



GÖTEBORGS UNIVERSITET

Kommunikation och krav- utvinning vid spelutveckling

En studie av systemvetares roll vid utvecklingen av
pedagogiska och underhållande spel

Communication and requirements elicitation during game development

A study of the role of systems analysts during the development of pedagogical
and entertaining games

DAVID GRÖNLUND
ROBIN KLOCKERMAN
NIKNAM MOSLEHI

Kandidatuppsats i informatik

Rapport nr. 2011:028
ISSN: 1651-4769

Sammanfattning

Rollen som systemvetare är traditionellt förknippad med utvecklandet av nyttoapplikationer som är till för att stödja processer inom olika organisationer. Den här studien vill undersöka om och i så fall hur systemvetare kan bidra i skapandet av ett pedagogiskt och underhållande spel, ett så kallat "playful learning"-spel, samt hur systemvetaren kan bidra som en medlare under denna process som onekligen innefattar flertalet olika kompetenser, vilket i sin tur innebär flera olika sätt att kommunicera på. Det finns mycket forskning som beskriver hur krav kan utvinnas och förmedlas i systemvetarens traditionella roll men mycket lite om hur systemvetaren bör agera vid utveckling av datorspel. Genom en analys av olika kravspecifikationer och en transkriberingsanalys kan vi redogöra för en metod som var oss behjälplig i ett specifikt projekt kallat CoDAC. Metoden är en uppsättning hjälpmedel som inkluderar en "playcentrisk" designprocess, flödesscheman, klassdiagram, prototypande och "mock-ups". För att undvika begreppsförvirring bland deltagare med olika bakgrund föreslås användning av ett gemensamt referensramverk, till exempel Björks & Holopainens (2005) designmönster eller ett spelbibliotek. Vi har slutligen kunnat se att systemvetaren även kan verka som en agent vid transformerandet av idéer till tekniska lösningar och vid eventuella intressekonflikter där fokus på underhållning eller pedagogik gör att den ena aspekten riskerar att försakats till förmån för den andra.

Nyckelord: kommunikation, kravspecifikation, kravutvinning, datorspel, lekfullt lärande, systemvetare

Abstract

The role as systems analyst (in Sweden called “systemvetare”) is often associated to the development of process supporting business applications. This paper aims to examine whether, and in such case how, the systems analyst can contribute to the development of a pedagogical and entertaining game, a so called “playful learning” game. Furthermore it aims to describe how the systems analyst can contribute as a mediator during this process which undeniably includes several different competencies, resulting in several different ways of communication. A literature review suggests that much research has been done related to requirements elicitation and the communication of requirements in general development situations, but there is a lack of research concerning requirements and communication in the development of computer games. Through an analysis of requirements specifications and an analysis of transcribed material we present a method that helped us in a specific project, called the CoDAC project. The method is a collection of aids including a “playcentric” design process, flow charts, class diagrams, prototyping and mock-ups. To avoid confusion concerning the language use among the project participants we suggest the use of a common framework of references, for example the design patterns presented by Björk & Holopainen (2005) or a game library. In conclusion we see that the systems analyst can take on the role as an agent in the transformation of ideas to technical solutions and in conflicts of interest where entertainment or pedagogical principles are at risk of being sacrificed in favor of one or the other.

This paper is written in Swedish.

Keywords: communication, requirements specification, requirements elicitation, computer games, playful learning, systems analyst

Förord

Vi vill rikta ett stort tack till vår handledare Wolmet Barendregt för goda råd, kloka hänvisningar och snabb återkoppling. Utan ditt engagemang hade vi stundtals onekligen famlat i blindo.

Tack även till samtliga deltagare i CoDAC-projektet för många värdefulla inslag till vår datainsamling.

Innehåll

| | | |
|-------|--|----|
| 1 | Inledning..... | 1 |
| 1.1 | Syfte..... | 2 |
| 1.2 | Frågeställning..... | 2 |
| 2 | Bakgrund..... | 3 |
| 2.1 | Kommunikationshistoria..... | 3 |
| 2.2 | Tyst kunskap..... | 3 |
| 2.3 | Behovet av ett bättre sätt att kommunicera krav..... | 4 |
| 2.4 | Allmänt om CoDAC-projektet..... | 4 |
| 2.5 | Utmaningen pedagogik kontra underhållning..... | 6 |
| 3 | Teori - Relaterat arbete..... | 7 |
| 3.1 | Kravspecifikationer..... | 7 |
| 3.2 | Designmönster som kravspecificeringsverktyg..... | 8 |
| 3.2.1 | Game Component Framework..... | 8 |
| 3.2.2 | Utmaning, fantasi och nyfikenhet..... | 10 |
| 3.3 | Sammanfattning av teoridelen..... | 11 |
| 4 | Metod..... | 12 |
| 4.1 | Bakgrund..... | 12 |
| 4.2 | Val av metod..... | 12 |
| 4.3 | Aktionsforskning..... | 12 |
| 4.4 | Deltagande observation..... | 13 |
| 4.5 | Datainsamling, transkribering och textanalys..... | 13 |
| 4.6 | En "playcentric" designprocess..... | 14 |
| 4.7 | Möten med projektgruppen..... | 16 |
| 5 | Resultat..... | 17 |
| 5.1 | Resultat av specifikationsanalysen..... | 17 |
| 5.1.1 | Första specifikationen..... | 17 |
| 5.1.2 | Systemvetaren kliver in..... | 18 |
| 5.1.3 | Simplifiering av budgetskalet..... | 26 |
| 5.2 | Resultat av transkriberingsanalysen..... | 26 |
| 6 | Diskussion..... | 32 |
| 6.1 | Resultatens konsekvenser..... | 33 |
| 6.2 | Förslag på fortsatt forskning..... | 33 |
| 7 | Slutsats..... | 34 |
| 8 | Referenser..... | 35 |
| 9 | Bilaga..... | 38 |

1 Inledning

Inom systemutveckling är det vanligt med många intressenter som vill påverka programvaran. Många intressenter medför många olika områden av expertis och kunskap. Detta kan naturligtvis vara en styrka då det ofta är bra att ha bred kunskap så att det är möjligt att täcka in olika aspekter av en programvara. Många intressenter kan dock göra utvecklingsprojekt problematiska eller som det gamla ordspråket lyder: "Ju fler kockar desto sämre soppa".

Vid systemutveckling är intressenterna ofta med och specificerar vad programvaran ska kunna göra. Den här processen brukar kallas kravutvinning. Kravutvinning är en svår process som ofta medför problem. Brooks (1987) menar till och med att det svåraste med att utveckla programvara är att bestämma exakt vad som ska utvecklas. Resultatet av en kravutvinning brukar resultera i en så kallad kravspecifikation, vilken fungerar som en kommunikationskanal mellan intressenter och används ofta för att bekräfta att utvecklarna har förstått vad beställaren vill ha. En undersökning av Al-Rawas m.fl. (1998) tar upp kommunikationsproblem som kan uppstå mellan intressenter vid kravutvinning. De visar att beställaren ofta har svårt att förstå och bedöma kravspecifikationer på grund av de notationstekniker som används och menar att specifikationsdokument behöver bli mer effektiva som kommunikationsmedel.

Det finns vissa roller inom systemutveckling som fungerar som mellanhand mellan intressenter. Systemvetare är ett exempel på en sådan roll. I rollen som systemvetare ingår ofta att verka som en sorts medlare eller kommunikatör mellan olika parter (SACO, 2010) och i den rollen försöka minska kommunikationsproblem. Historiskt sett har systemvetaren figurerat exempelvis inom affärsverksamheter vid införande av nya system och fyllt en konsulterande roll då verksamhetens affärsprocesser och -logik stått inför en förändring. Allt eftersom informations- och kommunikationsteknologi (IKT) innefattas i allt fler verksamhetsområden öppnar detta således upp för ett större arbetsfält för systemvetare.

Ett exempel är det svenska skolväsendet som idag går mer och mer mot att innefatta IKT i allt större del av sin undervisning, vilket är påtagligt inte minst i Göteborgs stad (Lineducation, 2010). Detta ökar kraven på kompetent programvara som kan komplettera den traditionella undervisningen. Med detta som bakgrund har ett projekt, CoDAC (Conditions and tools for Development of Arithmetic Competences), initierats. CoDAC-projektets huvudsyfte är att observera och utveckla barns aritmetiska kompetenser genom att bygga upp en så kallad mikromiljö, dvs. närmiljön som påverkar barnens, i detta fall, matematiska förmågor (Lindström, 2009). I CoDAC-projektet utvecklades för några år sedan ett spel som ska försöka uppfylla projektets huvudsyfte (Barendregt m.fl., 2009).

Det här spelet ska nu vidareutvecklas till att bli mer underhållande samtidigt som det fortfarande uppfyller CoDAC-projektets huvudsyfte då tidigare studier visat att spelet i vissa fall inte varit motiverande att spela (Barendregt m.fl., 2009). Spelet ska med andra ord vara både pedagogiskt och underhållande. Vid utförandet av projektet har både pedagoger och spelutvecklare givna roller. Då dessa kompetensområden är relativt olika

varandra är risken stor för eventuella kommunikationssvårigheter. Det kan därför hävdas att även systemvetare har en given roll i det här projektet.

1.1 Syfte

I vidareutvecklingen av spelet kommer pedagoger arbeta tillsammans med spelutvecklare och som tidigare nämnts så kan många intressenter medföra problem. I det här fallet ligger problematiken i att pedagoger och spelutvecklare har olika sätt att uttrycka sig och olika kompetenser samt använder olika terminologier.

Den här studien positionerar sig just här, i kommunikationen mellan pedagoger och spelutvecklare i syfte att se om det finns något som vi som systemvetare kan göra för att hjälpa dessa två grupper i kommunikationen. Det här leder oss in på studiens frågeställning.

1.2 Frågeställning

För att göra spel som är både underhållande och pedagogiska krävs det ett samarbete mellan olika parter, såsom pedagoger och spelutvecklare. Således är följande frågeställning formulerad:

Hur kan systemvetare stödja design- och utvecklingsprocessen av ett pedagogiskt och underhållande spel?

För att mer noggrant kunna besvara frågeställningen har vi formulerat två underfrågor:

1. Vilka specifika utmaningar, om några, finns med att skriva kravspecifikationer som ska kunna förstås och tolkas av både pedagoger och spelutvecklare?
2. Hur kan systemvetare underlätta kommunikationen mellan pedagoger och spelutvecklare?

Med den första frågan syftar vi på det praktiska arbete som vi utfört med kravspecifikationerna till skillnad från den andra frågan som syftar till den stödjande rollen vi har haft i den dagliga kommunikationen mellan pedagoger och systemutvecklare. Således är vår ambition att täcka in både systemvetarens formella och informella roll i ett specifikt projekt.

2 Bakgrund

Följande avsnitt presenterar inledningsvis generella studier i kommunikationshistoria. Avsnittet fortsätter sedan med att beskriva begrepp så som formell och informell kommunikation samt tyst kunskap. Vidare beskrivs behovet av en god kommunikation vid utvinningen av krav. Slutligen redogörs för det projekt inom vilket studien utförts samt projektets största utmaning, nämligen att förena pedagogik med underhållning.

2.1 Kommunikationshistoria

Människan har alltid varit intresserad av världen och de mysterier som följer den mänskliga naturen. De flesta saker som vi tar för givet kan bli oväntat komplexa när vi försöker förstå dem systematiskt. Kommunikation är en av de saker vi tar för givet eftersom det är så djupt rotat i oss och följderna blir att vi ofta bortser från hur viktigt och hur komplext det är. Det har skrivits en hel del om kommunikationshistoria varav följande är ett fåtal exempel: Peters (1999) försöker i sin bok förklara hur kommunikation blev ett problem för oss människor; Crowley & Heyer (2006) väljer att visa på relationen mellan människans historia och kommunikationshistoria; Solymar (1999) fokuserar på den tekniska delen av kommunikation och undersöker när man först lyckades sända information från punkt A till B, långt bort, utan att leverera meddelande fysiskt. För den här studien ligger dock fokus snarare på allmän kommunikation och vi kommer att behandla ämnen så som informell och formell kommunikation samt tyst kunskap.

När det gäller kommunikation allmänt, skiljer man ofta på formell och informell kommunikation som då är två olika former av kommunikation (Jacobsen & Thorsvik, 2008). Formell kommunikation är oftast i skriven form och kan vara regelverk, PM, mötesrapporter eller som i detta fall kravspecifikationer. En nackdel med formell kommunikation är att det finns risk för att missförstå budskapet. Informell kommunikation är den typ av kommunikation som faller utanför ramen för formell kommunikation och kan då exempelvis vara vanliga samtal mellan kollegor på en arbetsplats. Formell och informell kommunikation relaterar till vår studie eftersom vi kommer att försöka underlätta kommunikationen mellan pedagoger och spelutvecklare genom kravspecifikationer och samtal. Formell och informell kommunikation kan sägas representera kravspecifikationer respektive samtal.

2.2 Tyst kunskap

I det här projektet är det som sagt många involverade och dessa har olika kunskapsbakgrunder. En dimension av kunskap som kommer att vara av stor vikt under studiens gång är tyst kunskap. Definitionen av tyst kunskap (eng. tacit knowledge) varierar. Några menar att tyst kunskap står i kontrast till explicit kunskap (bl. a Spender, 1996) och grundat på detta har man försökt förklara hur interaktionen mellan dessa bildar en kunskapsspiral: explicit kunskap delas mellan individer och blir tyst kunskap genom en internaliseringsprocess (Nonaka & Takeuchi, 1995). Andra menar att all kunskap har en tyst dimension (Polanyi, 1966). Oavsett hur man väljer att definiera det så är tyst kunskap något som alla har och eftersom kunskap är, i alla fall delvis, baserat på erfarenheter är tyst kunskap också subjektiv (Leonard & Sensiper, 1998).

De vanligaste situationerna där tyst kunskap används är vid problemlösning, problemdefinition och när det gäller att förutse händelser. En stor skillnad mellan experter och noviser vid problemlösning är att experter har erfarenheter som kan sägas bilda ett mönster som kan appliceras på nya problem och underlätta finandet av lösningar. Tyst kunskap gör det möjligt att rama in problem på ett kreativt sätt och utesluta de uppenbara svaren på problemet och istället ställa frågor som belyser problemet från en annan vinkel. Vidare ger det en grundläggande förståelse för hur något fungerar. När en sådan förståelse är uppnådd går det alltså också att förutse troliga skeenden i en situation (Leonard & Sensiper, 1998).

Leonard och Sensiper (1998) skriver att det är främst i interaktionen med andra människor som innovation uppstår eftersom kreativitet kommer från medveten och omedveten sortering, gruppering, matchning och blandning, och interaktionen med andra människor stimulerar dessa aktiviteter. Personers tysta kunskap är det som gör en person värdefull i en innovationsgrupp eftersom det i interaktionen med andra bildas ny unik kunskap (Leonard & Sensiper, 1998). Om vi relaterar det här till studien är både pedagoger och spelutvecklare experter inom sina områden och bär på en stor mängd tyst kunskap. En utmaning för oss kommer därmed att vara att försöka underlätta delandet av tyst kunskap mellan dessa grupper.

2.3 Behovet av ett bättre sätt att kommunicera krav

Som vi tidigare varit inne på så har en undersökning utförts av Al-Rawas m.fl. (1998) som berör några problem som kan uppstå i kommunikationen mellan olika parter. Den visar att slutanvändaren ofta har problem att förstå och bedöma kravspecifikationer på grund av de notationstekniker utvecklarna använder. Det är dessutom, menar Al-Rawas (1998), allmänt erkänt att kommunikationsproblem är en av huvudfaktorerna till att mjukvaruprojekt blir försenade eller misslyckas. I utvecklingsprojekt är 44 % av specifikationerna som utvecklare får av analytiker (exempelvis Systemvetare) formella med traditionella notationstekniker, exempelvis UML. Många slutanvändare är inte insatta i den typen av notation och är sällan intresserade av att lära sig den. I samma undersökning säger 86% av de tillfrågade utvecklarna att slutanvändarna ofta behöver kompletterande information när de använt formella notationstekniker i specifikationerna. För utvecklaren är det å andra sidan ett problem när specifikationen kommer i form av löpande text eftersom det ofta tar längre tid att utvinna funktionerna än från ett diagram som beskriver funktionaliteten. Sammanfattningsvis menar Al-Rawas m.fl. (1998) att det finns mer man kan göra för att specifikationsdokument ska bli mer effektiva som kommunikationsmedel.

2.4 Allmänt om CoDAC-projektet

Det har under årens lopp funnits ett stort antal spel som haft som ambition att göra lärande underhållande för eleverna. Som exempel på sådana spel kan nämnas "Cheops Pyramid" (Lundeberg & Tengborn, 2000). Problemet med de här spelen har emellertid varit att nästan alla antingen är underhållande eller pedagogiska. Det är i sin tur ett problem eftersom det är viktigt för inlärningsprocessen att spelet är underhållande (Lepper & Cordova, 1992). Få spel har lyckats kombinera båda aspekterna och spelmomentet har, i matematikspelens fall, oftast varit uträkning av tal förflyttat från papper till en skärm medan ramberättelsen eller spelkontexten "levt sitt eget liv". En utmaning är således att

“väva in” spelmomentet i ramberättelsen och samtidigt behålla de pedagogiska principer som, sedan tidigare, arbetats fram.

Under hösten 2010 kom vi i kontakt med ett projekt kallat CoDAC (“Conditions and tools for development of arithmetic competences”) genom att en av författarna fick uppdraget att skriva ett program för att visualisera loggdata från olika spelsessioner av ett matematikspel. Spelet går ut på att snabbt känna igen olika mönster genom en kognitiv process kallad subitizing (förmågan att, utan att räkna, snabbt kunna känna igen delar som en helhet). Denna process innefattar vad som brukar kallas för del-helhetsrelationer (eng. part-whole relationship), vilket innebär att delarna relateras till den helhet (i detta fall en summa) som delarna utgör (Motschnig-Pitrik & Kaasbøll, 1999). Varje mönster består av ett antal symboler (i det här fallet frukter, bollar, glassar och löv) och intervallet är 1 - 10. Det pedagogiska målet är att träna upp barns förmåga att uppfatta del-helhetsrelationer och spelet riktar sig mot förskolan och barn med särskilda behov.

CoDAC-projektets huvudsyfte är att observera och utveckla barns aritmetiska kompetenser genom att bygga upp en så kallad mikromiljö, dvs. närmiljön som påverkar barnens, i detta fall, matematiska förmågor. En sådan miljö ger underlag åt didaktiska interventioner, som nämns vara stödjande vid utvecklingen av talbegreppsliga förmågor hos yngre barn (Lindström, 2009), vilka i sin tur är grundläggande för vidare utveckling av aritmetiska färdigheter.

Projektet syftar till att studera hur utveckling av grundläggande talbegrepp och aritmetisk kompetens sker i interaktion mellan barn, mellan barn och vuxna samt mellan barn och olika typer av artefakter och att beskriva denna utveckling i termer av ”learning trajectories” (ett sätt att beskriva utveckling som lärande över tid). Mer specifikt omfattar dessa studier analyser av:

- hur barn utvecklar förståelse av tal som del-helhetsrelationer
- vilka strategier de utvecklar för att hantera tal som del-helhetsrelationer
- strategiernas relation till utvecklingen av mer generiska aritmetiska färdigheter
- hur barnen använder fingrar som redskap i sitt matematiserande
- hur didaktiska miljöer ger villkor för och bidrar till utvecklingen av funktionella del-helhetsstrategier och talbegreppsliga förmågor (Lindström, 2009, s. 2).

Grundspelet som använts i projektet utvecklades i Flash för att det ska vara plattformsoberoende och har spelats på en vanlig arbetsstation med hjälp av två specialkonstruerade tangentbord med fem knappar var. De här knapparna är mappade mot spelarens fingrar och spelaren har alltså ett tangentbord per hand. Det finns emellertid vissa användbarhetsproblem med dessa tangentbord, främst till följd av att spelaren tvingas att anpassa sin fingersättning enligt tangentpositioneringen. Med dagens touchskärmar öppnas dock nya möjligheter upp. Genom att utveckla det nya spelet till Ipad, som i mångt och mycket kommer att vara en rak ”portning” av det gamla spelet, blir det både mer portabelt samtidigt som det öppnar upp för en vidareutveckling i form av mer intuitiva gränssnitt samt mer varierande spelformer.

Då tidigare studier visat att spelet i vissa fall inte varit motiverande att spela (Barendregt m.fl., 2009) vill vi positionera uppsatsen i samarbetet mellan pedagoger och spelutvecklare i syfte att frambringa egenskaper och finesser hos spelet som inte varit möjligt tidigare. De barn som tidigare har spelat spelet mest har även varit de som utvecklats mest (Barendregt m.fl., 2009), vilket talar för att spelet med fördel kan vara mer underhållande. Det här i kombination med det faktum att de gamla tangentborden orsakade vissa användbarhetsproblem (något som exempelvis en Ipad skulle kunna avhjälpa) motiverar ett samarbete mellan forskare inom pedagogik och datorspelsindustrin.

Från den akademiska världen ingår två medlemmar från Högskolan Kristianstad i projektgruppen; en universitetslektor i matematik, fysik och data, som även är CoDAC-projektets projektledare, samt en professor i pedagogik. Dessutom ingår två medlemmar från Göteborgs universitet; en professor i pedagogisk informationsvetenskap samt en universitetslektor i IT och lärande. Från spelindustrin ingår tre medlemmar från ett spelutvecklingsföretag som ska programmera och skapa grafiken till spelet. Dessa medlemmar innefattar företagets VD, företagets projektledare samt en spelutvecklare. Slutligen har även författarna ingått i projektgruppen i rollen som systemvetare.

2.5 Utmaningen pedagogik kontra underhållning

I samband med de många ansatser som gjorts mot att utveckla pedagogiska spel som är underhållande, har spel med dessa karakteristika teoretiserats med begreppet "edutainment games" (se exempelvis Egenfeldt-Nielsen, 2005; Hussain m.fl., 2003), framställt ur de engelska orden *education* och *entertainment*. Edutainment beskrivs som tillämpningen av utbildning och underhållning på ett stort urval mediaplattformar, däribland datorspel (Egenfeldt-Nielsen, 2005). I samband med detta betonas även begreppets bristande egenskaper, vilka nämns vara bland annat:

- Yttre, snarare än inre motivation. Exempel på detta är att få poäng för en avklarad spelnivå. Inre motivation hade istället varit känslan av att behärska spelet i samband med att en spelnivå avklaras.
- Ingen förening mellan inlärningsupplevelse och spelupplevelse, utan dessa är separerade från varandra.
- Ytinläring. Edutainment-spel begär ingen djupare förståelse för den problematik de presenterar för spelaren då problemen kan överkommas genom repetitiv användning och memorering. Ett exempel är att svaren till aritmetiska problem som $2 + 2$ kan memoreras då samma problem presenteras vid upprepade tillfällen och tillåter spelaren att memorera svaret snarare än att förstå de bakomliggande reglerna som tillåter att $2 + 2 = 4$.

Vidare argumenteras det för varför edutainment som begrepp borde formuleras om, då de två ord som inspirerat begreppet, nämligen *education* och *entertainment*, beskriver tjänster som tillhandahålls av någon annan (Resnick, 2004). En brukare av edutainment kan således betraktas som en passiv mottagare, vilket nödvändigtvis inte behöver stämma då rollen som mottagare eller deltagare bestäms av hur brukaren tolkar och interagerar med mediet. Därför föreslås, av Resnick (2004), begreppet "playful learning" som till synes lägger fokus på orden *play* och *learn*, det vill säga saker man gör snarare än blir tillhandahållen med.

3 Teori - Relaterat arbete

Det här kapitlet tar upp den teori som uppsatsen bygger på. En del i studien är att se om det är något speciellt med att utveckla kravspecifikationer för pedagoger och utvecklare och därför kommer vi att börja med att beskriva allmän teori om kravspecifikationer. Därefter beskriver vi ett ramverk för att kunna tala om och analysera spel samt tre principer för att skapa motiverande, pedagogiska spel. Det här för att den andra utmaningen är att underlätta kommunikationen inom projektgruppen.

3.1 Kravspecifikationer

Den här delen behandlar vad litteraturen säger om kravspecifikationer och vilka utmaningar som finns med skapandet av kravspecifikationer. Kravspecificering handlar i grunden om att upptäcka syftet med en produkt. I detta ingår att identifiera intressenter och deras behov och att dokumentera detta i en form som möjliggör analys, kommunikation och tillslut implementering av produkten (Nuseibeh & Easterbrook, 2000). Brooks (1987) är av uppfattningen att det svåraste med att bygga mjukvara är att bestämma exakt vad som ska byggas. Ingen annan del i det konceptuella arbetet är lika svårt och ingen annan del medför så stora konsekvenser om det görs dåligt. Han menar att det visserligen inte finns någon "Silver Bullet" i utvecklingen av mjukvara, men att kravspecificering - som iterativ process - trots det fortfarande är den viktigaste delen i utvecklingsprocessen, då det är av stor vikt att utvecklaren förstår vad det är kunden behöver. Vi kommer nedan beskriva ett antal hjälpmedel för kravspecificering.

Nuseibeh & Easterbrook (2000) beskriver ett antal olika metoder för att utvinna krav, bland annat traditionella tekniker, grupptechniker, prototyping, kognitiva, kontextuella och modelldrivna metoder. Kognitiva metoder syftar till att utvinna krav till kunskapssystem, kontextuella metoder ämnar att utvinna krav genom etnografiska studier och modelldrivna metoder syftar till att framställa mål som ska uppnås. Då kunskapssystem inte är en del av den här studien, då etnografiska studier skulle bli för tidskrävande samt då målen för spelet är definierade sedan innan, utelämnas dessa metoder från det teoretiska ramverket.

I traditionella tekniker ingår allmänna metoder för datainsamling såsom enkäter, intervjuer och analys av befintliga organisationsdokument exempelvis processmodeller eller organisationskartor. Grupptechniker försöker, genom gruppdynamik, främja att överenskommelser mellan intressenter kommer till stånd samt att få en ökad förståelse för kraven. I detta kan brainstorming, fokusgrupper eller RAD (rapid application development) användas som metoder. Prototyping används när det är stor osäkerhet kring kraven eller där det är viktigt med feedback från kunden tidigt i projektet. Prototyping kan också kombineras med andra tekniker, till exempel med en fokusgrupp för att få till en diskussion om produkten.

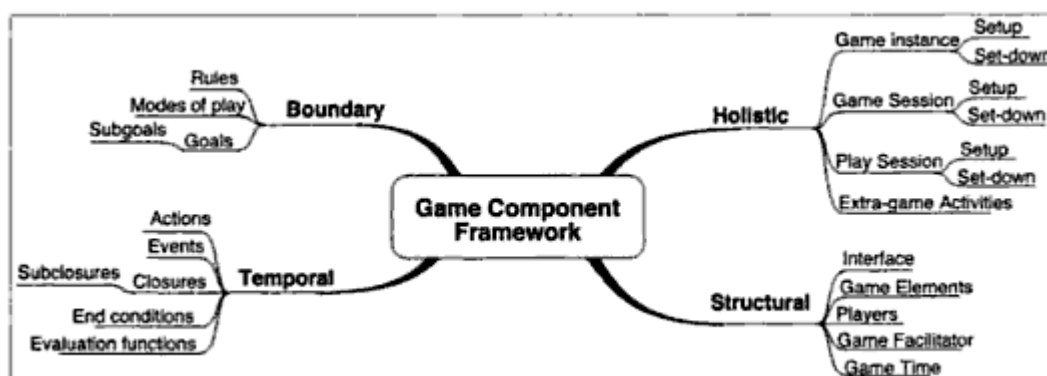
När det handlar om specifikt kommunikation och kravspecificering så har det forskats om det tidigare. Bland den tidigare forskningen har ett ramverk utvecklats som är tänkt att användas som beslutsunderlag för hur kommunikationen kan underlättas vid kravutvinning (Coughlan & Macredie, 2002).

3.2 Designmönster som kravspecificeringsverktyg

Nedan nämns de designprinciper som används som grund vid utvecklandet av kravspecifikationer. Björk & Holopainen (2005) beskriver ett antal designmönster som kan användas vid spelutveckling. Malone (1980) beskriver faktorer som gör pedagogiska spel motiverande.

3.2.1 Game Component Framework

Björk & Holopainen (2005) föreslår ett gemensamt "språk" för att kunna tala om och analysera spel. "Språket" består av ett antal designmönster grundade i ett allmänt ramverk. Ramverket består av fyra grundkomponenter indelade i olika subkomponenter (se figur 1).



Figur 1 Ramverk för spelkomponenter (Björk & Holopainen, 2005, sid. 8).

De *holistiska komponenterna* (Holistic) beskriver spelet på ett övergripande plan, eller en metanivå om man så vill. *Game instance* beskriver hela spelet och dess kontext från det att spelaren, eller spelarna, gör sig redo att spela spelet för första gången till det att spelet avslutas för sista gången samt allt däremellan. Det här inkluderar en *Setup*, där det som måste göras innan spelaren kan börja spela spelet ingår, och en *Set-down*, där det som görs efter att man slutat spela spelet ingår. En *Game Session* kan ses som en delmängd av en *Game instance* och beskriver en spelares spelande av ett spel. Även i den här komponenten ingår en *Setup* och en *Set-down*. En *Play Session* är i sin tur en delmängd av *Game Session* och beskriver en spelares spelande av en omgång, alltså den oavbrutna tid som en spelare spelar ett spel under en spelomgång. Liksom de två föregående komponenterna inkluderar även den här en *Setup* och en *Set-down*. Slutligen består de holistiska komponenterna även av *Extra-Game Activities*, en komponent som beskriver allt som man kan göra som egentligen inte har med själva spelandet att göra. Det kan t.ex. vara skapandet av en avatar eller registrerandet av ett så kallat "handle" på en "high score"-lista. De här komponenterna går inte helt sällan in i varandra och kan vara svåra att skilja på. T.ex. kan registrerandet av ett high score både vara en del av *Set-down* komponenten i en session men även en *Extra-Game Activity*.

Gränskomponenterna (Boundary) beskriver spelets logik och vad själva spelandet går ut på. Här ingår spelets regler (Rules), mål (Goals) med delmål (Subgoals) och olika typer av sätt att spela spelet på (Modes of Play). *Rules* beskriver, så klart, spelets regler och således vilka handlingar som får göras av spelaren och i vilken ordning dessa får utföras. Data- och tv-spel skiljer sig här från traditionella kort och brädspel genom att det

kan göras omöjligt att bryta mot reglerna. Vissa spel kan dessutom ha ett fokus på att hur det går i spelet är avhängt hur bra spelaren kan reglerna, i elektroniska spel kan man ofta testa sig fram och spelaren behöver därför inte kunna reglerna på förhand men ju bättre hon kan reglerna desto lättare är det att klara av målen och delmålen. Att "lista ut" reglerna kan därför bli ett spelmoment i sig. *Goals* och *Subgoals* är själva motivationen för att spela spelet och alltså vilka handlingar som bör utföras. Nästan alla spel har flera delmål som måste uppnås för att klara av huvudmålet. Till och med ett så pass enkelt spel som Sten, sax och påse har två delmål, det första är att vinna individuella matcher och det andra är att bli den första som vinner två matcher. Vissa spel tydliggör inte målen utan att lista ut vad det finns för mål med spelet är ett mål i sig. Den sista komponenten, *Modes of Play*, beskriver de olika sätt som spelaren kan spela spelet på och således vilka handlingar som kan utföras i de olika lägena. Exempelvis har *Pac-Man* ett *Mode of Play* när karaktären inte har ätit någon stor prick och därför måste akta sig för spökena och ett *Mode of Play* när karaktären har ätit en stor prick och därför kan äta upp spökena.

De *temporal komponenterna* (Temporal) beskriver spelets flöde så som det skulle beskrivas om en spelare berättar vad som hänt i en spelomgång. *Handlingar* (Actions) beskriver hur spelaren kan ändra på spelets *Game State*. Handlingarna kan antingen vara explicita eller implicita för spelaren. Explicita handlingar är sådana som syns för spelaren i form av t.ex. en joystick, knappar och menyer. Implicita handlingar återfinns framför allt i textäventyr och klassiska "peka och klicka"-äventyr. Här blir utforskandet av vilka handlingar som är möjliga en viktig del av spelet. Handlingar är associerade med spelets gränssnitt och beskriver vad som händer på skärmen när spelaren interagerar med gränssnittet. *Händelser* (Events) beskriver outputen från spelet, och även de är ändringar i spelets *Game State*. Händelserna kan men behöver inte vara associerade med spelarens handlingar. Vissa händelser "triggas" av andra spelare, algoritmer eller beror på hur länge spelaren spelat. *Avslut* (Closures) beskriver mätbara och, för spelaren, meningsfulla erfarenheter. *Avslut* kan, men behöver inte, vara associerade med mål. Ett avslut kan t.ex. vara när det blir uppenbart för spelaren att ett mål inte kan nås. En viktig skillnad mellan mål och avslut är att avslut alltid är knutna till en speciell tidpunkt i spelandet och kan även vara spelarens subjektiva erfarenheter. Mål, å andra sidan, är en del av spelets logik. *Slutvillkor* (End Conditions) definierar kraven för att ett avslut skall uppnås eller att en game instance, game session eller play session skall avslutas. Som sådana kan de användas för att "trigga" händelser. *Utvärderingsfunktioner* (Evaluation Functions) bestämmer hur ett slutvillkor behandlas. Det vill säga hur det gått för spelaren i spelet. Dels rör det sig om ifall spelaren klarat spelet (eller en del av spelet) men det kan även röra sig om hur väl spelaren klarat spelet. Ett exempel är funktionen i "Rock Star"-spel som "Grand Theft Auto" och "Bully" där det syns hur många procent av spelet som spelaren klarat. Det är möjligt att spela igenom spelet utan att ha behärskat hundra procent och därför går det att "klara" spelet olika bra.

De *strukturella komponenterna* (Structural Components) är de delar av spelet som manipuleras av spelarna och systemet. *Game Facilitator* är den agent som ser till att spelet flyter på och uppdaterar spelets *Game State* efter vilka handlingar som genomförs. I klassiska spel som brädspel och kortspel är det här spelarna själva, i analoga rollspel är det spelledaren och i elektroniska spel är det datorn. *Spelarna* (Players) är representationen av de som spelar spelet och alltså tävlar eller samarbetar om att uppnå spelets

olika mål. Även om spelaren oftast är en person som spelar spelet kan man behandla även datorkontrollerade element som spelare för att på så sätt vinna nya insikter i hur spel är uppbyggda. En representation av en spelare kan dessutom vara kontrollerad av flera olika individer som antingen hjälps åt att avgöra vilka handlingar som skall utföras eller som helt enkelt turas om att spela. *Gränssnitt* (Interfaces) förser spelarna med information om spelets innevarande game state och vilka handlingar som går att utföra. Det ger även spelarna möjlighet att utföra handlingar genom att t.ex. trycka på en knapp eller dra i en spak. *Spelelement* (Game Elements) är de fysiska och logiska komponenter som innehåller spelets game state och som blir manipulerade av spelarna för att de skall kunna uppnå sina mål. Spelelementen har ofta olika attribut kopplade till sig som anger på vilket sätt de kan manipuleras och som används i algoritmer för att avgöra vilka konsekvenser en handling får. Exempel på spelelement är spelvärlden, spelarens avatar, fiender och andra spelares avatrar. *Speltid* (Game Time) är den tidslinje som beskriver handlingarnas ordningsföljd under ett spelets gång. Speltid skiljer sig ofta från verklig tid i den bemärkelsen att den kan pausas, inte behöver vara konstant och i vissa fall till och med reverseras (se till exempel spelet "Braid").

Baserat på de ovanstående komponenterna har Björk & Holopainen (2005) skapat ett uppslagsverk av olika designmönster. Meningen är att man skall kunna använda dessa mönster för att hitta inspiration, analysera befintliga spel eller helt enkelt för att underlätta kommunikationen kring spel. Som exempel kan nämnas "new abilities"-mönstret som beskriver att spelaren får nya förmågor under spelets gång. Det här mönstret kan i sin tur ge upphov till nya mönster och instansieras av andra mönster. I Pac-Man finns mönstret "new-abilities". Det instansieras av "power-up"-mönstret. När avataren äter en "power-up" kan han äta upp de spöken som jagar honom. Det här ger upphov till "rewards"-mönstret då spelaren får extra mycket poäng för att äta upp spöken. På det här sättet hänger många av mönstren ihop och vissa går inte att kombinera.

3.2.2 Utmaning, fantasi och nyfikenhet

Malone (1980) har definierat ett antal principer för att göra instruktiva (pedagogiska) spel mer motiverande. Han definierar tre olika typer av motivation: utmaning, fantasi och nyfikenhet.

Utmaning

För att en spelmiljö ska vara utmanande behöver det finnas en osäkerhet kring hur ett spelmål ska uppnås. Det finns fyra sätt att skapa osäkerhet kring hur ett mål ska uppnås: variabel svårighetsgrad, flera spelmål för en nivå, gömd information och slumpmässighet. När det gäller spelmål så finns det inget som tyder på att det skulle vara fördelaktigt med bara två möjliga utfall (vinna eller förlora), det är dock ofta en större motivation om målen är kortsiktiga samt om fortlöpande nyttskalor konverteras till ett diskontinuerligt mål genom någon form av kriterienivå. Ett exempel på diskontinuerligt mål är när det är möjligt för spelare att ha som mål att slå sitt tidigare rekord eller ett rekord någon annan spelare har. Ett annat exempel skulle kunna vara att ha flera deluppgifter på en spelnivå så att det finns flera mål att uppfylla på en nivå istället för bara ett. I dessa fall har en kontinuerlig nyttskala konverterats till ett diskontinuerligt mål. Bandura och Schunk (1980) menar även att det är bättre med kortsiktiga mål än med långsiktiga

eller inga mål alls när det gäller att upprätthålla prestanda eller intresse i att göra en uppgift.

Fantasi

Malone (1980) nämner också fantasier som ett sätt att göra pedagogiska spel mer underhållande där fantasier definieras som "mentala bilder av saker som inte är märkbara för sinnena hos en person". Här skiljer han på yttre och inre fantasier. Yttre fantasier är beroende av en persons färdigheter men inte tvärtom. Ett spel där spelaren flyttar en bil längs en bana allt eftersom spelaren svarar rätt på aritmetiska problem är en yttre fantasi eftersom övningen inte är beroende av fantasin. När det gäller inre fantasier beror fantasin på en persons färdighet men färdigheten är också beroende av fantasin, vilket betyder att övningarna presenteras i samma form som den faktiska färdigheten i verkligheten. I ett dartspel är exempelvis färdigheten att bedöma avstånd applicerad på fantasivärlden med darttavlor. Färdigheten att bedöma avstånd används således både i spelet och verkligheten.

Nyfikenhet

Den tredje motivationen som nämns är nyfikenhet där en persons nyfikenhet kan väckas genom en optimal nivå av informationskomplexitet. Detta innebär att spelvärlden inte ska vara varken för simpel eller för komplex jämfört med spelarens nuvarande kunskap. Miljön bör vara ny och överraskande men inte oförståelig. Optimalt bör en spelare veta tillräckligt för att ha förväntningar, men där förväntningarna tidvis grusas.

3.3 Sammanfattning av teoridelen

Gällande kravspecifikationer generellt har vi i litteraturen sett att det är en stor utmaning att bygga kravspecifikationer och att ingen annan del i utvecklingsarbetet är lika svår som att bestämma vad som ska byggas. Vi har dock sett att det finns vissa hjälpmedel för att underlätta framtagandet av krav såsom olika grupptechniker och prototyping för att öka chansen att beställare och utvecklare har samma bild av vad som ska göras. Ett sätt att skapa kravspecifikationer för spel är att använda befintliga begrepp, såsom designmönster, som förklarar centrala funktioner i spel som kan göra att spelet blir lyckat. Vidare har vi sett att spel med tre motiverande egenskaper såsom att det är utmanande, fantasifullt och eggat spelarens nyfikenhet kan göra spel mer underhållande. Med detta avslutas teoridelen och vi går vidare med att förklara de metodval vi gjort för studien.

4 Metod

Det här avsnittet kommer först ge en bakgrund till valet av metod varpå vi går in på själva valet av forskningsmetod för studien samt valet av forskningsansats. Vi kommer också ta upp en metod för spelutveckling som har använts vid utvecklingen av kravspecifikationer. Metodavsnittet avslutas med ett stycke om de projektmöten vi deltagit i under projektets gång.

4.1 Bakgrund

I anslagsansökningen för CoDAC-projektet står att läsa att projektets grunddesign är så kallad "Design Based Research", DBR (Lindström, 2009). Metoden är iterativ och varje iteration består av följande moment:

- Barn erbjuds att spela ett "talspel" (se ovan).
- Etnografiska mikrolongitudinella studier görs på barnen när de spelar "talspelet".
- Kliniska intervjuer med barnen genomförs.
- Analyser av materialet med avsikt att beskriva utveckling av aritmetisk kompetens samt utgöra ett underlag för att revidera spelet.

Lindström (2009) menar att syftet framför allt är att designa en mikromiljö som erbjuder möjligheten att studera barns matematiska aktiviteter, genom att dels variera de verktyg och matematiska situationer barnen ställs inför och dels på ett effektivt sätt generera data. Utvecklingen kommer att beskrivas i form av "learning trajectories". Vår metodik måste således förhålla sig till den här övergripande metoden för projektet som helhet.

4.2 Val av metod

Vi kommer som sagt att delta i CoDAC-projektet, som ämnar utveckla ett pedagogiskt och underhållande spel. De mötestillfällen vi har med projektgruppen kommer ligga till grund för datainsamling.

Då vår roll i CoDAC-projektet kommer att bestå i att undersöka kravspecificering och kommunikation, är ett naturligt metodval *aktionsforskning* (se nedan) eftersom det handlar om just *ett* specifikt projekt. I vårt fall är ett alternativ till aktionsforskning att göra intervjuer med de olika projektdeltagarna. Vi anser dock detta vara överflödigt eftersom projektmötena ändå kommer att äga rum, samt då deltagarnas åsikter troligtvis inte kommer fram lika väl vid intervjuer.

4.3 Aktionsforskning

Aktionsforskning handlar om processer eller företeelser som inte skulle komma till stånd utan att forskarna påverkar eller startar ett skeende (Wallén, 1996). Aktionsforskning är inte en kombination av först forskning och sedan tillämpning och genomförande, utan vid aktionsforskning är genomförandet i sig ett sätt att bedriva underökning och används för både datainsamling och testning (Wallén, 1996). Det är därför viktigt, ur methodsynpunkt, att beskriva själva forskningsprocessen. Forskningsprocessen är också en läroprocess och bland de viktigaste resultaten från forskningen är vidgade erfarenheter hos

deltagarna. Detta är också ett av problemen med aktionsforskningen. Aktionsforskning betraktas av bland annat den anledningen ofta som ineffektiv jämfört med annan forskning och en del menar att om man kan undvika den och nå samma resultat så är det att föredra (Wallén, 1996). Å andra sidan är fördelen med aktionsforskning att man studerar något i en verklig situation där man genom detta får en sann bild av vad personerna tycker och tänker jämfört med i en intervju eller observationssituation där handlandet och åsikter ofta påverkas av att de är medvetna om forskningen (Wallén, 1996).

4.4 Deltagande observation

Vi kommer att gå in i projektet relativt förutsättningslöst, utan exempelvis frågor eller förberedda formulär, och gör därmed en kvalitativ ansats som kallas deltagande observation. Graden av forskarens deltagande vid dessa observationer kan variera från relativt utanförstående till att helt gå in i en arbetsuppgift (Wallén, 1996). Vårt fall är mer likt det senare eftersom vi har rollen som systemvetare i projektgruppen och är alltså helt deltagande. Fördelen med deltagande observation är att man får "inifrån-kunskap", kännedom om socialt samspel, "tyst" eller utsagd kunskap om sådant som tas för givet, samt erfarenheter och värderingar som inte kommer fram i intervjuer.

4.5 Datainsamling, transkribering och textanalys

Som tidigare nämnts kommer mötestillfällena delvis resultera i datainsamling i form av ljudinspelningar. I observationssammanhang är syftet med detta bland annat att registrera beteenden och informella förfaranden (Preece m.fl., 2007). Denna datainsamlingsmetod kommer därför att vara en väldigt viktig del i den process som förhoppningsvis resulterar i ett svar på underfrågan "*Hur kan systemvetare underlätta kommunikationen mellan pedagoger och spelutvecklare?*".

De informella förfaranden som äger rum vid mötestillfällena, exempelvis tolkning och förmedling av information mellan två parter, kan identifieras genom transkribering av ljudinspelningarna. En diskursanalys kommer sedan att utföras med den transkriberade texten som underlag. Målet med diskursanalysen är att tydligöra kommunikationsflödet mellan mötesdeltagarna samt framhäva vår roll som systemvetare i projektet.

I rollen som systemvetare ingår ofta att verka som en sorts medlare eller kommunikator mellan olika parter (SACO, 2010) och i den rollen minska kommunikationsproblem. Fördelen med att ha med ett systemvetarperspektiv vid projektgruppsmöten är att den kan hjälpa till med att hålla fokus vid de frågor som är viktiga för programvaran. I detta fall är det av vikt att spelet både är pedagogiskt och underhållande. När det transkriberade materialet analyserades tittade vi därför efter stycken med följande innehåll:

- Stycken där de medverkande talar förbi varandra.
- Stycken där de medverkande inte förstår varandra
- Stycken där vi som systemvetare förtydligar uttalande av de övriga medverkande
- Stycken där pedagogik eller underhållning försakas till förmån för det ena eller det andra.
- Stycken där någon kommer med ett koncept och vi föreslår en teknisk lösning.

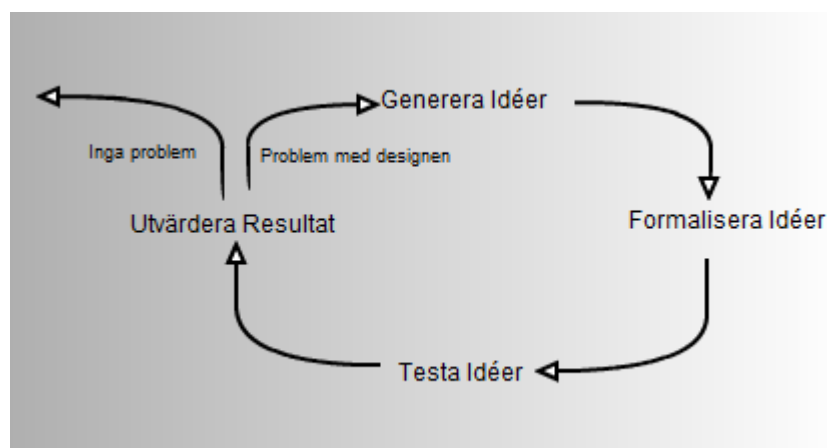
- Stycken där mötesdeltagarna använder sig av spelreferenser från befintliga spel för att förklara idéer och tankar på ett sätt som alla förstår.

För att underlätta granskningen av transkriberingen kommer ljudinspelningarna avlyssnas en gång innan analyseringen av det transkriberade materialet. Vid avlyssningen kommer anteckningar föras som anger något som matchar kriterierna ovan, samt den tidpunkt i inspelningen då de inträffar. Dessa anteckningar kommer sedan användas vid transkriberingen samt transkriberingsanalysen för att lättare kunna lokalisera relevant material.

4.6 En “playcentric” designprocess

Det finns en uppsjö av olika iterativa metoder för design och utveckling av mjukvara; se exempelvis: OOA&D (Mathiassen m.fl., 2001), Unified Process (Larman, 2004), SCRUM (scrum.org), XP (extremeprogramming.org). De flesta är emellertid inriktade på design och utveckling av diverse nyttoapplikationer som stöd för olika organisationer. Här blir studerandet av organisationens kultur och arbetsrutiner viktig. Vid design och utveckling av spel är det här hänsynstagandet av mindre vikt men icke desto mindre är användarperspektivet en mycket tungt vägande faktor. Tycker inte användarna (läs: spelarna) om att spela spelet är hela utvecklingen bara slöseri med tid. Vi har i arbetet med designen av spelet därför valt att använda oss av Fullertons (2008) metod för att skapa underhållande och innovativa spel.

Fullerton (2008) beskriver i sin bok “Game Design Workshop - A playcentric approach to creating innovative games” en process för att designa innovativa och underhållande spel. Metoden är iterativ och har sju olika steg. Stegen består av 1) brainstorming, 2) fysisk prototyp, 3) presentation, 4) mjukvaruprototyp, 5) designdokumentation, 6) produktion och 7) kvalitetsutvärdering. Med iterativ menar Fullerton (2008) helt enkelt att man designar, testar och evaluerar konstant under hela utvecklingsprocessen för att hela tiden förbättra spelet tills det möter de kriterier för spelupplevelse som satts upp (se figur 2). Nedan följer en redogörelse för ovanstående designprocess.



Figur 2 Den iterativa process som leder fram till att målen med spelupplevelsen uppfylls (Fullerton, 2008).

Det första som behöver göras är att sätta upp mål för spelupplevelsen och det görs under steg 1. Baserat på dessa mål fortsätter steg 1 med att man försöker komma på spelkoncept och mekanismer som uppfyller målen för att till slut skapa en lista med de

tre bästa förslagen. En kort beskrivning av alla de tre olika idéerna skrivs sedan ned i vad som brukar kallas ett konceptdokument för att kunna testas med potentiella spelare.

I steg 2 skapas en spelbar fysisk prototyp med hjälp av t.ex. papper och penna. Den här prototypen speltestas sedan och när den uppvisar fungerande spel som uppfyller de mål med spelupplevelsen som satts upp skrivs ett dokument som beskriver hur spelet fungerar på ca. 3 - 6 sidor.

Det tredje steget består av en presentation för att kunna försäkra sig om finansiering. Även om en finansiär inte behövs kan det vara bra att skapa en ordentlig presentation av spelet för att på så sätt få ett underlag för att tänka igenom spelet och kunna presentera det för övriga medarbetare och ledning. Att man löper risk att bli avvisad gör inte så mycket i det här skedet eftersom det inte lagts ner några enorma mängder arbete och en återgång till ritbordet är således fortfarande fullt möjlig.

Steg 4 består av mjukvaruprototypande och börjar med att utvecklarna skapar grova modeller av den grundläggande spellogiken. Oftast skapas flera olika prototyper som fokuserar på olika delar av systemet. Då huvudsyftet är att testa idéer är användandet av tillfällig grafik som kostar lite pengar att skapa är att föredra. Prototyperna speltestas sedan och när de uppvisar fungerande spellogik som uppfyller de mål som satts upp för spelupplevelsen går man vidare till nästa steg.

I steg fem börjar designdokumentationen att skapas. Under arbetet med spelet har det antagligen skrivits ned en hel del anteckningar och idéer, de används nu tillsammans med kunskapen som erhållits under prototypandet för att skriva ett första utkast av ett dokument som förklarar alla aspekter av spelet och hur de är tänkta att fungera. Det här dokumentet brukar benämnas designdokument men på senare tid har många designers börjat använda sig av wikis istället då dessa är mer flexibla och lättare att samarbeta i.

Det sjätte steget är själva produktionen av spelet. Här är det viktigt att arbeta med hela projektgruppen för att försäkra sig om att alla delar av designdokumentationen är korrekt beskrivna och möjliga att färdigställa. Vid det här steget börjar arbetet med att skapa den riktiga grafiken och koden. Det är därför viktigt att inte glömma bort att testa arbetet, i enlighet med den iterativa process som beskrivits ovan, under tiden det fortskrider. På det här sättet borde det bli allt färre problem och således färre förändringar ju längre arbetet fortskridit.

Det sista steget i designprocessen är kvalitetsutvärdering och när spelet kommit så här långt i utvecklingen bör spellogiken vara mycket stabil. Det är emellertid mycket möjligt att det fortfarande finns vissa problem så det är bra att fortsätta med speltester och framförallt titta efter hur användarvänligt spelet är. I det här skedet är det även viktigt att försäkra sig om att spelet är tilltalande för hela den publik av spelare som det riktar sig till.

Användandet av Fullertons (2008) designprocess har resulterat i två olika kravspecifikationsdokument. De här dokumenten kommer sedan att analyseras i resultatavsnittet för att få svar på den första underfrågan: *“Vilka specifika utmaningar, om några, finns med*

att skriva kravspecifikationer som skall kunna förstås och tolkas av både pedagoger och spelutvecklare?”

4.7 Möten med projektgruppen

Under projektets gång, och särskilt frekvent under initieringsfasen, har vi haft regelbundna möten med projektgruppen. Mötesförfarandet har gått till på sådant sätt att samtliga deltagare har slutit upp, antingen i en möteslokal eller på distans genom någon form utav internetbaserad kommunikation (såsom Skype eller Adobe Connect), och fört en gemensam diskussion om bland annat vilka element som bör ingå i spelet, hur spelprogressionen ska fortlöpa, vilken ramberättelse spelet ska ha, vilka pedagogiska aspekter spelet skall beröra, med mera. Större delen av det som diskuterades dokumenterades samtidigt i ett gemensamt Google-dokument, vilket fungerade som underlag för oss vid sammanställningen av spelspecifikationen som vi överlämnade till spelutvecklingsföretaget.

Mötena har, utöver att vara tillfällen att gemensamt spåna kring spelidén, även fungerat som utvärderingstillfällen där vi kunnat bedöma kvaliteten av de spelprogressionsmodeller som varit tilltänkta och koppla dessa till olika föreslagna ramberättelser (se figur 4).

5 Resultat

I det här avsnittet presenterar vi resultatet av dels en analys av kravspecifikationerna och dels en analys av det transkriberade materialet från ljudinspelningarna. Resultatet av specifikationsanalysen kommer först att beskriva den första kravspecifikationen som skrevs av pedagoger, för att sedan redogöra för hur vi använt oss utav Fullertons (2008) designprocess för att skapa underlag till den andra kravspecifikationen. Slutligen presenteras enstaka citat från det transkriberade materialet och kommenteras samtidigt i relation till sådant vi funnit vara relevant för att svara på våra frågeställningar.

5.1 Resultat av specifikationsanalysen

Som tidigare nämnts menar Al-Rawas (1998) att slutanvändare ofta har svårt att tolka kravspecifikationerna på grund av den notationsteknik som används av utvecklare. I vårt fall har utmaningen snarare varit att få spelutvecklingsföretaget, vilka har varit slutanvändare i detta fall, samt redan väl införstådda i notationsteknik för utveckling, att förstå de pedagogiska termer och begrepp som projektets forskare bidragit med till kravspecifikationen.

För att kunna besvara den första forskningsfrågan utgår vi från de skapade kravspecifikationerna för att försöka hitta några utmaningar med att skapa kravspecifikationer för pedagoger och spelutvecklare. Den här delen av resultatet är en redovisning av den arbetsprocess som har utförts när kravspecifikationerna har framtagits och de justeringar som gjorts allt eftersom det behövs samt eftersom förutsättningarna för projektet har ändrats. Vi inleder med att beskriva problem i den första specifikationen varpå vi visar resultatet från användningen av de tre första stegen Fullertons (2008) ramverk. Vi avslutar med att beskriva de ändringar som behövde göras efter det ställdes krav på en simplificering av kravspecifikationen på grund av budgetskäl.

5.1.1 Första specifikationen

Det första utkastet till specifikationen var inte fullständigt och till stor del skriven av pedagogerna i projektgruppen. Dessa har ingen stor vana av att skriva den här typen av kravspecifikationer, vilket fick som följd att språket var väldigt fackmannamässigt och bestod till stor del av funderingar och tankar utan någon förklaring. Följande stycken är exempel på ovanstående.

Exempel 1 är en alltför fackmannamässig förklaring på tanken om att bygga spelet på förmågan att känna igen och relatera delar som en helhet. Det vill säga förmågan att kunna förstå tal på olika sätt.

“Tanken är att man har delar som kombineras till en helhet och som man sedan transformerar till ett annat (?) del-helhetsmönster också i en annan modalitet.”

Exempel 2 är en något underutvecklad och flytande beskrivning av hur spelets feedback till spelaren bör fungera.

*“Formativ feedback --- om felsvar tala om vad de svarat och ange rätt svar. ----
Genom röst... Visa också vad man angivit för tal på skärmen.”*

Exempel 3 är en beskrivning av uppbyggnad och progression inom nivåer. Den här beskrivningen anger inte vad som behöver göras utan är snarare en nedteckning av tankar angående logiken.

“Inom uppgiftsset slumpas i följande dimensioner:

- 1. Helhetstal (summan). Grundtal (70&????) och variation runt om . Diskret fördelning, med vikter..Kan parametreras,*
- 2. Delantal ... $5=1+4=2+3=0+5$ (vissa kombinationer gå att mappa direkt (behöver inte transformeras). Kan vi införa detta som parameter. Klassa mönstren.*
- 3. Varje del kan ha olika mönster. vissa mönster är lättare än andra.. ”*

Angående inmatningsalternativ i spelet specificerades nedanstående i exempel 4. Det här medförde sedermera en oklarhet för spelutvecklingsföretaget då de vid ett möte eftersökte förtydligande på detta.

“Huvudalternativ

- *Fri fingersättning*
 - *Utan restriktioner*
 - *Med “affordanser” (illustration/fingerbild,markering av yta) som begränsar och riktar spelarens fingeranvändning? Detta kräver att spelet läggs på tvären med området för inmatning på undre halvan. Fungerar detta.*
- *Tangentbordsinmatning med rad av tangenter*
 - *Med numeriska symboler*
 - *Fingertalssymboler*
 - *Prickmönstersymboler (tärningsmönster)*
- *Virtuellt “Fingertangentbord” , motsvarande det gamla spelet”*

Som vi kan se från exemplen ovan är det otydligt vad som egentligen menas och det blir således svårt för utvecklarna att veta vad som behöver göras. I vår roll som systemvetare blev det därför vår huvuduppgift att skapa en begriplig och korrekt specifikation.

5.1.2 Systemvetaren kliver in

I arbetet med att skapa och strukturera upp de kravspecifikationer som spelutvecklingsföretaget ville ha för att kunna utveckla spelet tog vi hjälp av Fullertons (2008) “playcentrisk” designprocess. Emellertid har vår medverkan i projektet inom ramen för den här uppsatsen varit något begränsad. Vi kan därför, i dagsläget, inte beskriva steg 4 - 7 i processen då det arbetet helt enkelt inte har påbörjats än. Nedan följer alltså en redovisning av hur vi tillämpat steg 1 - 3:

Steg 1: Våra mål med spelupplevelsen, som vi kom fram till under möten med projektgruppen, var att det skulle vara pedagogiskt, på ett sådant sätt att det lär ut aritmetik, och underhållande på ett sådant sätt att spelaren vill spela om inte bara spelet utan även enskilda banor. Det var även viktigt för pedagogerna att spelet kan logga de olika

spelsessionerna på ett sätt som underlättar studier av hur enskilda spelare utvecklas. Eftersom det fanns ett ganska stort antal önskemål från pedagogernas sida som vi var tvungna att ta hänsyn till (som exempel på dessa kan nämnas att mönstren som visas måste röra på sig för att försvåra "räknande" och att det ska vara möjligt att expandera spelet med nya gränssnitt och högre summor) var det en utmaning att passa in allt. Som ett sätt att uppnå de här målen arbetades tre olika ramberättelser fram som sedan presenterades för projektgruppen. Värt att nämnas är att vi inte delat med oss av ett konceptdokument till potentiella spelare, vilket rekommenderas av Fullerton (2008). Detta eftersom vi redan hade ett koncept att bygga på samt att vi inte hade tillgång till försökspersoner. De tre ramberättelserna utvärderades istället inom projektgruppen. Dessa redovisas här nedan:

- *"Spelaren har ett "snöfort" i botten på skärmen. Mönstren är snöbollar som bombarderar fortet. När tillräckligt många har träffat fortet rasar det samman. Snöbollarna kan förstöras innan de når fortet genom att spelaren svarar rätt. Efter varje nivå kan fortet utökas/få ny inredning eller dylikt... Den här idén skulle kunna utvecklas till ett "turn-baserat" tvåspelarläge där det finns två fort. Den ena spelaren trycker då ned en fingerkombination och ett mönster av snöbollar motsvarande kombinationen kastas iväg mot det andra fortet. Motspelaren måste då trycka ner motsvarande antal fingrar för att försvara sitt fort.*
- *Karaktern i spelet håller i en glass och ska till cirkustältet som syns i fjärran. Mönstren svävar förbi i bakgrunden i exempelvis moln. För varje rätt svar kommer denna närmare tältet. Under tiden smälter glassen och spelnivån begränsas således till den tiden då glassen finns kvar. Denna variant har ett slut, och alltså även ett mål per nivå. Varje nivå kan ha en ny slutdestination.*
- *Djur som måste placeras i en linje för att ta sig över hinder i terrängen. En sorts Frogger-variant. Trycker man ner för få fingrar når man inte ända fram, trycker man ner för många kommer ett rovdjur och skrämmer bort dom. Målet är att ta sig någonstans. Måste finnas en morot för att klara uppgifterna snabbare."*

Den tredje ramberättelsen valdes ut av projektgruppen som en lämplig kandidat till vidareutveckling. Ramberättelsen är motiverande för barnen bland annat genom utmaningsaspekten *diskontinuerligt mål* från Malones (1980) ramverk. Efter en teknisk undersökning av Ipad:en, som syftade till att utröna huruvida det gick att registrera tio fingerned-sättningar samtidigt (se figur 3), gick vi vidare till steg 2.



Figur 3 Applikation som användes för att undersöka om Ipad:en klarar av att registrera tio fingernedsättningar samtidigt.

Steg 2: För att få en förståelse för vilka exponerings- och svarstider som skulle kunna vara rimliga i spelet gjordes en fysisk prototyp som bestod av en spelplan och mönster i papp (se figur 4). Vi skapade prototypen med hjälp av clipartbilder som vi skrev ut, klippade till och klistrade fast på bastantare papper. Vi skrev även ut alla mönster så att vi kunde testa oss fram till ett fungerande poängräkningssystem. En eftermiddag avsattes sedan för speltestning och vi kunde, baserat på den sessionen, skriva en detaljerad specifikation av spelprogressionen. Testningen gav en indikation på ungefär hur mycket tid man behöver på sig för att hinna "se" svaret, men också hur lång tid man behöver för att hinna räkna. Detta ger ett rimligt gränsvärde mellan första exponeringen och andra exponeringen på en nivå. Själva testningen gick till som så att en person "lekte dator" och förde mönstren nedåt i olika rörelsebanor varpå "spelaren" fick sätta ned fingrarna på samma sätt som skulle gjorts på Ipad:en.



Figur 4 Fysisk prototyp av spelet som användes för att bedöma kvaliteten av den tilltänkta spelprogressionen. Bland annat undersöktes vilka grafiska element som kunde öka spelkvaliteten samt på vilket sätt spelaren kunde belönas för sina framsteg.

Under testningen gick vi noga igenom spelförfarandet för att försöka hitta luckor i spellogiken. Vi fann att det inte är möjligt att visa uppgifterna (i mellansekvensen) med lång exponering med en färg och uppgifterna med kort exponering med en annan färg eftersom en uppgift med lång exponering kommer igen om den inte klaras vid första tillfället på en nivå. I samband med detta diskuterades poängsättning. Här föreslogs ett upplägg där spelaren skulle få mer poäng för en uppgift när exponeringstiden var kortare samt mer poäng ju högre målsiffran var. Vi diskuterade även en parameter som skulle bestämma hur många poäng spelaren förlorar per tickad sekund (5% per tidsenhet), samt om spelaren ska ha en tidsfrist på några sekunder innan poängen börjar räknas ner.

Det här arbetet resulterade i en utbrodering, ett förtydligande och en revidering av ramberättelsen och spelprogressionen som slutligen såg ut enligt följande:

“Spelkontext / Ramberättelse

Set-up

Spelet börjar med en välkomstskärm där namnet på spelet presenteras. Användaren kommer sedan att se en inloggningsskärm där det skall gå att välja mellan att logga in som spelare, lärare eller forskare. Lärare och forskare loggar in med hjälp av ett lösenord. Spelaren kommer till en skärm där denne ställs inför valet att antingen börja en ny session, fortsätta på en gammal eller välja en av de banor som redan spelats. Om spelaren väljer en ny session kommer en vy för att skapa en avatar att visas. Den här avataren skall ha ett namn, vara pojke eller flicka och det skall även finnas möjlighet att variera utseendet. När avataren är skapad, när spelaren väljer att fortsätta där den senast slutade eller när spelaren vill välja mellan de banor som redan spelats visas en “världskarta” där det fram-

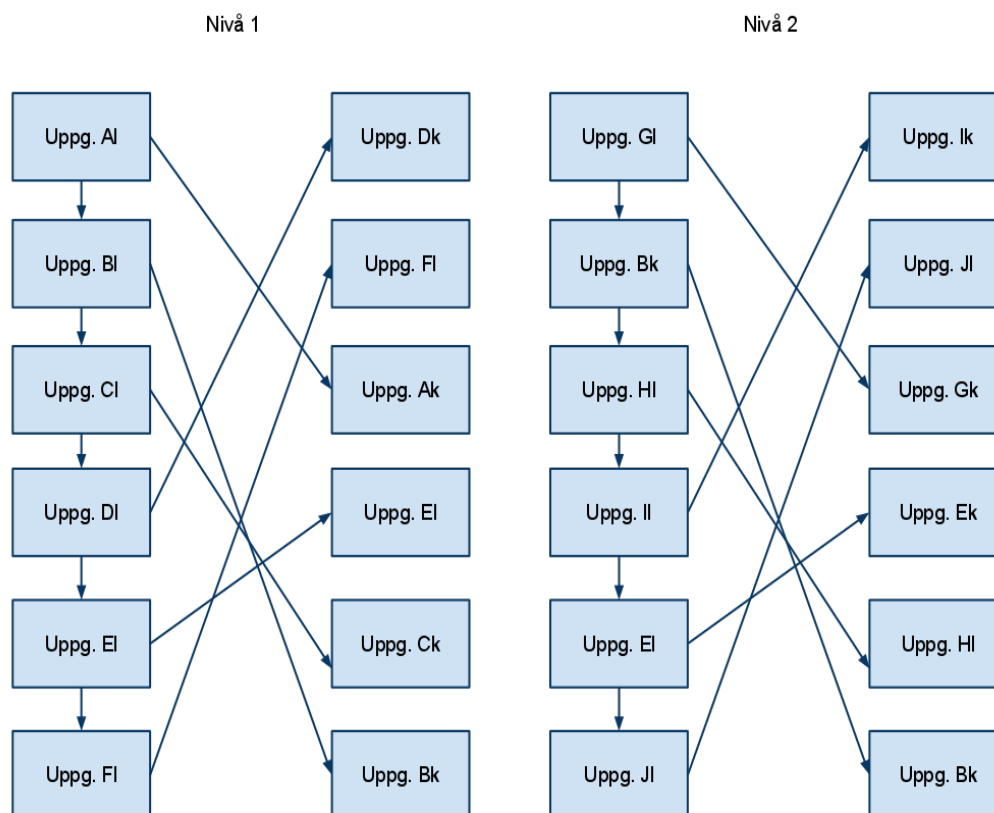
går var spelaren befinner sig i spelet genom att avataren står på banan där spelaren senast befann sig (i fallet av en ny session befinner sig avataren så klart på bana ett). Spelaren väljer att spela banan genom att trycka på den med fingret.

Spellogiken och ramberättelsen

Spelaren deltar i en löpartävling som går över en eländig terräng. Varje bana representerar olika etapper i loppet och skall ha olika karaktär. Det går att antingen klara av etappen och då få brons, silver eller guld (beroende på hur snabbt man klarat av de olika enskilda uppgifterna) eller att misslyckas med etappen och därmed inte komma i mål. Det finns sju olika banor och varje bana består av femton olika uppgifter som är fördefinierade men vars ordning slumpas ut. En uppgift går ut på att spelaren ställs inför ett hinder, t.ex. en flod, ett stup eller en öken. På skärmen visas även ett antal (mellan 1-10) djur som utgör en viss formation (antingen i ett ensamt kluster eller i kluster om två) och som rör sig över landskapet. Vid stupet kan djuren vara fåglar och vid floden flodhästar osv. Efter den här presentationen visas en svarsskärm. Om spelaren trycker ner lika många fingrar som det finns djur kommer djuren att forma en passage över hindret på nästa skärm. Om spelaren trycker ner för få fingrar kommer passagen inte att bli tillräckligt lång och om spelaren trycker ner för många fingrar kommer det på nästa skärm komma fram ett rovdjur och skrämna iväg de andra djuren så att det inte bildas någon passage alls. Om spelaren inte tryckt ner några fingrar alls försvinner alla djur på nästa skärm. Mellan varje uppgift räknas poäng ut beroende på hur snabbt man svarat och det här representeras av en mycket kort mellansekvens på några få sekunder där avataren springer vidare till nästa uppgift olika snabbt beroende på hur kort svarstiden var på den klarade uppgiften. Misslyckas spelaren med uppgiften visas en kort sekvens då avataren måste ta en omväg runt hindret. Proceduren upprepas tills alla uppgifterna är spelade och en totalpoäng räknas ut. Det här representeras i spelet genom att en prisceremoni äger rum där avataren antingen inte klarat banan och således inte kommer i mål eller att brons, silver eller guld delas ut på etappen beroende på hur bra det har gått. Samma procedur upprepas sedan på alla de sju olika banorna. Skulle spelaren "klara av" spelet blir det till slut en prisceremoni för hela loppet där avataren tilldelas brons, silver eller guld beroende på hur bra det gått på alla etapper sammanlagt.

Spelprogression

Spelet är organiserat med ett antal "spelnivåer". Varje spelnivå innehåller ett antal uppgifter (n , i dagsläget 15), som är fördefinierade. I varje spelnivåomgång presenteras N uppgifter två gånger. En gång med lång exponeringstid. Om svaret är rätt första gången uppgiften exponeras kommer den vid nästa exponeringstillfälle ha en kort exponeringstid. Om svaret var fel den första exponeringen kommer den vid nästa exponeringstillfälle återigen ha en lång exponeringstid. Ordningen mellan uppgifterna är slumpmässig. Om uppgiften blivit rätt eller fel besvarad skall sparas och "knytas" till spelarens profil så att uppgiften, om den dyker upp på en annan nivå, "vet" om den skall ha kort eller lång exponeringstid.

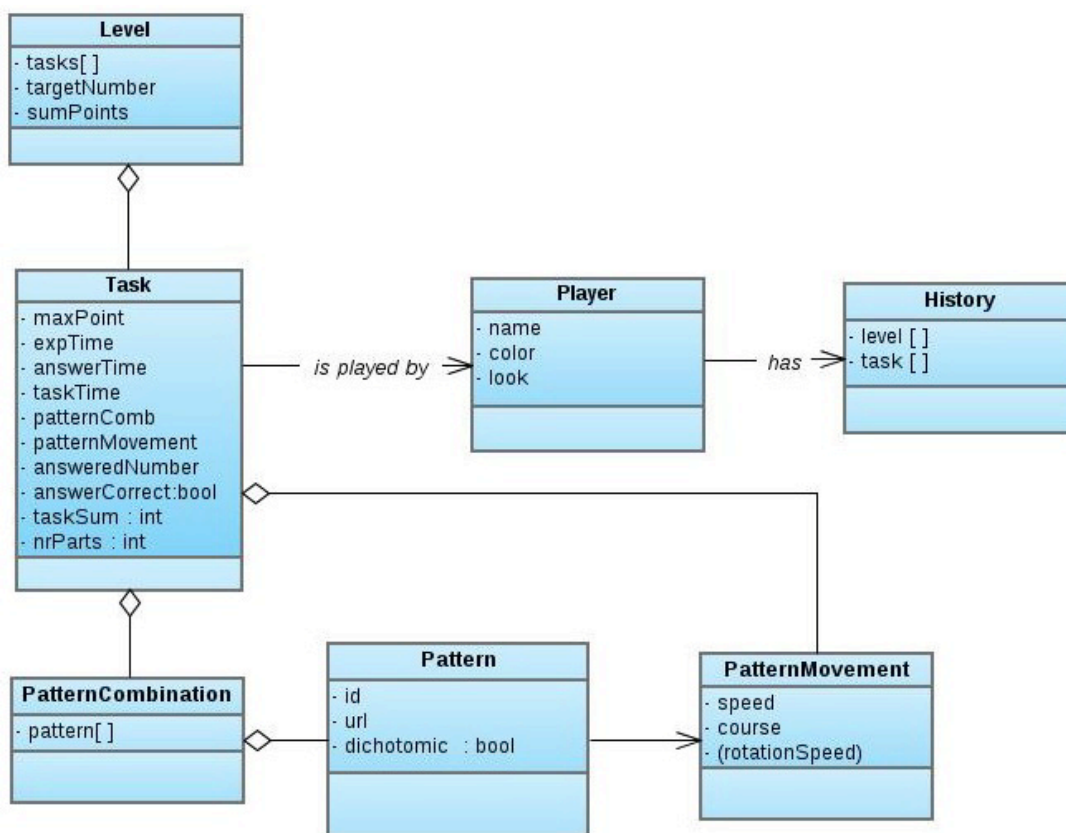


Figur 5 Bilden visar hur nivåerna är uppbyggda av uppgifter. Ordningen på uppgifterna slumpas fram och när en uppgift exponeras för andra gången kortas exponeringstiden om spelaren svarade rätt på uppgiften första gången. Svarade spelaren fel på uppgiften första gången förändras inte exponeringstiden för uppgiften nästa gång den visas. Uppgift A i det här exemplet går alltså från lång exponeringstid (l) till kort exponeringstid (k) under nivå 1. Spelaren har alltså svarat rätt på uppgift A första gången den exponerades. Uppgift E besvaras inte korrekt och exponeringstiden förändras därmed inte från första till andra exponeringen. När spelaren avancerar till nivå 2 återkommer uppgift B och eftersom spelaren svarade rätt på den uppgiften på den förra nivån visas den med kort exponeringstid (k).

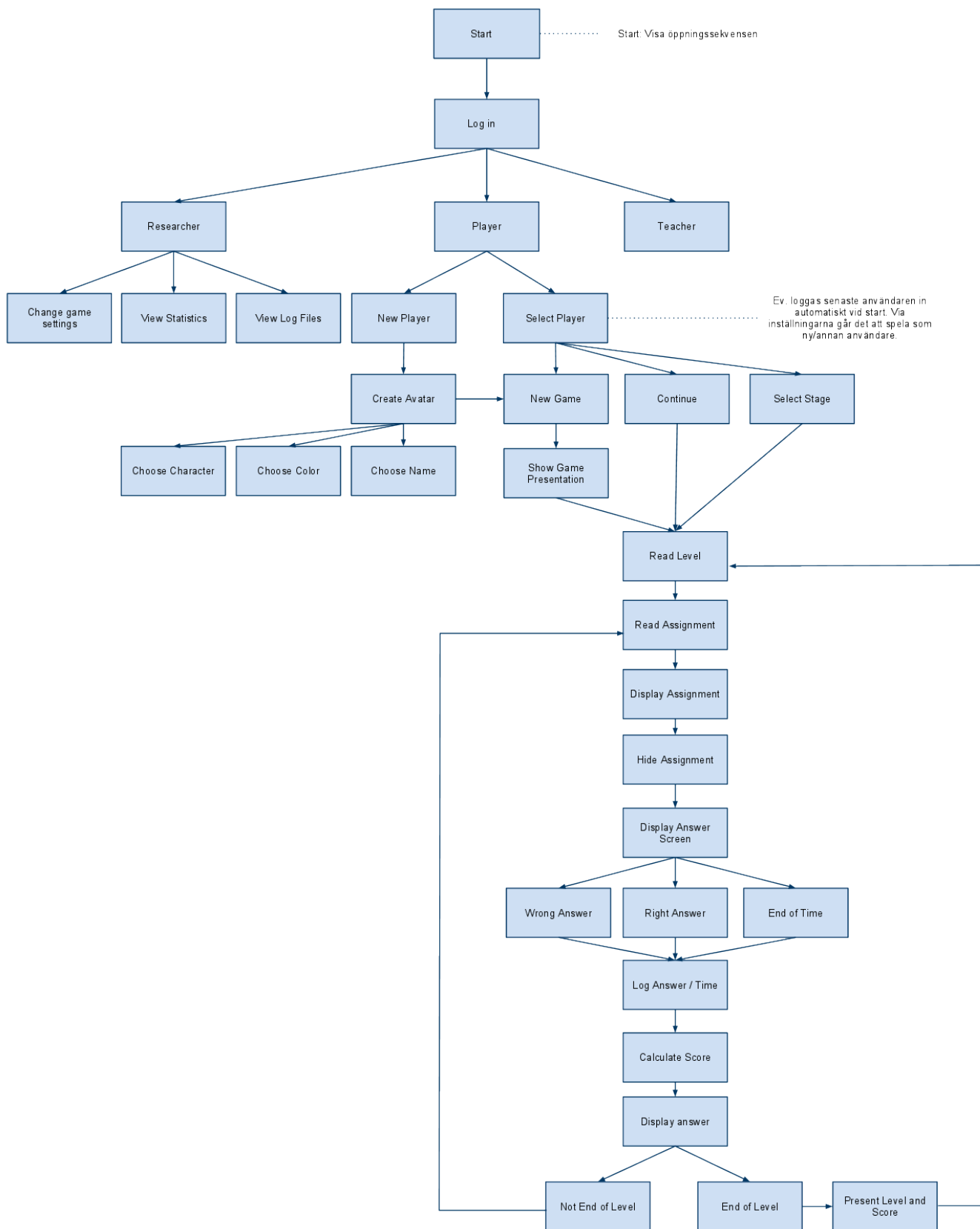
Kriteriet för att ha klarat av en nivå är att man "behärskar" de matematiska objekt man exponerats för. "Att behärska" beror på objektet och exponeringstid. Huvudkriterierna är att en viss procentsats av uppgifterna skall besvaras med rätt svar inom svarstiden. Dessutom skall spelaren kunna klara av uppgifterna olika bra beroende på hur snabbt spelaren svarar. En poängsumma knyts till uppgiftsobjektet. Den här poängsumman är olika stor beroende på uppgiftens svårighetsgrad. Ju längre tid spelaren tar på sig för att svara desto mer krymper poängen och poängen för uppgiften blir således mellanskillnaden maxpoäng - svarstid. Svarar spelaren rätt inom svarstiden är uppgiften dock avklarad på "grundnivå" och tiden kan således inte bli så lång att poängen försvinner. Exakt hur många poäng som behövs för att avancera till nästa nivå måste testas fram."

För att ytterligare förtydliga kravspecifikationen och översätta pedagogiska tankar och begrepp till ett språk som spelutvecklare förstår skapades ett klassdiagram (se figur 6) och ett flödesschema (se figur 7). Första versionen av flödesschemat tog sin form tidigt i projektet och baserades till stor del på samtal med projektgruppen och kan ses som en sammanfattning av den diskussion som varit med projektgruppen. Flödesschemat är i egentlig mening en beskrivning av spelets "game loop" och visar på ett konceptuellt plan

hur det är tänkt att spelet ska fungera. Klassdiagrammet kom till som ett önskemål från deltagare i projektgruppen i syfte att få en överblick över vilka klasser som skulle kunna vara aktuella i spelet och ge en indikation på ungefär hur mycket programmeringsjobb som skulle kunna krävas.



Figur 6 Föreslaget klassdiagram.



Figur 7 Flödesschema över spelprogressionen.

Steg 3: Vi skapade en presentation för spelutvecklingsföretaget. Meningen med presentationen var dels att reda ut vissa frågetecken och dels att ge spelutvecklingsföretaget ett bättre underlag för att kunna komma med en offert. Presentationen behandlade specifikationen, samt demonstrerande av hur spelet är tänkt att fungera. Som Fullerton

(2008) påpekar löper man här en risk att bli avvisad och det här fick vi erfara då vår ursprungliga idé visade sig vara för kostsam i dagsläget. Vi fick därför gå tillbaka till steg 1 och specificerade en mycket mer simpel variant av spelet.

5.1.3 Simplifiering av budgetska

Spelutvecklingsföretagets besked om att det spel vi tänkt oss skapa skulle bli dyrt att utveckla betydde att vi fick gå tillbaka till steg 1 i Fullertons (2008) designprocess igen. Det hade sedan tidigare funnits en överenskommelse mellan CoDAC-projektet och spelutvecklingsföretaget om att en portning av det gamla spelet till Ipad skulle kunna skapas på en månad och efter ett möte med projektgruppen bestämdes att arbetet skulle riktas mot det här målet istället. Specifikationerna som vi redan gjort är trots allt fortfarande aktuella någon gång i framtiden men inte i dagsläget.

Det första vi fick göra var att överge ramberättelsen. Specifikationerna blev därför mer fokuserade på att beskriva spelet skärm för skärm. Det här för att utvecklarna skulle kunna bedöma hur mycket grafik och hur mycket kod som behöver skapas. Spelprogressionen fick vara kvar med den lilla förändringen att spelaren nu ska ha ett visst antal "liv" eller försök på sig på varje nivå istället för det poängräkningssystem som tidigare arbetats fram. Det här framkom under steg 2 då vi visserligen inte arbetade med en fysisk prototyp men använde oss av en whiteboard.

I steg 3 skickade vi in de utarbetade specifikationerna till spelutvecklingsföretaget för bedömning och som ett underlag för att de skulle kunna lägga en offert. Offerten blev accepterad och arbetet kommer nu att fortskrida med skapandet av det här spelet grundat på våra specifikationer (specifikationerna återfinns i sin helhet i bilagan).

5.2 Resultat av transkriberingsanalysen

Nedan redovisas ett antal utdrag ur det transkriberade materialet. Det kan tilläggas att ljudinspelningarna uppgick till 3 timmar material varav allt transkriberades (53 sidor). Mötesdeltagarna benämns vid den titel de har samt det lärosäte de tillhör (Göteborgs universitet (GU) eller Högskolan Kristianstad (HKR)), om något alls.

För att få en uppfattning om hur vanligt det är med kommunikativa missförstånd och förvirring när det gäller termer sökte vi igenom den transkriberade texten efter stycken där medverkande talar förbi varandra. Det här är en av de aspekter som en systemvetare bör kunna vara behjälplig med. Nedanstående, där en diskussion om vad som skall loggas och i vilket format det skall loggas utspelar sig, är ett exempel på ett sådant tillfälle. Spelutvecklarna vill ha konkreta tekniska förslag som går att arbeta med och professorn tror att det är det som han förklarar. Slutligen mynnar diskussionen ut i att vad spelutvecklarna egentligen vill ha är ett format som loggfilen ska sparas i. Det uppfattas av systemvetarna som kommer med ett konkret förslag.

“Spelutvecklare: Det är lättare att ta fram en konkret idé och sedan ändra på den också diskutera enskilda delar så att man har en grund. För det är väldigt viktigt när vi har det här mötet sen att det liksom man har nånting som är konkret och så kan man ju liksom testa och justera efteråt.

...

Professor GU: Men så det ju också viktigt vad ni har för alltså ... [paus] ... alltså ni har naturligtvis ... saker som ni ... ja lösningar som är knutna till vissa idéer va? Det är liksom att pussla ihop det också så att ni får en massa onödigt merjobb va. Att man inte får något som väldigt konstigt...

Projektledare spelföretag: Något som tekniskt funkar.

Professor GU: Det måste tekniskt funka å... å så.

Spelutvecklare: Ja, precis.

Projektledare spelföretag: Men grunden finns ju. Den är ju ganska klockren egentligen. Det kommer funka alldeles utmärkt på Ipad:en som plattform.

Spelutvecklare: Ja, det är mer dom här detaljerna och sidogrejerna som det här med loggning och allt det där som vi måste snacka mer om.

Projektledare spelföretag: Ja, precis. Den behöver vi...

Spelutvecklare: ...bena ut ordentligt.

Projektledare spelföretag: Hur det skall funka egentligen.

Professor GU: Alltså grunden så är de så... Vi hade massa idéer om hur man skulle kunna displaya och såna saker... men i grunden så är det så det som finns i det gamla spelet är det som vi utgår ifrån. Det är att du måste få ut en fil som loggar den här... alltså varje uppgift i princip.

Projektledare spelföretag: Den loggar liksom just den användaren.

Professor GU: Ja. Och så. Och sen har vi pratat om det här tidigare att användaren måste liksom identifieras och man ska ha... Det har vi inte pratat om här. Det har inte ni gjort va? Alltså den måste ha nån sorts figur eller nån typ av... av...

Projektledare spelföretag: Eller identifikations...

Professor GU: Ja. Eller så. Och sen det här med när man loggar tangent... eller... när det gäller gränssnitt så kan man säga så här att då... och så ser det ut nu också... att man måste bygga det så att man har det här fingergränssnittet som är liksom poängen men att man också gör det så att du skulle kunna ha ett väldigt enkelt tangentbord...

Projektledare spelföretag: Går det att scanna in fingeravtrycken? Då blir dom unika...

[skratt]

Professor GU: Ja, men det var det som vi hade förra gången. Ja, men så dom där delarna är nog inte så svåra som... som... Men där vill man då ha loggat. Lite analogt som vi hade tidigare... Det är mer hur det ser ut va. Men det skall ut i en form som gör att det går att hantera sen...

Systemvetare: Ja, XML eller en vanlig textfil.

Spelutvecklare: Mmm.

Systemvetare: Alltså så att det går att köra... köra in i vad som helst.

Spelutvecklare: Precis. Ja, men det kan vi snacka mer om."

Ett relaterat område till det ovanstående är situationer där de medverkande inte förstår varandra, det vill säga när någon använder ett sätt att uttrycka sig på som medför svårigheter för förståelsen eller att en persons tacita kunskap gör att man förstår en sak på ett visst sätt. Nedanstående samtal handlar om att spelet innehåller en så kallad slumpfaktor, denna föreslås ett antal gånger under samtalet på ett sätt som visar att det inte riktigt gått in tidigare gånger. Vi kan också se att det här är ett exempel på begreppsförvirring då det finns oklarheter med vad "nivå" innebär i det här fallet. Systemvetaren bidrag är här att förtydliga vad diskussionen handlar om.

"Systemvetare: Och sen slumpa lite, i en annan procentsats, mönster och att nivån börjar på en viss tid, alla nivåer börjar på en viss tid, och sen går det fortare och fortare på varje nivå.

Universitetslektor GU: Ja, inom en nivå. För nu är de så att nivå hänger ihop med hur mycket tid man får för att svara, men man får också svårare mönster, som vi sa, egentligen behöver vi ge, egentligen samma tid i början av en, en nivå, för att då är det svårare så då måste man egentligen ha mycket tid och sen snabba upp det.

Spelutvecklare: Men när ni säger nivå, då är de inte oändlighetsläge utan man klarar den efter ett tag, att det är förutbestämt.

Universitetslektor GU: Ja, men sista nivån kan man egentligen inte klara. I det nuvarande spelet kan man inte klara sista nivån, för de är ingenting som är efter det.

Systemvetare: Men det vi har pratat om är att de skulle ha ett slut.

Universitetslektor GU: Ja, precis, de borde ha ett slut. Egentligen.

Spelutvecklare: Och då ska de antagligen vara nån slumpfaktor då när man spelar en bana att den, man får ungefär så här mycke går de ihop till totalt. Nu är de siffror som adderas upp till fem totalt men vilka siffror det är de slumpas kanske eller?

Universitetslektor GU: Ja, och vi sa också att de kanske inte är så att om man är på den nivå som representerar femman, att man inte får gå över femman, de får, ibland får man få en sexa för att se vad som kommer igen, men de är nivå fem som är fokus där.

Systemvetare: Man kanske kan slumpa lite tal bakåt också men att de ska vara övervägande... på den.

Spelutvecklare: Okej, för jag tänkte bara, till skillnad från att man har helt förutbestämda banor där man kan lära sig i vilken ordning de är för.

Systemvetare: Ja, precis, de ska vara ett slumpmoment.”

I rollen som systemvetare ingår att verka som medlare eller kommunikatör och som genom tvärvetenskaplig kunskap kan förtydliga situationer. Vi har därför sökt i transkriberingen eftersom situationer där medverkande gjort uttalanden som kan uppfattas som vaga eller otydliga och där vi har fått göra förtydligande. I nedanstående exempel pratar en medverkande om att spelet ska kunna göras i 3D, men får inte riktigt fram det han vill säga varpå en av oss försöker förtydliga uttalandet.

“Professor GU: Ja, jag säger det. Det var precis det jag var ute efter. Det kan man fundera när det gäller spel, alltså... I grunden tillhör detta inte det nuvarande spelet. Så det är liksom inte det vi kan börja med. Men å andra sidan kan fundera på när man designar det här att är det problematiskt att öppna upp detta för att ha andra typer av mönster av den här tiden. Eller, hur svårt är det, vad innebär det extra jobbet även om man inte gör själva så att säga expansionen? Och det är saker som man får ta ställning till när man designar det här. Är det här rimligt att...”

Systemvetare: Det blir ett helt annat grafiktänk.

Professor GU: Jo jo...”

Eftersom målet med samarbetet mellan spelföretag och pedagoger har varit att skapa ett spel som både är underhållande och pedagogiskt har det för vår roll som systemvetare varit viktigt att försöka hålla både de pedagogiska och de underhållande målen med spelupplevelsen i fokus. Vi har därför sökt i transkriberingen efter stycken där antingen pedagogik eller underhållning riskerar att försakats till förmån för det ena eller andra. Nedanstående diskussion om en eventuell förkortning av exponeringstiden samt en snabbare rörelse av de olika mönstren är ett exempel på ett sådant tillfälle:

“Universitetslektor GU: Fast om man kortar ner exponeringstiden så tar man också bort möjligheten att räkna.

Professor GU: Ja, jo...”

Systemvetare: Jag kan tänka mig att det här med att de rör sig fortare... Nu är inte jag någon expert på kognitivitet och sånt där, men alltså det känns som att mönstren kan bli blurrade och sen kan man inte se... Och då motverkar det ju meningen med det pedagogiska att man kan känna igen ett mönster som en siffra. Då kanske det är bättre att korta ner exponeringstiden istället.”

En annan viktig del i systemvetarens roll är att kunna förstå och beskriva idéer i tekniska termer. Det är därför intressant att uppmärksamma stycken där vi försöker ge förslag på hur en teoretisk idé skulle kunna realiseras i praktiken. De två nedanstående utdragen är exempel på sådana förslag. Det första är ett förslag på en lösning där man placerar kombinationer av mönster i en vektor, det vill säga en tvådimensionell talserie.

“Systemvetare: Fast man kan ju ha en tid knuten till varje mönster om det skulle vara så.

Universitetslektor GU: Ja, om vi säger att vissa mönster är lättare än andra...

Systemvetare: Du kan ju ha en vektor istället för en array...

Universitetslektor GU: Mhm.

Systemvetare: Så har du ju... [ritar] Du har ju ett mönster och sen så har du en tid på mönstret. Det är ju två dimensioner liksom.”

Det andra är en diskussion om hur det är möjligt att realisera spelprogressionen programmeringsmässigt.

“Professor GU: Då kan man faktiskt i den objektdefinitionen lägga in det här som är alltså... som du har noterat där alltså, det här med att om det är en mappning eller icke mappning.

Systemvetare: Men alltså rent programmeringsmässigt kan man ju ha en klass som heter typ Uppgift eller nåt sånt där. Och sen så har ju den klassen, då har ju den alltså [vissa egenskaper]. /.../ Till exempel då mönster. Mönster 1 då, och Mönster 2.”

För att visa på fördelen med spelreferenser som gemensamt språk har ett antal diskussioner identifierats i det transkriberade materialet. Det bidrar inte enbart till en ömsesidig förståelse bland mötesdeltagarna utan är också ett bra sätt att beskriva lösningar på olika problem som kan uppstå med spellogiken. Nedanstående citat är en diskussion som bidrog till att konkretisera idén om att det skall gå att klara av en nivå olika bra beroende på hur snabbt spelaren svarar. Som synes refereras det till Drop7, Tetris och Angry Birds.

“Universitetslektor HKR: Om man tar glassen som miljö, så skulle man kunna ha att man klarar av ett visst antal uppgifter på en viss tid. Alltså, glassen leder ju in tankarna på att den smälter på en viss tid.

Professor GU: Ja just det, men det här kopplar vi alltså... Det är ju spelidén som är viktig att fundera på. Vad har man för grundspelidé. Ungefär som Drop7 eller Tetris. Alltså man fyller ut...[ohörbart] Om man har... Ja, alltså det här går nog inte att diskutera i abstrakta termer bara utan att faktiskt koppla det till det.

Systemvetare: Ja. Om man kör på tidsspåret, alltså tidsspåret är ju, det är ju ett spår oavsett vad spelidén är. För då handlar det ju egentligen om... Alltså, antingen kan du klara av alla mönster, eller så kan du klara av alla mönster snabbt. Och då blir det ju extra bra. Det är väl det som är grundidén kanske.

Universitetslektor GU: Ja, precis. Men om man klarar av alla uppgifter tillräckligt snabbt innan tiden rinner ut, så måste man klara hela nivån egentligen. Sen om man gör det snabbare så blir det mer. Det är minimikravet egentligen, som i Angry Birds att man dödar alla grisar. Det måste man göra. Sen hur många poäng man får, det är... [diskussionen avbryts]”

Följande diskussion handlar om kriterier för att klara en nivå och la grunden till en lösning där vi använder medaljer som belöning istället för traditionella poäng. Här användes spelet “Angry Birds” som referens i projektgruppen för att försöka förklara en tanke.

“Universitetslektor GU: Vi pratade lite om att om man skulle kunna klara en nivå och få maxpoäng, man skulle kunna säga att måste komma ner till en viss svarstid för att klara nivån. Sen kan man bli snabbare och få extra poäng på nåt sätt eller... extra.

Universitetslektor HKR: De är lite grann som i de här “Angry Birds” va... “Angry Birds” så kan man liksom... där finns ett minimikrav på att man... för att man ska klara en nivå.

Universitetslektor GU: Ja, men minimikravet är egentligen att man svarar rätt, så är det också med “Angry Birds” egentligen.

Universitetslektor HKR: Ja... alltså... analogin håller inte helt... jag är fullt medveten om det, men där finns ändå möjligheten att... förbättra sig på varje nivå.

Spelutvecklare: Mmm, lätt att klara det och svårt att bemästra.”

Med det här avslutas resultatavsnittet. Vi har i resultatet sett exempel på olika situationer som uppstått i projektgruppen och sett hur systemvetarrollen har bidragit i dessa situationer. Vi går vidare med att i nästa avsnitt diskutera dessa resultat.

6 Diskussion

Det här stycket besvarar de två underfrågorna med utgångspunkt från resultatet för att sedan gå in på vad resultatet får för konsekvenser. Avsnittet slutar med förslag på fortsatt forskning.

Vilka specifika utmaningar, om några, finns med att skriva kravspecifikationer som ska kunna förstås och tolkas av både pedagoger och spelutvecklare?

Vad vi upplevt i vårt arbete med kravspecifikationer är å den ena sidan att de pedagoger som ingått i CoDAC-projektet inte har någon utarbetad metod för att utvinna krav utan snarare arbetar efter de pedagogiska principer som är viktiga. Å den andra sidan har det spelföretag som medverkat inte heller haft någon praxis eller standard för hur de vill ha krav specificerade. De här två faktorerna i kombination har gjort arbetet med en kravspecifikation som kan förstås och tolkas av både pedagoger och spelutvecklare utmanande. Samtidigt har det varit ett bra tillfälle för oss som systemvetare att testa en metod, Fullertons (2008) "playcentriska" designprocess, som förhoppningsvis bidrar till skapandet av ett spel som är både pedagogiskt och underhållande. Den största svårigheten för oss har varit att översätta de pedagogiska principer som ligger till grund för spelets ambition att lära ut aritmetik till fungerande spellogik som förhoppningsvis samtidigt är underhållande. I den utmaningen har vi tagit hjälp av fysiska prototyper, referenser till redan existerande spel, flödesdiagram, klassdiagram och Malones (1980) principer för pedagogiska spel. Det här angreppssättet har varit lyckat då vår slutliga specifikation nu bildar underlag för utvecklingen av det spel som vi i projektgruppen skapat.

Hur kan systemvetare underlätta kommunikationen mellan pedagoger och spelutvecklare?

Projektdeltagarnas olika bakgrunder har gjort det svårt att hitta ett gemensamt språk. Björk & Holopainens (2005) föreslagna ramverk har visat sig för komplext för att alla deltagare skall kunna tillgodogöra sig det på kort tid. Vår lösning på det här har varit att, med utgångspunkt från ovannämnda ramverk, transformera designmönster till verkliga exempel från spel. Istället för att tala om exempelvis "new abilities"-mönster har vi hänvisat till "Pac-Man" och så vidare. Som ett symptom på deltagarnas olika bakgrunder har ett tydligt inslag, som vi observerat i transkriberingsanalysen, varit begreppsförvirring. När någon pratar om t.ex. poäng eller nivå betyder det olika saker för olika mötesdeltagare (se "Resultat av transkriberingsanalysen"). Det här fenomenet bör kunna avhjälpas genom användandet av ett gemensamt ramverk antingen i form av Björks & Holopainens (2005) designmönster eller genom skapandet av ett refererbart spelbibliotek.

Bristen på ett gemensamt språk har även lett till att idéer och koncept inte kunnat förmedlas mellan de olika parterna på ett tillfredsställande vis. Vår roll som förmedlare har haft särskilt fokus då dessa idéer och koncept varit tvungna att tilldelas tekniska specifikationer för att kunna realiseras. Denna förmedlingsprocess är något som startat vid de första mötena med projektgruppen, och slutat vid färdigställandet av kravspecifikationen som överlämnades till spelutvecklingsföretaget. Under mötena framkom det ständigt mer eller mindre outtalade krav, och det var vi som systemvetare, eller närmare bestämt

kravanalytiker i det här fallet, som hade uppgiften att omvandla dessa krav till tekniskt genomförbara specifikationer. Parallellt med detta hade vi både spelets pedagogiska samt underhållande mål att ta i beaktande. Då antingen spelets pedagogiska eller underhållande egenskaper riskerats att försakats till förmån för det andra har vi bidragit med tekniska förslag som potentiellt kunnat avhjälpa problemet.

6.1 Resultatens konsekvenser

Trots att det finns snarlik forskning om kommunikation i den här typen av projekt har vi inte kunnat hitta någon forskning som specifikt fokuserar på systemvetarens roll. Vi anser därför att vi genom den här studien har bidragit till att definiera en del av systemvetarrollen. Vidare, som vi tidigare påpekat, finns det en uppsjö av metoder för att utvinna och kommunicera krav på programvaror. Majoriteten av dessa metoder är dock inte anpassade för det område vi valt att undersöka, det vill säga "playful learning"-spel. Vi anser därför att den här uppsatsen bidrar med ett förslag på en metod som visat sig fungera för att utvinna och kommunicera krav för ett pedagogiskt och underhållande spel. Påpekas bör dock att vi arbetat i ett specifikt projekt och underlaget, och därmed de eventuella slutsatser som kan dras är mycket begränsade. Spelet är heller inte färdigutvecklat än och därmed kan vi ännu inte dra några slutsatser om den färdiga produkten. Vad vi kan konstatera är däremot att vår specifikation har visat sig duglig nog för att ligga till grund för en accepterad offert och utvecklingen av mjukvaran. Vidare är en svaghet i studien att slutanvändaren inte har involverats. Användare är en viktig del i framtagandet av krav eftersom det är de som ska använda slutprodukten och bedömer huruvida produkten är användbar (Nuseibeh & Easterbrook, 2000). I vårt fall var slutanvändaren av spelet forskare och barn i förskoleåldern men eftersom det i vårt fall var viktigt att de pedagogiska principerna kom fram i spelet och låg därför prioritet på de krav som pedagogerna/forskarna ställde på produkten. På grund av detta och på grund av den begränsade tidsperiod vi hade för projektet involverades inte barnen i kravspecificeringen.

6.2 Förslag på fortsatt forskning

Som tidigare nämnts är de slutsatser som kan dras från resultaten begränsade och skulle behöva verifieras ytterligare, vidare har inte användare involverats i utvecklingsprocessen. Med det här som bakgrund skulle ett förslag på fortsatt forskning vara att dels testa föreslagen metod ytterligare samt att utvärdera om spelet uppfyllt kraven på att vara pedagogiskt och underhållande genom att låta barn utvärdera vad de gillar och inte gillar med produkten. Vidare, eftersom vi inte har haft möjlighet att involvera användare i designprocessen, skulle det också vara intressant att utveckla ett spel från grunden och i detta involvera barn som medutvecklare. Där skulle barnen till exempel kunna anta ett flertal olika roller, beroende på hur involverade barnen är tänkta att vara i framställningen av designkrav för det nya spelet (Markopoulos m.fl., 2008). Dessa roller kan vara användare, testare, informant och designpartner, där de senare nämnda har en större ansvars- och deltagande roll då de ärver rollegenskaper från föregående roller (Druin, 2002). Det här angreppssättet skulle då också kunna tillföra en ytterligare dimension i utmaningen med att kommunicera krav.

Det här avsnittet har behandlat de två underfrågorna, vad resultatet har för konsekvenser och förslag på fortsatt forskning. Nästa stycke summerar studiens resultat.

7 Slutsats

I det här avsnittet summeras studien genom att ge förslag på metoder som kan tillämpas vid utvecklingen av ett pedagogiskt och underhållande spel, och samtidigt besvara studiens huvudfrågeställning.

Hur kan systemvetare stödja design- och utvecklingsprocessen av ett pedagogiskt och underhållande spel?

Vi har i den här studien kunnat visa på att systemvetaren kan stödja design- och utvecklingsprocessen av ett pedagogiskt och underhållande spel på många olika sätt. Systemvetare har ett brett utbud av modelleringsverktyg och designmetoder att arbeta med för att underlätta både kravutvinning och kommunikationen av dessa krav i form av kravspecifikationer. De flesta av dessa är inte direkt anpassade för utveckling av pedagogiska spel men genom att hålla fokus på de mål med spelupplevelsen som satts upp och genom att ta hänsyn till de olika deltagarnas respektive referensramar har en kombination av olika metoder föreslagits. De här metoderna är: Fullertons (2008) "playcentriska" designprocess, flödesscheman, klassdiagram, prototypande och "mock-ups". För att undvika begreppsförvirring bland deltagare med olika bakgrund föreslås användning av ett gemensamt referensramverk, till exempel Björks & Holopainens (2005) designmönster eller ett spelbibliotek. Vi har slutligen kunnat se att systemvetaren även kan verka som en agent vid transformerandet av idéer till tekniska lösningar och vid eventuella intressekonflikter där fokus på underhållning eller pedagogik gör att den ena aspekten riskerar att försakats till förmån för den andra.

Studiens begränsningar har dels legat i omfattningen och att det därmed inte går att generalisera slutsatserna från resultatet och dels i att barn i egenskap av slutanvändare inte har involverats i designen och utvärderingen av spelet. Vi har därför gett som förslag på fortsatt forskning att metoden testas genom ytterligare studier samt att låta barn utvärdera spelet för att utröna om det uppfyllt kraven på att vara pedagogiskt och underhållande. Det här skulle sammantaget bli ett steg i att försöka generalisera den här studiens slutsatser.

8 Referenser

Bandura, A. & Schunk, D (1980) *Cultivating competence, self-efficacy, and intrinsic interest through proximal self-motivation*. Unpublished manuscript, Stanford, California: Stanford University.

Barendregt, W., Emanuelsson, J. & Lindström, B. (2009) The number practise game: Visual recognition of numerosity and children's early development in mathematics. *SAGSET 39th Conference Proceedings*.

Björk, S. & Holopainen, J. (2005) *Patterns in game design*. Boston: Charles River Media, INC.

Coughlan, J. & Macredie, R.D. (2002) Effective communication in requirements elicitation: A comparison of methodologies. *Requirements engineering*, 7(2), 47-60.

Crowley, D. & Heyer, P. (2006) *Communication in history: Technology, culture, society* (5:e upplagan). Boston: Allyn & Bacon.

Druin, A. (2002) The role of children in the design of new technology. *Behaviour and information technology*, 21(1), 1-25.

Egenfeldt-Nielsen, S. (2005) *Making sweet music: The educational use of computer games*. Doktorsavhandling. IT University of Copenhagen.

Fullerton, T. (2008) *Game design workshop - A playcentric approach to creating innovative games* (2:a upplagan). Burlington: Elsevier

Hussain, H., Embi, Z. C., & Hashim, S. (2003) A conceptualized framework for edutainment. *Informing science: InSite – Where parallels intersect*, 1077-1083.

Jacobsen, D. I., Thorsvik, J. (2008) *Hur moderna organisationer fungerar*. Lund: Studentlitteratur.

Larman, C. (2004) *Applying UML and patterns: An introduction to object-oriented analysis and design and the Unified Process* (3:e upplagan). Upper Saddle River: Prentice Hall.

Leonard, D. & Sensiper, S. (1998) The role of tacit knowledge in group innovation. *California Management Review*, 40(3), 112-133.

Lepper, M. R., Cordova, D. I. (1992) A desire to be taught: Instructional consequences of intrinsic motivation. *Motivation and Emotion*, 16(3).

Lindström, B. (2009) *Vilkor och redskap för utveckling av aritmetiska kompetenser (Bilaga A)*. Institutionen för pedagogik och didaktik vid Göteborgs universitet och Sektionen för Lärarutbildning vid Högskolan Kristianstad.

- Lineducation (2010) *Göteborg Stad skriver ramavtal med Lin Education*. (Elektronisk)
Tillgänglig: <http://www.lineducation.se/default.asp?sid=30> (2011-05-12)
- Lundeberg, J. & Tengborn, C. (2000) *Interaktiva läromedel i ämnet matematik – Finns det plats för dem i skolan?* Kungliga Tekniska Högskolan. Avdelningen för medieteknik och grafisk produktion.
- Malone, T. W. (1980) What makes things fun to learn? A study of intrinsically motivating computer games. *Technical Report CIS-7 (SSL-80-11)*. Palo Alto, California: Xerox Palo Alto Research Center.
- Markopoulos, P., Read, J., MacFarlane, S. & Höysniemi, J. (2008) *Evaluating children's interactive products* (s. 45-46). Burlington: Morgan Kaufmann Publishers.
- Mathiassen, L., Munk-Madsen, A., Nielsen, P. A. & Stage J (2001) *Objektorienterad analys och design*. Malmö: Studentlitteratur.
- Motschnig-Pitrik, R. & Kaasbøll, J. (1999) Part-whole relationship categories and their application in object-oriented analysis. *IEEE Transactions on Knowledge and Data Engineering archive*, 11(5).
- Nonaka, I & Takeuchi, H. (1995) *The knowledge creating company* (s. 51). New York, NY: Oxford University Press.
- Nuseibeh, B. & Easterbrook, S (2000) Requirements engineering: A roadmap. *Future of Software Engineering*.
- Peters, J. D (1999) *Speaking into the air: A history of the idea of communication*. University of Chicago Press.
- Polanyi, M. (1966) *The tacit dimension* (s. 4). New York, NY: Doubleday.
- Preece, J., Rogers, Y. & Sharp, H. (2007) *Interaction design – Beyond human-computer interaction*. Chichester: John Wiley & Sons, Ltd.
- Resnick, M. (2004) Edutainment? No thanks. I prefer playful learning. *Associazione Civita Report on Edutainment*.
- SACO (2010) *Systemvetare / Om yrket* (Elektronisk). Tillgänglig:
<http://www.saco.se/templates/Ocupation.aspx?id=3954&epslanguage=SV> (2011-05-12)
- Solymar, L. (1999) *Getting the message - A history of communications*. New York: Oxford University Press Inc.
- Spender, J.C (1996) Competitive advantage from tacit knowledge? Unpacking the concept and its strategic implications. I Bertrand Mosingeon och Amy Edmondson, ed.

1996. *Organizational learning and competitive advantage*. London: Sage Publications, 56-73.

Wallén, G. (1996) *Vetenskapsteori och forskningsmetodik*. Lund: Studentlitteratur.

9 Bilaga

Följande bilaga är den slutliga revideringen av kravspecifikationen som låg till underlag för offerten som sedermera accepterades inför utvecklingen av spelet.

Specifikation

I denna specifikation är spelet egentligen inte mer än en portning av det gamla spelet så när som på några egenskaper som exempelvis spelprogression. Några egenskaper har även tagits bort från det gamla spelet.

Pedagogisk idé

Barn skall utveckla en förmåga att "se" tal som del-helhetsrelationer, primärt en helhet som konstrueras av två delar med olika modalitet (ex. $7=6+1=5+2=4+3$). Att förstå talet sju handlar om att se alla möjliga sätt att konstruera talet sju och att relatera till det. Att förstå ett tal handlar således om att förstå ett tal i ett "talsammanhang" (system av tal), som en relationell konstruktion. Räknande är ett led i utveckling av talbegrepp, men bör undertryckas. Barn skall uppmuntras att "se" möjliga konstruktioner och hindras från att räkna för att utveckla aritmetiska förmågor på ett bra sätt.

Fokus på talbegrepp inom talområdet 1-10, men detta bör/måste kunna utökas.

Barnen skall svara genom "fingertal" och inte genom att använda talord/symboler. Grundidén med detta är det simultana erfandet av talbegrepp i flera modaliteter.

Fingrarna används för att representera tal (som avbildning), men primärt inte som symboler.

Variationsteoretiska grundprinciper - variation i vissa dimensioner

- a. antal
- b. del-helhetsrelationer (givet antal) - invarians över delar
- c. modalitet (visuell - taktill) - invarians över modalitet
- d. rörelse i rummet - invarians i rummet

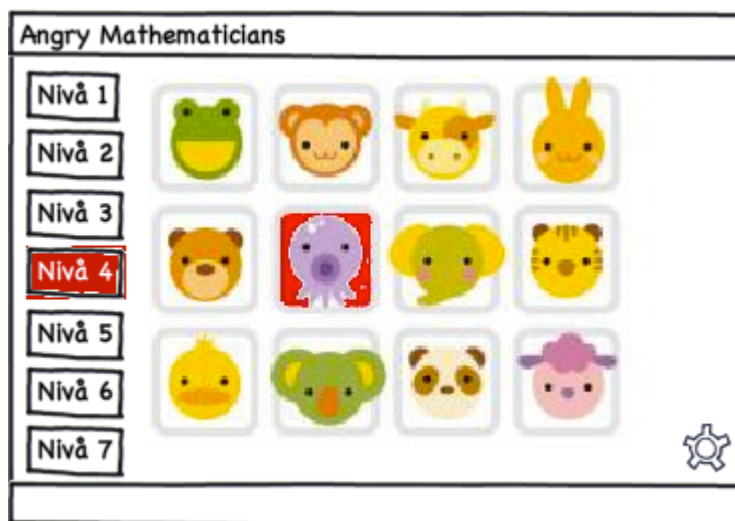
Bakgrunder / Nivåer

Spelet består av sju olika nivåer. I dagsläget behövs endast en bakgrund till alla sju olika nivåer i spelet. Möjligtvis kan den varieras i form av t. ex. färg. Det är viktigt att bakgrunden inte stjäl uppmärksamheten från uppgifterna som skall besvaras. Den får därför inte vara för plottrig eller för intressant.

Skärmar

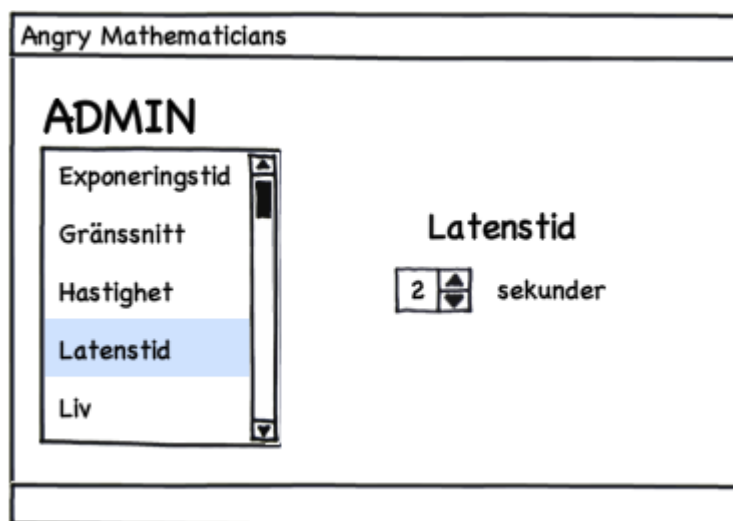
Startskärm: Den första skärm som visas när man startar spelet. Här skall spelaren kunna välja sin egen "ikon". Ikonerna är fördefinierade och en unik spelaridentifikation knyts

till ikonerna per barngrupp. På det sättet kan de återvändas i andra barngrupper. Spelaren skall även kunna välja nivå upp till den nivån som senast spelades. Avslutade spelaren sin förra session på nivå fyra kan hon alltså välja att börja på nivå ett till fyra. Default-nivån är den nivå som senast spelades av spelaren i fråga. Slutligen skall det finnas en ikon för att komma till administratörsinställningarna.



Figur 1 Spelaren väljer sin avatar och ser därefter vilken nivå den påbörjade senast.

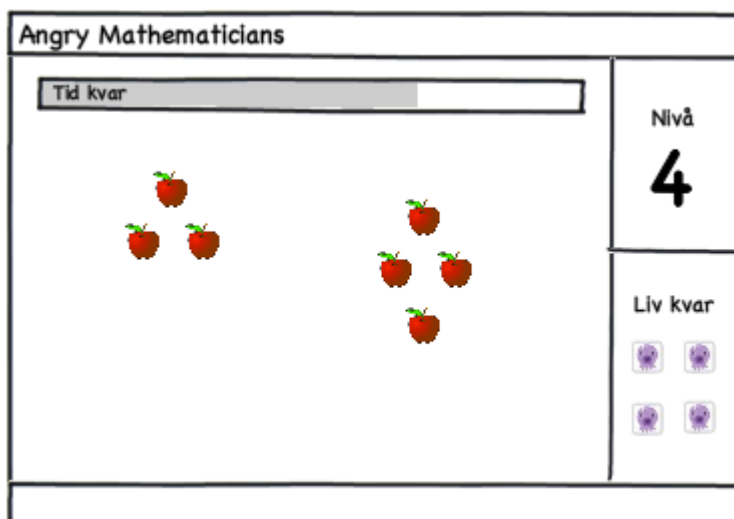
Administratörsinställningar: Den här skärmen skall vara lösenordsskyddad och består av ett antal inställningar (se rubriken Inställningar).



Figur 2 Lösenordsskyddade administratörsinställningar.

Spelskärmen: På spelskärmen skall, förutom de två olika mängderna i form av mönster, visas antal "liv" (representerade av spelarens ikon) spelaren för tillfället har till förfogande på nivån, vilken nivå spelaren befinner sig på för tillfället samt en mätare som visar hur mycket tid spelaren har kvar innan pågående uppgift måste besvaras. Efter att varje uppgift besvarats skall spelaren ges visuell feedback om ifall uppgiften besvarades korrekt (se figur 4), felaktigt (se figur 5) eller om tiden tog slut (se figur 6). Om uppgiften besvarats felaktigt försvinner ett "liv", om tiden tog slut försvinner ett halvt "liv". För att klara av nivån krävs att spelaren inte har slut på "liv" när alla uppgifter spelats igenom. Spelaren skall kunna avbryta sin spelomgång genom att trycka på Ipad:ens "hemknapp"

och då skall en bekräfta-ruta visas där spelaren måste bekräfta om hon verkligen vill avsluta spelomgången. När spelaren sätter ner sina fingrar på skärmen skall det ges visuell feedback på detta genom att cirklar visas där fingrarna befinner sig.



Figur 3 Spelskärmen innehåller information om vilken nivå spelaren befinner sig på, hur många liv den har kvar, hur mycket svarstid det är kvar (förloppsmätare som töms) samt aktuellt mönster.



Figur 4 Pop-up som visas då spelaren angett rätt svar.



Figur 5 Pop-up som visas då spelaren angett fel svar.



Figur 6 Pop-up som visas då svarstiden gått ut.

Nivå avklarad: När spelaren spelat igenom alla uppgifter på en nivå och inte har slut på "liv" är nivån avklarad och en skärm med en målflagga (eller liknande) visas. På skärmen visas också hur många "liv" spelaren lyckades behålla och uppmuntrande audio-feedback skall ges. Efter den här skärmen avancerar spelaren till nästa nivå.



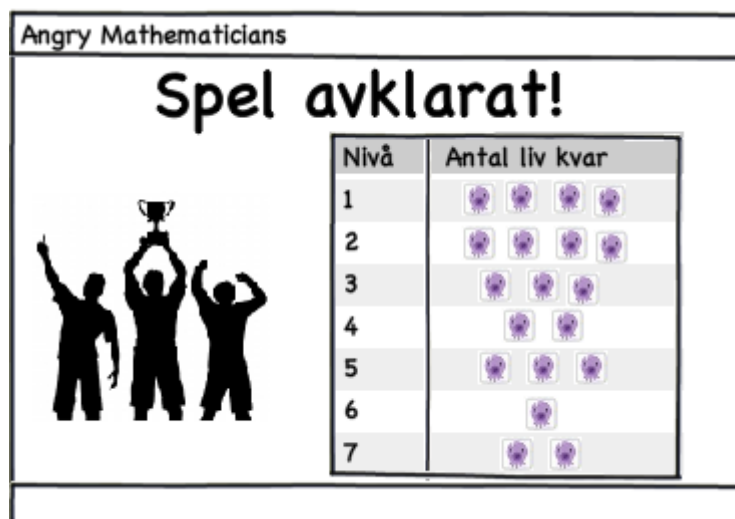
Figur 7 Skärm som visas då spelaren klarat av en nivå med ett eller flera liv i behåll.

Nivå ej avklarad: När spelaren spelat igenom alla uppgifter på en nivå men har slut på "liv" har nivån inte klarats av. En skärm med en uppmaning att försöka igen (helst i form av en bild) skall då visas och spelaren ges audio-feedback. Efter den här skärmen börjar spelaren om på samma nivå igen.



Figur 8 Skärm som visas då spelaren gått miste om samtliga liv under en nivå.

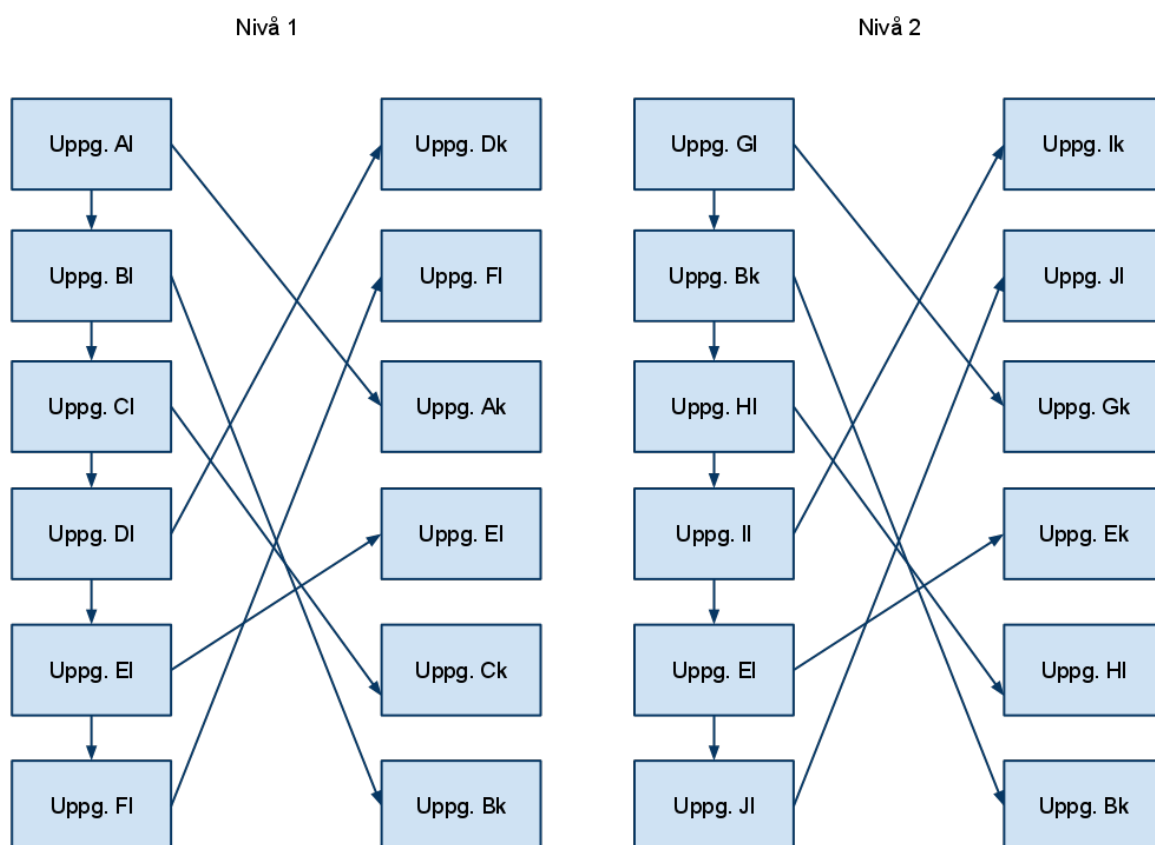
Spelet avslarat: Om spelaren lyckas klara av alla nivåer visas en skärm där en “festlig” bild och hur många “liv” som spelaren lyckats behålla på varje nivå presenteras. En glad “trudelutt” skall även spelas.



Figur 9 Skärm som visas då samtliga nivåer är avslarade med ett eller flera liv i behåll.

Spelprogression

Spelet är organiserat med ett antal "spelnivåer". Varje spelnivå innehåller ett antal uppgifter (n , i dagsläget 15), som är fördefinierade. I varje spelnivåomgång presenteras N uppgifter två gånger. En gång med lång exponeringstid. Om svaret är rätt första gången uppgiften exponeras kommer den vid nästa exponeringstillfälle ha en kort exponeringstid. Om svaret var fel den första exponeringen kommer den vid nästa exponeringstillfälle återigen ha en lång exponeringstid. Ordningen mellan uppgifterna är slumpmässig. Om uppgiften blivit rätt eller fel besvarad skall sparas och "knytas" till spelarens profil så att uppgiften, om den dyker upp på en annan nivå, "vet" om den skall ha kort eller lång exponeringstid.



Figur 10 Bilden visar hur nivåerna är uppbyggda av uppgifter. Ordningen på uppgifterna slumpas fram och när en uppgift exponeras för andra gången kortas exponeringstiden om spelaren svarade rätt på uppgiften första gången. Svarade spelaren fel på uppgiften första gången förändras inte exponeringstiden för uppgiften nästa gång den visas. Uppgift A i det här exemplet går alltså från lång exponeringstid (l) till kort exponeringstid (k) under nivå 1. Spelaren har alltså svarat rätt på uppgift A första gången den exponerades. Uppgift E besvaras inte korrekt och exponeringstiden förändras därmed inte från första till andra exponeringen. När spelaren avancerar till nivå 2 återkommer uppgift B och eftersom spelaren svarade rätt på den uppgiften på den förra nivån visas den med kort exponeringstid (k).

Kriteriet för att ha klarat av en nivå är att man "behärskar" de matematiska objekt man exponerats för. Att behärska beror på objektet och exponeringstid. Huvudkriterierna är att en viss procentsats av uppgifterna skall besvaras med rätt svar inom svarstiden. Dessutom ska det antal liv spelaren har vara knutet till spelarens svar. Svarar spelaren

fel förlorar den ett liv, och om tiden går ut förloras ett halvt liv. När spelesessionen är över presenteras en tabell över de antal liv spelaren lyckats behålla efter varje nivå.

Loggning av spelprogression

Vid virtuell tangentbordsinmatning inmatning loggas intryckt symboltangent.

Vid fri fingernedsättning loggas fingernedsättningen i realtid, med angivande av positioner på skärmen.

Metoder för input-registrering skall kunna ta emot data från olika "källor" såsom touch-skärm, "tangentalternativet" samt loggfiler. På så sätt går det att mata in en loggfil istället för att spela och sessionen i fråga "spelas upp".

Inställningar

En administratör ska kunna tillgå spelinställningar genom att ange ett lösenord. Följande inställningar ska inkluderas:

- **Sessionsidentifikation:** Tresiffrigt, inställningsbart löpnummer
- **Spelaravatar:** Visar vilken bild (avatar) som tillhör vilken spelare
- **Gränssnitt (input):** Här anges hur inmatningen ska gå till (fingeranvändning/touch eller virtuellt tangentbord)
- **Loggning:** Här kontrolleras om loggningsfunktionen ska vara på eller av.
- **Ny session:** Ny session skapas och gamla inställningar nollställs
- **Latenstid:** Tid som förflyter innan svaret registreras (bör kunna varieras av experimentella skäl)
- **Hastighet:** Mönstrens hastighet, låg/medel/hög (global inställning)
- **Exponeringstid:** Kort eller lång exponering av mönstren (global inställning)
- **Svarstid:** Tid spelaren har på sig att svara (global inställning)
- **Liv:** Antalet liv spelaren har per nivå

Saker som behöver utvärderas

Nedan listat element som bör gå genom en utvärdering innan de kan tilldelas ett slutgiltigt värde, och som tills vidare tilldelas ett temporärt värde.

- Default-nivån spelaren börjar på vid återupptaget spel. Denna nivå är förslagsvis den senast påbörjade nivån i historiken.
- Latenstid
- Antalet "liv" en spelare har per nivå (kan variera mellan nivåer).
- Hur länge fingeravtrycken hänger kvar efter spelaren har lyft sina händer.

Inmatning och feedback

- Spelare placerar fingrarna på Ipad:en med fri fingersättning utan restriktioner förutom på den del av skärmen där menyn är. För närvarande nöjer vi oss med högsta nummer 10, därför behövs ingen lösning på hur fingersättningen skulle fungera med högre nummer.

- (Senare version) Virtuellt tangentbord längst ned på skärmen.

Registrering av svar

Eventuellt kan det vara en idé att inte begränsa inmatningstiden genom att registrera svaret i samma ögonblick som fingrarna rör vid skärmen. En stunds betänkningsstid kan erbjudas genom att exponera en förloppsmätare av något slag då minst ett finger trycks ned. När förloppsmätaren har nått sitt slut registreras svaret. Förloppsmätarens livslängd blir därmed lika med latenstiden.

Förslag på förloppsmätare, eller snarare visuell återkoppling som indikerar latenstiden, kan vara att fingeravtrycken (vilka representerar varje nedtryckning) skiftar i färg och slutligen blir gröna och låsta på skärmen. Detta indikerar att svaret är registrerat, och kan förstärkas med annan visuell, och eventuellt även auditiv, återkoppling.

Latenstid är den tid som förflyter från att ett antal fingrar hålls mot skärmen tills programmet registrerar ett svar. Förslagsvis sätts latenstiden till 2 sekunder. Tiden mäts från en punkt där ett visst mönster är registrerat och ingen förändring sker.

Feedback

- Rätt svar - Smiley och ljud
- Felaktigt svar - Frowney och ljud
- Uppgiftstiden går ut - Till exempel timglas som rinner ut eller en klocka. Detta ackompanjeras av ljud.
- Spelaren har ett antal "liv", dessa representeras av ikonerna spelaren valt (fem liv = fem ikoner). När spelaren svarar fel försvinner ett "liv". Om tiden går ut försvinner ett halvt liv.
- Någon form av fingeravtryck där fingrarna sätts ned ska visas för att visa att skärmen reagerar.
- När svaret är registrerat och spelaren lyfter sina händer är fingeravtrycken kvar i 1 sekund (tid kan behöva testas).
- Information om när svarstiden går ut ges i form av en progressbar.
- Nivån fortsätter tills spelaren trycker avbryt

Loggning

Från v1

Vi behöver en händelselogg för den fria fingernedsättningen. Om möjligt önskas en händelsevektor där varje tangentnedtryckning registreras med tid och koordinater, och där uppsläpp av ett finger gör en tidigare nedtryckning falsk. I sådana fall behövs också en rutin som avläser antalet sanna nedtryckningar.

Utöver detta önskas loggning av svarstid samt latenstid för varje uppgift.

Händelselogg: Kontinuerlig loggning av fingernedsättning (upp ner) , med positioneringskoordinater, tid och svarssumma.

Loggningmodulen skickar (?) svarssumma och ts till SM.

Loggar sparas per uppgift.

Individ/uppgiftsnummer....

Fil med samtliga uppfitsloggar för varje omgång sparas ner....

Inledning och avslut för varje uppgift....

Vilka objekt visas?

Små cirklar, spelar inte så stor roll. Måste hänga ihop med resten av designen, men måste vara tydliga.

Spelloggning

| Id | Spelare | Play session | Löpnummer | Nivå | Expnr |
|-----|---------|--------------|-----------|------|-------|
| 000 | 07 | 0002 | 0022 | 3 | 0034 |

| Datum | Tid |
|------------|----------|
| 2008:05:26 | 13:34:05 |

| Exptid | Uppgiftstid | Reaktionstid? |
|--------|-------------|---------------|
| 11.00 | 20.00 | |

| MönsterID | Delsumma1 | Delsumma2 | Summa |
|-----------|-----------|-----------|-------|
| 06031032 | 3 | 3 | 6 |

| Svar | Rätt | SvarsTid | Antal rätt | Antal fel |
|------|------|----------|------------|-----------|
| 04 | 1 | 15.23 | 12 | 3 |

Tangentloggning

Vi vill veta för varje uppgift:

| Id | Spelare | Play session | Löpnummer | Nivå | Expnr |
|-----|---------|--------------|-----------|------|-------|
| 000 | 07 | 0002 | 0022 | 3 | 0034 |

Slutsvaret (hur många)

x,y koordinater för varje finger i slutresultatet

händelselogg: x,y koordinater för varje finger som sätts ner eller tas bort.