



# TDP005 Projekt: Objektorienterat system

## Refelektionsdokument

Författare

Jonathan Dandemar, [jonda429@student.liu.se](mailto:jonda429@student.liu.se)



Höstterminen 2015  
Version 1.0

2015-12-19

## Revisionshistorik

| Ver. | Revisionsbeskrivning | Datum      |
|------|----------------------|------------|
| 1.0  | Skapade dokumentet   | 2015-12-19 |

# 1 Arbetsgång

Vi skulle egentligen vara tre personer i gruppen men ena medlemmen var frånvarande under i stort sett hela projektets gång. Det ledde till att början av projektet blev ganska jobbigt eftersom det var svårt att veta hur vi skulle dela upp arbetet när ena medlemmen var med ena dagen och borta nästa. Vi insåg dock ganska tidigt att det var enklast att räkna med att vi skulle göra projektet på två personer och efter det så fortgick arbetet bättre. Det var dock ett tråkigt problem att behöva ha att göra med och jag tror tyvärr inte att jag är så mycket klokare av hur liknande situationer ska hanteras. Det känns som att jag möjligtvis hade kunnat hjälpa till mer med att få in medlemmen i gruppen men samtidigt så fick han en uppgift att göra precis som oss utan att göra den vilket ledde till att jag fick göra den istället och sådana situationer är av naturliga skäl ganska frustrerande. Självklart vill man förstå och hjälpa någon som inte är helt med men samtidigt så tycker jag att alla borde kunna ta sitt egna ansvar för att bidra till ett projekt på den här nivån, eller iallafall försöka, vilket inte var fallet den här gången.

Vi började projektet med att göra en OOA för att få en överblick av klasserna vi trodde att vi skulle behöva i spelet, klasserna skrev vi upp på post-it lappar som vi använde för att visualisera hierarkin vilket jag tyckte var enormt givande då det gjorde så många andra saker lättare jämfört med att börja koda direkt. Nu behövde vi i slutändan ändå lägga till och ändra ganska mycket men den initiella analysen var ändå en av de viktigaste momenten i hela projektet.

Vi försökte dela upp arbetet så mycket som möjligt genom att arbeta parallellt med olika delar av projektet vilket fungerade bra och fungerade så att vi helt enkelt tog en post-it var från övre delen av klasshierarkin och kodade den. Det fungerade bra första dagarna men när det inte var så många klasser kvar och vi började komma till implementeringen av klasserna så delades projektet upp i två huvuddelar där vi då ansvarade för varsin. Det här var ingenting vi egentligen diskuterade utan det föll sig naturligt i arbetsgången. Det ska sägas att vi satt bredvid varandra på campus och kodade under hela projektet minimerade möjliga kommunikationsmissar om den här typen av uppdelning av arbete vilket har varit bra.

Den huvudsakliga uppdelningen av arbetet blev så att jag min partner skrev mer av den underliggande arkitekturen i spelet medan jag skrev allt som syntes på skärmen, jag fick göra allt det roliga arbetet kan man säga. Med underliggande arkitektur menar jag främst gameloopen som i vårt spel sköter i princip allt i spelet genom att dirigera mellan olika gamestates. Så arbetet blev så att min partner skapade den och dess tillhörande gamestates medan jag då implementerade klasserna för spelaren och de olika fiendetyperna. Eftersom vi jobbade parallellt så fanns det såklart ingen gameloop att köra vid testning så under arbetets gång så hårdkodade jag i princip hela spelet i en separat mainfil för att kunna se om rörelser och kollisionshantering fungerade som det skulle i spelet.

Arbetet blev rätt så tidspressat dels för att andra kurser har tagit tid men också för att vi började några dagar sent samt att vi som nämnt hade vissa besvär med en gruppmedlem i början av projektet. Tidspressen i sig har dock inte varit något större problem utan det har snarare varit ganska roligt att det har varit intensivt från början till slut, det beror antagligen mycket på att arbetet har gått bra de flesta dagarna och det hade antagligen varit jobbigt annars. Vi hann iallafall klart till slut även om mycket om tilläggen som vi hade velat implementera inte har hunnits med.

Överlag så har arbetsgången och arbetsuppdelningen varit bra under projektet, eftersom vi har suttit bredvid varandra hela tiden så har vi kunnat hjälpas åt mycket och det har varit väldigt skönt att alltid ha nån att bolla idéer med.

## 2 Ny programmeringskunskap

Projektet har varit väldigt lärorik för min utveckling som programmerare då det har gjort att jag har fått mer kunskap och vana med att programmera i C++. Det har varit givande att använda SDL för att få en bättre förståelse för hur grafik och programmering kan vara sammankopplat, jag vet inte hur andra liknande ramverk fungerar men SDL verkar som en bra grund att arbeta med på den här nivån. Det känns även kul att veta att jag nu kan utveckla den här typen av program som faktiskt syns mer än som text i ett terminalfönster och som kan reagera på flera typer av input från användaren.

Spelet som vi har skapat innehåller väldigt mycket rotation och vinklar då olika objekt flyger åt olika håll och snurrar och byter riktning hela tiden. Det har inneburit att jag har behövt läsa upp på gamla trigonometrikunskaper vilket har varit förvånansvärt intressant och framförallt så känns det som att det kan vara bra kunskap att ha nära till hands för liknande projekt i framtiden eftersom matematiken är densamma vid byte av programmeringsspråk.

Många koncept som inte var helt klara i C++ innan har även blivit lite tydligare under projektets gång, främst då hur man kan använda pekare och vektorer vilket känns som en väldigt viktig del av C++.

Den största och mest centrala lärdomen handlar däremot om objektorientering och klasser vilket jag känner att jag har lärt mig enormt mycket om jämfört med det jag visste innan. Jag hade inte riktigt förstått poängen av objektorientering innan vi började implementera klasserna som vi hade definierat i vårt spel och vi började få igång arv och polymorfism och jag förstod precis hur användbart det kan vara. Det som jag tyckte var allra bäst med det i vårt projekt var att vi använde en virtuell update funktion i alla objekt, det ledde till lite frustation i början just för att vi ofta kände att vi ville kunna skicka med argument som behövdes för de olika objekten. Vi höll dock fast vid att ha en update som inte tog några argument och arbetade runt alla problem genom att använda setters, getters och medlemsvariabler och tur var väl det för det gjorde allting så grymt fint i gameloopen sen eftersom det enda vi behövde göra för att uppdatera spelet mellan iterationer var att gå igenom en vektor med pekare och köra update på objektet den pekade på. Hur smidigt det är tycker jag är helt fantastiskt såhär på efterhand. Självklart så har kursen även gett mycket praktiskt kunskap om objektorientering men för min del så är den största lärdomen från projektet en mer abstrakt förståelse av hur bra abstraktion är i programmering och hur stor del objektorientering måste ha i större system med tanke på hur stor del det hade av vårt lilla spel.

En sak som kan vara värd att nämna är att vi inte använde eclipse i projektet överhuvudtaget, vilket såhär på efterhand är lite tråkigt men samtidigt så har jag skrivit lite kod i visual studios förut så jag hade iallafall lite erfarenhet av att jobba med en IDE sedan tidigare. Istället för Eclipse så använde vi oss av Emacs och terminalen.

## 3 Kort sammanfattning

Vi hade vissa problem med en gruppmedlem vilket jag tyvärr inte tycker att jag har lärt mig så mycket av mer än att jag har fått erfarenhet av en situation där alla parter inte riktigt gör vad de ska. Arbetsgången gick dock bra ändå och det var ju egentligen inget problem att vara två i en grupp eftersom de flesta grupperna var två.

Det jag tar med mig från projektet är främst en mycket större förståelse för objektorientering, en större kunskapsbas i C++ och SDL rörande både koncept och syntax och även lite förnyade trigonometrikunskaper som kan vara nyttiga i framtiden.