



AKADEMIN FÖR UTBILDNING OCH EKONOMI
Avdelningen för kultur-, religions- och utbildningsvetenskap

Har man gjort det själv så spelar man det

Lärarattityder kring spelprogrammering i gymnasieskolans
grundläggande programmeringskurser

Mikael Bergström

2012

Examensarbete, Avancerad nivå (yrkesexamen), 30 hp
Datavetenskap med ämnesdidaktisk inriktning
Lärarprogrammet

Handledare: Fredrik Lindstrand
Examinator: Elisabet Björklund

Bergström, Mikael (2012). *Spelprogrammering som pedagogiskt instrument i programmeringsämnet på gymnasiet*. Examensarbete i didaktik. Lärarprogrammet. Akademin för utbildning och ekonomi. Högskolan i Gävle.

Sammanfattning

Elektroniska spel har gått från att vara en hobby för en handfull nördar i en källare till att bli en oerhört stor del av vår kultur. Mobilspel, datorspel och TV-spel spelas av allt fler.

Om ungdomar nu är intresserade av spel, och intresse gör att man blir mer engagerad och lär sig mer, så borde det innebära att den pedagog som lyckas använda sig av elevernas spelintresse i sin undervisning får stor fördel av detta. I det här examensarbetet var mitt mål att undersöka programmeringslärare på gymnasienivås attityder till, användning av och resonemang kring att använda spelprogrammering som pedagogiskt verktyg. Syftet med detta var att skapa kunskap kring huruvida spelprogrammering skulle kunna spela en större roll än vad det gör idag.

Arbetets data kommer från två källor; en enkät och en serie intervjuer. Genom att använda både kvalitativa och kvantitativa metoder skapas förutsättningar för en djupare förståelse. Först genomfördes enkäten. Den gav data som sedan användes för urval av personer att intervjua samt som kunskapsgrund genom vilken intervjuvaren belystes. Enkäten gav en bred överblick över attityder och samband, medan intervjuerna gav djupare förståelse för de resonemang som låg till grund för attityderna, vilka tankar och upplevelser lärarna hade i ämnet.

Innan jag skrev det här arbetet trodde jag att attityden, framför allt hos äldre programmeringslärare, skulle vara negativ. Jag föreställde mig att de som använde spelprogrammering i sin undervisning framför allt var yngre lärare med eget spelintresse och gedigna spelvanor, samt att de i lägre grad än äldre lärare såg programmeringsämnet som besläktat främst med matematiken.

De resultat som framkommit visar att det egentligen saknas entydiga samband eller heterogena grupper. Det enda som löper som en röd tråd är att programmeringslärare upplever sig sakna fortbildning och gärna skulle uppdatera sina kunskaper inom till exempel spelprogrammering. Att många lärare inte använder spelprogrammering handlar alltså mer om att man saknar de erforderliga kunskaperna än om någon starkt negativ attityd mot spelprogrammering som arbetsmaterial.

Nyckelord: Elektroniska spel, programmeringsdidaktik, spelprogrammering

Innehåll

Kapitel 1: Bakgrund	1
1.1 Inledning	1
1.2 Syfte och problem	2
Kapitel 2: Teoretiskt perspektiv	3
2.1 Centrala begrepp	3
2.1.1 Spel	3
2.1.2 Grundläggande programmeringskurs	3
2.1.3 Programmeringstermer och språk	4
2.2 Tidigare forskning	5
2.2.1 Intresseforskning	5
2.2.2 Programmeringsdidaktisk forskning	5
2.2.3 Spelforskning	6
Kapitel 3: Metod	8
3.1 Urval	8
3.2 Upplägg och genomförande	8
3.2.1 Enkäten och följebrevet	9
3.2.2 Intervjuerna	10
3.3 Materialbearbetning	10
3.4 Tillförlitlighetsfrågor	10
3.5 Etik	11
Kapitel 4: Resultat	12
4.1 Enkäten	12
4.1.1 Användning av spel i undervisningen (enkätfråga 9)	12
4.1.2 Antal år i yrket (enkätfråga 2)	12
4.1.3 Hobbyprogrammering (enkätfråga 12)	14
4.1.4 Spelintresse och speltimmar (enkätfråga 10,11)	16
4.1.5 Likhet med andra ämnen (enkätfråga 4,5,6,7,8)	19
4.1.6 Interesse vad gäller fortbildning i spelprogrammering (fråga 13)	22
4.2 Intervjuerna	23
4.2.1 Intervju A	23
4.2.2 Intervju B	24
4.2.3 Intervju C	25
4.2.4 Intervju D	26
4.2.5 Intervju E	27
Kapitel 5: Diskussion	29
5.1 Metoddiskussion	29
5.2 Spel som pedagogiskt verktyg	29
5.3 Attityder om spel	29
5.4 Förväntade samband och verklighet	30
5.5 Vikten av fortbildning	30
5.6 Sammanfattning och konsekvenser	31
Referenser	32
Bilaga A: Mailet	34
Bilaga B: Följebrevet	35
Bilaga C: Enkäten	36
Bilaga D: Intervjuguiden	39

Kapitel 1: Bakgrund

1.1 Inledning

Programmeringsämnet är ganska ungt, i jämförelse med till exempel matematik. Det har forskats en hel del kring programmeringsdidaktik tidigare, men eftersom datorutvecklingen går så fort som den gör hinner verkligheten inte sällan förändras innan analysen är klar. Varje ny förändring i det rådande programmeringsparadigmet, varje innovation bland yrkesprogrammerare har sedan också debatterats som didaktisk praktik.

En av de saker som förändrats de senaste decennierna är hur vanliga privatägda datorer blivit. 1994 hade ungefär 25 % tillgång till dator i hemmet (Statistiska Centralbyrån, 2011), 2012 är siffran 94 % (Statistiska Centralbyrån, 2012). Mellan 1998 och 2007 ökade tillgången till internet i hemmet från 38 % till omkring 78 % (Statistiska Centralbyrån, 2011). Idag är datoranvändandet bland både ungdomar och vuxna stort, liksom internetanvändandet - man kan rentav påstå att de två delvis blivit samma sak.

För femton år sedan ansågs den som spelade dator- eller tv-spel vara lite speciell, en del av en subkultur "spelnördar". Idag spelar nästan alla, framför allt ungdomar. Spel är inte längre en smal hobby, utan något genomgående som förenar många olika subkulturer.

Det här innebär att förutsättningarna för lärare i datorämnena förändrats drastiskt på förhållandevis kort tid. Ungdomar använder datorer i stor utsträckning, och de allra flesta spelar någon form av elektroniskt spel några gånger i veckan. Elektroniska spel har blivit populära, och det finns ett stort intresse bland ungdomarna för denna form av underhållning.

Detta väcker många frågor: hur tar programmeringslärare tillvara detta intresse hos eleverna? Kan det göras i större utsträckning? Finns det en vinst att utnyttja elevernas spelintresse? Vilka möjligheter finns, och vilka hinder? Min förhoppning är att genom detta examensarbete kunna bidra till detta förhållandevis unga kunskapsområde.

1.2 Syfte och problem

Syftet med det här examensarbetet är att undersöka om spelprogrammering skulle kunna spela en större roll inom programmeringsdidaktiken än vad det gör, och vad som i så fall hindrar att så sker idag.

För att kunna nå det syftet behöver följande frågor besvaras:

- Hur använder programmeringslärare spelprogrammering som pedagogiskt verktyg?
- Kan man skönja någon form av mönster i programmeringslärares egna erfarenheter och intresse av spel kontra deras attityder till och användning av spelprogrammering i klassrummet?
- Vilka faktorer upplever programmeringslärare bidrar till graden av spelprogrammering de använder i klassrummet?

Mina antaganden är:

- Att använda spelprogrammering som pedagogiskt instrument är effektivt och skapar intresse och engagemang.
- Det är förhållandevis ovanligt att spelprogrammering används som pedagogiskt instrument i programmeringskurser.
- De lärare som i högre grad använder spelprogrammering som pedagogiskt instrument gör det för att de själva har intresse i spel.
- De lärare som i lägre grad använder spelprogrammering som pedagogiskt instrument är framför allt äldre, saknar eget spelintresse och betraktar programmering främst som ett matematiskt, naturvetenskapligt ämne snarare än ett humanistiskt eller estetiskt ämne.

Det programmeringslärare framför allt saknar för att i högre grad använda spelprogrammering i sin undervisning är ett bredare utbud av och större utrymme för programmeringsdidaktisk fortbildning, med fokus på spelprogrammering.

Kapitel 2: Teoretiskt perspektiv

För att kunna sätta det här examensarbetets frågeställningar, metod och resultat i ett sammanhang går jag i det här kapitlet igenom dels grundläggande begrepp och vilka definitioner jag använder i arbetet, och dels relevant tidigare forskning. Tanken är att dessa ska ge en förståelse för det teoretiska utrymme examensarbetet arbetar i.

2.1 Centrala begrepp

Begreppen "spel" och "grundläggande programmeringskurs" är väldigt centrala för examensarbetets frågeställningar. Därför redovisas här de resonemang och källor som legat till grund för hur de används i examensarbetets kontext.

2.1.1 Spel

Det är inte alldeles enkelt att definiera elektroniska spel, eller spel över huvud taget; hur skiljer sig till exempel ett spel från en sport, eller en lek? Raph Koster, en av ludologins (spelveenskapens) pionjärer, säger i ett blogginlägg (Koster, 2012) att hans definition är att "playing a game is the act of solving statistically varied challenge situations presented by an opponent who may or may not be algorithmic within a framework that is a defined systemic model" (att spela ett spel är att lösa statistiskt varierande utmanande situationer som skapas av en motståndare, algoritmiskt styrd eller ej, inom ett ramverk som är en definierad systemisk modell).

Denna definition är naturligtvis mycket vid och kan tänkas inkludera en mängd aktiviteter av olika slag. Koster påpekar dock att det finns en poäng med detta; han eftersöker en definition som kan hjälpa honom att förklara spel som helhet och vad det är som gör att spel är roliga.

Matthew Gallant skapar på sin blogg en annan, mer specifik definition:

Software which displays images on a video screen, interacts with a player or players and is intended to provide challenge and/or produce an aesthetic response in the viewer. (2009)

Alltså, att elektroniska spel är mjukvara som visar bilder på en skärm, interagerar med en eller flera spelare och är tänkt att erbjuda utmaning och/eller skapa estetisk respons i betraktaren.

Jag har valt att kombinera dessa för att på så sätt få en definition som är bättre anpassad för klassrumsmiljön och det didaktiska sammanhanget. I den följande texten definieras således ett elektroniskt spel på följande sätt: mjukvara som...

- ... interagerar med en eller flera spelare.
- ... är designad att ha någon form av utmaning.
- ... innebär möjligheten för spelaren eller spelarna att lära sig någon form av mönster.

"Mönster" är här tänkt att tolkas väldigt löst; alla spel där någon form av upprepning sker har mönster spelare kan öva på och bli bättre på, oavsett om det handlar om att hitta tekniker för att klara sig så länge man kan i "Snake" eller att lyckas hoppa lagom långt mellan plattformarna i "Super Mario Bros".

2.1.2 Grundläggande programmeringskurs

I det här examensarbetet används begreppet "grundläggande programmeringskurs" återkommande. Eftersom programmeringskurserna förändrats i och med Gyll så kommer detta begrepp att behöva problematiseras något.

Den kursplan som använts för programmering A och B inom ramarna för läroplanen Lpf94 inrättades år 2000. Fram till vårterminen 2013 går fortfarande elever kvar i gymnasiet som påbörjat sin utbildning med denna kursplan. Med Gy11 skapades nya, mer uppdaterade kursplaner.

I de gamla kursplanerna fanns Programmering A respektive B (Skolverket, 2000a), som båda låg på en termin, sammanlagt 100 poäng. I Gy11 finns istället kursen Programmering 1 (Skolverket, 2012), som ensam ligger på 100 poäng och innehåller allt det de gamla kurserna gjorde, samt utökade kunskaper som motsvarar utvecklingen som skett utanför skolans väggar.

I Lpf94 ingick Programmering A i det Naturvetenskapliga gymnasieprogrammet med inriktning matematik och datavetenskap. Den fanns också med som valbar kurs inom Teknik- och Elprogrammen. B-kursen fanns som valbar kurs på alla tre programmen (Skolverket, 2000b). Utöver detta fanns kursen också tillgänglig som individuellt val.

I Gy11 ingår Programmering 1 och 2 i Teknikprogrammets inriktning mot informations- och medieteknik (Skolverket, 2011). Elever i övriga program kan få välja kurserna som individuellt val.

Jag har valt att låta begreppet "grundläggande programmeringskurs" innehålla både Programmering A+B och Programmering 1; de är någorlunda jämbördiga. Jag kommer att försöka precisera innehållet för att lättare kunna svara på mina frågeställningar.

I jämförelser med internationella studier och artiklar kompliceras frågan en smula. Jag har utifrån sammanhanget i artiklarna försökt bedöma vilken kunskapsnivå det rör sig om; till exempel tycks den amerikanska CS1, eller Computer Science 1 åtminstone delvis motsvara de svenska grundläggande programmeringskurserna.

2.1.3 Programmeringstermer och språk

Genom arbetet nämns några olika programmeringsspråk. De beskrivs här översiktligt.

2.1.3.1 Java

Ett programmeringsspråk som utvecklades under tidigt 90-tal (Lewis & Loftus, 2009, s. 55). Det är mycket populärt och används inom många områden. Språket har en strikt grammatik och program som skrivs i Java kan sedan köras i de flesta operativsystem.

2.1.3.2 Javascript

Ett programmeringsspråk som framför allt används för att skapa dynamiska hemsidor. Trots namnet är Javascript inte släkt med Java (Thau, 2000).

2.1.3.3 Pascal

Ett äldre programmeringsspråk som först släpptes 1970. I början av 80-talet användes Pascal ofta i programmeringsundervisning (Hill, 1981).

2.1.3.4 Python

Ett programmeringsspråk som skapades kring 1990 (Lindblad, 2009). Språket har en enkel grammatik och de program som skapas i Python kan köras i många operativsystem.

2.1.3.5 Flash

Ett program för grafisk multimedieproduktion. I Flash används grafiska verktyg för att åstadkomma enkla animeringar medan programmeringsspråket Actionscript används för mer komplexa beteenden. Spel och filmer som görs i Flash har generellt webben som målplattform (Chun, 2009).

2.1.3.6 C#

Ett programmeringsspråk utvecklat av Microsoft, som bland annat kan användas tillsammans med biblioteket XNA för att utveckla spel till spelkonsollen X-Box 360 (Dolan, 2006).

2.2 Tidigare forskning

I det här arbetet utgår jag från att:

- a) De flesta elever är intresserade av spel.
- b) Elever som arbetar med arbetsuppgifter de finner meningsfulla och som anknyter till något de är intresserade av presterar bättre och lär sig mer aktivt.
- c) Elever som får arbeta med uppgifter där de får skapa spel därmed borde prestera bättre.

För att förklara och försöka leda detta i bevis beskriver jag i det här delkapitlet dels forskningsläget kring intresse och lärande, dels vilken forskning som gjorts kring programmeringsdidaktik i allmänhet, och dels vilken forskning som gjorts kring elevers spelintresse.

När jag sökt efter artiklar och böcker har jag framför allt använt mig av tre sökmotorer: Svenska Libris, databasen Discovery via Högskolan i Gävle, och Google. Google är den mest problematiska av de tre; dock helt nödvändig för att hitta information om ludologin eftersom så många av de som idag arbetar inom området inte är publicerade i akademisk mening. Ett urval har gjorts för att undvika trovärdighetsproblem vad gäller de källor som hittats via Google.

2.2.1 Intresseforskning

Ett grundläggande antagande i sammanhanget är att elevers resultat och föreståelse för ett ämnes stoff, struktur och teoretiska innehåll ökar om stoffet och det teoretiska innehållet kopplas till områden de är intresserade av. Att intresse leder till bättre lärande torde vara förhållandevis trivialt att bevisa, och det finns också en hel del forskning kring elevers intresse och lärande.

I en artikel sammanfattar Hidi (1990) forskningen från många olika källor, och hon delar upp intresse i "individuellt intresse", alltså sådant eleven själv bidrar med, och "textbaserat intresse", som då är intresse som väcks genom att eleven läser en viss passage. Båda dessa typer av intresse, konstaterar hon, "have a profound effect on cognitive functioning and the facilitation of learning" (har en djupgående effekt på kognitiva funktioner och på underlättandet av inläring) (s.565). Hon visar också att det är stor skillnad på de processer som sker när man går igenom intressant information kontra information som inte är intressant.

2.2.2 Programmeringsdidaktisk forskning

Till övervägande del tycks forskningen kring pedagogik och didaktik i programmerande klassrum vara fokuserad antingen på specifika områden i det teoretiska stoffet som upplevs som problematiska för eleverna eller på effekten av olika paradigmer inom programmeringen på den pedagogiska praktiken. Ett genomgående tema är problemet att många elever inte väljer att fortsätta studera programmering efter att ha genomgått en grundkurs.

Robins, Rountree och Rountree (2003) skriver till exempel om ett antal områden där problem uppstår; de diskuterar kring "ineffektiva" och "effektiva" nybörjare, där skillnaden är att de senare arbetar aktivt för att nå en djupare inläring, och de visar också några av de centrala problem nybörjare behöver lösa. En viktig kärnkompetens är, enligt författarna, att inte bara förstå syntax utan att kunna forma

strategier, att till exempel inte bara veta hur en for-loop ser ut, utan vilka sorters problem som kan lösas med hjälp av den.

2.2.3 Spelforskning

Spelforskningen är ett förhållandevis ungt ämne, men det är trivialt att visa att väldigt många ungdomar spelar spel. Enligt Nordicoms internetbarometer 2010 spelar c:a 50% av ungdomar som är mellan 15 och 24 år internet- konsol- eller datorspel en genomsnittlig dag/vecka (Nordicom 2011). I en undersökning Medierådet gjorde 2010 bland yngre barn och ungdomar angav 67 % av de tillfrågade barnen mellan 12 och 16 år att de brukade spela elektroniska spel. År 2009 genomfördes en enkätundersökning (Nilsson 2009) bland eleverna på ett gymnasiums IT-program som bland annat visade att 94 % av de som svarade på enkäten spelade elektroniska spel mer än två timmar i veckan.

Frågan är om detta intresse för att spela spel också innebär att eleverna är intresserade av att skapa spel. Svenska studier eller rapporter kring detta tycks vara svårfunna, och det gäller intressant nog också internationellt. Många har forskat på användningen av spel i undervisningen, framför allt då s.k. edutainmentspel, där olika ämnen lärs ut via specialutformade pedagogiska spel, men betydligt färre tycks ha forskat på programmeringen av spel.

I ett amerikanskt försök att väva in speldesign i en universitetskurs i Computer Science testade Rankin, Gooch och Gooch (2008) flera olika speldesignrelaterade övningar. Överlag fick kursen ett bra mottagande av eleverna, även om bara 40 % angav att de lärt sig en "avsevärd mängd" av vad det innebär att designa ett spel. Observera dock att detta mått inte gäller hur mycket programmering de lärt sig. Rankin, Gooch och Gooch menar att kursmomenten antagligen var för "öppna" för en del elever, som behövt mer exakta instruktioner.

Även Becker (2001) har använt spelprogrammering i sina grundläggande programmeringskurser på universitetsnivå (CS1+CS2). Hennes elever har fått programmera en textbaserad kopia av spelet "minesweeper", som på svenska kanske är mest känt som "MS Röj". Hon menar att:

If students can be inspired and excited by the things they learn and the work they do early on, these attitudes will often carry them through the difficult times and the "all-nighters". It has been said that much of what is learned in computer science is learned by doing, particularly when it comes to programming, so if students can be motivated to do more it seems reasonable to conclude that students will end up learning more as well (s. 24).

Hon säger alltså att om eleverna ska kunna ta till sig de svåra delarna av programmeringen så är det viktigt att de inspireras och engageras av det de gör. Hon upplever också en markant skillnad i hur eleverna tog till sig det nya spelfokuserade lärandet; de blev betydligt mer engagerade vilket också förbättrade deras inläring.

Det bör nämnas att detta visserligen var 2001, men de textbaserade spel hennes "minesweeper" liknar snarare tillhör 1980-talet. Som spel betraktat var det med andra ord väldigt omodernt jämfört med samtidens spel, men det fungerade ändå väldigt effektivt ur pedagogisk synpunkt. Spel är, som Becker skriver, icketriviale program som ställer stora krav på utvecklaren. Samtidigt innebär de en konkret slutprodukt eleverna känner igen och kan förstå.

I en powerpointpresentation visar John Nordlinger och Andrew Phelps (2007) några andra projekt som utförts på universitetsnivå. Enligt de resultat de presenterar är spelprogrammering något som gör eleverna aktiva och engagerade, samt att det kan leda till att en högre andel av de som gått den första programmeringskursen (Computer Science 1) väljer att fortsätta till den andra (Computer Science 2).

Leutenegger och Edgington (2007) rapporterar också från försök på universitetsnivå, där de använt flashspelsprogrammering för att lära ut grundläggande programmering. I en utvärdering deras elever gjort framkommer att de känner sig motiverade och att de upplever att de nått en överlag god förståelse för programmeringens grunder. Deras spelfokus har också hjälpt dem att locka nya elever, och kursen har blivit mycket uppskattad och populär.

I dagsläget verkar forskningsläget för den nivå som det här examensarbetet gäller, alltså svensk gymnasienivå, förhållandevis eftersatt. Jag hittar ingen relevant svensk forskning, men inte heller någon amerikansk eller annan forskning om användningen av spelprogrammering som pedagogiskt verktyg på motsvarande utbildningsnivåer.

Kapitel 3: Metod

Eftersom det här arbetet behandlar attityder och erfarenheter hos lärare, och vilka resonemang som förs i den praktiska verkligheten, så har jag valt att ha ett i grunden kvalitativt förhållningssätt. Målet är att kunna gå "på djupet" (Holme & Solvang, 1997, s.78) och nå en förståelse för de resonemang och upplevelser som ligger bakom de didaktiska val programmeringslärare gjort.

För att skapa en grundförståelse för den variation och de tendenser som finns bland programmeringslärare valde jag att kombinera den kvalitativa grunden med en inledande kvantitativ undersökning i form av en enkätundersökning.

Efter enkätundersökningen genomfördes en serie intervjuer med fem av de som svarade på enkäten. På så vis skapas både en "generell översikt och en speciell inblick" (Holme & Solvang, 1997, s.87) i de frågor jag föresatt mig att undersöka.

Enkätundersökningen, som genomfördes elektroniskt bidrog med ett brett urval av erfarenheter och attityder, vilket gav ett diskussionsunderlag trots att resultaten knappast blir statistiskt säkerställda. Intervjuerna bidrog i sin tur med en djupare förståelse för de intervjuades situation och bakgrund, samt en inblick i hur de resonerade kring frågorna.

3.1 Urval

I brist på allmänt tillgänglig central databas över Sveriges samtliga gymnasielärare som undervisar eller har undervisat i programmering har jag istället utnyttjat den så kallade SITSNET-listan. Denna är en stor maillista sammansatt på initiativ av Lennart Rolandsson, och ska enligt uppgift från densamme innehålla cirka 250 programmeringslärare (personlig kommunikation, 7:e september 2012). Det första urvalet skedde alltså genom att undersökningen endast vänt sig till de programmeringslärare som är anslutna till denna lista.

Till listan skickades ett mail innehållandes en kort presentation av projektet (bilaga A) samt en länk till enkäten i digitalt format (bilaga C). I mailet länkades också ett längre och mer utförligt följebrev (bilaga B) Att uteslutande använda elektronisk kommunikation och en elektronisk enkät innebar å ena sidan att endast de potentiella respondenter som a) var med i SITSNET-listan, b) läste sin mail under den period enkäten var ute och c) väljer att delta i elektroniska enkäter svarade. Å andra sidan innebär detta inte nödvändigtvis några problem i en kvalitativ undersökning. Det blir ett betydande bortfall, men det var knappast praktiskt genomförbart att resa runt och besöka samtliga programmeringslärare personligen med en enkät, och svarsfrekvensen hade knappast blivit högre om enkäten istället skickats ut via post.

Av de som svarade på enkäten bjöds sedan fem stycken in till intervju. Urvalet här gjordes baserat på bredd och geografisk spridning. De som tillfrågades har varit lärare olika länge och givit vitt skilda svar på enkätens frågor. På grund av tidsmässiga begränsningar har urvalet också begränsats till Stockholmsområdet. Jag har också undvikit att välja respondenter jag har en relation till sedan tidigare.

3.2 Upplägg och genomförande

I det här delkapitlet förklaras de metoder jag använt för att samla in information, och vilka principer som använts för att konkretisera metoderna till en enkät respektive en intervjuguide.

3.2.1 Enkäten och följbrevet

I arbetet med enkäten har "Att få svar" (Kylén, 2004) och "Enkäten i praktiken" (Ejlertsson, 2005) använts som utgångspunkter när det gäller utformning och formulering av frågor och svarsalternativ. Enkätfrågorna utformades först i enlighet med Ejlertssons kapitel om frågekonstruktion (2005, s.51).

Med andra ord lades vikt vid att använda ett enkelt och funktionellt språk, entydighet i frågeställningarna, att undvika ledande frågor och dubbla negationer, att undvika alltför långa frågor och att vara exakt och avgränsad i frågor som gäller tid. För frågor med svar på en skala användes en femgradig skala med någorlunda likvärdiga alternativ på vardera sidan om en neutral mittposition.

Sedan genomfördes en mindre pilotstudie med två kollegor där de fick läsa och ställa frågor kring frågornas utformning och svarsalternativ. I det läget skedde en del kosmetiska förändringar, och enkätens sista fråga tillkom.

Denna fråga är egentligen inte helt överrensstämmande med goda principer för enkätfrågors utformning, eftersom den är framåtsträvande, hypotetisk och därtill till intet förpliktigande. Det kostar inget för respondenten att ange en hög grad av villighet att genomgå fortbildning i spelprogrammering. En möjlig farhåga är också att respondenter som själva upplever att de "borde" använda mer spelprogrammering använder denna fråga för att kunna väga upp sitt dåliga samvete.

Samtidigt innebär även detta ett användbart svar; lärarna skulle då uppenbarligen vilja använda mer av den sortens uppgifter, men eftersöker fortbildning. Det riskerar alltid att bli känsligt när pedagogers egen praktik undersöks.

Frågorna behandlar följande områden:

- Respondentens bakgrund (antal år som programmeringslärare, fortbildning)
- Respondentens pedagogik (hur stor andel som är spelprogrammering)
- Respondentens egna spelerfarenheter (intresse, hur mycket han eller hon spelar)
- Grad av likhet respondenten upplever mellan programmering och ett antal andra ämnen (matematik, teknik, engelska et cetera)

Det utökade följbrevet som länkades från enkäten utgår i stort sett från Ejlertssons exempelföljbrev (2005, s.40). Liksom exempelföljbrevet beskriver det följande:

- Hur urvalet gått till (att jag skickat till alla SITSNET-deltagare).
- De positiva konsekvenserna av respondentens deltagande.
- Att deltagande givetvis är frivilligt.
- Att svaren behandlas anonymt.
- Hur respondenten gör för att kontakta mig som skickat ut enkäten.

Anonymiteten kommer sig av att det digitala verktyg som använts inte sparar vem som skickat in varje svar. Hypotetiskt innebär detta att respondenterna kunnat svara flera gånger samt att det egentligen inte går att kontrollera att alla svar kommer från deltagare på SITSNET-listan.

Det enda sätt som funnits för att identifiera enskilda respondenter har varit de som genom att skriva sin mailadress i fältet där de anger att de kan tänka sig att bli intervjuade. I följbrevet gavs också möjligheten att anmäla intresse för intervju via ett separat mail till mig, så de som skrev sin mailadress valde själva att ge upp sin anonymitet.

3.2.2 Intervjuerna

Inför intervjuerna skapade jag en intervjuguide i enlighet med trattmodellen (Kylén, 2004, s.31), där strukturen ser ut som följande:

1. Öppning; presentation av intervjuaren, frågor kring anteckningar och inspelning.
2. Fri berättelse; respondenten får berätta fritt kring området.
3. Precisering; respondenten får följdfrågor.
4. Kontroll; respondenten får frågor om sådant som upplevts som oklart.
5. Information; diskussion kring undersökningen, frågor och svar.
6. Avslutning; intervjun avslutas.

Steg 2-4 upprepas för varje frågeområde. Jag valde att utgå från en liknande modell även på makro-nivå; det första området är relativt allmänt och låter respondenten berätta fritt kring sin egen lärarroll och bakgrund. De efterföljande områdena blir gradvis mer precisa.

Eftersom intervjuerna framför allt handlar om lärarnas egna upplevelser och praktik utgår intervjuerna från öppna frågeställningar, med några exempel på följd- och delfrågor i punktform (bilaga D). Intervjuerna spelades in med hjälp av en mobiltelefon och Android-applikationen RecForge Pro. Respondenterna tillfrågades på förhand, via mail, huruvida de godkände inspelning av intervjun. Samtliga godkände inspelning.

Anteckningar i form av stödord fördes också under varje intervju. Inspelningarna laddades direkt efter varje intervju upp till molnplattformen Dropbox för att säkerställa att de skulle finnas kvar även om någon olycka drabbade mobiltelefonen.

De tillfrågades också efter intervjuerna huruvida de ville läsa igenom materialet innan publikation.

3.3 Materialbearbetning

Enkätresultatet bearbetades genom att gruppera svaren på olika sätt och se om till exempel ett lågt spelintresse också indikerade en låg användning av spel i klassrummet. De olika grupperingarna ställdes upp som jämförande stapeldiagram för att synliggöra eventuella mönster. Intervjumaterialet analyserades för att hitta gemensamma teman, återkommande attityder och tendenser. Varje intervju sammanfattades sedan i kortare form tillsammans med viss analys.

3.4 Tillförlitlighetsfrågor

Det grundläggande problemet med attitydundersökningar är naturligtvis att respondenterna ganska snabbt räknar ut ungefär vad den som genomför undersökningen är ute efter, vilket riskerar att snedvrida resultatet. I det här fallet finns naturligtvis risken att respondenterna anpassat sina svar till att bli mer "spelvänliga", eftersom den underliggande tesen om spel som produktiv och positiv pedagogisk kraft inte är särdeles svår att se även utan att det skrivs ut.

Dessutom kan man utgå från att de som har ett intresse för spel och spelprogrammering, och de som använder produktion av elektroniska spel i sin undervisning, har en högre tendens att besvara en enkätundersökning som rör just tillverkning av elektroniska spel i programmeringsundervisningen än de som saknar sådant intresse eller som inte använder spel.

Hade enkäten varit en del av en kvantitativ undersökning finns en uppsjö av problem. Förutom ovanstående finns också omfattande problem med bortfall (41 av ett okänt antal kring 300 svarade). Enkätsvaren bör med andra ord inte tolkas som tillförlitlig statistik, utan som diskussionsunderlag som indikerar attityder och erfarenheter hos specifika lärare.

3.5 Etik

Arbetet uppfyller Vetenskapsrådets rekommenderade forskningsetiska principer (Vetenskapsrådet 2002) genom att följa de fyra huvudpunkterna:

- **Informationskravet.** Respondenterna till både enkäten och intervjuerna informerades om undersökningens syfte och inriktning. De informerades också om vilken roll deras svar skulle komma att spela, och om de villkor som gällt för deras deltagande.
- **Samtyckeskravet.** Enkäten skickades ut i elektronisk form till en maillista, och deltagandet var helt frivilligt. De som i enkäten angav att de var intresserade av att bli intervjuade kontaktades och gavs vidare information om vad en intervju skulle innehålla. De intervjuade hade goda möjligheter att när som helst avbryta sin medverkan, och erbjöds att ta del av arbetet innan det publiceras. På så vis har de haft möjligheten att dra tillbaks sin medverkan in i det sista.
- **Konfidentialitetskravet.** Enkäten genomfördes anonymt. De som anmälde intresse för att bli intervjuade valde i samtliga fall att göra detta genom att skriva sin mailadress i enkäten istället för att skicka ett separat mail, vilket fanns med som alternativ i instruktionerna. I mailkonversationen framkom deras namn, men den har inte lagrats elektroniskt på annat sätt. Intervjuerna spelades in, och ljudfilerna är döpta efter datum och klockslag och kan således inte knytas till någon person. Informationen i ljudfilerna kan dock gissningsvis användas för att knyta intervjun till en person, dock med stora svårigheter.

Under arbetets gång har enkätsvaren samt information om vem som skulle intervjuats när lagrats i min elektroniska kalender (Google Calendar), som är markerad privat. Ljudfilerna har legat på min telefons lagringsyta samt på min molnlagringsyta som jag har via tjänsten Dropbox. Detta för att säkerställa att ingen data skulle förloras utifall något skulle hända med mobiltelefonen. Både googlekontot och dropboxkontot har varit skyddade via s.k. two-step verification. Det betyder att för att utomstående ska kunna komma åt informationen måste vederbörande ha tillgång dels till mina inloggningsuppgifter och dels till de SMS som skickas till min mobiltelefon.

Sammantaget har alltså informationen lagrats på ett säkert sätt; utan att stjäla min utrustning (mobil) går det inte att koppla samman personer med deras svar varken i enkäten eller intervjuerna. Även med informationen man då kan få fram kommer identifiering att vara försvårad, då inga namn användes under intervjuerna och inga namn står med i enkätsvaren.

Därutöver är arbetets ämne sådant att de uppgifter som sparats knappast kan betraktas som känsliga.

Nyttjandekravet. Det material som insamlats kommer inte att få nyttjas till något annat än vetenskapliga syften.

Kapitel 4: Resultat

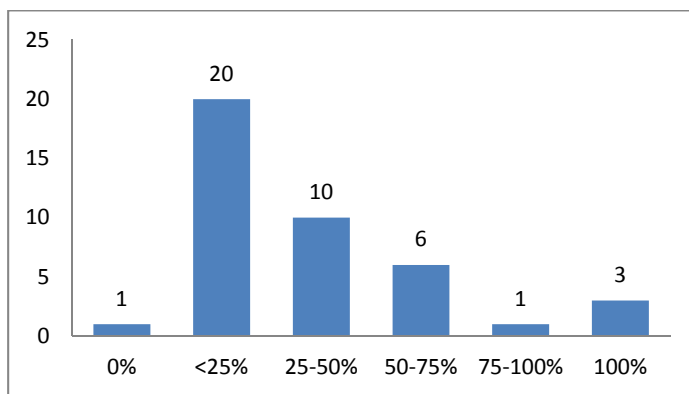
I det här kapitlet presenteras resultatet från examensarbetets två datakällor. Enkätsvaren presenteras i form av siffror och statistik, och intervjuerna i berättande och sammanfattad form.

4.1 Enkäten

Av de c:a 250 personer som är med i SITSNET-maillistan svarade sammanlagt 41 personer på enkäten. Det enda interna bortfallet var en respondent som inte svarade på hur många timmar vederbörande spenderade med att programmera för eget bruk per vecka.

4.1.1 Användning av spel i undervisningen (enkätfråga 9)

Av 41 respondenter anger 31 att 50 % eller mindre av deras uppgifter handlar om spelprogrammering, när de undervisar i grundläggande programmeringskurser. Den överlägset största gruppen, 20 respondenter, anger att de använder spelprogrammering i mindre än 25 % av sina uppgifter. Tre respondenter använder uteslutande spel i samtliga uppgifter, och endast en enda använder aldrig spel.

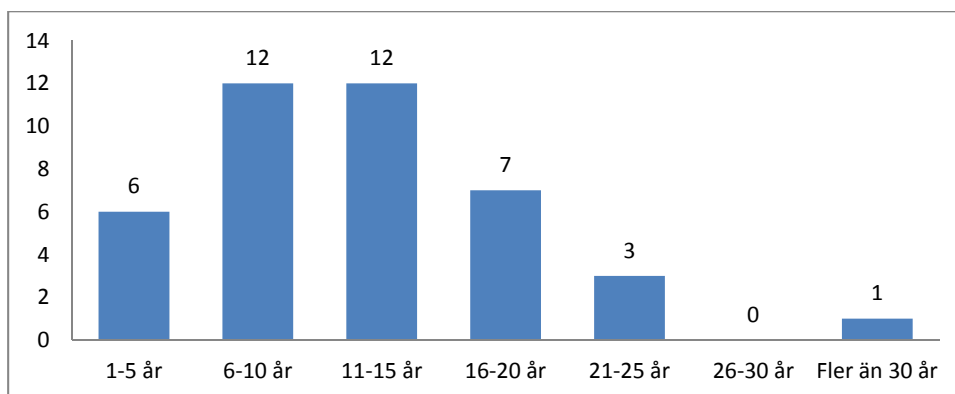


Figur 4.1 Antal lärare som använder olika antal procent spelprogrammeringsuppgifter.

Svaren ligger i linje med tesen om att spelprogrammering i hög grad inte används. De allra flesta av respondenterna använder väldigt få sådana uppgifter. Däremot finns spel med som element för i stort sett samtliga.

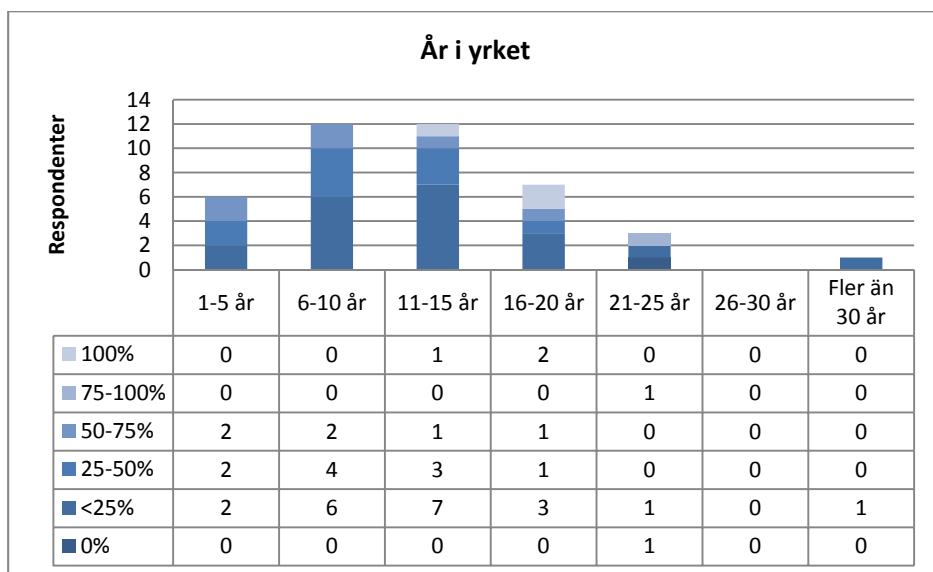
4.1.2 Antal år i yrket (enkätfråga 2)

De flesta av mina respondenter, totalt 24 stycken, har arbetat som programmeringslärare i 6-15 år. Sex stycken är förhållandevis nya i yrket med 1-5 års erfarenhet och sju har längre erfarenhet, 16-20 år. Sammanlagt fyra respondenter har arbetat i 21 eller fler år.

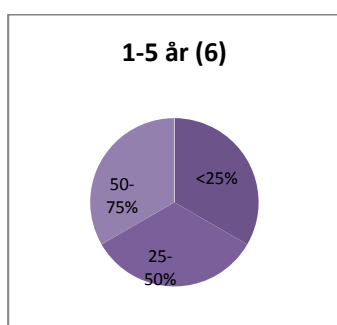


Figur 4.2 Antal år de svarande arbetat som programmeringslärare.

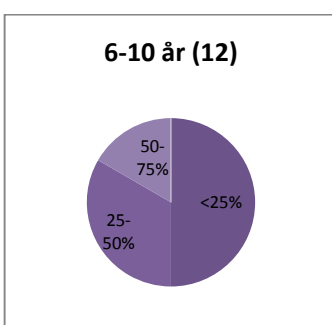
När respondenternas svar på enkätfråga 9 grupperas efter antal år i yrket blir resultatet som i figurerna 4.3-4.8.



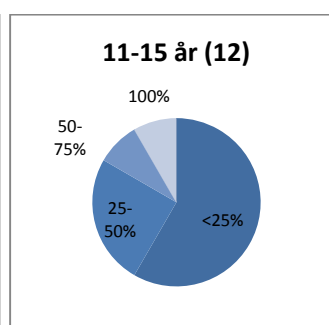
Figur 4.3 Antal år som programmeringslärare och andel spelprogrammeringsuppgifter i undervisningen i procent.



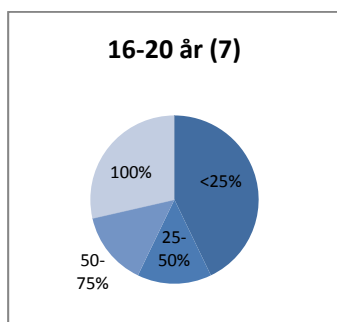
Figur 4.4 Andel spelprogrammeringsuppgifter hos svarande som arbetat 1-5 år.



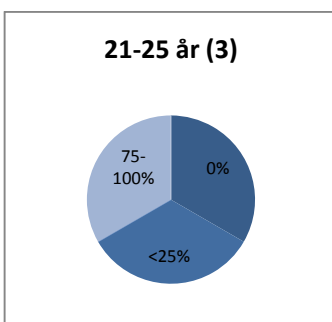
Figur 4.5 Andel spelprogrammeringsuppgifter hos svarande som arbetat 6-10 år.



Figur 4.6 Andel spelprogrammeringsuppgifter hos svarande som arbetat 11-15 år.



Figur 4.7 Andel spelprogrammeringsuppgifter hos svarande som arbetat 16-20.



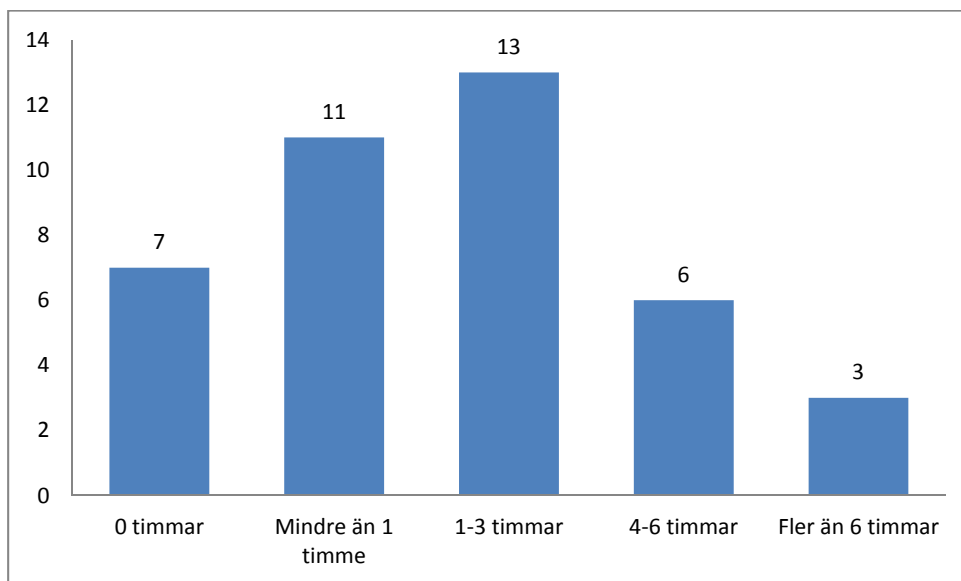
Figur 4.8 Andel spelprogrammeringsuppgifter hos svarande som arbetat 21-25 år.

Detta innebär alltså att de respondenter som hade mindre än 25 % spelprogrammeringsuppgifter är utspridda över alla grupperna, samt att det fåtal som använde en hög grad av sådana uppgifter (75 % eller mer) framför allt uppträdde bland de som varit programmeringslärare i 11 år eller mer, även om gruppen som använde 50-75 % även finns bland de med färre år inom yrket.

Resultat är intressant, och går delvis på tvärs med tesen att äldre programmeringslärare skulle vara mindre benägna att använda spelutveckling som pedagogiskt verktyg. Det bör dock nämnas att de som är nya i yrket inte alls avhåller sig; en god portion av de som arbetat 1-10 år tillhör också de som använder mer än 50 % den sortens uppgifter.

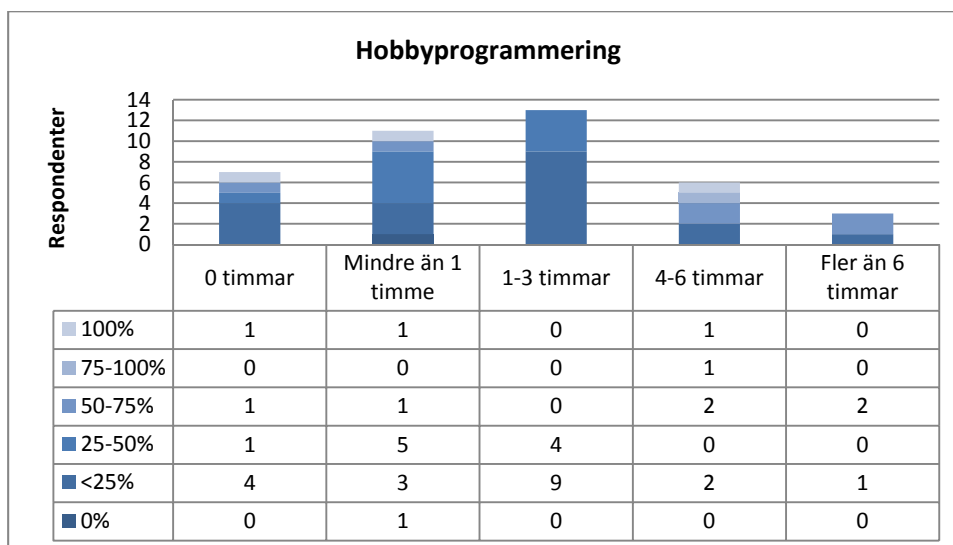
4.1.3 Hobbyprogrammering (enkätfråga 12)

Av 41 respondenter är det bara sju som inte alls programmerar för eget bruk en vanlig vecka.

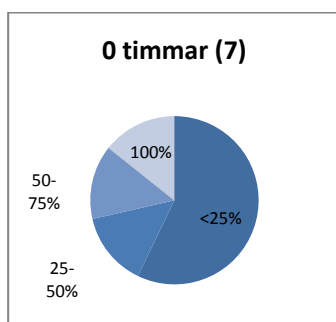


Figur 4.9 Antal timmar de svarande spenderar med att programmera för eget bruk en vanlig vecka.

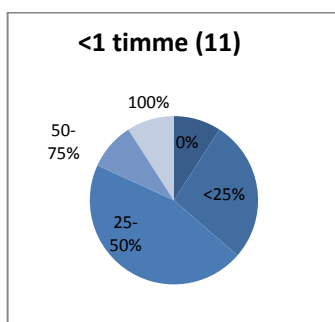
När respondenternas svar på enkätfråga 9 grupperas efter mängd hobbyprogrammering blir resultatet som i figurerna 4.10-4.15.



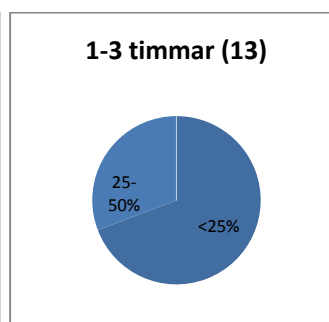
Figur 4.10 Antal timmar spenderade med hobbyprogrammering och andel spelprogrammeringsuppgifter i undervisningen.



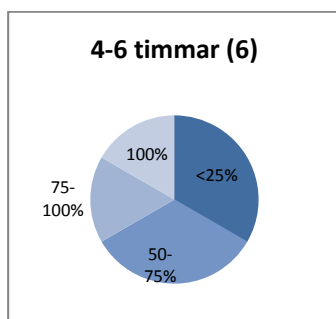
Figur 4.11 Andel spelprogrammeringsuppgifter hos svarande som inte spenderar någon tid per vecka.



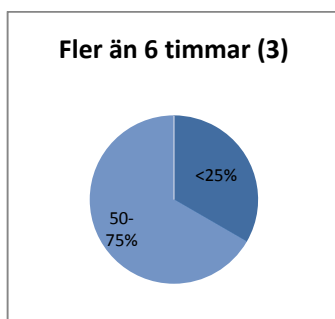
Figur 4.12 Andel spelprogrammeringsuppgifter hos svarande som spenderar mindre än en per vecka.



Figur 4.13 Andel spelprogrammeringsuppgifter hos svarande som spenderar 1-3 timmar per vecka.



Figur 4.14 Andel spelprogrammeringsuppgifter hos svarande som spenderar 4-6 timmar per vecka.



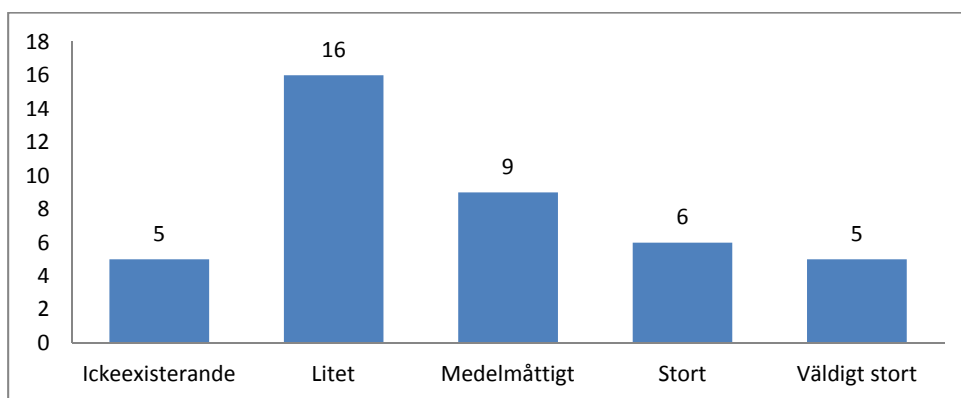
Figur 4.15 Andel spelprogrammeringsuppgifter hos svarande som spenderar fler än 6 timmar per vecka.

Andelen som använder mindre än 25 % spelprogrammeringsuppgifter är med andra ord något större bland de respondenter som inte alls hobbyprogrammerar samt bland de som spenderar 1-3 timmar på hobbyprogrammering än övriga. De som använder 75 eller fler procent spelprogrammeringsuppgifter återfinns över i stort sett hela spektrumet.

4.1.4 Spelintresse och speltimmar (enkätfråga 10,11)

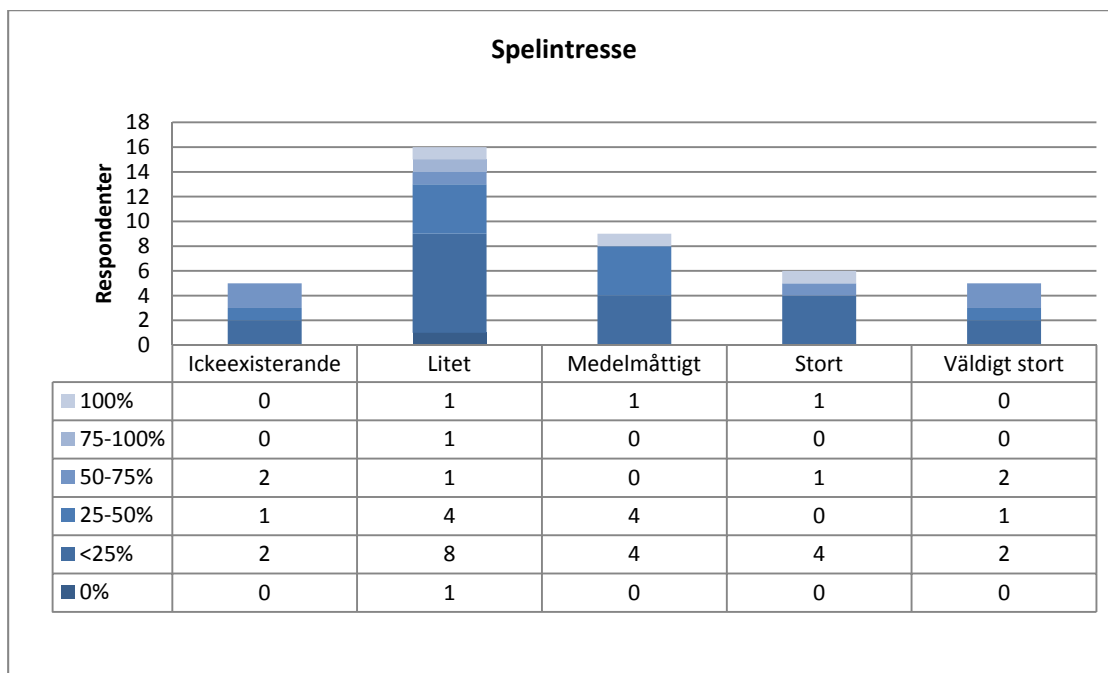
Respondenternas bedömning av sitt eget spelintresse har en viss övervikt mot det lägre hållet. 21 av 41 anser att de har ett litet eller icke existerande spelintresse medan bara 11 anger att de har ett stort eller väldigt stort.

En av respondenterna angav svaret "Stort intresse för att tillverka spel, mindre för att spela". Detta svar har räknats in i gruppen "stort" eftersom frågan gällde spelintresset i stort.

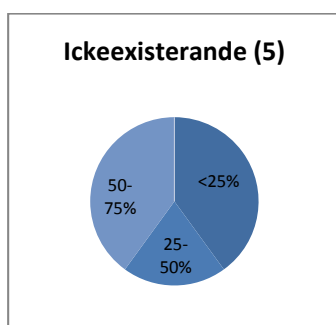


Figur 4.16 Det svarandes beskrivning av det egna spelintresset.

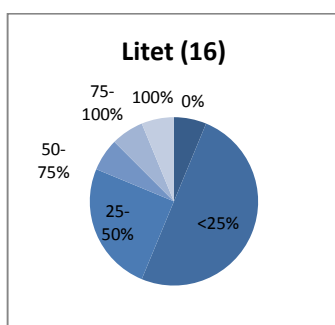
När respondenternas svar på enkätfråga 9 grupperas efter deras uppgivna spelintresse ser det ut som i figur 4.17-4.22.



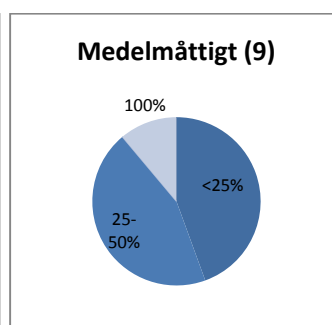
Figur 4.17 De svarandes beskrivning av sitt eget spelintresse och andelen spelprogrammeringsuppgifter i undervisningen.



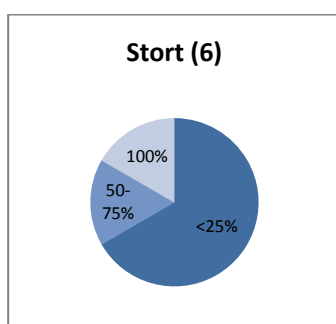
Figur 4.18 Andel spelprogrammeringsuppgifter hos svarande som beskriver sitt spelintresse som ickeexisterande.



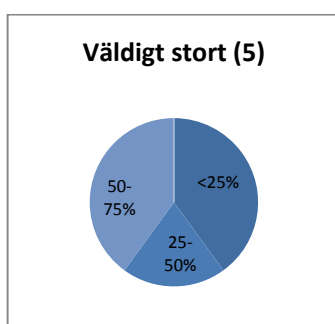
Figur 4.19 Andel spelprogrammeringsuppgifter hos svarande som beskriver sitt spelintresse som litet.



Figur 4.20 Andel spelprogrammeringsuppgifter hos svarande som beskriver sitt spelintresse som medelmåttigt.

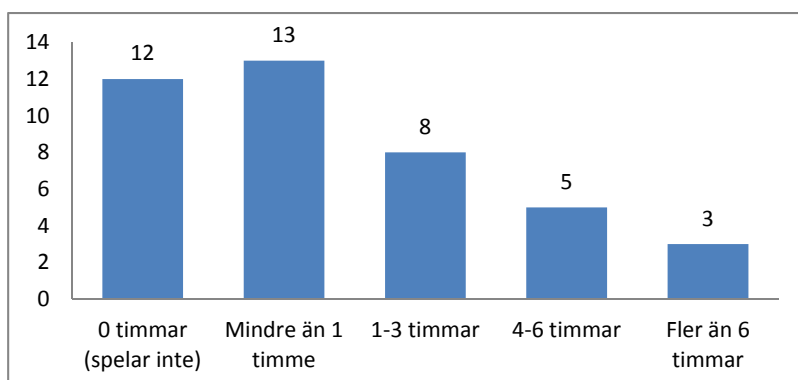


Figur 4.21 Andel spelprogrammeringsuppgifter hos svarande som beskriver sitt spelintresse som stort.



Figur 4.22 Andel spelprogrammeringsuppgifter hos svarande som beskriver sitt spelintresse som väldigt stort.

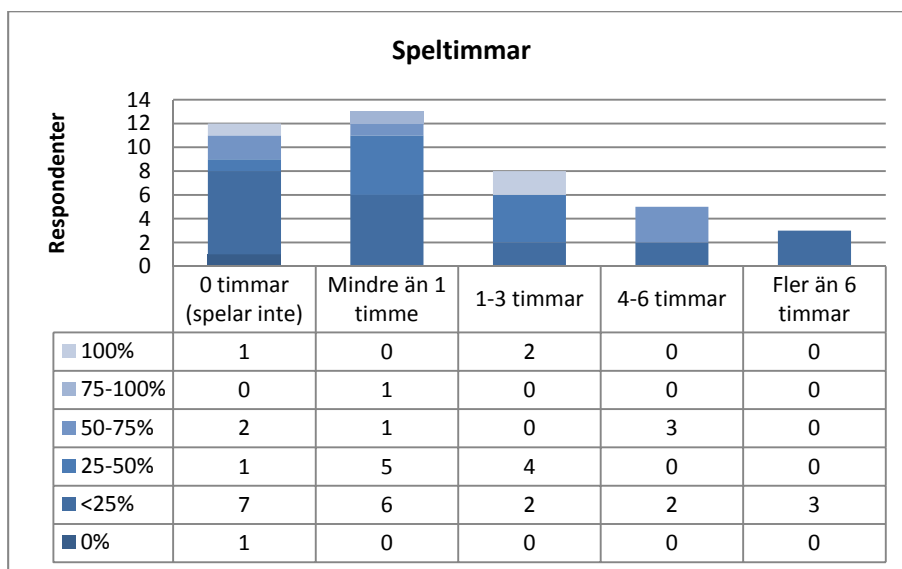
När det gäller hur mycket de svarande spelar på fritiden ser svaren ut som följer:



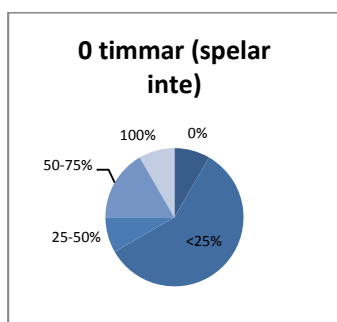
Figur 4.23 Antal timmar de svarande spenderar med att spela elektroniska spel på fritiden

12 av 41 spelar alltså inte alls, och sedan kan man konstatera en fallande kurva där de flesta spelar ganska lite (1-3 timmar/vecka) och bara tre spelar fler än 6 timmar per vecka.

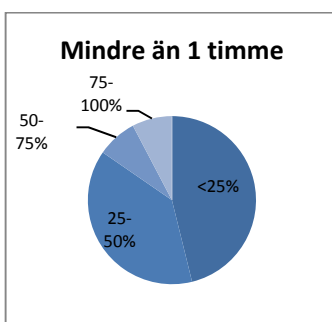
Det som framkommer om respondenternas svar på enkätfråga 9 grupperas efter antal timmar de spelar elektroniska spel per vecka återfinns i figurerna 4.24-4.29.



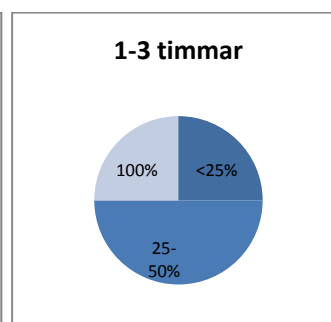
Figur 4.24 Antal timmar de svarande spenderar med att spela elektroniska spel och andelen spelprogrammeringsuppgifter.



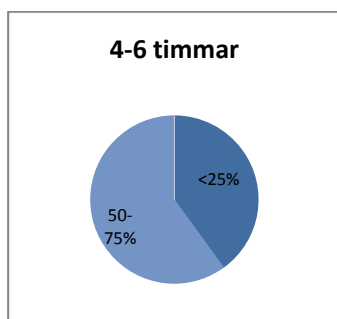
Figur 4.25 Andel spelprogrammeringsuppgifter hos de svarande som inte spelar elektroniska spel.



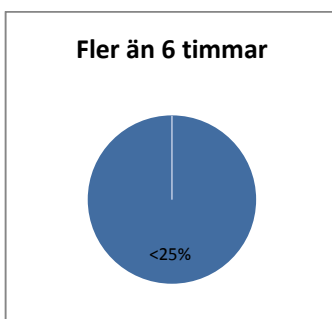
Figur 4.26 Andel spelprogrammeringsuppgifter hos de svarande som spelar mindre än 1 timme i veckan.



Figur 4.27 Andel spelprogrammeringsuppgifter hos de svarande som spelar 1-3 timmar i veckan.



Figur 4.28 Andel spelprogrammeringsuppgifter hos de svarande som spelar 4-6 timmar i veckan.

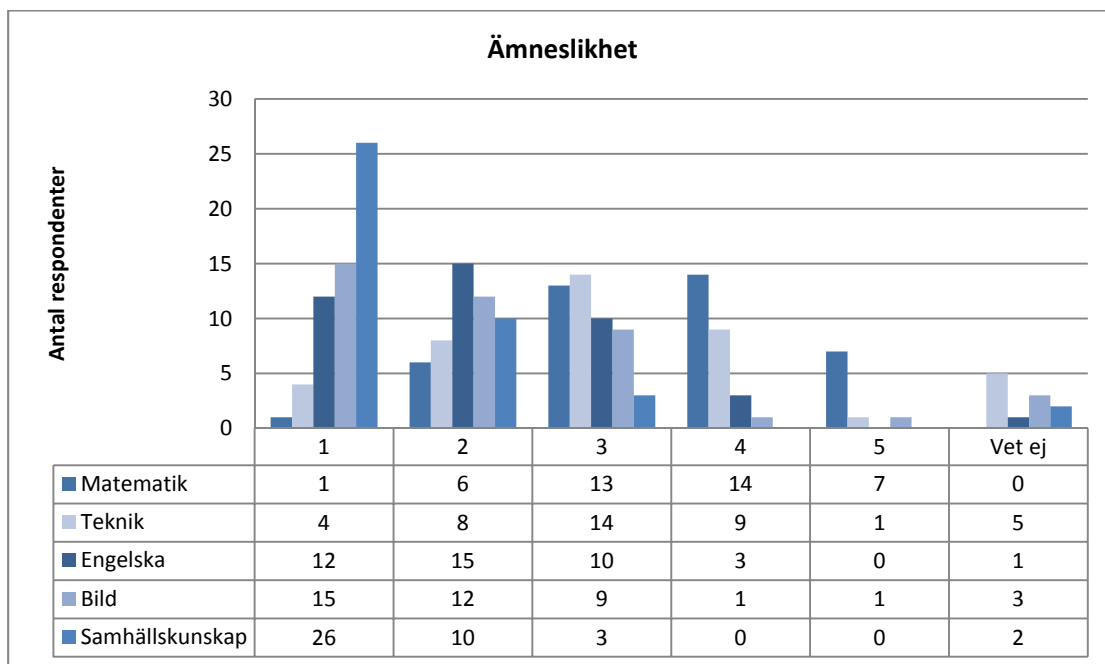


Figur 4.29 Andel spelprogrammeringsuppgifter hos de svarande som spelar fler än 6 timmar i veckan.

Det verkar inte finnas något tydligt samband mellan uppgivet spelintresse och användning av spelprogrammeringsuppgifter. Däremot framkommer att alla som använder sådana uppgifter till 75 % eller mer själva spelar mindre än tre timmar per vecka. Därmed inte sagt att de som spelar spel inte alls använder den typen av uppgifter.

4.1.5 Likhhet med andra ämnen (enkätfråga 4,5,6,7,8)

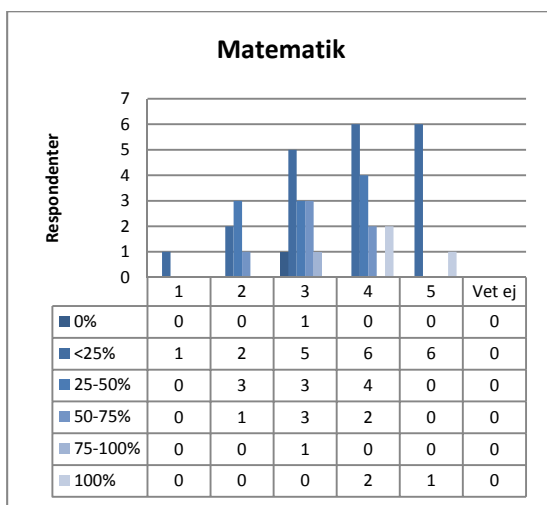
Enkätfrågorna 4-8 handlar om programmeringsämnets likhet med andra ämnen. Respondenternas svar presenteras i figur 4.30.



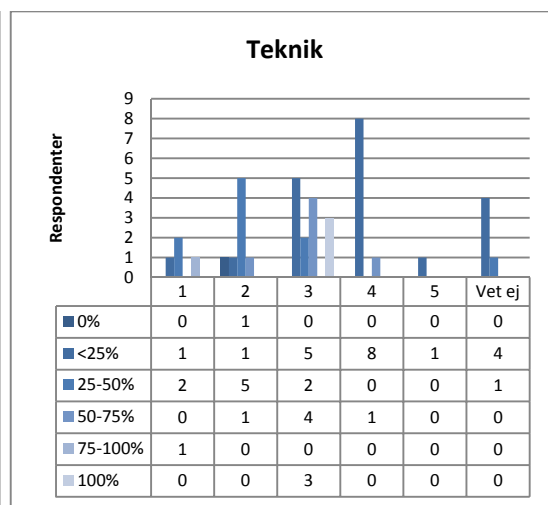
Figur 4.30 Graden av likhet de svarande upplever mellan ämnet Programmering och andra ämnen.

De flesta menar med andra ord att Samhällskunskap inte alls liknar programmeringsämnet. Det är i stort sett bara matematiken där fler än 50 % av respondenterna givit likheten en fyra eller femma. Tekniken ges en medelmåttig likhet, medan engelskan och bilden övervägande hamnar i den lägre delen av skalan.

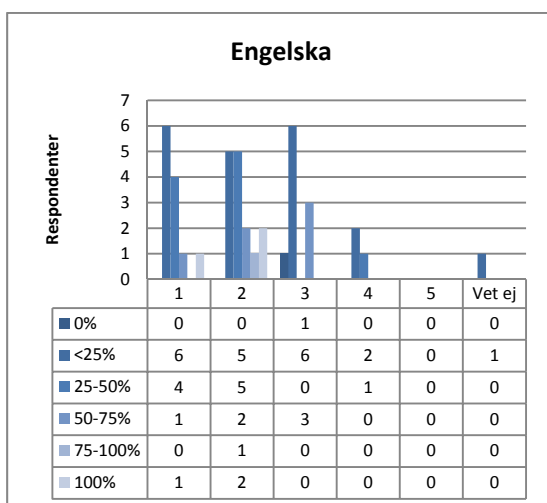
Uppbrutet efter ämne och grupperat efter svaren på enkätfråga 9 blir resultatet som i figurerna 4.31-4.35.



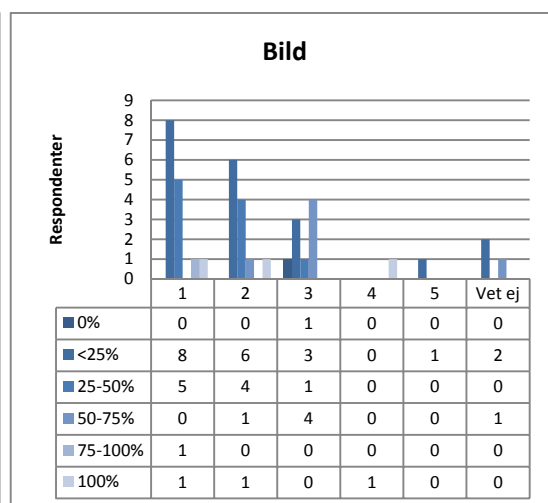
Figur 4.31 Likhet mellan programmering och matematik kopplat till andel spelprogrammeringsuppgifter i undervisningen.



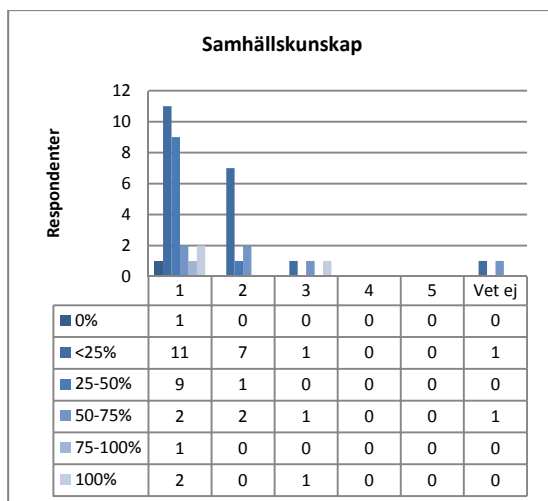
Figur 4.32 Likhet mellan programmering och teknik kopplat till andel spelprogrammeringsuppgifter i undervisningen.



Figur 4.33 Likhet mellan programmering och engelska kopplat till andel spelprogrammeringsuppgifter i undervisningen.



Figur 4.34 Likhet mellan programmering och bild kopplat till andel spelprogrammeringsuppgifter i undervisningen.



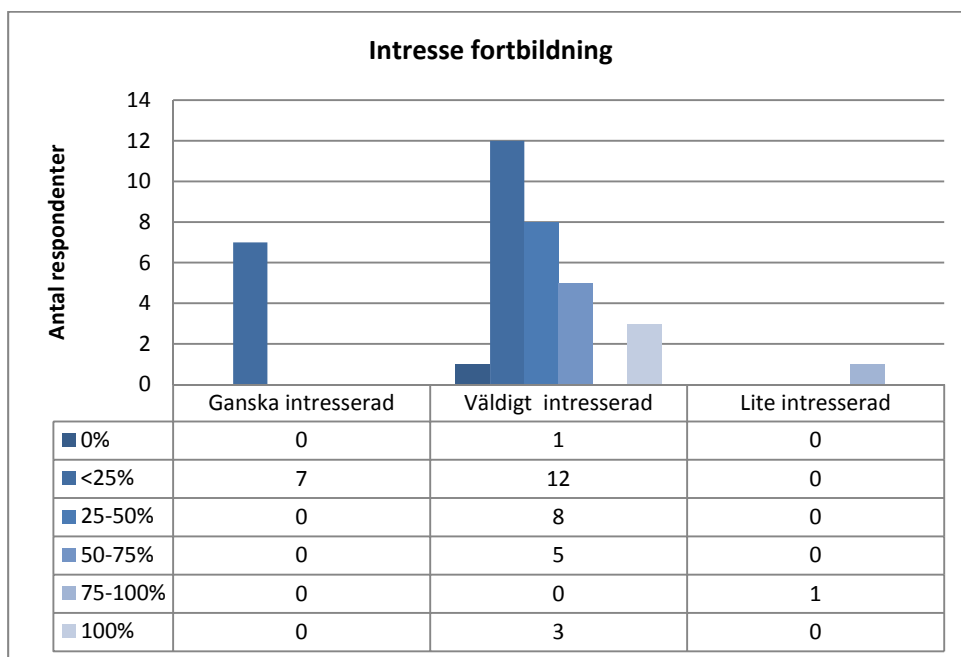
Figur 4.35 Likhhet mellan programmering och samhällskunskap kopplat till andel spelprogrammeringsuppgifter i undervisningen.

Sammantaget finns här inga överraskningar; teknik och matematik anses av de flesta respondenter ha större likheter med programmeringsämnet än samhällskunskap, engelska och matematik. Bland de som har mindre än 50 % rankar 20 av 31 engelska som en etta eller en tvåa; motsvarande bland de som har mer än 50 % är 7 av 10. Motsvarande för bildämnet är 23 av 31 och 4 av 10. För samhällskunskapen är siffrorna 30 av 31 och 7 av 10.

I jämförelse får matematiken 6 av 31 respektive 1 av 10. Det finns med andra ord inga tydliga kopplingar mellan vilka ämnen som anses mer eller mindre lika programmeringen och valet att använda spelprogrammeringsuppgifter.

4.1.6 Intresse vad gäller fortbildning i spelprogrammering (fråga 13)

Av de 41 enkätsvaren faller ett bort från sammanställningen nedan. Personen ifråga svarade att "sådan får man fortbilda sig själv i", vilket kan tolkas antingen negativt eller pragmatiskt.



Figur 4.36 De svarandes intresse i fortbildning kopplat till andel spelprogrammeringsuppgifter i undervisningen

Som syns i figur 4.36 är de allra flesta ganska eller väldigt intresserade av att genomgå fortbildning i spelprogrammering, om de skulle få utrymme för det i sina tjänster. Endast en av de 40 är "lite intresserad". Samtliga av de sju som är "ganska intresserade" använder spel i låg utsträckning (mindre än 25 % av tiden), medan de 29 som var "väldigt intresserade" är spridda över hela spektrat.

4.2 Intervjuerna

De intervjuade har anonymiserats och fått namnen A, B, C, D och E. Sedan tidigare har jag utbytt studiematerial med C och varit aktuell som ersättare när E bytte arbetsplats samt varit på samma spelträff som E vid ett tillfälle. Jag har ingen relation till någon av de intervjuade utöver denna korta kontakt.

4.2.1 Intervju A

4.2.1.1 Bakgrund och utbildning

A har arbetat som programmeringslärare sedan 2002, med "lite uppehåll" ungefär mitt i perioden, fram till 2006. Sedan tidigare var han lärare i matematik, CAD, teknik och naturkunskap. Får åtta år sedan, alltså 2004, valde han att fortbilda sig och skaffa sig behörighet som datorlärare. Detta gjorde han eftersom han ville bredda sin kompetens. "Som naturlärare jobbade jag mycket med datorämnet", säger han, och han ville få bättre koll och bättre formell behörighet.

Sedan dess har han inte haft någon fortbildning utöver de träffar för programmeringslärare i SIT-SNET-nätverket han varit på. Han beskriver fortbildningssituationen som att det är "klent" och "skamligt" hur få möjligheter det finns. "Datavetenskapen utvecklas så fantastiskt mycket och fort [...] Det borde vara krav att man som datorlärare, oavsett vilka datorämnen man har, att man borde skickas på regelbunden fortbildning". Han jämför med matematiken, där han menar att det inte händer så mycket; "det är samma matte, liksom". Däremot "om programmeringskurserna ska kunna fortsätta vara ett levande ämne så behöver man fortbildning".

Han har, med stöd från sin skolledning i form av en ledig dag i veckan, påbörjat en kurs i androidprogrammering, men han upplever att han inte har tid att slutföra den eftersom han inte känner att han har de förkunskaper som krävs.

4.2.1.2 Didaktik och uppgifter

A går i en grundläggande programmeringskurs igenom först sekventiell programmering (variabler, datatyper, loopar, villkor). Därefter jobbar han med metoder, att skriva egna klasser, variabler och klassers synlighet samt enkla UML-diagram (relations, struktur och flödesdiagram). Han har börjat fundera på hur han ska lägga in rekursion, som är nytt i och med Gyll.

A använder enbart Java, som är lite av en industristandard idag, som språk i undervisningen. Tidigare utgick han, med hans egna ord, mer "slaviskt" från det läromedel som fanns tillgängligt. Det innebar mest ickegrafiska uppgifter. Numera använder han grafik tidigt i form av klassen "Simplewindow" som han fått från läromedlet "Objektorienterad programmering och Java" (Holm, 2007). Redan tidigt låter han eleverna tillverka enkla spel, såsom "pong", "snake" och andra klassiker. De får sedan jobba vidare med mer avancerad grafikhantering, där de gör egna frågesporter och liknande. Mot slutet av de grundläggande programmeringskurserna har han en uppgift som utgår från ett spel några elever tidigare skapat; ett bristfälligt "tre i rad". Eleverna får då arbeta med att förbättra spelet.

Anledningen till att han bytte till att arbeta mer grafiskt var att han upplevde att det bättre fängade elevernas intresse, "att hela tiden får feedbacken grafiskt är ju så mycket mera stimulerande än att få textmeddelande". Däremot säger han om spel, "att det just är spel är ju mindre intressant, Det är ju bara det att spel i sig har något slags lockelse... Det hade lika gärna kunnat vara att grafiskt rita ett hus, men det är inte samma lockelse i det". Med andra ord är det främst det grafiska han själv är ute efter, men han anser ändå att just själva spelformen fungerar extra bra med eleverna. Det blir, säger han, som att köra på riktigt snarare än att köra med "attrapp". Spelen eleverna bygger vill de visa för andra: "kolla, ändå, jag lyckades med det här..."

4.2.1.3 Spel

Han ser stor fördel i att kunna arbeta grafiskt; "man ser hela tiden grafiskt resultatet av alla sina ansträngningar". De områden där han tycker att det blir svårt att arbeta med spelprogrammering är dels rekursion och dels sökning och sortering. I övrigt ser han inga hinder.

Själv har han inget spelintresse alls privat; han använder spel för att motivera eleverna men spelar inte alls själv. Han ser mest de negativa aspekterna av spel: "Eftersom våra elever är teknikelever på media-information[sprogrammet], så har de oftast det gemensamt att de tycker om datorer, och att de tycker att dom kan mycket om datorer, och att deras syokonsulenter och föräldrar har sagt att dom ska välja något som har med data [att göra]... Kanske hälften av mina elever är spelmissbrukare... och får en väldigt snäv social kompetens av att sitta hemma och spela, typ fyra till tolv timmar om dagen." De eleverna tappar mycket skoltid, och "Betygsmässigt hamnar de ofta i bottenskiktet".

Han säger att han hört att spelande tränar en "massa andra förmågor", men är ganska skeptisk. I skolsammanhang har han själv bara negativa erfarenheter.

När han jobbar med spel finns det alltid de som blir otåliga, de blir "lite missnöjda för att... Att en så enkel sak som att få två kvadrater att röra sig och inte krocka, eller att någonting ska hända när de krockar, är liksom... skitsvårt. Att göra på ett bra sätt, tycker de, såklart. [...] De tänker sig att de ska göra ett litet "Counterstrike", liksom, på två veckor. [...] Det är väl bara att göra ett spel, och så blir de lite förbannade..." Han betonar att eleverna måste vara medvetna om att det krävs en hel del arbete och abstrakt tänkande för att klara kursen.

4.2.2 Intervju B

4.2.2.1 Bakgrund och utbildning

B gick sin första kvällskurs i programmering på 60-talet. Den var, med hans ord, "inte så effektiv" eftersom eleverna inte fick se eller hantera en dator förrän slutet av kursen. På 80-talet började han undervisa i datorämnen. På den tiden rörde det sig mest om kvällskurser, eftersom de stora universiteten och högskolorna inte var intresserade av datorkurser. Det fanns, berättar han, "en stor mängd små företag som undervisade i dels datoranvändning, dels programmering, webbdesign, sådana saker. Det fanns ett stort tryck då, folk ville lära sig men det fanns inga kurser inom de här vanliga etablerade vägarna".

I slutet av 80-talet började han arbeta som programmeringslärare vid den företagsekonomiska institutionen vid en högskola. Efter detta bestämde han sig för att bli gymnasielärare och studerade då vid lärarhögskolan för att få behörighet. I början av 90-talet började han undervisa på olika gymnasier. Programmering, säger han, fanns inte som lärarinriktning. "De flesta programmeringslärarna var själv-lärda".

Efter 90-talet har han fortsatt jobba som gymnasielärare, och även bland annat skrivit läromedel i programmering.

4.2.2.2 Didaktik och uppgifter

Det viktiga, anser B, är att det finns en progression i kunskaperna. Man börjar med sekventiell programmering, iteration (loopar) och så vidare. Vi kommer under intervjun aldrig riktigt in på konkreta "moment" så som diskuterades med A. Däremot talar B om det viktiga i att välja ett språk läraren behärskar till fullo.

Han arbetar mest med småuppgifter, i varje fall i början av kurserna, till exempel enkla frågesporter. "Det går ju ganska snabbt att lära sig att göra enkla datorspel... Visserligen är det ju inte den sortens kunskaper som industrin eller webbdesignföretagen efterfrågar..." Det är tydligt att han framför allt ser programmering som något kopplat till industrins eller webbutvecklingsföretagens behov, och att spelen egentligen inte har något värde i sig. Han påpekar också att olika kompetenser krävs för industriell programmering kontra spelprogrammering.

Däremot menar han att spelen som uppgifter betraktade fungerar oerhört bra och ger motivation. "Om man skulle sätta en elev på att programmera ett stålverks olika industriprocesser så funkar inte det, för det finns inget stålverk att göra det mot... Men om man ger eleven i uppgift att göra en liten frågesport med ungefär samma syntax med moduler och så vidare så går det att få det klart på en timma eller två, och köra det på några kompisar eller köra det själv, det blir en helt annan motivation om man kan göra något man känner är färdigt och klart, man kan felsöka och börja diskutera". Spel är heller inget nytt som didaktiskt grepp menar han; "så har det ju alltid varit, ända sedan början av 80-talet, att det var spel var något som ingick i de vanliga inlämningsuppgifterna".

B menar också att programmering borde vara allmänbildning: "grundkunskap inom programmering är en nödvändighet för alla som vill förstå modern teknik och modern vetenskap och det har tyvärr inte slagit igenom varken i Sverige eller i andra länder".

4.2.2.3 Spel

Själv spelar B väldigt sparsamt med elektroniska spel. När han spelar blir det mest schack, othello eller liknande. Möjligen Tetris någon gång. "Men som här stora datorspelet, som säljs mycket, är ett slags filmkonst, har inget med programmering att göra. Programmering tycker jag mer är ett hantverk, som ska som man måste ha när man ska göra datorspel men också ett hantverk som man kan använda i andra sammanhang som inte har med datorspel att göra"

Det han tycker är negativt med spel är framför allt att man idag inte har möjligheten att själv modifiera spelen i någon större utsträckning. Man får spela på producentens villkor. Han skulle överlag vilja se mer av "End-User Software Engineering", alltså att slutanvändaren skulle kunna modifiera och anpassa programmen till "något som bättre svarar mot de behov eller arbetsuppgifter man har".

4.2.3 Intervju C

4.2.3.1 Bakgrund och utbildning

C studerade först vid universitetet, och läste sedan upp lärarbehörigheten separat 1996. Därefter arbetade han som programmerare i fem år. Därefter sysslade han med helt andra projekt några år, för att sedan söka sig till läraryrket. Han har alltså inte arbetat länge alls som programmeringslärare.

4.2.3.2 Didaktik och uppgifter

När C först började arbeta som programmeringslärare blev det framför allt med språket Java, och en typ av övningar han kallar "torrsim" eller "matte med syntax". Det passade vissa elever, de som var intresserade av matematik, men långt ifrån alla. Därför jobbade han först med det lite enklare språket Python och det grafiska biblioteket PyGame, vilket han upplevde fungerade bättre. Responsen från eleverna var positiv, det var lättare att komma in i och eleverna kom snabbt in i det grafiska och i att få direkt visuell återkoppling.

Han har också testat språket C#, som bland annat används för att bygga spel till spelkonsollen Xbox 360, men upplevde att det var lite krångligt att komma in i om man började med det redan i de grundläggande kurserna. Nu jobbar han med Actionscript, som används för att göra Flashspel och liknande.

Han har också testat på Javascript, som mest används för att skapa dynamiska hemsidor. Överlag har C med andra ord hunnit med att experimentera en hel del under de år han varit programmeringslärare.

När han lägger upp en grundläggande programmeringskurs går han igenom variabler, funktioner, händelsehantering, klasser, objekt, arrayer (listor), loopar samt sökning och sortering. Det enda han har svårt att få in via speltillverkning är just sökningen och sorteringen, samt rekursion som tillkommer nu i Gy11. Han jobbar annars med spel så tidigt som möjligt, även om det ibland finns risk för att det bara blir att han står och skriver kod via projektor och att eleverna sedan skriver av. Det krävs en del abstrakta, logiskt tänkande för att skapa även de enklaste spelen.

Tidigare, när han jobbade med Java, kom grafiken inte in förrän i slutet av Programmering C. Grafik, och speltillverkning, ger eleverna motivation. Det är också något eleverna hela tiden efterfrågar. "De jobbar mycket hellre hemma om det är något spel de håller på med än om de ska göra något matterelaterat..."

Han upplever att han fått blandade reaktioner från sina kollegor, men att han har stöd från sin skolläring som ser att en spelinriktning kan hjälpa till att locka elever till skolan.

4.2.3.3 Spel

C spelar inte mycket själv, framför allt blir det "Super Mario Bros" med sonen. Han har inte tiden eller intresset, och upplever det som en nackdel att han inte har tillräcklig koll på "spelvärlden". De kunskaper han har kommer från egna studier: "Man har fått tagit av fritiden och läst in sig... jag kan inte göra något häftigt spel själv, och det känner eleverna av..."

Han ser stort värde i spelen som instrument, han menar att "de flesta barn och ungdomar är jätteintresserade" vilket ger högre motivation. Dessutom tycker han att det är "bra för dem också att se, på programmeringen, hur mycket kod och hur mycket logik det krävs för de där små enkla spelen". Eleverna förstår mer av spelen de spelar.

Nackdelen med spel ur skolperspektiv menar han är att de tar så mycket av elevernas tid - "eleverna ovana vid att göra läxor, de spelar istället".

Han tycker dock att spel är något som känns nytt och fräscht som pedagogiskt grepp; han menar att "Det är väl lite gamla skolan kanske som vill köra algoritmer och rekursiva och matte med syntax... [...] Sen kommer det ganska många nya programmeringslärare som känner av elevernas intresse, och själva kanske spelar också."

4.2.4 Intervju D

4.2.4.1 Bakgrund och utbildning

D är i grunden högskoleingenjör. Efter sin examen arbetade han som datortekniker på komvux och gymnasieskolor i sju år. I samband med detta kom han i kontakt med undervisning i datorämnen, och undervisade sina första kurser i AutoCAD, programmeringsspråket Pascal och operativsystemet MS-DOS.

1997 fick han en heltidstjänst som lärare, då framför allt på "nätverkssidan", med kurser i operativsystem och databaser. Senare kom han också in på programmeringsämnet. Han kompletterade sin utbildning med en lärarexamen och blev behörig 2010. Han upplever att han trivs i yrket och att hans elever är studiemotiverade.

Han beskriver sig själv som självlärd, självständig och "lite ensam" eftersom han är den ende programmeringsläraren. Han skulle gärna vilja ha mer utbyte, diskutera uppgifter och bedömning med andra programmeringslärare.

4.2.4.2 Didaktik och uppgifter

I kursen Programmering A börjar D med grundläggande syntax, med variabler, datatyper, villkor, loophantering och funktioner. I B-kursen tillkommer sedan filhantering, länkade listor och vektorer. Han jobbar framför allt med små uppgifter, men ju mer eleverna lär sig desto större uppgifter får de. Det kan till exempel handla om ett program som räknar ut arean på en rektangel, eller ett enkelt kassa-program.

I de större uppgifterna väljer eleverna själva inriktning, och många väljer då att göra enkla spel. Som exempel nämner han kortspelet "21", "Yatzy" och "Mastermind". Själv introducerar han inga speluppgifter i de grundläggande kurserna eftersom han då inte upplever att elevernas kunskaper är tillräckliga; han säger att "oftast kan det ju vara lite svårt det här med spelprogrammering och därför har inte jag det som element i programmering A och programmering B-kursen utan jag brukar vilja lägga in det i programmering C, när de har de grundläggande bitarna".

I C-kursen upplever han att elevernas intresse blir större när de får göra spel och jobba mer grafiskt. Han menar att det "kanske inte så kul att mata in namn i en länkad lista" och att spel är något eleverna är mer intresserade av. På frågan om han skulle vilja använda spel i högre grad än han gör nu svarar han "nej, jag tycker det är bra nu. Eventuellt om man skulle hitta någon ingång till app-programmering". Vi kommer inte djupare än så.

4.2.4.3 Spel

Själv spelar D en hel del spel, och ganska avancerade sådana. Han är ett stort fan av "Mass Effect"-serien och berättar att han helst spelar spel med "story", alltså berättande spel. Hans elever spelar en hel del spel, framför allt "Fifa" och "World of Warcraft". Han ser en fördel i elevernas spelande i relation till programmeringsundervisningen; "de vet hur ett bra spel ska designas, de kommer ju inte så långt i programmeringen, men kan ändå se hur de kan förbättra saker och ting i ett enkelt spel, att göra det lite roligare."

Nackdelen med elevernas spelande, menar han, är att det finns risk för att det går "till överdrift". Men han tillägger att de allra flesta av hans elever håller det på en "bra nivå".

4.2.5 Intervju E

4.2.5.1 Bakgrund och utbildning

E är från början utbildad och behörig gymnasielärare i historia, och arbetade tidigare som lärare i historia vid ett universitet. Efter det gick han diverse programmeringskurser och fick 1999 arbete som gymnasielärare i programmering. Det jobbet har han, på olika skolor, haft sedan dess. Han upplever att det är otroligt mycket enklare att få jobb som programmeringslärare än historielärare, och tänker skaffa sig behörighet i ämnet.

E har haft tur och kunnat skaffa sig viss fortbildning genom att studera på halvtid via lärarlyftet. Han menar att eget intresse, eget driv, är en förutsättning för att man ska kunna fungera som lärare i programmering: man blir "inte långlivad utan eget intresse".

4.2.5.2 Didaktik och uppgifter

E spenderar i stort sett hela Programmering A med att gå igenom variabler, selektion (if-satser) och iteration (loopar). I B-kursen går han sedan igenom datastrukturer, filhantering och liknande. Klasser och objekt kommer han inte in på förrän i C-kursen.

I början använder han mer "slutna" uppgifter, där varje program har ett väldigt begränsat användningsområde, till exempel att skriva ut texten "Hello, world!" på skärmen. Allt eftersom får eleverna göra större och mer öppna uppgifter, till exempel kan de få konstruera ett enkelt kassasystem eller en miniräknare. I C-kursen arbetar han just nu med spel.

Senast han hade en A-kurs jobbade han tidigt med spel för att visa på grundläggande selektion. Han gav eleverna ett enkelt spel där man gick ner i en grotta, och eleverna kunde själva förbättra, komplettera och utöka spelet. Han upplever att även enkla, textbaserade spel ger väldigt positiv respons från eleverna.

Han upplever att han använder mer spel i sin undervisning ju längre han arbetar, att han vågar "ta ut svängarna" när han blir mer bekväm i ämnet. Han ser ingen anledning till varför man inte skulle kunna använda spelprogrammering redan från början i en grundläggande programmeringskurs, men tillägger att det inte fungerar i vissa grupper. Det finns elever, menar han, som betraktar spel som något "töntigt" och där det snarare är till nackdel att använda dem. Det är ofta de eleverna, säger han, som man måste vara som mest försiktig med. I allmänhet menar han dock att man "absolut [...] kan köra spel på varje nivå". Även de elever som själva inte spelar "har ändå en förståelse för att de finns där", och dessutom är det antagligen, enligt E, sällan de som inte spelar spel som väljer att läsa datorkurser.

Han upplever att spel motiverar eleverna överlag. Det ger något att kunna producera ett spel: "Hur dåligt spelet än är, har man gjort det själv så spelar man det". Spel tenderar också att göra eleverna förhållandevis självgående; "det räcker med att sätta igång det, sedan driver de varandra".

4.2.5.3 Spel

På frågan om han spelar elektroniska spel svarar E skämtsamt att han "inte gjort det på säkert en timme". Han säger att han är väldigt intresserad av spel, och spelar en hel del ganska avancerade tur-baserade strategispel och liknande. Dessutom spelar han onlinerollspelet "Lord of the Rings Online" och diverse brädspel som konverterats till elektroniskt format t.ex. "Blood Bowl".

De nackdelar han ser med elevers spelande är när det går "överstyr", när de spelar för mycket, för länge, och till för sent på natten. När det går ut över deras skolarbete. Han menar att det handlar om en mognadsfråga, och att alla elever inte är så mogna när de går i gymnasiet. En annan nackdel han ser är att elever som spelar mycket tenderar att röra sig mindre.

Han passar också på att argumentera emot idén att spel skulle göra ungdomar våldsamma; de grupper som spelade våldsamma spel t.ex. "Doom" eller "Wolfenstein 3D" tidigare är knappast de som begår våldsbrott idag.

Han ser många fördelar i spelen som hobby. De innebär en form av social träning, eleverna blir bättre på engelska, eleverna kan också få ett större intresse för att modifiera eller bygga spel själva vilket då leder dem till programmeringsämnet. Det kan också ge en god stämning menar han, och exemplifierar genom en ganska heterogen klass dom fick god sammanhållning när det visade sig att de alla spelade på samma server i "World of Warcraft".

Kapitel 5: Diskussion

I det här kapitlet sammanfattas resultaten och de slutsatser jag dragit utifrån dem.

5.1 Metoddiskussion

Kvalitativa undersökningar medger, till sin själva natur, inte att resultaten tas för allmänna sanningar. Däremot kan de fylla en funktion som djuplodande problematiseringar kring ett givet ämne. Det är också så intervjuerna fungerat i det här arbetet.

Resultaten från enkäten, den kvantitativa undersökningen, bör inte heller ses som statistiskt säkraställda, framför allt på grund av det stora bortfallet. Med 41 svar från ungefär 250 potentiella svarande kan det konstateras att enkätens värde framför allt ligger i att den kunde ligga till grund för urvalet av intervjurespondenter samt att den i någon mån kunde informera formuleringen av intervjufrågor samt analysen av intervjuvaren.

5.2 Spel som pedagogiskt verktyg

Att döma av både den tidigare forskningen på området och svaren i de intervjuer som ingår i den här undersökningen så kan speltillverkning fungera bra som medel för att få elever motiverade i programmeringsundervisningen.

Om eleverna får arbeta med spelprogrammering blir det mer sannolikt att de jobbar självständigt och frivilligt arbetar hemma. Elevernas egen drivkraft är en viktig motor för inläring. Motivation och intresse leder till positiva effekter för lärandet, vilket då i sin tur leder till att vi kan sluta oss till att spelprogrammering borde förbättra elevernas resultat. Detta bekräftas av de undersökningar som gjorts i andra länder.

Det finns också nackdelar med spelprogrammering. Först och främst finns det risk för frustration hos eleverna, när deras förväntningar kring hur svårt det är att skapa ett spel möter verkligheten. Eleverna har, enligt de intervjuade, ofta en felaktig bild där det är enkelt och går snabbt att bygga spel.

En annan nackdel är att spelprogrammering inte nödvändigtvis leder in eleverna mot kunskaper de kommer att behöva ifall de ska arbeta som programmerare inom industrin eller på webbyråer. Frågan är i vilka andra ämnen eleverna förväntas vara fullvärdiga yrkesutövare efter att ha genomgått gymnasieutbildning.

Ett par av de intervjuade lärarna använder spel tidigt, i stort sett från första lektionen. Detta motsäger det andra av de intervjuade lärarna säger, om att spel är "för avancerat" för de grundläggande kurserna. Det går att se två huvudfåror; dels de som menar att grundkunskaperna måste finnas för att spel- och grafikprogrammering ska fungera, dels de som menar att det bästa sättet att skaffa sig grundkunskaperna är att lära sig med hjälp av grafik- och spelprogrammering.

Däremot krävs naturligtvis mer forskning för att avgöra vilka grupper metoden bassar bättre eller mindre bra för. Det skulle också behöva forskas mer på vilka pedagogiska grepp som finns när det gäller just hur man lär gymnasieungdomar att göra spel. De flesta böcker om spelprogrammering är inte anpassade efter skolvärlden.

5.3 Attityder om spel

Ingen av de intervjuade såg spelprogrammering som något negativt. Vissa hade en uppfattning om spelprogrammering som något komplext och traditionen tycks vara att man inte arbetar med spel eller

grafik förrän i den mer avancerade C-kursen. Samtliga upplevde dock att elevernas motivation ökade när de fick arbeta med speltillverkning.

Vad gäller elevers spelintresse talade flera om det som ett problem. Ofta har de mestadels sett spelandets baksidor i form av spelberoende elever och elever vars spelvanor i förlängningen drabbar närvaron och skolarbetet negativt.

Det verkar inte finnas något direkt samband mellan egna spelvanor och didaktisk användning av spelprogrammering. I enkätsvaren kan man rentav se ett negativt samband; att de som själva spelar mycket tenderar att använda spelprogrammering i lägre grad. Spel och grafik ses inte som kulturform eller mål i sig, utan som ett sätt att få elever intresserade. Lärarna använder alltså inte spel för att de själva är intresserade av spel.

5.4 Förväntade samband och verklighet

I den här undersökningens grundläggande frågeställning fanns en förväntning kring hur de grupper som i hög respektive låg grad använde spel såg ut.

De som använde spel i hög grad förväntades vara unga lärare med eget spelintresse, som betraktade programmering som ett brett ämne som liknade både engelska och bild.

De som använde spel i låg grad förväntades vara äldre lärare utan eget spelintresse, som betraktade programmering som ett rent naturvetenskapligt ämne som mest liknade matematiken.

De här förväntningarna har också speglats en aning i intervjuerna, även om ingen uttryckligen identifierat sig med den senare gruppen.

Intervjuerna och enkätsvaren visar däremot att verkligheten är mer komplex, även om tendenserna säkerligen finns där. Alla intervjuade verkar ha en tydlig bild av "hur det var förut", de flesta har börjat sin bana som programmeringslärare med att lära ut "matte med syntax". Spel är något de upptäckt allt eftersom, eftersom elever efterfrågat det eller eftersom de själva tänkt att det borde kunna motivera eleverna.

Det har inte varit alldeles okomplicerat att försöka göra några grupperingar alls utifrån det begränsade underlaget. Det enda som kan sägas är egentligen att i stort sett alla programmeringslärare, unga som gamla, anser att programmering liknar matematiken. Bland intervjupersonerna finns flera som ser programmering som något som är kopplat till industrin i första hand, men även de använder spel i sin undervisning.

Bland de som svarade på enkäten kan man också se att de allra flesta inte använder spel i någon större utsträckning i de grundläggande programmeringskurserna. Detta gör att det blir svårt att egentligen uttala sig statistiskt eller svepande om de som använder spel.

5.5 Vikten av fortbildning

Något som bekräftas både av intervjuer och av enkätsvar är att det finns ett stort intresse av och sug efter fortbildning. Av de 41 som svarade på enkäten är det bara två som inte är "ganska" eller "våldigt" intresserade av fortbildning inom spelprogrammering. Att döma av intervjuerna finns det ett dug efter både ökat samarbete och mer fortbildning överlag, inte nödvändigtvis begränsat till just spelprogrammering.

Det verkar samtidigt vara väldigt viktigt med stöd från arbetsplatsen. Lärarna vill få tid i sina tjänster, och en av intervjupersonerna talade rentav om att skolledningar borde tvinga ("skicka") skolans programmeringslärare på fortbildning. Programmeringsämnet beskrivs som att det ständigt förändras och fortbildning beskrivs som närmast en nödvändighet.

Problemet verkar mest handla om resurser och tillgång. Dels kan kurserna kosta pengar, men sedan upplever de som blivit intervjuade att det inte heller finns särdeles mycket vad gäller bra fortbildning i programmeringsdidaktik. De upplever också en brist på mötesplatser; flera av respondenterna nämner SITSNET-träffarna som sker 1-4 gånger per år som den enda de känner till och har varit på.

5.6 Sammanfattning och konsekvenser

Den här studien indikerar att uppfattningen att användning av spelprogrammering som didaktiskt verktyg handlar om en generationsfråga inte stämmer. Istället verkar spelprogrammeringens användning begränsas i första hand av programmeringslärares kunskaper, samt deras förutsättningar och förmåga att skaffa sig dem.

Spelprogrammering är ett potent verktyg i programmeringslärares verktygslåda, och det finns ett skriande behov av fortbildning hos lärarna. Idag styrs användningen av spelprogrammering enbart av programmeringslärares egna intresse, drivkraft och vilja att lägga ner arbets- och fritid på att lära sig hur man bygger spel.

Konsekvensen av detta blir alltså att många spelintresserade elever får en programmeringsutbildning som i låg grad eller inte alls utnyttjar deras spelintresse. Detta riskerar i sin tur att leda till att de blir omotiverade och får lägre betyg och lägre benägenhet att fortsätta studera programmering.

Referenser

- Becker, K. (2001). Teaching with games: the Minesweeper and Asteroids experience. *J. Comput. Sci. Coll.*, 17(2), 23–33.
- Chun, R. (2009). *Adobe Flash CS4 Professional : classroom in a book*. Berkeley, Calif.: Peachpit.
- Dolan, M. (2006). XNA: Make your own XBOX games in 10 steps. *FierceDeveloper*. Retrieved December 21, 2012, from <http://www.fiercedeveloper.com/story/xbox-make-your-own-games-xna-10-steps-diy>
- Ejlertsson, G., & Axelsson, J. (2005). *Enkäten i praktiken: en handbok i enkätmetodik*. Studentlitteratur.
- Gallant, M. (2009, January 3). Defining Video Games. *The Quixotic Engineer*. Retrieved December 8, 2012, from <http://gangles.ca/2009/01/03/defining-video-games/>
- Hidi, S. (1990). Interest and Its Contribution as a Mental Resource for Learning. *Review of Educational Research*, 60(4), 549. doi:10.2307/1170506
- Hill, M. (1981, April 2). Why Pascal is Not My Favorite Programming Language. Retrieved December 21, 2012, from <http://www.lysator.liu.se/c/bwk-on-pascal.html>
- Holm, P. (2007). *Objektorienterad programmering och Java*. Lund: Studentlitteratur. Retrieved from <http://www.studentlitteratur.se/omslagsbild/artnr/6735-03/height/320/width/320/bild.jpg>
- Holme, I. M., & Solvang, B. K. (1997). *Forskningsmetodik: om kvalitativa och kvantitativa metoder*. Studentlitteratur.
- Koster, R. (2012, March 13). “X” isn’t a game! *Raph’s Website*. Retrieved December 8, 2012, from <http://www.raphkoster.com/2012/03/13/x-isnt-a-game/>
- Kylén, J.-A. (2004). *Att få svar : intervju, enkät, observation*. Stockholm: Bonnier utbildning.
- Leutenegger, S., & Edgington, J. (2007). A games first approach to teaching introductory programming. *SIGCSE Bull.*, 39(1), 115–118. doi:10.1145/1227504.1227352
- Lewis, J., & Loftus, W. (2008). *Java software solutions : foundations of program design*. Boston ;: Pearson/Addison-Wesley.
- Lindblad, E. (2006). *Programmering i Python*. Lund: Studentlitteratur.
- Medierådet. (2010). *Unga & Medier 2010*.
- Nilsson, M. (2009). *Världens största klassrum : Gymnasieelevers attityder till användandet av dator- och TV-spel i undervisningen*. Retrieved from <http://lnu.diva-portal.org/smash/record.jsf?pid=diva2:273980>
- Nordicom Sverige. (n.d.). *Sveriges Internetbarometer 2010*. Retrieved from http://www.nordicom.gu.se/?portal=publ&main=info_publ2.php&ex=340&me=10
- Nordlinger, J., & Phelps, A. (2007). Games First Pedagogy: Using Games and Virtual Worlds to Enhance Programming Education. Retrieved from <http://research.microsoft.com/en->

us/um/redmond/events/fs2007/presentations/nordlinger_john_phelps_a_games_research_faculty_summit_071607.ppt

- Rankin, Y., Gooch, A., & Gooch, B. (2008). The impact of game design on students' interest in CS. In *Proceedings of the 3rd International Conference on Game Development in Computer Science Education* (pp. 31–35). New York, NY, USA: ACM. doi:10.1145/1463673.1463680
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13(2), 137–172. doi:10.1076/csed.13.2.137.14200
- Rouse, M. (2007, November). What is C#? *TechTarget*. Retrieved December 21, 2012, from <http://searchwindevelopment.techtarget.com/definition/C>
- Skolverket. (2000a). Ämne - Datorteknik. *Skolverket*. Retrieved December 8, 2012, from <http://www.skolverket.se/forskola-och-skola/gymnasieutbildning/gymnasieskola-fore-ht-2011/kursplaner/sok-amnen-och-kurser/subjectKursinfo.htm?subjectCode=DTR>
- Skolverket. (2000b). *Gy2000. 2000:19, Programhandledning : programöversikter samt förteckning över ämnen och kurser : [programstrukturer, kurser och kurskoder m.m.]*. Stockholm: Statens skolverk.
- Skolverket. (2011). *Gymnasieskola 2011*. Stockholm: Skolverket. Retrieved from <http://www.skolverket.se/publikationer?id=2597>
- Skolverket. (2012). Ämne - Programmering. *Skolverket*. Retrieved December 8, 2012, from http://www.skolverket.se/forskola-och-skola/gymnasieutbildning/amnes-och-laroplaner/sok-program-och-amnesplaner/subject.htm?subjectCode=PRR&courseCode=PRRPRR01#anchor_PRRPRR01
- Statistiska centralbyrån. (2011). Undersökningarna av levnadsförhållanden (ULF/SILC). Retrieved December 8, 2012, from http://www.scb.se/Pages/ProductTables___12209.aspx
- Statistiska centralbyrån. (2012). Tillgång till IT-utrustning och Internet i hemmet för personer i åldern 16-74 år. År 2003-2012. Retrieved December 8, 2012, from http://www.scb.se/Pages/SSD/SSD_SelectVariables___340487.aspx?rxid=a8b8fd9f-923a-45f0-92b1-034d56487dcc&px_tableid=ssd_extern%3aLE0108T08
- Thau. (2000). *The book of JavaScript : a practical guide to interactive Web pages*. San Francisco, Calif.: No Starch Press.
- Vetenskapsrådet. (2002). *Forskningsetiska principer inom humanistisk-samhällsvetenskaplig forskning*. Stockholm: Vetenskapsrådet. Retrieved from http://www.cm.se/webbshop_vr/pdf/etikreglerhs.pdf

Bilaga A: Mailet

Hej!

Jag heter Mikael Bergström, och just nu arbetar jag med mitt examensarbete i didaktik - mer specifikt programmeringsdidaktik.

Programmering är ett klurigt ämne, som kräver ganska mycket av både eleverna och oss lärare. Med ett samlat grepp från flera lärare skulle man kunna se vad som är bra och lära av varandra. I mitt examensarbete tänker jag försöka undersöka vilka attityder och erfarenheter som finns kring att använda programmering av elektroniska spel som pedagogiskt verktyg i programmeringsundervisningen.

Det är här du kommer in. Jag är intresserad av att få veta dina åsikter om och erfarenheter av spel och spelprogrammering. Först och främst skulle jag vilja att du tog dig tiden att fylla i en enkät om ämnet. Enkätsvaren kommer att behandlas anonymt. Enkäten genomförs mellan den 10:e och den 24:e oktober, och beräknas ta ca 5-15 minuter att besvara.

Du kommer till enkäten genom att klicka på följande länk:
(länk)

Ett komplett följebrev till enkäten hittar du genom att klicka på den här länken:
(länk)

Om länkarna inte fungerar, så kan du kopiera dem till adressfältet i din webbläsare istället.

Utöver enkäten har jag också för avsikt att under hösten genomföra en serie intervjuer med programmeringslärare om samma ämne, och skulle uppskatta ditt deltagande även där. Om du kan medverka i en intervju ber jag dig att fylla i din mailadress i enkäten eller att du skickar ett mail märkt "intervju" till (mailadress), så tar jag kontakt senare.

Om du har problem med att besvara frågorna eller har synpunkter på frågeformuläret så hoppas jag att du tar kontakt med mig. Målet med undersökningen är att skapa kunskap kring hur vi i vårt yrke arbetar, så att vi kan lära av varandra och få fler verktyg i vår pedagogiska verktygslåda, och i det arbetet är du en nyckelperson.

Tack på förhand!

Mikael Bergström
E-mail (mailadress)
Mobilnummer (nummer)

Bilaga B: Följebrevet

Till dig som undervisar i grundläggande programmeringskurser

Programmering är ett klurigt ämne, som kräver ganska mycket av både eleverna och oss lärare. Med ett samlat grepp från flera lärare skulle man kunna se vad som är bra och lära av varandra. Resultaten från den här enkäten kommer att användas som underlag i ett examensarbete rörande användandet av speltillverkning som pedagogiskt instrument i programmeringsundervisningen.

Den här enkäten skickas ut till alla som anslutit sig till SITSNETs lista och deltagande är givetvis helt frivilligt. Dock uppskattas den tid du lägger ner på att fylla i den, då ditt deltagande gör att resultatet blir bättre och mer verklighetsnära.

Enkätresultaten kommer att behandlas anonymt, och det finns inget sätt att identifiera just ditt svar om du inte själv skriver ditt namn.

Under hösten har jag för avsikt att genomföra en serie intervjuer med programmeringslärare om samma ämne, och skulle uppskatta ditt deltagande även där. Om kan medverka i en intervju ber jag dig att fylla i din mailadress i enkäten eller att du skickar ett mail märkt "intervju" till (mailadress), så tar jag kontakt senare.

Om du har problem med att besvara frågorna eller har synpunkter på frågeformuläret så hoppas jag att du tar kontakt med mig. Målet med undersökningen är att skapa kunskap kring hur vi i vårt yrke arbetar, så att vi kan lära av varandra och få fler verktyg i vår pedagogiska verktygslåda, och i det arbetet är du en nyckelperson.

Nynäshamn 2012-09-26

Mikael Bergström

E-mail (mailadress)

Mobilnummer (nummer)

Bilaga C: Enkäten

Enkät om programmeringsdidaktik och elektroniska spel

Programmering är ett klurigt ämne, som kräver ganska mycket av både eleverna och oss lärare. Därför är det intressant att undersöka vilka metoder att undervisa i programmering som egentligen används, för att se vad vi kan lära oss av varandra. Dessutom är det bra att utvärdera nya metoder för att se om de skulle kunna passa in i en modern pedagogik. Resultaten från den här enkäten kommer att användas som underlag i ett examensarbete rörande användandet av speltillverkning som pedagogiskt instrument i programmeringsundervisningen.

Syftet med denna enkät är alltså att undersöka erfarenheter och attityder kring spel och programmeringsuppgifter hos programmeringslärare på gymnasienivå.

När enkäten talar om kurser i "grundläggande programmering" menas framför allt kurserna Programmering A+B och Programmering 1.

Alla svar är anonyma, och resultatet kommer att fungera som diskussionsunderlag i ett examensarbete med samma tema.

För fullständigt följebrev och mer information:

https://docs.google.com/document/d/1_1DFf4e4wqBNMZ75qNz42a0F2f4Fmm01Ka9A4G2B7OY/edit

Om du har frågor:

Mail: mikael.bergstrom@tekniknashamn.se

Telefonnr: 070-2811063

* Required

1. Skulle du kunna tänka dig att bli intervjuad om ditt arbetssätt och tänkande i rollen som programmeringslärare?

Fyll isåfall gärna i din mailadress nedan, eller skicka ett mail till mikael.bergstrom@tekniknashamn.se. Det är naturligtvis helt frivilligt.

2. Hur många år är det ungefär sedan du först började undervisa i programmering på gymnasienivå? *

- ☐ 1-5 år
- ☐ 6-10 år
- ☐ 11-15 år
- ☐ 16-20 år
- ☐ 21-25 år
- ☐ 26-30 år
- ☐ Fler än 30 år

3. Har du genomgått någon utbildning, fortbildningskurs eller motsvarande i programmering de senaste fem åren? *

- ☐ Nej
- ☐ Ja, någon enstaka
- ☐ Ja, flera

4. I hur hög grad tycker du att ämnet ENGELSKA liknar programmeringsämnet?

Gradera från 1 (inte alls likt) till 5 (väldigt likt)

▼

5. I hur hög grad tycker du att ämnet MATEMATIK liknar programmeringsämnet?

Gradera från 1 (inte alls likt) till 5 (väldigt likt)

▼

6. I hur hög grad tycker du att ämnet SAMHÄLLSKUNSKAP liknar programmeringsämnet?

Gradera från 1 (inte alls likt) till 5 (väldigt likt)

▼

7. I hur hög grad tycker du att ämnet BILD liknar programmeringsämnet?

Gradera från 1 (inte alls likt) till 5 (väldigt likt)

▼

8. I hur hög grad tycker du att ämnet TEKNIK liknar programmeringsämnet?

Gradera från 1 (inte alls likt) till 5 (väldigt likt)

▼

9. Hur stor del av undervisningstiden skulle du uppskatta att dina elever spenderat med uppgifter som handlade om att tillverka elektroniska spel, när du undervisat i grundläggande programmering? *

Utgå från senaste gången du var lärare i en grundläggande programmeringskurs (Programmering A+B, Programmering 1 eller motsvarande).

- ☐ Aldrig
☐ Mindre än 25% av tiden
☐ 25 % eller mer, men under 50 % av tiden
☐ 50 % eller mer, men under 75 % av tiden
☐ Aldrig
☐ Mindre än 25% av tiden
☐ 25 % eller mer, men under 50 % av tiden
☐ 50 % eller mer, men under 75 % av tiden
☐ 75 % eller mer, men under 100 % av tiden
☐ 100% av tiden
☐ Other:

10. Hur många timmar utanför arbetstid uppskattar du att du spenderar på att spela elektroniska spel en vanlig vecka?

- ☐ 0 timmar (spelar inte)
☐ Mindre än 1 timme
☐ 1-3 timmar
☐ 4-6 timmar
☐ Fler än 6 timmar
☐ Other:

11. Hur skulle du beskriva ditt spelintresse?

- ☐ Ickeexisterande
☐ Litet
☐ Medelmåttigt
☐ Stort
☐ Väldigt stort
☐ Other:

12. Hur många timmar uppskattar du att du spenderar på att programmera för eget bruk (hobbyprogrammering) en vanlig vecka?

- ☐ 0 timmar
☐ Mindre än 1 timme
☐ 1-3 timmar
☐ 4-6 timmar
☐ Fler än 6 timmar
☐ Other:

13. Hur intresserad skulle du vara av att genomgå forttbildning i spelprogrammering, om du skulle få tid till det i din tjänst?

☐ Inte alls intresserad

☐ Lite intresserad

☐ Måttligt intresserad

☐ Ganska intresserad

☐ Veldig intresserad

☐ Other:

Bilaga D: Intervjuguiden

BAKGRUND: Hur ser din bakgrund ut?

- Hur länge har du jobbat som programmeringslärare?
- Hur kommer det sig att du blev det?
- Vad har du gått för utbildning/ar?

DIDAKTIK: Hur lägger du upp en grundläggande programmeringskurs?

- Vilka områden brukar du ta upp (t.ex. datatyper, klasser, etc)?
- Hur examinerar du?
- Har du använt spel i undervisningen, isf hurdå?
- Hur tänker du kring spelprogrammering? Erfarenheter? Resultat/effekt?
- (Vad kan inte läras ut via spelprogrammering?)

SPEL: Hur ser du på elektroniska spel?

- Vad är de bra/dåliga till?
- Spelar du själv, isf vad?