och begränsningar.

Detta beskriver hur utvecklingsprocessen går till. När man kör sitt spel blir modellens förutsättningar och begränsningar väldigt tydliga. När man programmerar reflekterar man hela tiden över sina erfarenheter av begrepp och metoder och man tolkar och värderar lösningarna. Man fixar och justerar koden tills det fungerar och ser ut som man tänkt sig.

Utveckla sin förmåga att tolka förklara och använda matematikens språk, symboler, metoder, begrepp och uttrycksformer.

Programmering är en suverän träning i att använda sig av formella språk med dess symboler. Du måste formulera precis det du vill ska hända med uttryck på det formella språket. Det finns inte utrymme för olika tolkningar och missförstånd, kompilatorn har bara en tolkning.

Vi tror att en bidragande orsak till att man inte programmerar fler spel i skolan är en föreställning om att det krävs avancerad matematik och stora kunskaper i programmering. Men det är delvis fel. Med grundläggande kunskaper i aritmetik, algebra och funktioner går det att programmera ett riktigt bra 2Dspel. De programmerings kunskaper som krävs lär sig eleverna på några månader. För att programmera ett 3Dspel krävs matematiska begrepp som vektorer, matriser, koordinatsystem i tre dimensioner och en mängd operationer på vektorerna som geometriska omvandlingar som skalning, rotation och förflyttningar (se Frank D.Luna 2006) .

Exempel på matematik i ett 2D spel.

Två grundläggande momenten i alla spel är att först måla upp en värld på skärmen och sedan få något att röra sig i den världen. Många 2D spel är rutbaserade vilket innebär att bakgrunden är indelad i rutor. Rutsystemet är bra när man ska hålla reda på placeringar händelser mm. Redan uppgiften att placera ut rutorna på skärmen innehåller flera roliga matematiska problem.

64 rutor ska placeras på skärmen i ett koordinatsystem. Rutorna bildar en kvadrat med sidan 8 och man behöver därför instruera datorn att lägga rutorna i åtta rader om åtta i varje.

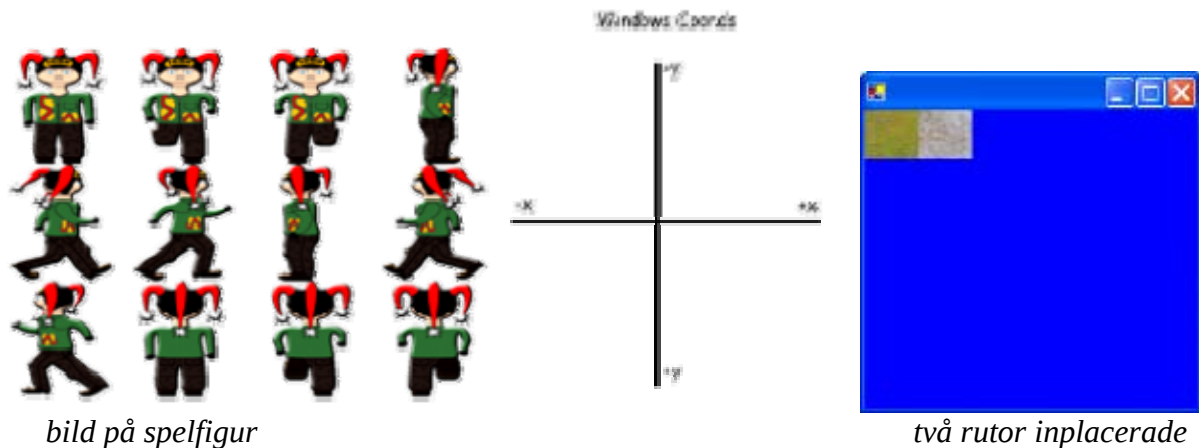
För att tala om för datorn att den ska börja på ny rad när den har lagt 8 rutor behöver man

använda division med rest. När rutorna har nummer som är jämnt delbara med åtta är det dags att börja på ny rad osv.

För att få en rörelse behöver man animera vilket kräver räkning i bråk och procent. Vid animering av en figur klipper man ut olika delar av bilder som innehåller olika positioner (se nedan). De urklippta bilderna visas sedan alternerande på skärmen vilket ger illusionen av en rörelse. Man måste ange vilken del som ska klippas ut och var på skärmen den ska visas.

För att få bilden att röra sig åt höger behövs en funktion där x värdet ökar. Storleken på ökningen anger steglängd.

void goright(int x){x=x+1;} kod för funktion som flyttar bild åt höger



Slutsats

Sammanfattningsvis så anser vi att spelprogrammering är ett väldigt spännande komplement till matematikboken bland annat på grund av att:

Spelprogrammering är väldigt intressant för dagens gymnasieelever.

Spelprogrammering ger suveräna möjligheter för eleverna att tillämpa matematiken på ett meningsfullt och kreativt sätt.

Spelprogrammering konkretiserar de matematiska begreppen och ger eleverna en ökad förståelse för dess betydelse.

Spelprogrammering tränar elevernas förmåga att uttrycka sig och kommunicera med hjälp av formella språk och därmed även elevernas förmåga att tänka matematiskt.

Spelprogrammering ger hela tiden upphov till nya situationer som stimulerar eleverna till att lära sig nya matematiska metoder.

Referenser

Emanuelsson, G., Johansson, B., Ryding, R. & Wallby, K. (Red.) (1996). *Matematik – ett kommunikationsämne*. Nämnaren TEMA. Göteborgs universitet. Institutionen för ämnesdidaktik

Frank D. (2006) Introduction to Gameprogramming with DirectX 9.0c: A shader approach. Texas: Wordware Publishing, Inc.

Kucheman, D (1981). Algebra. In K. M. Hart (Ed.), *Children's understanding of mathematics* 11-16 (pp. 102-119). London: Murray

Nilsson, J. (2001). Havsörnsvals. *Nämnaren*, 28(3), 36-39