

Slutrapport kursen Individuellt utvecklingsprojekt, 1DV430, LNU

Projekt: Chaser

Johan Söderlund, 20180601

Abstrakt

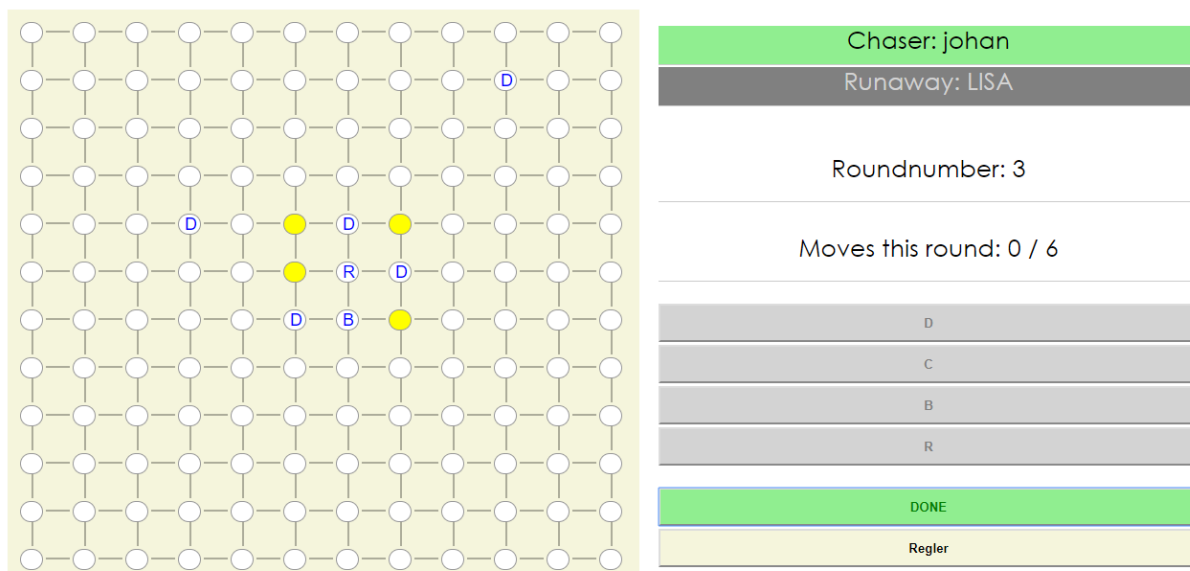
Denna rapport behandlar det projekt som utmynnat i ett strategispel för webben som jag döpt till Chaser och de erfarenheter jag dragit i samband med det. Inspirationen till projektvalet är mitt intresse av brädspel och spelutveckling. Att bygga en realtidsapplikation med hjälp av websockets med NodeJS och ReactJS visade sig vara helt görbart även på den relativt korta tid som fanns till förfogande för detta projekt.

Inledning och bakgrund

Chaser spelas mellan två spelare där en spelare har rollen "chaser" och en rollen "runaway". Spelet går ut på att chaser ska försöka fånga in rymlingens spelpjäs genom att förflytta en detektiv till samma position på spelbordet som rymlingen. Rymlingen vinner genom att hålla sig undan genom samtliga spelrundor.

Målet har varit att göra ett spel som är lätt att lära, går snabbt att spela men som samtidigt premierar strategiskt tänkande. Typisk målgrupp är folk som åker buss eller tåg till eller från arbete eller studier.

Projektet har bedrivits genom ett agilt arbetssätt med korta iterationer. De tekniker jag använt är [Node.js](#), [Reactjs](#), [Websockets](#), [MongoDB](#), [Mocha](#), [Chai](#).



Spelets användargränssnitt.

Positiva erfarenheter

Jag är väldigt nöjd med att på denna relativt korta tid komma i mål med en fungerande applikation som täcker in baskraven och kravspecifikationen. Jag är också glad över att jag tycker den har ett ganska högt spelvärde. En typisk förströelseapplikation som är enkel att lära och går snabbt att spela.

Projektet har fortlöpande gett mig nya ideer för hur jag vill vidareutveckla applikationen för att ta den från denna prototypnivå till mer kommersiell nivå där slutmålet alltså är en mobilapplikation. En anledning till det tror jag är det iterativa arbetssättet där produkten succesivt växer fram. Ideerna finns listade under product backlog i projektets wiki.

För utvärdering har jag också låtit ett antal personer spela spelet vilket också gett mig värdefull återkoppling för kommande versioner.

Websocket var smidigt och fungerade bra redan i ett tidigt skede. Här ska sägas att jag hade god hjälp av de exempel och tutorials som finns på webben.

Utveckling med React gör till sin natur att logiken delas mellan servern och klienten jämfört med en arkitektur där logik och vy är striktare separerad. Just genom att det blir så invävt i skapandet av själv domen (GUI) gör att det blir svårare att isolera och automat testa. Den positiva erfarenheten är att väl utbyggda manuella tester då visat sig ha ett stort värde. Jag har under utvecklingen identifierat flera buggar i spellogiken genom att kontinuerligt återupprepa dem manuella testerna vid större kodförändringar eller tillägg. Jag valde tidigt bort att försöka skapa automatiserade tester av klienten till förmån för mer omfattande användarfall kopplade till motsvarande manuella testfall.

Server och databashanteringen har jag skrivit testsviter för enhets- och beteendetester. Även dessa avslöjade ett antal buggar och brister.

Negativa erfarenheter

Att använda React hade lite större tröskel än jag förväntat mig. Koden blev initialt något klumpig och dåligt modulariserad innan jag blev mer bekväm med hur det fungerar och den hierarkiska strukturen det bygger på.

Driftsättningen hade jag lite större problem med än väntat. Prövade flera olika sätt men landade på att skaffa en VM på Amazon och köra nginx där. En fallgrop här var att det var förhållandevis enkelt att driftsätta en indexsida vilket jag gjorde tidigt i projektet, men svårare med en klient-server arkitektur med en websocketkommunikation. Konfigurationen av "reverse proxy" och brandväggs regler blev därmed mer komplex. Egen felsökning blev lösningsmetoden då tillgängliga turtorials och exempel inte var lika tydliga för hur det skulle implementeras. Totalt tog det mig nästan två dagar att få ordning på driftsättning och den initiala riskminimeringen med att driftsätta en indexsida skapade därmed en falsk trygghet.

Skriva, utföra och dokumentera de manuella testfallen tog betydligt längre tid än jag först uppskattade. Det beror till viss del på att de blev mer omfattande och att jag höjde ambitionsnivån när det gäller dokumentationen av dem med alla bilder jag klippte och klistrade in. I den absoluta slutfasen gjorde jag en del CSS förändringar av GUI så bilderna i testrapporten stämmer inte längre riktigt med nuvarande utseende.

Andra erfarenheter

Spellogik: Med React blir konsekvensen gärna att logiken delas mellan server och klient. Att de båda spelarna dessutom har olika regler gör implementationen rejält svårare eftersom det ändå ska ligga i samma kodbas. Samma Reactkod skickas upp till båda spelarna oavsett roll. Däremot distribuerar

servern delvis olika information vid uppdateringar till respektive klient. Totalt sett har det tagit längre tid att få spellogiken bugfri, vilket jag tror den är nu, än jag trodde.

GUI: Är inget jag haft höga ambitioner med i detta projekt, vilket är helt enligt plan men absolut något jag funderat på för kommande versioner. Det är också kopplat till en framtida portering till mobila plattform.

Vad kunde jag gjort bättre?

Mockups och design: det är lätt att fastna i kodning och göra för stora bitar i taget istället för att lägga mer tid på att planera för hur man kan implementera mindre delar och testa av dem successivt. Har jobbat iterativt men det finns förmodligen förbättringspotential i just planeringen av hur det ska genomföras på ett bra sätt. Jag vill gärna komma igång snabbt med att implementera mina ideer och testa dem och få kvitto på om de håller, så snabbt som möjligt.

Att arbeta iterativt och inkrementellt kräver en bättre planering i inceptionfasen. Jag måste bygga erfarenhet av att det lönar sig och den här kursen till en del går ut på.

Sammanfattning

Jag har huvudsakligen positiva erfarenheter av de tekniker jag valt att arbeta med. Tutorials, forum och bloggar på webben förenklar och utgör ett ovärderligt stöd vid utveckling, i synnerhet när man använder tekniker som man inte är helt familjär med som var fallet i detta projekt.

Mitt slutintryck av ReactJS är väldigt positivt då det förenklar bygget av vyn när man väl kommit förbi den initiala inlärningströskeln och det är något jag kommer att använda mig av igen. Jag har tidigare haft en preferens för ASP .NET och C# men har i andra projekt gått in i väggen med dessa tekniker beroende på att de förändrats under projektens gång och att använda nylanserade tekniker inom .NET har för mig visat sig medföra stora problem.

ES6 har definitivt gjort att jag tagit till mig Javascript i större utsträckning. NodeJS är en enkel och bekymmersfri plattform. Websockets fungerar enkelt och är användbart och egentligen nödvändigt för att bygga många typer av applikationer över webben.

På det hela taget ett roligt och i mina ögon lyckat projekt.