

TDP005 Projekt: Objektorienterat system

Reflektionsdokument

Författare

Denis Ivan Blazevic, denbl369@student.liu.se

Innehåll

1	Revisionshistorik	2
2	Vad har jag fått ut av kursen?	2
3	Hur gick det att arbeta?	3
4	Vilka svårigheter stötte jag på?	3

1 Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.0	Första versionen av reflektionsdokumentet.	161218

2 Vad har jag fått ut av kursen?

Enligt studiehandboken ska följande punkter följa med mig efter avslutad kurs:

- Goda kunskaper i att använda ett objektorienterat språk och dess bibliotek
- Konstruerat ett mer omfattande program som följer den objektorienterade metodiken
- Kunskaper om god källkoddesign
- Förmåga att förbättra programvara genom att hålla och delta i kodgranskningsmöten

Det som står ovan tycker jag stämmer rätt så bra med den kunskap jag har fått med mig under kursens gång. Jag har fått utnyttja den kunskap jag har fått i C++, och skapa ett spel tillsammans med en annan student.

Att gå från att programmera direkt utan att tänka till på vad som ska göras, till att programmera efter att man har lagt upp hur gången under projektet kommer se ut - är ett väldigt stort kliv som kommer bara underlätta för mig framöver. Detta är första gången jag verkligen planerar ut hur allt ska se ut och vad som ska göras innan jag programmerar, och det kommer jag fortsätta med att göra eftersom det är otroligt mycket enklare att göra det på detta sätt.

Genom att konstruera spelet har jag och min projektpartner lärt oss mycket om att samarbeta ihop och hur man gör det på bästa möjliga sätt. Till exempel satte vi oss ner i början av projektet och pratade om koduppbyggnad och hur den ska se ut. Vi beslöt oss att skapa vårt projekt genom att använda oss utav komponentbaserad uppbyggnad.

Jag och min projektpartner har försökt att hålla en god källkodsdesign genom att alltid försöka skriva självförklarande kod. Det vill säga att vem som helst ska kunna kolla på koden och ungefär förstå vad som händer utan att kommentarer behövs. Självklart har vi även kommentarer i koden som förklarar vissa saker där det inte är så självklart längre.

Samtidigt som vi försöker skriva självförklarande kod, skriver vi även enligt den kodstil vi har beslutat att använda. För att verkligen följa en och samma kodstil använder vi oss utav Reformat Codefunktionen som finns i CLion. Genom att mata in den kodstil man vill använda i projektet genom projekt inställningarna i CLion, kan man med hjälp av funktionen formatera koden så att den följer den kodstil man har satt. Detta underlättade väldigt mycket för oss då vi både hade olik kodstil, men i slutändan så ser koden exakt likadan ut.

Veckomöten måste jag säga att det inte gav mig så mycket. Jag tyckte att det var onödigt i denna kurs. Det var inte så givande. Däremot vet jag utav personlig erfarenhet att veckomöten är himmelskt i projekt där man arbetar väldigt nära kunden, eller där alla grupper delar med sig vad som sker.

Anledningen till varför just dessa möten inte var givande tror jag beror på det inte var så verklighetstroget med en beställargrupp som inte faktiskt beställde en produkt. Visserligen läste vi igenom varandras spelidé och hjälpte varandra när det behövdes men det var inte samma sak som att gå till beställaren och ställa frågor om produkten som de kan ge konkreta svar på. Egentligen var vi beställare till vår egen produkt så att säga, och då brast möten.

3 Hur gick det att arbeta?

Det gick väldigt bra att arbeta ihop med Frans Bergström! Vi visste direkt hur vi ville lägga upp arbetet och vem som skulle göra vad. Eftersom jag arbetar bättre på kvällar och nätter så jobbade jag för det mesta på kvällar och nätter medan Frans jobbade då på dagarna och kvällar.

Genom att använda oss utav Git på ett lämpligt sätt undvek vi konflikter när vi arbetade. Med hjälp av Git använde vi oss utav branches för nya features, buggfixar och allmän fix som behövdes.

Eftersom vi fokuserade på kursen TDP004 fick vi det stressigt sista veckan innan projektets deadline. Det som vi gjorde då var att sitta med projektet mer än vi gjort förr, och kodade samtidigt bredvid varandra så att vi kunde dela med oss all information och kunskap den andre kanske inte hade.

Hursomhelst så gick det bra och vi fick ihop ett spel som man vill köra konstant för att få bra slutpoäng. Vi fick ändra lite i våra krav, men vi ändrade inget stort, så spelet följer fortfarande kravspecifikationen men vissa funktioner är inte implementerade eller så är de implementerade på andra sätt än vad vi tänkt från början.

4 Vilka svårigheter stötte jag på?

De svårigheter som uppstod var bland annat att disponera tiden rätt mellan TDP004 och TDP005. Vi fokuserade väldigt mycket på TDP004 labbarna och för-tentamen. När vi väl var färdiga med dem så märkte vi att tiden för TDP005 började ta slut, och då satte vi igång på den högsta växeln och kodade nonstop.

Det jag kommer göra i framtiden är att göra ett mer tydligare schema för mig själv när det gäller kurser som går parallellt med projektkurs så att samma problem inte uppstår.

Ännu en svårighet som dök upp var att lära sig vad och hur komponentbaserad koduppbyggnad innebar och hur man implementerar det på bästa sätt. Detta var inte utav den dåliga typen av svårigheter, istället var det en svårighet som gav mig en del kunskap i området. Nu förstår jag skillnaden mellan vanlig koduppbyggnad med klassarv och komponentbaserad koduppbyggnad mycket bättre, och kan med en säkerhet välja rätt koduppbyggnad till nästa projekt.