

Rabbid Wars

Spelet för dig som vill ha en paus i vardagen.

Projektrapport: David Grenmyr 2014-06-06.

Abstrakt:

Denna rapport handlar om mitt projekt i kursen Individuellt Mjukvaruutvecklingsprojekt (VT2014).

Mitt projekt är ett webbaserat spel för i denna release Firefox eller Google Chrome. Spelet är turbaserat och går ut på att genom placera sig i bra positioner sedan skjuta och förinta motståndaren. Som i denna release är en dator (AI).

I rapporten reflekterar jag över mitt avslutade projekt. Ni kommer få ta del av positiva respektive negativa reflektioner från projektiden. Sista sidan kommer vara en sammanfattning av projekttiden som helhet.

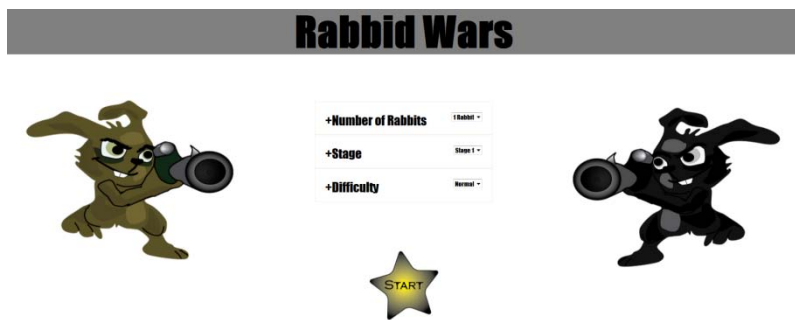
Inledning/Bakgrund:

Visionen med Rabbid Wars var att skapa ett turbaserat spel med retrovibbar ifrån spelet Worms. Spelet fyller sin roll som i pausen på arbetet eller som tidsfördriv mellan lektionerna. Tanken med spelet är att genom en lättnavigerad meny ska användare lätt slussas in i spelet. Och eftersom det är turbaserat kan man enkelt lämna spelet obevakat när pausen är över.

Projektet har utvecklats iterativt med kontinuerlig testning emot gällda krav. Genom kontinuerligt testa produkten så har i efterhand en god dokumentation erhållits. Igenom projektet har jag även försökt lämna delleranser varje vecka till min arbetsgivare genom att pusha på Github Gh-Pages gren.

Projektet har utvecklats med JavaScript och HTML5 Canvas som tekniker och i en utvecklingsmiljö bestående av Webstorm och Visual Studio.

Vy ifrån Meny



Vy ifrån Spel



Positiva erfarenheter:

Enhetstestning var något jag testade på . Jag använde ett ramverk som heter Mocha ihop med biblioteket Chai med stöd av Sinon och Sinon Chai i min miljö för enhetstestning. Se [länk](#) för förklaring. Jag har verkligen bara sniffat på ytan här. Även om Chai ger än väldigt intuitiv syntax så är startsträckan på Mocha och Sinon för mig väldigt lång. Jag hade gärna fördjupat mig inom detta men på grund av bristande tid och kunskap hann jag bara [testa](#) min Gun. JS

Bild ifrån enhetstest.

Gun

update function check

- ✓ should decrease x and y value of projectile with vx and vy

clear function test

- ✓ Test to check it use clearRect method to clear the Gun object from canvas with given parameters

draw function test

- ✓ Test to check draw use fillRect method to draw Gun object from canvas with given parameters

object test

- ✓ Test to check Gun properties of object get correct values on usage of shooter obj. Also test that canvas get initialized.

Funktionstester var väldigt nyttigt också. Att testa implementerad kod emot gällda ställda baskrav och givna testfall gjorde att jag fick en bra dokumentation och det hjälpte mig även i veckoplaneringen då misslyckade testfall är perfekt att lägga in som uppgift i nästa veckas sprint.

Tack vare att genom arbeta med Scrum har jag kunnat planera min tid på ett bra sätt. När vardagslivet har varit väldigt uppbokat har jag ändå kunnat planera och anpassa mig på ett bra sätt gentemot förändringar i tid. Jag har använt en [mall](#) som jag modifierat när jag har sammanställt min dokumentation och planering.

Negativa erfarenheter:

Jag har varit alldeles för ambitiös i hur många baskrav jag la i min vision från början. Det medförde att jag fick ett väldigt stort fokus på uppfylla dessa krav funktionellt i min applikation. Vilket i sig gjorde att jag tappat jättemycket tid för implementering av gränssnittet och spelupplevelsen. Eftersom spelupplevelsen och balans i spelet är väldigt viktigt för slutprodukten så skulle jag behöva kasta baskraven åt sidan ett par veckor och istället fokusera på balans i spelet och spelupplevelse.

På ett sätt har det genom att endast använda JavaScript och Canvas varit stimulerande att få räkna ut vinklar och AI själv. Men med facit i hand så skulle jag försöka ladda ner en fysikmotor som hanterar fysiken i spelet. Att själv skriva kod för kollision och gravitation har varit extremt svårt skapat många buggar och varit tidskrävande för mig. Om jag gör fler spel så kommer jag fokusera mer på spelupplevelsen och mindre på spelets fysiska lagar.

Eftersom jag aldrig utvecklat ett spel innan har jag fått refaktorera kod hela tiden. Jag försökte att skapa en bra struktur men igenom hela projektet har jag gång på gång fått refaktorera. Detta har tagit lång tid och ibland har även oväntade buggar inträffat efter refaktorerings. Förhoppningsvis har jag nu en stabilare grund att stå på så jag tidigare kan sätta upp struktur för kommande applikationer om vad de olika komponenternas uppgift är.

Det har även varit väldigt svårt att få spelets alla funktionaliteter att fungera i de 3 stora webbläsarna. När det funkar med en funktionalitet i en webbläsare bryter det ihop med en annan i en tredje. Just nu i nuvarande release fungerar till exempel inte ljud i Internet Explorer.

Sammanfattning:

Det första som slår mig är hur svårt det är att arbeta mot redan utsatta mål. Jag skrev baskraven tidigt i projektet och har medvetet implementerat kod emot att uppfylla dom. Någon gång i mitten av projektet inser jag att dels är en del baskrav felaktigt formulerade och en del skulle kunna tas bort och flera läggas till. Jag kan tänka mig dilemmat i en release för kund när de ställt upp vissa krav och när produkten är färdig så har man fått en taskig produkt för baskraven var felaktigt ställda. Jag utgår ifrån att man med erfarenhet och kommunikation med kund kan undvika dessa misstag. Jag hade själv i mitt projekt kunnat ändra eller slopa ett par baskrav för ett snabbare och eventuellt bättre resultat av produkt. Men jag valde att ha dom kvar.

Det känns som denna release är en prototyp att arbeta vidare på. Som man kan testa spelets balans, upplevelse och kodstruktur med för att sedan implementera om i en ny release. Det finns mängder mer roliga spel objekt man skulle kunna krydda produkten med. Andra mål i framtida version skulle vara att införa spel mot andra användare.

Tiden jag har använt har verkligen gett mig en ljupare kunskap om programmering i JavaScript generellt. Men särskilt i hur man bäst implementerar olika former av media. Eftersom det spänt allt i mellan rendering av fps, bilder, ljud, animering så känner jag en mycket bättre generell kunskap där.

Andra lärdomar som jag kommer ha nytta av i framtida kurser/arbeten är enhetstestningen av Gun. JS som jag gjorde. Trots att jag bara har glipat lite på dörren när det gäller enhetstestning så kommer jag absolut ha nytta av den kunskap och generella insikt om hur det fungerar som jag fick. Till att arbeta testdrivet är det en lång väg, men att testa på enhetstestning fick mig att inse dilemmat med funktioner som gör flera saker. Man har ju blivit predikad att en funktion ska leva sitt egna liv och endast uppfylla sin funktion. Men genom enhetstesta har jag förstått nyttan av detta. Dels så blir det mycket lättare att testa. Men även kodstruktur blir mycket tydligare och förhoppningsvis behövs kod inte refaktoreras lika mycket i mitt nästa projekt.