

M Game

Projektrapport

Av: Lowe Raivio
lr222gw
WP13 Distans
Linnéuniversitetet, Individuellt mjukvaruutvecklingsprojekt
2014-06-03

Abstrakt:

Rapporten beskriver spelet MGame som är ett deckarspel på webbläsaren, skriven med JavaScript (vanilj) och HTML5 Canvas. Jag kommer gå igenom allt som har gått bra men även hinder jag har stött på när jag har byggt själva motorn till spelet. Jag kommer också nämna problem som jag har stött på gällande testning som inte blivit klar mm.

Spelet syfte är att få reda på vem mördaren är och allt utspelar sig i ett hus, varje gång man startar upp spelet på nytt så rollsätts olika karaktärer, spelet blir alltså alltid olika varje gång du startar det.

Spelet blev aldrig klart, men det finns en beta version som du kan komma åt här:

https://dl.dropboxusercontent.com/u/57362949/MGameWebSite/MGame_Beta_v.1.0/index.html

Innehållsförteckning

Innehåll	Sid nummer
Abstrakt	S. 2
Inledning/Bakgrund	S. 4-6
-Projektets Syfte/Mål	S. 4
-Arbetssätt	S. 5
-Genomförande/Teknik	S. 6
Positiva Erfarenheter	S. 7
Negativa Erfarenheter	S. 8-9
Sammanfattning	S. 10

Inledning/Bakgrund

Projekts Syfte/Mål

Målet var att bygga spelet MGame, ett spel där spelaren sätter sig in i rollen som detektiv och ska undersöka ett mord. Idén var att bygga en 'motor' som tog fram ett unikt mordfall för varje gång spelaren startar ett nytt spel, en spelrunda skulle kunna vara mellan 20-40 minuter, eller längre beroende på flera faktorer (Kommer in på det mer senare).

MGame motorn behöver något som jag kallar 'gamecards', vad det i grund och botten är är data rörande ledtrådar, vad intervjupersoner ('bottarna/NPC, datastyrda figurer i spelet) ska svara gällande ledtrådarna och intervjudata som intervjupersonerna behöver för att spelaren ska kunna intervjua dem om saker som inte är en ledtråd. Utöver 'gamecards' så behövde spelet också något som jag kallar 'motivdata', motivdata är helt enkelt en samling med ID'n till gamecardskort, dessa ID:n placeras sedan i en av motivdata objektets arrayer(array är en samling med data..). De olika Arrayerna som motivdatan har representerar vilka typer av roller som kan ha detta kort, dvs. den som designar gamecardkorten designar kort för de olika roller som spelet erbjuder, dessa är rollerna; Murder: Personen som har begått mordet.

Victim : Person som har fallit offer för mördaren.

Other : Person som inte har något med mordet att göra (kan dock veta något om mordet).

Actor1 och enda upp till Actor4 : Roller som kan användas i ett specifikt mordfall, en av dessa actors kan t.ex. spela rollen som någons flickvän, eller som någons kompis, vad actors1-4 används till är helt upp till personen som skriver mordfallet.

Tillbaka till motivdatan, så motivdatan innehåller alltså de gamecards som tillsammans bildar ett specifikt mordfall, utöver korten som läggs in i motivdatan så kommer andra kort att användas i spelet, dessa kort används för att förvirra spelaren, vilseleda den mot andra spår för att göra spelet mer realistiskt och svårare (alltså, falska ledtrådar, eller helt enkelt ledtrådar som inte har någon betydelse för mordfallet..).

Genom att bygga denna motor som kan hantera dessa 'gamecards' så kunde jag på ett enkelt sätt göra att spelet hela tiden slumpas. Först slumpar man fram ett 'motivdata' sen slumpar man rollerna, efter det så kommer slumpade gamecards att kastas ut på planen för att vilseleda spelaren. Några fix behövs fortfarande göras på motorn så att varje rum inte blir fullt med gamecards utan att bara några placeholders (platser som gamecards kan ligga på) blir utnyttjade.

Samtidigt som varje spelrunda skulle bli en unik upplevelse ville jag också att det skulle vara roligt att spela, så några nya idéer sattes in i spelet.

Karaktärer går omkring i rum: varje 5 minut så byter alla levande karaktärer rum, detta för att få spelet mer varierande och roligt.

En annan sak som är en viktig kärna i spelet är tidspoäng, tidspoäng används för att begränsa spelarens frihet. Jag vill lämna så mycket frihet som möjligt hos spelaren, och det gör jag (*det finns inget som spelaren MÅSTE göra, målet med spelet är att tänka själv och göra det mest logiska valet*), men för att göra spelet roligt så måste det finnas ett hinder, detta hinder är alltså tidspoängen. Tidspoäng symboliserar den tiden det tar att göra någonting, att undersöka en ledtråd kostar 5 tidspoäng, att intervjua en karaktär kostar 5 tidspoäng (plus de tidspoäng spelaren spenderar på att ställa följdfrågor).

Genom att lägga till tidspoäng så blir spelet mer koncentrerat, som spelare blir man genast mer varsam om hur man ska gå tillväga. Man vill t.ex. inte gissa sig fram tills man hittar mördaren utan man måste gå på någonting. Om tidspoängen tar slut så kommer en 'Game Over' skylt upp med texten 'Sorry, You Know Too Much' vilket som berättar att mördaren märkt att du börjar närma honom på spåren och därför beslutat att döda dig innan du tar fast honom/henne.

Det här spelet är inte menat att passa vem som helst, det är meningen att vem som helst ska kunna njuta av det, men det krävs väldigt mycket från spelaren (tålamod och inlevelse). Mycket spel som man ser på dagens marknad är rätt 'straightforward', spel där målet är att vara snabbast och skjuta den andra i huvudet föst, inte för att jag tycker att sådana spel nödvändigtvis är tråkiga men jag ville bygga något som kändes mer unikt och där jag lade all 'makt' åt spelaren.

Spelet består av 2 JS filer och en HTML fil samt massa bilder i en mappstruktur. Anledningen till att jag valde att använda 2 JS filer var för att jag ville skilja på själva motorn och datan till spelet. Jag ville göra det så lätt som möjligt för moddare eller bara nyfikna att själv lägga in 'gamecards' i spelet. Därför finns det 1 js-fil som är själva motorn och 1 som innehåller all data om 'gamecards'.

Utöver själva spelet så kan man ifrågasätta mitt val att hårdkoda in 'gamecards' istället för att bevara dem i en databas. Jag gjorde detta valet medvetet för jag ville att spelet skulle vara så enkelt som möjligt att modda och sätta upp, jag ville inte att okunskapen om databaser skulle göra det svårt för någon att ladda ner och sätta upp sin version av spelet. Det hade vart fördelaktigt att använda en databas till korten, då det hade gjort spelet lite snabbare. I framtiden kanske ett stöd för databaser kan implementeras.

Arbetssätt

Jag har jobbat iterativt enligt modellen skolan visat.

Varje vecka (tisdag morgon) har jag haft iterations möte med min grupp och gått igenom vad som har vart problematiskt, vad jag ska göra nästa gång, vad jag har gjort och hur mycket jag har kvar. Under detta möte har jag fört anteckningar som jag använt mig av vid nästa steg. Nästa steg var att skriva en ny iteration, den nya iterationen bestod av 2 filer, en tidsplan och en iterationsplan. I iterationsplanen skrev jag hur föregående iteration hade gått (samt en bifogad version av min tidsrapport från föregående iteration) och vad jag planerar att göra på denna iteration.

Tidsrapporten har jag fyllt på med krav från min productbacklog, men i många fall så hade jag tagit på mig för stora 'tasks' och vart tvungen att slutföra uppgifter från föregående iteration.

Jag har jobbat olika tider olika dagar, jag har satt upp ett antal 'tasks' som skulle bli klara under den iteration jag jobbade på och som hade ett ungefärligt antal timmar som jag kände att jag kunde jobba med under den iterationen. Ungefärlig tid lagd per vecka är 20-35h.

Jag tycker att det var ett bra upplägg för då kunde jag sätta mig och jobba de tider som jag kände skulle bli extra produktiva, istället för att tvinga mig att jobba timmar som jag visste att jag bara skulle sitta och filosofera framför skärmen (Ni vet hur det är när man verkligen inte är på jobbhumör och ändå försöker jobba, man svävar lätt iväg...).

Det har funnits tider då jag har känt att arbetssättet inte fungerat speciellt bra och tider då det har fungerat mycket bra. Att ha uppgifter i backloggen och välja ut dem för varje iteration är ett utmärkt sätt att jobba på, det blir väldigt lätt att se vad man kan göra. Olyckligtvis har jag vart väldigt dålig på att dela upp uppgifterna från backloggen vilket resulterade i att jag tog på mig för mycket på en gång och helt enkelt hade för mycket att göra för att hinna bli klar med projektet i tid. Jag tror att en bra lösning hade vart att dela upp uppgifterna i backloggen innan man lade in dem där, då skulle jag kunna ha en uppgift som heter 'Bygg hudden' som skulle bestå av flera små uppgifter, detta skulle

ge mig ett bättre perspektiv på hur lång tid det faktiskt skulle ta att utföra.

Genomförande/Teknik

Jag har använt mig av javascript (och endast javascript, inga externa bibliotek..) och HTML5 Canvas. Canvas var helt nytt för mig och jag behövde läsa på om det, snart efter att ha kollat på några videor och läst om det så hade jag det jag behövde för att kunna bygga mitt spel. Det jag behövde kunna inom canvas: läsa in bilder, göra knappar, rita upp rutor.

Jag har jobbat lokalt på min dator men har också testat spelet genom dropbox ([länken](#)) och stött på lite problem som jag inte hade lokalt. Problemen hade att göra med att laddningen gick lite segare än väntat och att tecken encodingen inte blev som den skulle.

Jag har använt IDen WebStorm, den har funkat väldigt bra och är ett verktyg som var väldigt effektivt och enkelt att använda! (jag kan rekommendera det). Jag började skriva mitt projekt i Aptana men upptäckte snart att aptana bara kraschar och ville inte ge mig intellisense när jag ville ha den. När jag bytte till WebStorm så märkte jag att den markerar ut 'fel' i min kod som kan vara bättre, det är därför vissa delar av koden innehåller mer 'korrekt' kod än andra delar (majoriteten är dock bra kod.. lite mycket deklarationer på fel ställen kanske..).

Positiva Erfarenheter

Det finns många delar av projektet som jag känner att jag kommer ha lärdom av, så även om jag inte hann klart med det till deadline så har jag ändå lärt mig nya saker och lärt mig uppskatta iterativt utvecklande en hel del. Att jobba iterativt känns som något jag inte gillade till en början, men genom att bryta upp uppgifter började det kännas riktigt användbart då man får en väldigt bra överblick på vad som är kvar och ungefär hur lång tid det kommer att ta.

Jag har fått en bra bild av hur en liten spelmotor kan se ut, jag hade brutit upp min kod i 'maskiner' och 'classer' sedan hade jag lite kod utanför det som stod för att initiera data och spara vissa delar. Men snart upptäckte jag att min uppbyggnad inte var så bra, för efter att koden hade blivit väldigt lång så blev den också svårare att hitta i, om jag hade delat upp koden i maskiner mer så skulle jag kunna fortsätta med samma struktur fast lättare att navigera och hitta i.
(maskiner var till funktioner och klasser var för klasser..).

Jag känner att jag har blivit lite bättre på javascript, innan kände jag att jag kunde javascript hyfsat bra. Men nu har jag fått mer förståelse för det och det känns bra! Jag har inte direkt utvecklats i form av att min javascript har blivit för komplex och jätte kort, men det känns som att viss kunskap som hade försvunnit har nu kommit tillbaka och blivit lite bättre. Jag är dock inte helt nöjd med min kod, det finns många hål för buggar och funktioner som inte är så optimala. I nuläget skulle jag inte ändra på något annat än att försöka fixa buggarna, spelet fungerar fortfarande att "spela" även om det inte ser ut som jag vill.

Jag har kommenterat väldigt mycket av min kod, det finns vissa delar som saknar kommentarer men det är mest for-loopar som (jag ansåg) var självklara. Kommentarererna har hjälpt mig en hel del, både för att påminna mig själv om hur jag kan använda funktioner och för att underlätta vid hårdkodning av 'gamecards'.

När jag insåg att jag inte skulle bli klar med projektet i tid så blev jag först väldigt ställd, det var svårt att acceptera det och jag mårde rätt dåligt över det. Idag när jag skriver det här så känns det lite bättre, jag inser att jag har fått otroligt mycket lärdom av hela projektet. Jag har fått en väldigt tydlig bild av hur man kan jobba iterativt och det känns som om jag har blivit bättre desto längre in i projektet jag har kommit. Jag har förstått vikten av att testa, hur lätt det kan vara. Jag har förtått att uppdelning av uppgifter är väldigt bra både för att få en överblick över vad man ska göra och dessutom ger det en bättre bild av var man kan börja!
Den större delen av projektet har varit roligt, det känns skönt att ha fått göra något som är eget även om det inte blev klart.

Under stress har jag kollat upp saker för att underlätta dem, jag har lärt mig att om man inte kan lösa en uppgift på en kvart så är det ingen idé att sitta längre med den. Ta en annan uppgift och jobba med den tills du kommer på en lösning till det andra problemet.

Jag har fått nya erfarenheter av HTML5 Canvas, allt jag har gjort med canvas har jag lärt mig under denna kurs (i början). Det var inte så svårt som man kunde tänka sig, dock har jag fått bredare uppfattning om hur animering i canvas fungerar vilket som kan bli användbart för att göra en mer avancerad spelmotor.

Handledningspassen har varit bra, jag tror att jag hade varit mycket mer stressad om jag inte hade haft dem. Även om jag inte har behövt så mycket hjälp relaterat till projektet så har man fått en uppfattning om hur andra ligger till och på så sätt har man kunnat känna att man ligger rätt bra till. Dessutom har det varit skönt att man får feedback på vad man har hunnit med.

Negativa Erfarenheter

Testning:

Under projektet gång så är det en sak som jag har haft väldigt mycket problem med, det är testning. Testning är något som låter skrämmande och väldigt tråkigt, men det är bara innan man förstår vad det är. Jag designade mitt spel att kunna köras igång om jag hade tillräckligt många spelkort, detta blev ett problem när jag skulle göra testningar, jag hade helt enkelt inget jag kunde testa på grund av att jag inte hade någon data i spelet. Att lägga till data kändes som något jag kunde skjuta upp till senare, men om jag inte hade gjort det så hade jag snart upptäckt en stor mängd problem som jag stötte på i slutet och som resulterade till att jag inte hann klart.

Det var inte förrän senare jag förstod att jag ska testa en funktion som jag precis skrivit klart, då ska jag testa att skicka in förväntad data och oförväntad data och alla möjliga saker för att se om jag behöver göra några justeringar på funktionen, samtidigt som jag antecknade detta. Jag fick inte många test gjorda, och tyvärr insåg jag detta i ett sent skede.

För stort projekt:

Hela projektet känns nu i efterhand rätt ambitiöst. Jag borde inte tagit på mig ett så stort projekt från första början, jag blev varnad men lyssnade inte. Jag skulle kunna gjort ett liknande projekt utan att behöva så mycket innehåll, för att vara beroende av mycket innehåll gör att onödiga delar tar längre tid. Tror det är bäst att designa ett spel som kräver lite innehåll som man istället kan lägga till mer innehåll vartefter. Men när jag designade spelet så tänkte jag inte att det skulle ta så mycket tid som det tog att implementera en funktion, inte heller att skriva dialoger till gamecardKorten. Jag har fått bättre uppfattning om hur lång tid en "liten" funktion kan ta.

Hårdkodning:

När jag skummar igenom min kod så ser jag att jag har hårdkodat en hel del, jag kan tänka mig att vissa saker har jag bara hårdkodat för att försöka vara snabb, vilket som är ett rätt stort misstag. Flera gånger fick jag gå tillbaka och ändra på hårdkodning på flera olika ställen, detta gäller speciellt när jag angav storlek eller färger till olika föremål. Jag har dock lärt mig av det och beslutat att jag ska bli mer varsam med hårdkodning. Speciellt när det är så mer lönsamt att använda variabler. *(Undantaget är dock den hårdkodade datan för gamecards, dessa skulle vara hårdkodade för att underlätta för moddare som inte skulle behöva krångla med databaser..).*

Duplicerad kod:

Med hårdkodning kommer också duplicerad kod, och jag har en hel del duplicerad kod. I vissa fall har det varit mer nödvändigt än andra, ibland har jag velat behandla mina gamecards på olika sätt, och då har jag haft två funktioner som är väldigt lika varandra. Det finns funktioner som är lika varandra, jag kunde ha löst detta genom att använda mer variabler och parametrar som påverkar datan på olika sätt (istället för att ha två funktioner..).

Strukturen på iterationen:

Jag har haft en rätt bra struktur genom alla mina iterationer. Men det känns som om jag hellre skulle vilja ha haft ett program som hjälper mig att hålla strukturen, det skulle vara skönt med något som hjälpte mig att ta bort från backloggen och lägga in i sprintbackloggen etc.

Jag tror att till nästa projekt så ska jag använda mig av något sånt program för att lättare strukturera mina iterationer, backlog, iteration reviews etc.!

Problemet var att jag ibland fick spendera lite för mycket tid med att hoppa tillbaka mellan olika filer och klippa ut och klistra in. Detta blev rörigt och hjälpte inte när man var lite stressad.

Knappt någon design blev implementerad:

Det som är fixat är till större del är positionerna (det finns några titlar som saknar padding..) av boxar och knappar. Men varken designen på knapparna eller färgen på boxarna är den slutgiltiga, viss boxar var tänkt att bytas ut mot snyggare bilder med text och grafik på, men detta blev inte av då jag lade den mesta tiden på att bygga funktionaliteten. Jag tycker design är otroligt viktigt, men jag ångrar inte min prioritet, utan den funktionaliteten jag kämpat med skulle det helt enkelt inte vara något att lägga designen på.

Fonten i spelet är lite fel, i alla fall i titlar och rubriker, där skulle jag använda samma font som den i menyn. Dock är fonten i dialogerna rätt, den behöver egentligen bara vara läsbar så att byta ut den mot fonterna i menyn skulle bara vara dumt. Även fast jag inte har snygga till designen så fungerar det, det ger inte riktigt den känslan jag vill förmedla men det är lätt att förstå och det funkar.

Sammanfattning

Det finns en hel del att säga, jag tycker att projektet i helhet har gått bra. Jag tror att om jag inte hade gjort mina felsteg gällande att inte lägga in riktig data och att inte testa så skulle nog projektet gått bra! Jag har en hel del buggar som behöver fixas innan spelet är helt spelbart, några som jag redan vet var dem beror på och några som är helt skumma (att bilder laddas in sakta är en av dem... men det kanske beror på dropbox).

Utöver att spelet inte blev klart så finns det en hel del idéer som jag inte ens kunnat fantisera om att implementera men ändå tänkt på. Jag hade till exempel tänkt att lägga till musik, göra det mer synligt när tidspoäng försvann, designa en trigger som gör att vissa dialoger aktiveras när man har gjort en sak i spelet. Lägga till ett anteckningsblock, lägga till stöd för databas där ledtrådar effektivare kunde lagras, etc.

Att ha fått jobba med ett spel är otroligt roligt, jag har inte tagit mycket hjälp av varken guider eller personer när det gäller att ha tagit fram spelmotorn utan allt kommer från mitt huvud. Det jag har tagit hjälp ifrån är hur canvas fungerar. Så man kan säga att jag har 'återuppfunnit hjulet', detta tycker jag är en nyttig erfarenhet som ger en ny vinkel på hela idén med spel och det passar mig då jag gillar att tänka utanför ramarna.

I slutet så är jag ändå nöjd av mig själv och det jag har åstadkommit, jag skulle dock inte göra om detta projekt på det sätt som jag har jobbat. Jag skulle ta en helt annan approach.