VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Sampo Kaleva Tapani Rintanen

Spelutveckling i Flash As3

Företagsekonomi och turism
2011

FÖRORD

Detta lärdomsprov har skrivits under våren 2011 i syfte att erhålla tradenomexamen inom utbildningsprogrammet för informationsbehandling vid Vasa yrkeshögskola. Börje Harju har fungerat som handledare.

Vasa 24 maj 2011

Sampo Rintanen

VASA YRKESHÖGSKOLA

Utbildningsprogrammet för informationsbehandling

ABSTRAKT

Författare	Sampo Rintanen
Lärdomsprovets titel	Spelutveckling i Flash As3
År	2011
Språk	svenska
Sidantal	43 + 1bilaga
Handledare	Börje Harju

Detta lärdomsprov behandlar spelutveckling i Flash med hjälp av språket ActionScript 3.0 och genomförandet av spelprojektet Stundars tidsresa, ett lärorikt spel ämnat för barn beställt av Stundars museum.

Lärdomsprovet går i teorin igenom verktyg och allmänna spelutvecklingsmetoder i Flash och ActionScript 3.0. Den praktiska delen av lärdomsprovet går igenom planeringen och genomförandet av spelprojektet Stundars tidsresa.

Resultatet av detta projekt är en fin och välfungerande spelhelhet som tagits i bruk av beställaren. I det sista kapitlet sammanfattas slutresultatet.

Ämnesord Spelutveckling, Flash, actionscript 3

VAASAN AMMATTIKORKEAKOULU

UNIVERSITY OF APPLIED SCIENCES

Utbildningsprogrammet för informationsbehandling

ABSTRACT

Author	Sampo Rintanen
Title	Game development in Flash with As3
Year	2011
Language	Swedish
Pages	43 + 1 appendix
Name of Supervisor	Börje Harju

This thesis describes game development in Flash using the object-oriented language ActionScript 3.0 and the realization of the game project "Stundars tidsresa", a instructive and fun game for children commissioned by the Stundars museum.

The thesis goes through tools and common game development methods in theory using Flash and ActionScript 3.0. The practical part of the thesis looks at the planning and the realization of the game project "Stundars tidsresa".

The result of this work is a fully-functional and beautiful game that has been taken into use by the client. In conclusion the last chapter summarizes the outcome of this project.

Keywords Game development, Flash, Actionscript 3

INNEHÅLL

ORDFÖRKLARINGAR

Flash

Ett redskap för visuell design och multimedia som integrerar grafik, video, audio, animation och interaktion. I Adobe Flash skapar utvecklare interaktivt innehåll genom att kombinera ActionScript kod med animation, manuellt skapat inehåll och inbyggda tillgångar. (Moock 2007, 25.)

ActionScript

ActionScript 3.0 är ett objektorienterat programmeringsspråk med en syntax lik Java och C#, syntaxen torde därför vara bekant för erfarna programmerare. Kärnan av språket ActionScript 3.0 baserar sig på ECMAScript 4 och i framtiden förväntas ActionScript helt anpassa sig efter det. Det populära webbläsarspråket Javascript är även som ActionScript baserat på ECMAscript. (Moock 2007, 22.)

Objekt orienterad programmering (OOP)

Programmerare som vill ha mera modularitet och återanvändning av kod lutar sig ofta mot objektorienterad programmering. Objektorienterade programmerings-språk skapar program som är en samling objekt. Objekten är individuella instanser av klasser.

Klasser

Samlingar av kod som är självständiga och som inte påtagligt ändrar eller avbryter varandra. En annan viktig funktion inom objektorienterad programmering är arv, det vill säga förmågan att härleda klasser från en överklass till underklasser, och därmed föra vidare specifika kännetecken från överklassen.

Scrolling

Förflyttning av en bakgrund på skärmen i en glidande rörelse för att skapa en illusion av att området tittaren ser är större än själva skärmen. Text och bilder kan också röra sig glidande över skärmen mot en statisk bakgrund , alltså scrolla. (Griffith 2010, 10.)

Scenen

Scenen alltså ”the Stage” är huvudarean för bearbetning av grafik i Flash. Den är en representation av området användaren ser på skärmen när han spelar ett spel. (Rosenzweig 2008, 16.)

Tidslinjen

En Flashfilm är uppdelad i bildrutor. Tidslinjen låter användaren välja vilken bildruta som är visad på scenens arbetsyta. (Rosenzweig 2008, 18.)

Skärmobjekt

Ett skärmobjekt är vilket som helst objekt som har en visuell representation och kan placeras på scenen. Det finns många olika typer av skärmobjekt i Flash. Erfarna utvecklare känner till de vanligaste objekten som är knappar, sprites och filmklipp. Dom delar alla några egenskaper; de har alla x, y och z positioner på skärmen samt skalnings- och rotationsattribut.

Filmklipp

Filmklipp är individuella instanser, alltså objekt, av klassen filmklipp. Denna klass har många inbyggda egenskaper som passar spelprogrammering.

Paket

Paket innebär en samling klasser med kategoriskt liknande eller relaterad funktionalitet. Klasser inom samma paket kan refferera varandra utan specifik kod, däremot bör klasser inom olika paket importera varandra. (Griffith 2010, 36.)

Events (Händelser och lyssnare)

Händelser spelar en central roll i ActionScript 3. Händelser kan beskrivas som meddelanden som sänds mellan objekt för att notifiera varandra om just händelser. När ett objekt är uppsatt för att mottaga händelser kallas det för lyssnare. (Griffith 2010, 49.)

1 INLEDNING

Detta lärdomsprov tar både allmänt upp spelutveckling i Flash med actionscript 3 och genomförandet av ett beställt spelprojekt åt Stundars museum. Efter min praktiktid på Img-media erbjöds jag arbeta på ett spelprojekt ”Stundars virtuella tidsresa” för företaget Img-media beställt av Stundars museum. Spelet är ett lärorikt och roligt spel ämnat för barn i åldern 6 – 12 år. Från Januari 2010 fram till Maj 2010 gick projektet långsamt framåt vid sidan av mina studier, men fr.o.m maj 2010 ökade takten och arbetet pågick intensivt över sommaren och fram till den sista september. Vissa detaljer i spelet har även finslipats nu som då efter det. Den praktiska delen av detta lärdomsprov, alltså själva programmeringen av spelet tog sammanlagt 586 timmar att slutföra och resulterade i över 11000 kodrader. Spelet finns tillgängligt på internet. Adressen är http://stundars.img.fi/main.html

1.1 Syfte, målsättning och avgränsningar

Syftet med detta lärdomsprov är att förklara vanliga metoder inom spelutveckling i Flash och actionscript 3, att dokumentera spelutvecklingsprojektet ”Stundars virtuella tidsresa” och att jämföra slutresultatet med målsättningarna i planeringsskedet. Detta lärdomsprov kommer inte att ta upp grundläggande koncept inom programmering som variebler, datatyper, loopar, operatorer och flödeskontroll.

2 PLANERING

2.1 Syfte och mål för tidsresan

Syftet med den virtuella tidsresan är att på ett lustfyllt sätt lära sig om livet på landsbygden i svenska Österbotten under början av 1900-talet. Den virtuella tidsresan ska ge kunskap om försörjning (jordbruk, boskapsskötsel, hantverk, binäringar), boende, familjens sammansättning och de olika familjemedlemmarnas uppgifter, skolgång, mathushållning, textiltillverkning, klädsel, manlig hemslöjd samt kommunikation. Materialet ska vara roligt, tilltalande och väcka intresse för historien. Tidsresan ska på ett pedagogiskt sätt lära ut baskunskap om tidens sätt att leva. Utöver kulturhistorisk kunskap kan den virtuella tidsresan stimulera fantasin och inlevelseförmågan samt bygga upp barns historiemedvetenhet.

Lärandet sker audiovisuellt med hjälp av spel, rörliga och icke-rörliga bilder, fotografier, text, och ljud. Förklarande text finns tillgänglig för den som är intresserad av att fördjupa sin kunskap om livet för 100 år sedan.

Den virtuella tidsresan lämpar sig både som stöd till undervisningen i skolan och som fritidsaktivitet. Målsättningen är att ge en större medvetenhet, kunskap och förståelse för den kulturmiljö barnen lever i. Hur ser närmiljön ut och hur har den förändrats? Hur kan historien påverka oss som lever idag?

Målsättningen är en professionell produkt, både när det gäller det pedagogiska innehållet samt de uppgifter som skall utföras. Stor vikt läggs även på produktens gränssnitt och illustrationer som skall inspirera barnen att återkomma till tidsresan.

2.2 Målgrupp

Tidsresan riktar sig framför allt till barn i åldern 6-12 år. För yngre barn bör historieberättandet vara beskrivande samt relatera till nu - då. Äldre barn kan förstå

den bakomliggande meningen med olika företeelser och kronologi med årtal. Givetvis kan även unga och vuxna ha glädje av tidsresan.

2.3 Detaljplan

Den virtuella tidsresan utgår från en tecknad karta över Stundars område. Utgångspunkten är en sommarvardag i Solf under början av 1900-talet. Tidsresan formas som en helhet med en klar början åtföljd av en handling utgående från en ramberättelse samt en avslutning. Användaren kan alltså utföra olika praktiska vardagssysslor och övriga uppgifter under den virtuella tidsresan.

Den virtuella tidsresan finns tillgänglig på både svenska och finska. Kulturarvet i Österbotten under början av 1900-talet har många gemensamma drag för båda språkgrupperna, därför lämpar sig spelet utmärkt även för finskspråkiga.
Resenären rör sig med genom att klicka till valfritt illustrerat område/byggnad på museiområdet. För att komma in i en byggnad klickar man på dörren (dörren öppnas så att man förstår att man kan gå in).

Interiören i byggnaden är illustrerad sedd snett uppifrån. Man ser alltså väggen till vänster, rakt fram och till höger. Vissa föremål får man veta mera om (i antingen textform eller som text + bild) genom att klicka på dessa.

Illustrationerna görs trevliga i typisk spelstil/”seriefigurs stil” som tilltalar barn. Miljöerna görs trevliga genom att saker rör sig och något smått hela tiden händer. T ex. Väderkvarnen snurrar sakta, groda hoppar på marken, rök stiger upp ur skorsten, fjäril flyger, katt som tvättar sig osv.

I alla miljöer finns svaga bakgrunds- / miljöljud. Exteriörer: Vind, fågelkvitter, katt som jamar, hund som skäller mm. Interiörer: Sorl från människor, mat som kokar på spisen, stolar som flyttas, tallrikar som skramlar osv. Hustomten återkommer i varje exteriör/ interiör. Tidsresan innehåller både motoriska och icke motoriska speluppgifter.

Det finns möjlighet att besöka: Exteriören för 1. Smedens stuga, 2. Lanthandeln/Skolan, 3. Bondgården, 4. Kvarnen (eventuellt även Skomakarens stuga i samma bild) 5. Kloka gummans stuga. Interiören för 1. Lanthandeln, 2. Bondgården och 3. Smedens stuga.

2.4 Tidsresa

Spelaren börjar tidsresan genom att välja spelfigur vilken antingen är bondens dotter Hilda eller bondens son Erik. Det första delspelet påminner om en pappersdocka och spelaren ska klä sin spelfigur tidsenligt i 1800-tals kläder. I delspelet ligger klädesplagg utspridda, bland dessa moderna kläder. Spelaren kan dra och släppa olika kläder på sin spelfigur med markören. När spelaren funnit rätt plagg indikeras detta och spelet går vidare till nästa delspel.

Delspel två går ut på att spelaren parar ihop olika föremål som hör ihop genom att dra linjer mellan föremålen. Exempel på föremål som hör ihop är ägg - höna, hartass – griffeltavla, kardor – får, ett rågax – säck med rågsäd, kopphorn – koppyxa, kräckla – gryta. Om spelaren parar ihop fel föremål med en linje så blir linjen röd och försvinner långsamt. Då spelaren parar ihop rätt föremål blir linjen grön och lämnar kvar på sin plats. När spelaren parat ihop alla föremål går tidsresan vidare.

Efter det andra delspelet anländer spelaren till spelets huvudsakliga spelplan. Där kan spelaren vandra omkring mellan olika delområden av stundars och besöka olika hus på området. Tidsresans intrig går ut på att spelaren ska utföra olika uppdrag åt Maria Smeds som är en rollfigur i spelet. Spelaren startar vid smedens exteriör (gård) och möter rollfiguren Maria Smeds omedelbart när den huvudsakliga spelplanen laddats in. Hon bemöter spelaren genom att säga: Jag heter Maria Smeds och kallas Smedis Maria. Jag bor i det här huset. Min son Alfred kommer hem efter 5 år i Amerika och jag tänkte ordna en fest för honom. Kan du hjälpa mig? Smedis Maria ber om att få följande ärenden och uppdrag utförda av spelaren:

1. Äggen ska säljas i lanthandeln.
2. Handla kaffebönor och risgryn i lanthandeln.
3. Glöm inte att det är skoldag idag!
4. Besök bondstugan
5. Hämta mjöl från kvarnen
7. Kloka gumman hjälper sjuka människor.
8. Till festen ska gröten kokas, brödet bakas och smöret kärnas.

Vid smedens exteriör kan spelaren även hitta en slangbella på marken. Om spelaren klickar på denna slangbella startar ett delspel där spelaren får skjuta prick med slangbella på bleckmuggar. Inne i smedens stuga kan spelaren koka gröt. Gröten ska vispas med kräcklan tills ”aktivitetsmätaren” i gränsnittet nått gränsen ”klar”. Spelaren får även nagga brödet tills ”aktivitetsmätaren” nått gränsen ”klar”.

I lanthandelns interiör kan spelaren handla kaffebönor och risgryn. Spelaren kan även leverera ägg till handeln. Kaffebönor och risgryn inhandlas på kredit. Mera info om valda föremål i lanthandeln får man genom att klicka på dem. I lanthandeln finns även en kvast vilken spelaren kan klicka på för att starta ett delspel som innebär att jaga bort möss från lanthandeln genom att jaga bort dem med en kvast.

I skolan deltar spelaren i en skollektion genom att se på Stundars skoldagsfilm. I skolan kan spelaren även skriva på griffeltavlan och sudda med en hartass. Skrivstilsbokstäver från början av 1900-talet finns som modell.

Vid bondgårdstunet framträder namnet på byggnaderna när de markeras. Inne i bondstugan finns moderna föremål inplockade i storstugans interiör. Spelaren ska ta bort det som inte hör till tiden (slutet av 1800-talet).

Från kvarnen ska spelaren hämta en säck mjöl som behövs till bakningen. Mjölsäcken får spelaren när han kommunicerar med rollfiguren mjölnaren. Då

spelaren går in i kvarnen visas en film i genomskärning vad som händer när säden mals till mjöl.

Vid kloka gummans stuga strövar en höna omkring på gården. Då spelaren kommunicerar med rollfiguren kloka gumman så startas ett memoryspel. Par som finns i spelet är t.ex. Kopphorn och koppyxa (koppning ansågs bota många sjukdomar) Groblad (användes som förband på sår), lök (Saften från en färsk lök lindrade bi- eller getingstick), vinruta, älgört eller potatis (mot huvudvärk), rölleka (bad för stukade leder), ormbrännvin (smordes mot värk eller på bölder), beckolja, backtimjan eller hästhov (mot hosta), pepparrot (mot feber), vinruta (mot löss), humle (mot tandvärk), maskrosblad (mot vårtor), renfana (mot springmask) .

Under spelets gång dyker det nu som då upp en popup-knapp som spelaren kan klicka på för att få veta mera om livet på landsbyggden förr. T.ex. då spelaren pekar på skolans dörr första g ången dyker knappen upp, genom att då klicka på knappen tas spelaren till gränssnitten där olika informationstexter om livet förr finns samlade. Mera texter fylls på under spelets gång när spelaren besöker olika platser och spelaren kan närsomhelt gå tillbaka till texterna för att läsa mera.

När alla uppgifter är utförda presenteras ett färdigt dukat bord och festen kan börja. På bordet finns ett gemensamt lerkärl eller träkärl med gröt, ev. porslinstallrikar, dricksglas, var sin träslev, en gemensam dryckeskanna med svagdricka, brödhög, smör i smörbytta. Tidsresan avslutas. Smedis Maria tackar för all hjälp. Sonen stiger in i stugan och har med sig spännande gåvor från Amerika som delas ut. Spelfigurens klädsel ändras till nutida kläder. Efter avslutningen dyker det upp ett sista delspel som är en frågesport för att se vad spelaren lärt sigt.

I de olika exteriörerna kan spelaren stöta på Stundarstomten. Stundarstomten är en kortväxt figur (som ett 4-5-årigt barn), bastant byggd karl i grå kläder, grått skägg, sockertoppsformad, röd eller grå mössa, enögd (ögat mitt i pannan), med bred mun, godmodigt leende. Blir synlig när han vänder sin mössa avig.

3 SPELUTVECKLINGSTEORI I FLASH

I detta kapitel behandlas några centrala aspekter inom spelutveckling i allmänhet och hur de hanteras i Flash och ActionScript 3.0.

3.1 Användar input i Flash

Användarinteraktion sker oftast med tangentbordet eller musen eftersom de är standard input tillbehören som följer med moderna datorer. När du vill fånga interaktion med användaren. Detta sker genom att fånga MouseEvent och KeyboardEvent händelser. Dessa två klasser och deras egenskaper är arbetshästarna av användardrivna Flash applikationer. Överklassen InteractiveObject har flera attribut som kan användas till för att ställa in beroende på om man lyssnar efter dessa mus- eller tangentbordshändelser. Flashspelaren sänder en mushändelse till alla lyssnare varje gång användaren klickar på en musknapp, rör på musen eller scrollar musen över ett objekt. (Braunstein, Wright & Noble 2008, 351.)

Vad är ett KeyboardEvent? Jo det är naturligtvis varje gång man trycker på en tangent. KeyboardEvent liksom MouseEvent sänds av ett objekt som ärver av InteractiveObject. Detta objekt måste vara i fokus för att höra händelsen, så om KeboardEvent meddelandet vill göras globalt så ska man lyssna till scenen, annars lyssnar man till det objektet man är intresserad av. KeyboardEvent definierar ett KeyCode attribut som returnerar tangenten användaren trycker på. (Braunstein, Wright & Noble 2008, 362.)

3.2 Grafik och animation

Grafik och figurer som ska användas i ett spel görs till filmklipp i Flash bibliotek. När ett filmklipp ska införas i spelet från biblioteket finns det två sätt att göra det på. Det första sättet är att dra filmklippet och släppa ner det på scenen, och sedan

namnge instansen i attributen. Det andra sättet att införa ett filmklipp i spelet är genom att använda ActionScript kod. Då bör filmklippet först göras tillgängligt genom att ändra dess länkningsattribut. Export for ActionScript ställs in och klassens namn anges. Nu kan kopior av filmklippen skapas genom att endast använda ActionScript. Detta görs genom att skapa en variabel som innehåller instansen av objektet och sedan används addChild för att lägga objektet på skärmen. Filmklippet kommer nu att dyka upp på kordinaterna 0,0 på scenen. Denna position kan ändras genom att använda x och y attributen av instansen. (Rosenzweig 2008, 43-44.)

Varje filmklipp i Flash kan inehålla flera bilder av grafik, inte bara en. Dessa bilder kan animeras genom att byta grafik med jämna mellanrum alltså flytta fram en bild i taget på tidslinjen. Hastigheten av denna animation går dock inte att ställa in direkt i flash genom att höja standardvärdet 24 bilder i sekunden till t.ex. 60 bilder i sekunden. Denna inställning betyder nämligen bara att spelet försöker visa 60 bilder i sekunden, men om användarens dator är av äldre modell kan spelet bli långsammare och då saktar spelet in, därför behövs tidsbaserad animation.

Riktig tidsbaserad animation innebär animationssteg beroende på hur mycket tid som passerat och inte på flash egna tidsintervall eftersom de logiska beräkningarna kan ta olika länge för varje gång programmet loopar. Varje gång programmet går igenom en loop räknas först tiden ut sedan det senaste animationssteget. Sedan förflyttas objekten i överensstämmelse till denna tidsskillnad. T.ex. om det första intervallet är 0.1 sekunder och det andra 0.2 sekunder bör objektet förflytta sig dubbelt så långt efter det andra intervallet för att animationen ska hållas konsistent. (Rosenzweig 2008, 62.)

3.3 Slumptal

Slumptal används i nästan alla spel. Slumptal tillåter oändlig variation och hjälper till att hålla koden simpel. Att skapa slumptal i ActionScript 3.0 görs med

Math.random funktionen. Denna returnerar ett tal mellan 0.0 och 1.0 bortsett från själva talet 1.0. (Rosenzweig 2008, 76.)

3.4 Kollission och timers

Då det rör sig objekt på skärmen i spelet är det mycket vanligt att de testas mot varandra för kollision. ActionScript 3.0 innehåller två funktioner för upptäckt av kollisioner. Funktionen hitTestPoint tester en punkt för att se om den är inom ett objekt. Funktionen hitTestObject testar två objekt mot varandra för att kontrollera om de sammanfaller. (Rosenzweig 2007, 68.)

Timers kan förstås som små klockor som skickar meddelanden. Den skapas och startas, och sedan tickar den på och skickar meddelanden vid förutsatta tidsintervall. T.ex. kan en timer ställas in att kalla en viss funktione varje sekund. En timer ställs in på det viset att en tal som anger millisekunder skickas till funktionen. En till parameter kan också skickas som anger hur många händelser funktionen genererar före den stannar. (Rosenzweig 2008, 61.)

3.5 Audio

Det finns två olika sätt för att spela upp audio med ActionScript 3.0, som interna ljud i Flash bibliotek eller externa filer. För spel är troligtvis den bästa metoden att importera ljudfilerna till spelets bibliotek. För att använda ett ljud i Actionscript bör ljudefilens attribut länkas till ActionScript. Sedan skapas en variabel för ljudet och en variabel för en ljudkanal. Ljudkanalen startas sedan genom att kalla på ljudvariabelns namn. (Rosenzweig 2008, 74.)

Objekt av klassen Sound är egentligen bara behållare av det ljudets data. När ljudet spelas genereras ett objekt av typen SoundChannel och kommandon härefter ska skickas till denna kanal. Detta leder till att utvecklarern måste hålla reda på många olika objekt för att överhuvudtaget kunna ha kontroll över ett ljud som utlöses. (Griffith 2010, 102.)

3.6 Fel och buggar

Det förekommer fel och buggar i koden nu som då. Flash kompilator och runtime engine fångar många fel skapade p.g.a fel syntax, felstavningar och flera andra vanliga programmeringsmisstag. Dessa fel ger direkt återkoppling och är normalt väldigt lätta att korrigera. Buggarna en programmerare inte vill stöta på är de som uppkommer när det inte tekniskt är någonting fel på koden, men spelet uppför sig okorrekt. Exempel på dessa är försämrad prestande med tiden, läckande minne eller oförväntat output. (Griffith 2010, 293.)

Fel bör alltid korrigeras. Oftast beror felen på misstag i koden men ibland kan fel uppstå p.g.a att en funktionalitet har andra beroendeförhållanden som externa filer. I dessa sammanhang kan man förhindra att hela koden stoppas genom att fånga felen med try – catch. (Griffith 2010, 56.)

Den vanligaste formen för att visa information från din kod i debugningssyfte är användningen av trace. Trace skriver ut alla argument du ger det till Flash output fönster. (Griffith 2010, 294.)

4 VERKTYG

Utöver de verktyg som räknas upp i detta kapitel bör nämnas att online internet forum, bloggar och programmeringssidor var min största hjälp och min viktigaste källa när jag utförde programmeringen i praktiken. Utan dessa skulle programmeringen av spelet ha varit omöjligt.

4.1 Flash CS4

Orsaken att vi valde Flash som utvecklingsmiljö och actionscript 3 som programmeringsspråk är många.

Möjligheten att nå ut till spelare är för tillfället mycket god. Ungefär 98% av alla användare på internet har någon version av Flash player, och vanligtvis uppdaterar 80% av användarna till den nyaste versionen inom ett år. Endast storleken av den möjliga spelarskaran är enanstående. Eftersom Flash player är tillgänglig för Windows, Mac och Linux kopplar den samman alla de kändaste operativsystemen. (Griffith 2010, 14.)

Detta är goda tider att vara en Flash spelutvecklare. Just nu finns det inget bättre verktyg för skapande av spel från liten- till medium-storlek. Flash är snabbt, kraftfullt och enkelt att utveckla spel med. Nyckeln till detta är ActionScript 3.0, det fantastiska nya programmeringsspråket som stöder Flash. (Rosenzweig 2008, 2.)

Flash är kapabelt att göra många saker. Det går att göra tecknade filmer, effekter, presentationer, banners, reklamer, websidor, applikationer och förståss spel. Flash är också en väldigt visuell utvecklingsmiljö som är lätt att närma sig. (Griffith 2010, 14.)

Flash gör det möjligt att mycket lättare snabbt få sitt spel på marknaden som i andra programmeringsspråk skulle behöva mycket kod. Bestyr som animation och

uppspelning av video sköts väldigt galant av Flash. P.g.a sitt arv som animationsprogram gör Flash det väldigt enkelt att visa grafik på skärmen. Detta kan låta självklart men i jämförelse med andra utvecklingsmiljöer är detta en stor fördel. C++, Java och andra språk framställer allting till skärmen programmatiskt vilket leder till många rader av kod för simpla grafiska åtaganden. Flash tar hand om framställningen under ytan så att utvecklaren inte behöver oroa sig för det. (Griffith 2010, 14.)

Eftersom Adobes produkter är centrerade kring design uppstår en viss frustration av de mest extrema grafiska designerna och programmerarna i båda ändårna av spektrumet, men det är denna sammansmältning av teknik och design som gör Flash unikt. (Griffith 2010, 15.)

Flash har många fördelar men är inte utan problem när det gäller spelutveckling. Den största bristen är programkodseditorn. Den enklaste lösningen är att sköta all ActionScript i en annan applikation och hantera resten i Flash. När Flashspel växer i storlek och komplexitet behöver de bastantare hårdvara att köras på. De flesta andra moderna utvecklingsmiljöerna erbjuder verktyg för övervakning av hur mycket cpu- och minnesresurser spelet använder. Flash har inte detta så det är svårare att förutspå hur väl spelet presterar på olika system. Windows task manager är dock ett bra verktyg för att övervaka en applikations minnes- och cpusnålhet. (Griffith 2010, 17.)

Traditionella spelutvecklare försöker ibland slåss emot Flash karaktär när de först går över till Flash, men det bästa sättet att uppnå resultat är att spela på dess styrkor. Hemligheten är att veta vad Flash gör bäst och när du behöver ändra dess beteende. (Griffith 2010, 20.)

4.2 Actionscript 3

ActionScript är ett mycket bra programmeringsspråk för spelutveckling. Det är lätt att lära sig, spelutvecklingen går snabbt och språket är mycket kraftfullt. ActionScript 3.0 introducerades 2006 . ActionScript 3.0 är kulmineringen av flera

års utveckling. Denna version tog speciellt i beaktande till vad utvecklare använder Flash till. Nu har vi en en utmärkt utvecklingsmiljö för 2D speluveckling. Du kommer att märka att en av dess styrkor är att det går snabbt att göra fungerande spel. (Rosenzweig 2008, 8.)

Mest viktigt är dock att ActionScript 3.0 är omskrivet från botten upp och använder sig av en annan kodbas än tidigare versioner av språket. Denna optimering har ökat prestandan relativt dramatiskt. Inlärningskurvan för ActionScript 3.0 är brantare men detta beror på att språket är mera robust inte svårare än tidigare versioner. I jämförelse med sina tidigare versioner erbjuder ActionScript 3 mera detaljerad felrapportering, föbättringar i syntaxen, sammanslagna metoder för att dynamiskt lägga till skärmobjekt , sammanslagna metoder för events och förbättrad objekt-orienterad programmering. (Shupe & Rosser 2007, 4.)

4.3 Övriga verktyg

Notepad ++

Eftersom Flash inbyggda kod-editor är svag (Griffith 2010, 17.) behövde jag en bättre kod-editor. Jag beslöt mig för att använda Notepad++ för att det är gratis, för att jag var van vid dess gränssnitt från förr och för att Notepad++ har ett behändigt plug-in som listar programmets alla funktioner.

Adobe Illustrator CS4 och Adobe Photoshop CS4

Grafiken i spelet gjordes av en medarbetare, men jag använde ändå flitigt både Adobe Photoshop och Adobe Illustrator i processen. Oftast för små ändingar i grafikens storlek eller för urklippning och delning av olika grafiska element som behövdes.

Propellerhead Reason

Största delen av ljudeffekterna i spelet kommer från betalade paket eller från gratis lujdeffekts-samlingar på internet. Vissa effekter gjorde jag dock i musikprogrammet Propellerhead Reason.

Anpassat gränssnitt för databas

"Anpassat gränssnitt för databas" är ett skräddarsytt online program som användes för att mata in värden i en databas utan att direkt behöva använda MySql. Databasen matades med text och info om föremål och texter om livet förr i tiden som sedan hämtas till spelet.

5 GENOMFÖRANDE

Stundars tidsresa är det största och mest ambitiösa projekt jag någonsin jobbat med. Det är det första beställda spelet jag programmerat, tidigare har jag programmerat spel som en hobby i några år. Min tidigare erfarenhet med Flash och ActionScript 3 var en liten speldemo och ett ambtiösare platformsspel. Jag programmerade hela Stundars tidsresa på egen hand och projektet innebar tusentals rader med kod och hittills har jag lagt ner 586 timmar på projektet. Grafiken skapades av Benny Fors och internetrelaterad programmering som php gjordes av en annan medarbetare.

5.1 Förladdare

Flash spel utrustas vanligtvis med en förladdare. Detta innebär att spelaren kan få titta på någonting intressantare än en vit skärm medan spelet laddas. Ofta kan en växande balk användas för att visa hur stor del av spelet för tillfället laddats in. Flash har inbyggda egenskaper för att sköta detta lätt om man använder sig av flera än en fil. Ironiskt nog vill de flesta stora och kända spelsidorna på internet som erbjuder Flashspel ha sina spel i en enda fil. Detta leder till att Flash inbyggda system måste kringås på ett invecklat sätt för att upnå en förladdare i samma fil. Som tur kunde jag i detta projekt använda mig av Flash eget förladdningssystem eftersom spelet inte var ämnat att laddas upp till någon känd spelsida. Vår förladdare och också laddningar inne i själva spelet betäcknas av roterande kvarnvingar. Genomförandet av förladdaren gick relativt simpelt eftersom jag testat det tidigare i speldemon jag gjort.

5.2 Startmeny

Startmenyn i Stundars tidsresa är klar och tydlig och innehåller endast knappar för språkval (finska och svenska) och startknapp. Menyn var lätt att genomföra p.g.a. min tidigare erfarenhet med Flash. Språkvalsknappen ändrar värdet för en språkvariabel vilken via laddaren skickas vidare till de olika spelsektionerna.

5.3 Laddning

Laddningen var för mig den överlägset svåraste utmaningen i skapandet av Stundars tidsresa. Jag äger många böcker angående ActionScript 3 men ingen av dessa tar upp laddningen och interaktionen mellan olika swf filer på et konkret sätt så jag var helt tvungen att förlita mig till internet och jag hittade till sist några programmeringsbloggar som tog upp saken och praktiskt visade hur vissa saker bör göras. Jag hittade också biblioteket BulkLoader som förenklar in- och urladdningen. Bibliotekets metoder fungerar likt Flash egna, skillnaden är att den grundläggande laddningskonstruktionen är färdigt programmerad. Efter en del försök och misstag bemästrade jag slutligen in och urladdningen med hjälp av BulkLoader.

Jag visste från början att spelet skulle ladda in en huvudinstans dvs. exteriören i spelet, och därifrån skulle mindre spel (swf filer) laddas in och tömmas ur minnet vartefter spelaren rör sig mellan de olika delarna av spelet. Eftersom ActionScript 3 inte direkt stöder globala variabler mellan olika inladdningsinstanser bör olika globala variabler skickas från exteriören till laddaren som sedan skickar variabeln till den inladdade instansen. Om någon av dessa variabler sedan ändras i den inladdade instansen, som t.ex. att en viss uppgift utförs, bör variablerna sedan skickas tillbaka via laddaren när instansen avslutas innan den laddade instansen och laddaren töms ur minnet. Jag studerade först enklare alternativ till detta men de visade sig vara opålitliga och det är ändå allmänt sett bättre programmeringssed att skicka variabler mellan funktioner än att använda globala variabler. Denna laddning och variabelstafett visade sig dock inte helt oförväntat vara en av de långsammaste och betungande delarna att programmera.

5.4 Minneshantering

För att undvika fel och för att undvika att spelet ska äta minne var jag extra noggrann med att se till att alla händelser och lyssnare avlägsnas innan olika instanser av spelet töms ur minnet. En viss automatik av detta ingick i BulkLoader. Flash har också en egen automatiserad garbage collection som tömmer minnet av oanvända instanser vartefter.

Förutsättningen att denna ska fungera är dock att programmeraren bör komma ihåg att stänga av lyssnare, annars lämnar dom och lyssnar efter händelser i all evighet.

5.5 Exteriörerna

Huvudarean i spelet består av stugornas exteriörer, alltså själva området Stundars. Här kan spelaren gå mellan de olika husen och rollfigurerna och söka efter spel och uppgifter. Ur ren programmeringssynpunkt är exteriören den centrala delen av spelet som knyter samman de andra småspelen till en helhet. Filen växte enligt sin centrala roll under programmeringens gång och blev den största filen med den längsta koden i hela spelet.

Exteriörerna är också det ända stället i spelet där spelaren ser sin spelfigur. Spelfiguren kan styras antingen med piltangenterna på tangentbordet eller med mus. Styrningen av spelfiguren sköts från spelfigurens egna klass. Eftersom det finns både statiska och scrollande exteriörer behövs det fyra skillda system för styrandet av spelfiguren.Styrning med mus i exteriör både med statisk eller rörlig bakgrund och styrning med tangentbord både vid statisk och rörlig bakgrund. Ett bekvämt styrande av spelfiguren är naturligtvis viktig och funktionen finslipades genom hela produktionen. Programmeringen av scrolling alltså rörlig bakgrund är relativt invecklat eftersom utöver bakgrundens förflyttning bör alla objekt på spelplan förflyttas och alla objekt bör kollisionskontrolleras mot spelfiguren så att spelaren t.ex. inte kan röra sig igenom hus. Dessutom bör spelfigurens z kordinat tas i beaktande i förhållande till x och y kordinaterna för att kontrollera om spelfiguren ska visas bakom eller framför andra objekt på spelplanen.

Beroende på vilken exteriör spelaren befinner sig i spelas olika omgivningsljudeffekter upp, som t.ex. kraxande kråkor, susande vind och kacklande hönor. När spelaren pekar på en dörr dit spelaren kan gå in, öppnas dörren och en ljudeffekt spelas upp. I vissa fall då spelaren pekar på en dörr dycker även en pop-up knapp upp, vilken leder till lärorik information om livet förr.

5.6 Gränssnitt

I spelets centrala fil, vilken alltså består av spelets exteriörer, befinner sig också spelets huvudmeny eller gränssnitt. Menyn består av sex knappar. En knapp för audio, en hjälpknapp, en knapp för information om livet förr, en knapp som visar uppdragen, en knapp som visar kartan över området och en knapp för att återgå till spelets startmeny. De enskillda delarna av detta gränssnitt var inte i sig svåra att programmera men eftersom knapparna alltid är synliga då spelaren befinner sig i exteriören behövdes ett relativt invecklat system för att förhindra knapparnas användning i olägliga situationer. T.ex. förhindras spelaren att ge förflyttningsinput efter att kartan tagits fram. Ett annat exempel är objektens z-position, när man kontrollerar uppdragen bör uppdragsboken naturligtvis befinna sig framför alla andra objekt. Allt detta är småsaker, men de många kombinationerna som knapptryckningarna sammanlagt kunde leda till var många och en uteslutningmetod programmerades.

Bild 1. Smedens exteriör, spelets meny, rollfiguren Smedis Maria och spelaren.

5.7 Rollfigurer

De olika rollfigurerna i spelet så som Smedis Maria, kloka gumman, handlaren och mjölnaren ville jag programmera så att de alla skulle bygga på samma basklass för att standardisera konversationer och interaktioner mellan rollfigurerna och spelfiguren. Detta visade sig dock vara väldigt invecklat och lyckades endast delvis eftersom de olika spelsituationerna krävde att vissa funktioner skräddarsyddes. Det visade sig också att i vissa situationer var det lättare att sköta konversationerna via Flash tidslinje och i andra situationer helt och hållet via ActionScript.

5.8 Figurval

Från startmenyn tas spelaren till figurvalet. Där väljer spelaren rollfigur mellan en pojke och en flicka och klär därefter rollfiguren tidsenligt. I påklädningsspelet blandades pojk- och flickkläder, nya och gamla kläder huller om buller och spelaren ska sedan dra och släppa olika klädesplagg på rollfiguren. Drag and drop är relativt enkelt att genomföra i ActionScript och exempel gick lätt att hitta på internet. Enligt spelplanen var flickan äldre och längre än pojken vilket ledde till ett problem eftersom alla kläder behövde vara lika stora för att passa oavsett vilken rollfigur spelaren valt. Detta löstes dock lätt genom att skala ner pojken i rollfigursrutan och skala upp pojken i och med klädspelet. För övrigt gick programmeringen av figurvalet och påklädningsspelet efter plan. En del kordinat beräkningar behövdes eftersom klädesplaggen var olika långa. Flest problem orsakade klädesplaggens z kordinat igen beroende på plaggens längd, men detta var endast en ordningsfråga. Några andra finesser som behövde omtanke var att om spelaren drar ett plagg utanför skärmen skuttar det tillbaks in på spelplan, och då spelaren släpper ett plagg nära dit plagget passar på figuren, så hoppar plagget automatiskt på rätt plats.

5.9 Parspel

Att para ihop två sammanhängande föremål med linjer är ett klassiskt roligt barnspel. I och med detta spel fick jag ta reda på hur ActionScript arbetar med linjer och geometri.

Grundprinciperna för att programmera dessa linjer hittade jag på internet. Vissa problem stötte jag på eftersom bilderna placeras slumpmässigt och programmet bör hålla reda på vilket föremål spelaren trycker på, och om det hör ihop med följande föremål spelaren trycker på. Detta låter väldigt simpelt, men det var svårare än förväntat att programmera detta dynamiskt.

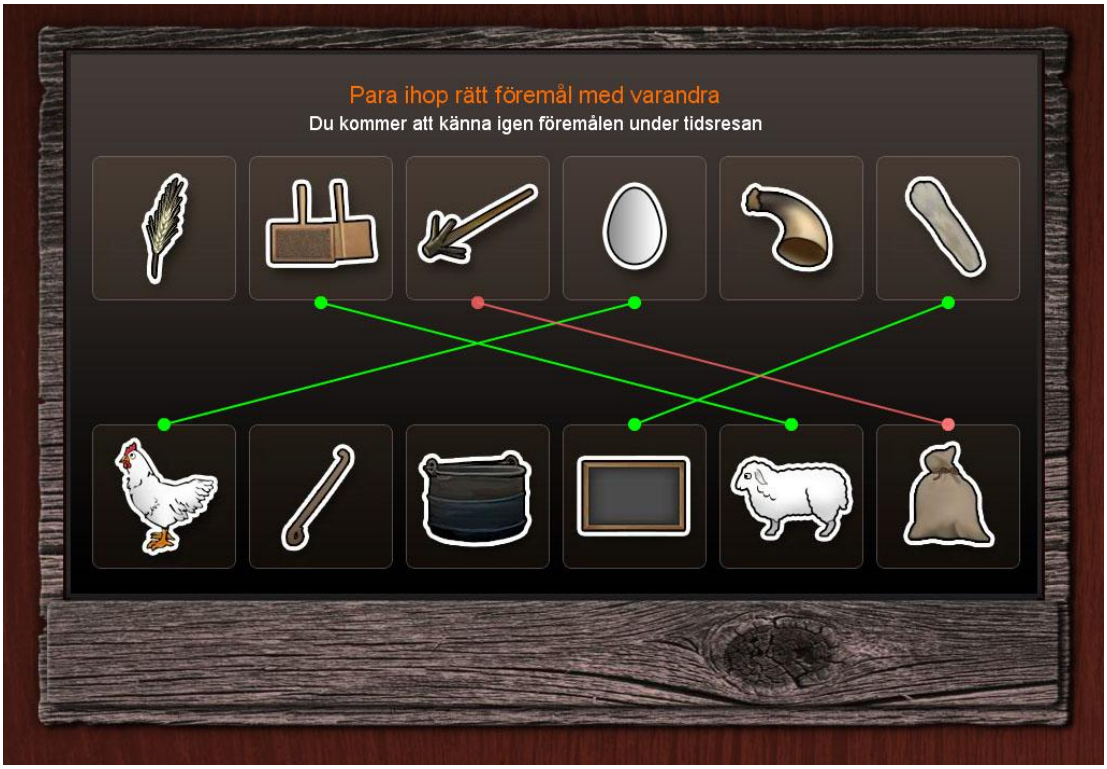

Bild 2 Parspel.

5.10 Skyttespel

Skyttespelet går ut på att bleckmuggar kastas framför spelaren från båda sidorna av rutan och spelaren ska försöka träffa dessa med en slangbella som styrs med musen. Spelaren ska träffa så många muggar som möjligt under en viss tid. Muggarna slungas från sidorna med olika infallsvinklar, infallsvinkeln räknas ut genom att ett slumptal beräknas mellan ett minimum och maximum värde. Kraften vilken muggen skjuts iväg med räcknas på samma sätt ut med hjälp av ett slumptal. Minimum och maximum

värdena testade jag ut genom trial and error för att förhindra muggarna att flyga iväg utanför spelskärmen. Det mesta av skyttespelets programmering gick smärtfritt. Jag fick först en del problem med kolissionsdetektionen mellan muggarna och slangbellan eftersom den kunde genomföras på flera olika sätt. Det bästa sättet att sköta kollisionsdetektionen fick lösas med speltestning, alltså åter igen trial and error. Eftersom muggarna rör på sig relativt snabbt över skärmen och dessutom roterar blev kollisionsdetektionen för tung då jag först använde många jämförelsepunkter på objekten. Problemet löste sig då jag ändrade kollisionen att gå efter bara en enda refferenspunkt. När spelaren träffar en bleckmugg spelas en klingande ljudeffekt upp. Även när spelaren skjuter med slangbellan spelas en ljudeffekt upp som låter som ett gummiband. I detta spel märks en av den objektorienterade programmeringens starka sidor, endast en klass behövde programmeras för bleckmuggen och efter det kan många likadana objektinstanser lätt skapas utifrån denna klass vartefter spelet framskrider.

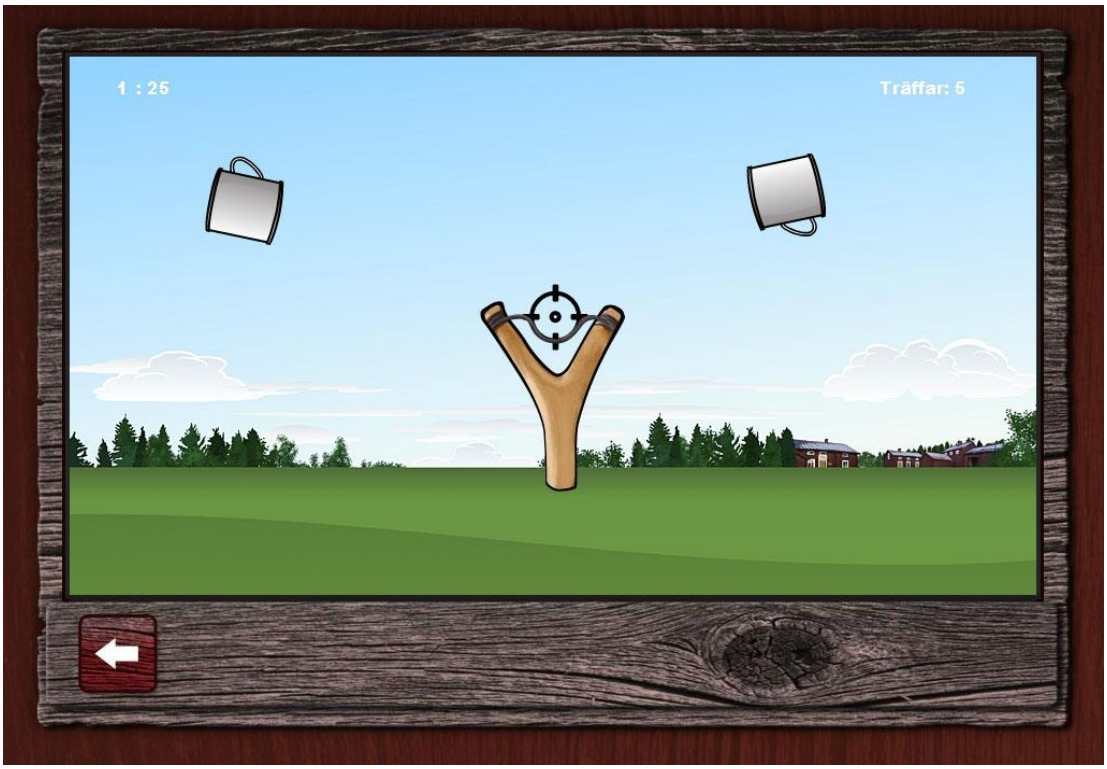

Bild 3. Skyttespel.

5.11 Skolfilm

När spelaren klickar på skoldörren kan han välja att se på en skolfilm, vilket också är ett av uppdragen i spelet. Skolfilmen hade tidigare producerats av Stundars. Inladdning och uppspelning av film på ett enkelt sätt ingår i Flash och ActionScript 3 vilket gjorde att detta var lätt att genomföra. Även spelarens utseende var lätt att modifiera så att den skulle passa in i spelet så bra som möjligt.

5.12 Griffeltavla

Det andra spelet som gömmer sig bakom skoldörren är griffeltavlan. Spelet går ut på att spelaren kan rita på tavlan med en krita eller sudda ut det han ritat med en hartass. Spelet är i princip ett primitivt ritprogram. På internet hittade jag många skriverier om hur detta kunde utföras, men de berörde ämnet endast ytligt och jag var tvungen att lista ut många detaljer på egen hand. Att avbryta linjerna före ramen var t.ex. ett måste, annars kunde spelaren rita utanför tavlan.

5.13 Handeln / jaga möss

I handeln kan spelaren köpa behövliga föremål av handlaren, jaga bort möss med en kvast och få information om föremål i handeln genom att klicka på dem. I handeln kan spelaren scrolla från sida till sida för att se den fina interiören och för att utforska alla föremål. Scrolling är som sagt relativt invecklat att utföra, men jag började ha vanan inne vid dethär laget från att ha programmerat scrolling i vissa exteriörer och bondstugan. Interaktionen med handlaren blev tyvärr gjord i viss brådska och var mer ambitiös i planen. Jag programmerade interaktionen så enkelt som möjligt, och den är i princip en bildsekvens.

Bild 4. Lanthandeln.

Genomförandet av musspelet gick relativt enkelt och det påminde till en del om genomförandet av skyttespelet. Spelet går på tid och spelaren ska försöka sopa bort så många möss som möjligt med kvasten under denna tid. Med jämna mellanrum strömmar det in mera möss på golvet (spelplanen). Antalet möss som kommer in är ett slumptal mellan ett minimum och maximum värde. Om spelaren lyckas sopa bort alla möss släpps det omedelbart ut mera möss enligt samma princip. När spelaren träffar en mus med kvasten spelas ett pipande läte upp. Detta spel visar igen styrkan av objektorienterad programmerin genom att jag behövde bara programmera funktionaliteten av ett enda möss i en musklass, och efter det kunde jag göra massvis instanser av denna klass för att få spelplanen full av springande möss. Genomförandet av information om föremål tas upp i kapitel 6.13.

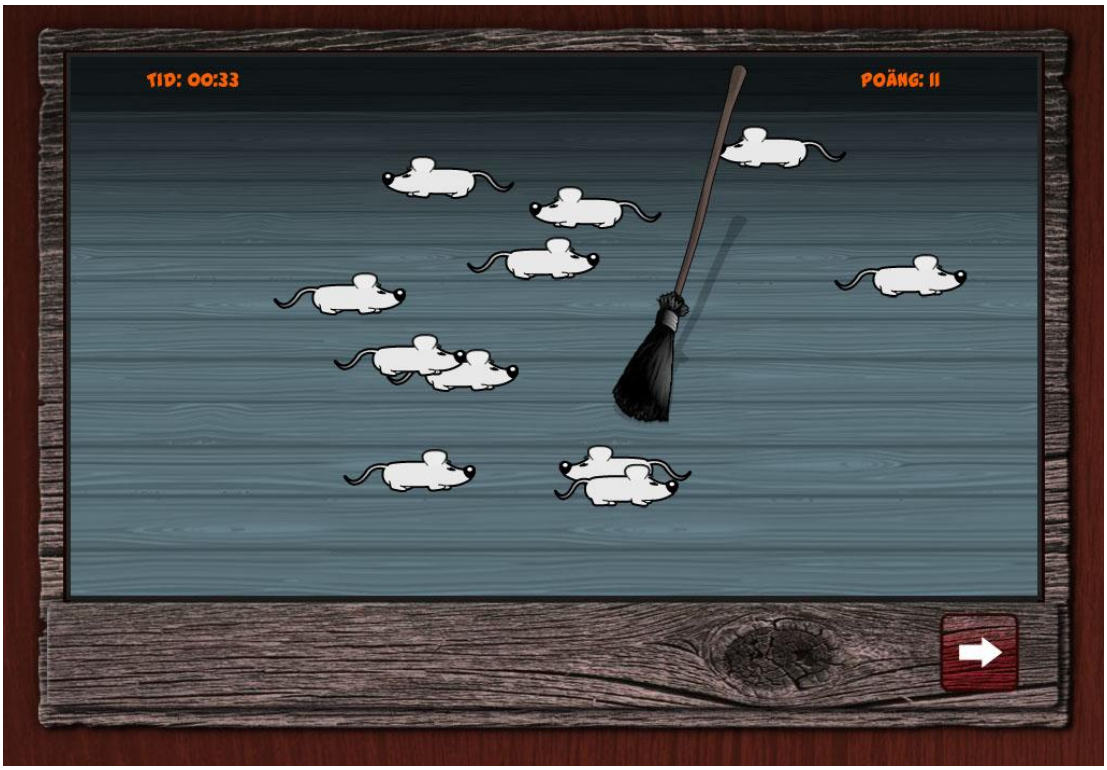

Bild 5. Musspel.

5.14 Bondstugan

Spelet går ut på att ta bort moderna föremål ur bondstugan. Genom att klicka på ett föremål kan spelaren ta bort det. Om spelaren råkar trycka på ett gammalt föremål som hör till bondstugan spelas ett varningsljud upp, men om spelaren trycker rätt på ett modernt föremål tas föremålet bort och en plingande ljudeffekt spelas upp. Efter spelet kan man även scrolla fram och tillbaka i bondstugan och läsa om de olika gamla föremålen genom att klicka på dem. Programmeringen av bondstugan innebar att spara alla föremål i tabeller och jämföra spelarens klick med dessa. Det var relativt rakt på sak, och vissa tabellfunktioner kunde förenklas genom att använda loopar.

Bild 6. Från bondstugans interiör bör moderna föremål plockas bort.

5.15 Information om föremål

Både i bondstugan och handeln skulle det enligt planen vara möjligt att klicka på föremål för att få se på en bild och läsa information om dessa i en inforuta. Bilderna sattes in i Flash men infotexten laddades in från en xml fil som skapas med ett skräddarsytt gränssnitt vilket är till för att spelets värdar lätt kan ändra på informationstexten om så skulle behövas. Inladdning av xml var nytt för mig men mycket väghjälp fanns på internet så detta kunde genomföras. Eftersom jag visste att denna finess behövdes både i handeln och bondstugan gjorde jag en egen klass Thinginfo i bondgårdsspelet som sedan kunde återanvändas i handeln bara genom små ändringar och utbyte av bilderna.

Bild 7. Föremålens inforuta.

5.16 Kvarnen

Vid kvarnen ska spelaren prata med mjölnaren för att få en säck mjöl. Interaktionen med mjölnaren är i princip en sekvens. Vissa problem med objektens z kordinater uppstod eftersom kvarnen har stora roterande vingar som i vissa fall hamnade ivägen då spelaren ska klicka på dörren eller mjölnaren. Detta gick naturligtvis att ordna upp. När spelaren väl går in i kvarnen laddas det in en film som visar kvarnens funktion från insidan vilket sköts smidigt inom Flash.

Bild 8. Kvarnens exteriör, spelarens rollfigur och mjölnaren.

5.17 Kloka gummans stuga / minnesspel

Vid kloka gummans stuga kan spelaren spela memory (det klassiska minnesspelet) och få information om läkemedel som användes förr i tiden. Spelet går på tid och desto bättre spelaren spelar desto mera poäng får han. I boken ActionScript 3.0 game programming university av Gary Rosenzveig finns ett exempel på hur ett minnesspel kunde programmeras och jag baserade spelet på denna version. Eftersom basen av spelet var programmerat annorlunda än min egen standard var jag tvungen att klura ut en del modifikationer i in- och urladdningen.

Bild 9. Kloka gummans memoryspel.

5.18 Smedens stuga

Smedens stuga var det sista delspelet som jag integrerade och vid det skedet var jag under tidspress så jag designade spelena för att kunna programmera dem så snabbt som möjligt och ändå uppnå resultatet i planen. Spelen i smedens stuga går ut på att koka gröt och nagga bröd. När spelaren rör på musen fylls en aktivitetsmätare. Dessa spel genomfördes utan desto mera finess men de fyller sin uppgift enligt planen, men kunde utan tvekan förbättras och göras intressantare. I smedens stuga kan spelaren också scrolla bakgrunden för att se hela interiören.

5.19 Tomten, fjärillarna och hönan

Dessa spelobjekt är till för att göra miljöerna mera levande. Fjärillarna är egentligen bara en klass som med hjälp av slumptal bestämmer vilken av fyra olika fjärilsanimationer som spelas upp. Med hjälp av sluptal bestäms också om fjärilen ska komma in från

vänster eller från höger på rutan. Slumptal avgör också om en fjäril ska skapas eller inte då en spelare byter exteriör. Speciella funktioner behövdes också för att få fjärilen att flyga på rätt sätt i exteriörer med scrollande bakgrund.

Tomten kan visa sig på vissa förutbestämda ställen. Om han visar sig bestäms med hjälp av slumptal. Chansen att han ska visa sig är betydligt lägre än fjärilens. Om spelaren klickar på tomten spelas en ljudeffekt upp som om tomten skulle skratta, därefter försvinner han.

Hönan går och pickar i marken utanför kloka gummans stuga. Hönans animation och rörelse är inte dynamiskt kodad, därför går den alltid i samma mönster. Om spelaren klickar på hönan så kacklar den till.

5.20 Frågesport

Frågesporten i slutet av spelet genomfördes med hjälp av tabeller och långa switch satser. Switch funktionerna behövdes p.g.a att spelet är tvåspråkigt. En del jobb gick också in i att ändra färgen på texten för att göra frågesporten så klarläslig som möjlig. För övrigt följer spelet en förutsagd sekvens vilket inte var något problem att programmera.

5.21 Information om livet förr

När spelaren pekar på en dörr eller utlöser andra funktioner dyker det upp en pop-up knapp som leder till information om livet förr då spelaren trycker på denna. Själva gränssnittet för denna information kan nås från menyn genom att trycka på knappen betäcknad med ikonen utropstecken. I detta gränssnitt listas olika ämnen upp vartefter spelet framskrider. Informationen hämtas på samma sätt som informationen om föremål från en xml fil. Detta visade sig vara relativt krångligt att få Flash att formatera xml texten på specifika sätt, och jag gick igenom en hel del trial and error förrän jag fick det att fungera. Överlag blev denna funktion den näst svåraste funktionaliteten att programmera för mig efter laddninsfunktionaliteterna. Fördelen är naturligtvis att

informationstexterna kan behändigt ändras när som helst online från det anpassade gränssnittet. Om texten är längre än det som kan visas på spelskärmen går den att scrolla upp och ner med hjälp av pilknapparna i informationsgränssnittet.

Bild 10. Visste du att? Gränssnitt.

5.22 Säkerhet

Flash och ActionScript har en sträng och noggrann security sandbox för att skydda användarna. Detta kan leda till en del problem för programmeraren, speciellt om spelet behöver ladda in olika resurser från olika servrar. Alla filer som ingår i Stundars tidsresa lagras lyckligtvis på samma server så detta ledde inte till några nämnvärda svårigheter i detta projekt.

5.23 Speltestning

Mina medarbetare gjorde sitt bästa för att hitta buggar och förhoppningsvis blev de värsta buggarna borttagna ur spelet. Vår mest värdefulla speltestning skedde dock vid Stundars barnkalas på sommaren 2010. Vi hade en egen stationering med en tidigare version av spelet i Stundars medan hundratals barnfamiljer besökte området. In i vår stuga kom massor barn som fick prova på spelet som var kopplat till en videokanon. Då kunde vi studera barnens reaktion och se vilka uppgifter som var svårast och som kunde behöva ändring. Till största del var reaktionerna positiva och de flesta barnen kunde spela problemfritt, och de barn som inte spelat spel förr lärde sig snabbt. Jag skulle kalla speltestningen en succé.

5.24 Publicering

Flash är mycket behändigt eftersom det är så enkelt att publicera spel gjorda i Flash. Publiceringsmenyn i Flash innehåller alla de attribut som programmeraren kunde tänkas behöva. Överlag använde vi oss av ganska hög standard på bildfiler för att bibehålla den fina grafiken. Också ljudfilernas kvalitet ville vi hålla ganska hög. Detta kan naturligtvis leda till lite längre laddningstider eller att spelet saktar in och börjar knycka på äldre datorer. Spelet testades dock på många olika datorer av olika ålder och processorstyrka och de flesta klarade av spelet utan problem. De få problem vi stötte på berodde inte på datorerna utan ovanliga inställningar i webbrowsers. Integreringen av spelet i html behövde jag inte blanda mig i eftersom det sköttes av en medarbetare.

6 RESULTAT OCH SAMMANFATTNING

Stundars tidsresa var från första början ett ambitiöst och stort spelprojekt med tanke på att det skulle slutföras av endast två personer. Spelet hade ingen deadline i frågan om timmar men personligen fick jag under projektets gång flera deadlines när vissa funktioner skulle vara färdiga. Några gånger var det bråttom men överlag var arbetstakten lämplig. Stämningen på arbetsplatsen var dessutom mycket god vilket naturligtvis är en stor lättnad och bra för vilket projekt som helst.

När deadlinen som Stundars givit nåddes hade vi en version av spelet färdig som sattes upp på deras hemsida. Arbetet på spelet fortsatte dock i ett par månader efter det. Mera funktionalitet sattes in och spelet finslipades. Eftersom projektet skulle genomföras under en så lång tidsperiod blev de första planerna kanske lite överambitiösa, men under projektets lopp slopades det överflödigaste bort. I detta projekt var kvaliteten viktigare än kvantiteten. Vi hade också små planeringsstunder minst en gång i veckan för att se hur spelet framskrider och för att planera nya finesser. Vi hade också flera gånger större möten med stundars personal och arbetet framskred i högsta samförstånd och till goda miner. Jag är ingen politiker men min uppfattning är att alla var nöjda med projektet från början till slut och inte en enda konflikt uppstod under hela projektets gång. En av höjdpunkterna inom spelplaneringen var att vi med vår spelerfarenhet kunde förbättra många spelideer som först lades fram för att göra delspelen roligare och intressantare. Spelet är dessutom programmerat såpass dynamiskt att om Stundars vill beställa utökning till spelet så är det lätt att programmera mera delspel och mera funktionalitet.

Största besvikelsen gällande slutresultatet ur min personliga synvinkel var att jag aldrig hade tid att göra vissa delspel så bra som jag skulle ha kunnat p.g.a. tidsbrist. Interaktionen med handlaren kunde ha varit intressantare, och det var också meningen enligt planen att interaktionen skulle inneburit ett vägningsspel. Spelen i smedens stuga var dåliga koncept från första början men tiden räckte helt enkelt inte till för att göra dem intressantare. Spelprojekt kunde fortgå i all evighet om det fanns resurser, för det finns alltid någonting som kunde finslipas och göras bättre.

Det mest påfrestande med programmeringen var att jag ofta funderade på problemlösningar hemma, speciellt vid sovdags låg jag och funderade över hur nästa problem kunde lösas, och inom programmering finns det alltid ett eller flera problem att lösa varje dag. Det mest tidskrävande inom programmeringen var de små detaljerna som man inte tänker på förhand. För varje funktionalitet som programmeras finns det flera små detaljer som man inte tänkt på. Dessa detaljer är dessutom så små att det vore helt tokigt att beröra dem i en omfattande plan.

För övrigt är jag personligen extremt nöjd och stolt över slutresultatet. Projektet var mycket utmanande men det var väldigt roligt att få jobba med att programmera ett spel. Dessutom har denna erfarenhet nu bevisat att jag personligen kan programmera ett relativt stort spelprojekt helt på egen hand vilket stärker självförtroendet.

KÄLLFÖRTECKNING

Rosenzweig, Gary 2008. ActionScript 3.0 Game Programming University. Indianapolis, Indiana, USA. Que Publishing.

Griffith, Chris 2010. Real World Flash Game Development – How to follow best practices and keep your sanity. MA, USA. Focal Press.

Mook, Collin 2007. Essential ActionScript 3.0. USA. O'Reilly Media, Inc.

Braunstein, Roger, Wright, Mims H., Noble, Joshua J. 2008. ActionScript 3.0 Bible. Inianapolis, IN, USA. Wiley Publishing Inc.

Shupe, Rich, Rosser, Zevan 2007. Learning ActionScript 3.0 A Beginner's Guide. Canada. O'Reilly Media, Inc.