TDT4100 – JAVA – Prosjekt

Sjakk

Planlegging av prosjekt

Del 1

Beskrivelse av appen: Kort om spillet/appen, hva skal den gjøre? Hva er målet med spillet?

I sjakk har vi to spillere, hvit og sort, som spiller mot hverandre på et brett med 8x8 firkanter. Hver spiller har brikkene bonde (8), tårn (2), løper (2), hest (2), dronning (1) og konge (1). Hver type brikke har en unik måte å bevege seg på brettet. Alle brikker har mulighet til å «ta» motspillerens brikker.

Spillet vinnes ved at kongen står i sjakk matt (står i sjakk og kan bare sette seg selv i sjakk). Remis (likt) kan oppnås ved at begge spillere blir enige om remis og f.eks begge lag har bare sin konge igjen, eller får tilfellet patt (kongen står ikke i sjakk, men kan bare sette selg selv i sjakk).

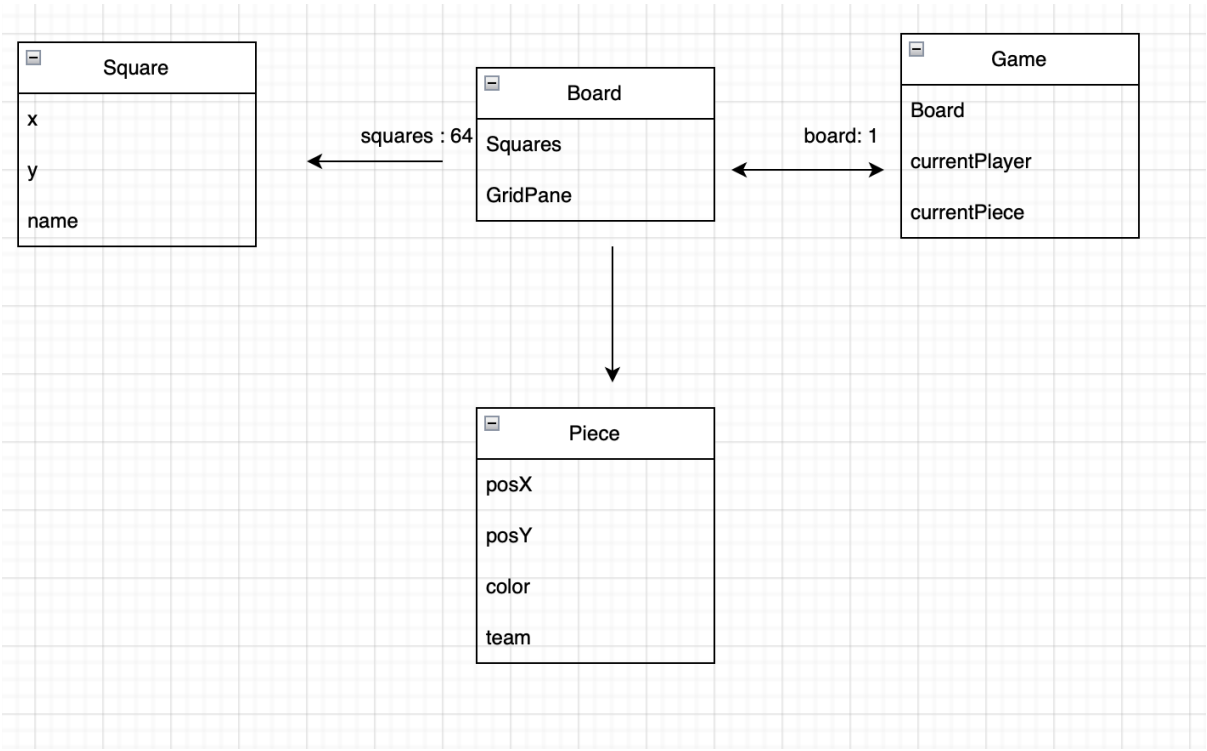
Appen min er et sjakk spill bygd på klassene i Java FX. Vi har en GridPane som vil være fundamentet til brettet. Hver rute er en StackPane lagt oppå GridPane-et. Vær brikke arver fra ImageView som er en klasse hvor man kan ha et eget bilde som det «fysiske» objektet til klassen. Disse legges oppå rutene og endrer rute den har okkupert når spillet utvikler seg. Alle brikker har vær sin klasse hvor deres karakteristiske trekk er definert.

I Board klassen vår Legge vi til et felt som skal være av typen GridPane som er en container i SceneBuilder. I konstruktøren kjører vi metoden makeBoard som kjører for-løkker for å opprette et «grid». For hver rute i vårt GridPane (som vi oppretter en ny til i neste iterasjon i for-løkken) legger vi til en Square i GridPane. Vi må da gjøre slik at klassen Square arver fra StackPane (en annen type container i SceneBuilder) slik at den kan fungere som en Node som legges på toppen av være rute i vårt GridPane.

Sjakk brettet er expandert med et ScoreBoard til høyre. Disse to grensesnittene er separert og har vær sin controller for å initialisere tilstanden. Begge ligger under samme SplitView som blir lagd i app klassen. Når et trekk blir gjort skrives trekket til en fil av klassen FileWriter, deretter er den en annen klasse GamesFileReader som leser av filen og skriver det den finner til scoreboard grensesnittet.

Del 2

Klasse Diagram



Del 3

Prosjektet dekker deler av pensum som arv, app programering, objektet og klasser, osv. En veldig viktig del åssen prosjektet er mulig er ulik bruk av arv. For eksempel så består mye av modellen av manuell programmering av javaFX klasser. Det som gjør det. Mulig for meg å sette opp brettet er at Board klassen er en arving av GridPane klassen, og Square klassen er en arving av StackPane klassen. Det gjør det da mulig for meg å opprette et 8x8 brett hvor 64 stackpanes ligger oppå en gridpane. Også er hver individuell brikke klasse arving av Piece klassen som igjen er arving av ImageView klassen som kan legges oppå en StackPane. Prosjektet tar også bruk av forskjellige måter for filbehandling, både lesing og skriving til fil.

Kunne kanskje ha hatt flere interfacer for å skape mere struktur i modellen. F.eks. en Square interface. Det kunne kanskje også vært bruk av lambda uttrykk i logikk delen av modellen. Også mer bruk av standard teknikker som f.eks. observert teknikken mellom hver brikke og hver rute. Mye av spillets funksjonalitet er bygd på ting utenfor pensum. Kanskje den viktigste delen er bruk av EventHandlers og MouseEvent som tar hånd om musetrykk. Dette gjør det da mulig å spillet et digital sjakk spill med musen og klikking som styrer.

All logikk og tilstand er satt i modellen. Controllerne brukes kun til å sette sammen modellen med grensesnitt objektene og initialisere tilstanden. Dette gjøres at det er et klart skille mellom dem og skaper struktur.

Testene er valgt ut ifra det jeg mener er viktigste for at appen skal fungere slik jeg vil. Som da er at ruters og brikker koordinater på brettet stemmer overens med endringen av deres tilstand. Har prøvd å lage tester for spill logikken, men fant fort ut at det kreves mye ekstra koding fordi hele start tilstanden til spillet blir satt i modellen og endringer skjer med tastetrykk istedenfor metodekall slik at det kreves en spiller for å teste logikken.