*Wafari - Word Safari*

Et bilde som inneholder kalender

Automatisk generert beskrivelse

Vi hadde lyst til å lage en app som kan være interessant for mange, og som man til og med kunne publisert. Da var en ordspill app den perfekte ide. Wafari (word-safari), er en applikasjon hvor man skal finne ord i en stor bolk med bokstaver. Brukeren velger enten tema og antall ord, eller så lager man sine egne ord til et tema man velger selv. De temaene man kan velge i menyen har alle en rekke ord som passer til temaet. Når man velger antall ord og et tema vil det bli valgt antall ord som er tilfeldig valgt fra det temaet. Basert på disse ordene produserer en stor tekstbolk som inneholder ordene. Det samme vil skje om man skriver inn egne ord. Når du har tekstbolken på skjermen får du også en “input field” hvor du skal skrive inn ord du ser av det gitte temaet. Når du har skrevet inn alle ordene her er du ferdig, og har “vunnet”.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Spørsmål 1:

Vi har brukt klasser og objekter i prosjektet vårt ved alle klassene vi har laget. I disse klassene har vi metoder, som alle er innkapslet og validert på riktig måte. Dersom de får feil tilstand vil de gi ut feilmeldinger. Interface er blitt brukt i Grid klassen. Den implementerer metodene som er viktige å ha for en grid. Filbehandling er blitt brukt mye i prosjektet vårt. Alle temaene og deres ord ligger i en tekstfil som både leses til og fra. Når man skal lage en grid og velger seg et tema og antall ord er det første som skjer at FileManager klassen leser all informasjon ned fra words.txt filen. Til denne txt filen blir også nye temaer med ord lest inn. Dette er en viktig del av prosjektet, siden det gjør oppgradering av appen mye enklere. Delegeringen er noe annet som er helt essensielt for prosjektet. Når du vil ha en ny grid og bestemmer hvilket tema og antall du vil ha er det ikke grid klassen som lager en “solution”. Det arbeidet blir delegert til en Solution klasse. Denne klassen delegerer igjen arbeidet om å lese innholdet fra en fil til FileManager klassen. FXML er tatt i bruk i “frontenden” av prosjektet, for at utseende på appen skal bli slik vi vil ha den. I “Backenden” har vi regelmessig brukt testing for å sikre at ting fungerer slik det skal.

Spørsmål 2:

Vi har ikke tatt i bruk interface av typen comparator i prosjektet vårt. Dette hadde egentlig ikke passet inn, ettersom vi ikke har noe vi vil at skal sorteres. Iterator interface er heller ikke tatt i bruk. Dette kunne blitt tatt i bruk ved flere anledninger. Når man skal ha en solution itererer man for eksempel gjennom en liste med ord. Når man skal utheve ordene som har blitt funnet i griden må man iterere gjennom den doble listen for å finne bokstavene som skal bli uthevet.

Observatør observert er ikke tatt i bruk. Dette kunne vi tatt i bruk ved hjelp av en “score” klasse følger med på hvor mange ord du har klart å finne. Dersom et ord blir “highlighta” i griden vil den få beskjed, og legge på en på scoren.

Spørsmål 3.

Koden vår forholder seg meget bra til Model-View-Controller-prinsippet. Da vi begynte med prosjektet vårt, startet vi med logikken til applikasjonen, altså modellen til applikasjonen. Først deretter begynte vi med den grafiske delen. Ved hjelp av Scenebuilder ble det grafiske brukergrensesnittet realisert. Det siste som gjensto var å binde dette med det logiske, ved hjelp av kontroller. Det er dermed minimalt med logikk i kontrollene våre.

Spørsmål 4.

Underveis i prosjektlagingen har vi testet mye. Samtlige klasser har blitt testet om de funker som de skal. Ved filhåndtering testet vi om dataen ble skrevet ut riktig. Det gjorde vi ved å printe ut resultatet da vi henta dataen i filen. Det var det samme vi gjorde da vi var i startfasen av «Solution» og «Grid» klassene. Da var vi sikre på at backend delen av prosjektet fungerte slik det skulle. Da vi testa ut javafx delen av prosjektet var det veldig greit å kjøre prosjektet får så å se hva som funka og ikke. Vi fikk også venner til å teste for å se om alt var implisitt.

Testene i prosjektet våre omfatter modellen av prosjektet, og ikke så mye av brukergrensesnittet. Dette er fordi vi tenkte at det var viktigst at logikken faktisk fungerte i applikasjonen. Testene tester Grid klassen, Solution klassen, og FileManager klassen. Vi har dermed ikke testet alle deler av koden, med alt det som omfatter modellen.