Model View Controller

Kompleksitet

Allerede på sprint 1 diskuterte gruppa hvordan vi skulle takle komplekstiten i kodebasen. Siden ingen hadde erfaring med større prosjekter fra tidligere var ingen spesielt opplyst på strategier for å takle den slags. Vi hadde fått hint fra student assistent om at Model View Controller var hensiktsmessig, så vi bestemte oss for at alle medlemmer skulle lese seg opp på temaet MVC. Det var dog ikke før sprint 4 at vi endte opp med å implementere MVC – behovet for struktur meldte seg ikke før vi skulle implementere db med grensesnitt.

Øivind fant ut at MVC kunne implementeres på forskjellige måter i java, men at hensikten til MVC var å systematisk strukturere kodebase i 3 deler, slik at programflyt og logikk er i controller, grafisk grensnitt befinner seg i view og all data som programmet bruker ligger i model. Derfor gir faglitteratur heller ingen absolutt implementasjons-mal, og poengterer at vi står fritt i valg av hvordan implementasjonen gjøres avhengig av programmets krav og programmeringsspråkets egenskaper. MVC er kort sagt et arkitektonisk mønster som strategisk takler kompleksiteten, men programmereren står fritt likevel fritt til å velge en taktikk for hvordan dette skal implementeres.

De tre første sprintene ble brukt til å prototype database, grafisk grensesnitt og gjøre oss kjent med verktøyene. Ikke før vi begynte å sette sammen løsningene meldte behovet seg for MVC. Hvordan skulle vi elegant separere GUI-kode fra databasen? Gruppen hadde nå blitt delt inn i 2 fraksjoner – GUI og database. Dette gjorde at behovet for å strukturere kode i henhold til MVC ble enda mer trykkende. Vi valgte derfor å lage mappestrukturen i henholdsvis tre java pakker, model, view og controller. Pakken ’model’ inneholdt all database-spesifikke filer, ’view’ inneholdt alle GUI filer. Pakken ’controller’ hadde på dette tidspunktet ingen filer.

Vi så oss nødt til å lage controlleren etter hvert som grensesnitt-sidene ble ferdig-lagd. Øivind og Sigurd fra database-gruppen ble enige om Øivind skulle lage kontrolleren til view, men i mangel av javadoc på daværende tidspunkt, valgte vi å parprogrammere mange av controller-logikken, hvor Sigurd inntok rollen som model-ekspert og veileder for Øivind. Forsinkelser i sprint 5 fra gui-gruppa førte til at vi måtte avvente med å ferdigstille controller inntil alle hadde lagd ferdig grensesnittene. Sigurd brukte sprint 5 og 6 til å oppdatere deler av model, og controller etterhvert som vi ble bedre kjent med behovene i grennsesnittene.