



NTNU

Kunnskap for en bedre verden

Kollokvium 2 - Oppgaver

TDT4100 Objektorientert programmering

Mandag 17.02.2020



Oppgaver

Vi skal utvide implementasjonen vår av en forenklet versjon av spillet [Snakebird](#) til å være en grafisk versjon.

Koden som er utdelt er en implementasjon av en halvferdig grafiskversjon av spillet gjort i foregående øvingsforelesninger og kollokvium. Koden består av to klasser, **Tile** som representerer en rute på spillbrettet og **Game** som representerer hele brettet og logikken i spillet. Samt en kontroller og app klasse med tilhørende FXML.



Spillregler

En oversikt over spillets regler finnes i forrige øvingsforelesning.

Pakker

Utdelt kode og oppgavene finnes i mappen foreksempel/src/kollokvie2.

`foreksempel/src/kollokvie2.kode`

Her finner dere utdelt kode og kan skrive deres egen kode.

`foreksempel/src/kollokvie2.underveis`

Her blir kode lagt inn rett etter felles gjennomgang.

`foreksempel/src/kollokvie2.test`

Her ligger kode som kan brukes til å teste at en oppgave er løst riktig

`foreksempel/src/kollokvie2.lf`

Blir gjort tilgjengelig i etterkant og inneholder et løsningsforslag for oppgavene.



Oppgave 1

 7 min

Motivasjon

Vi er nå klare til å fargelegge brettet.

Oppgavetekst

Lag en funksjon **drawBoard** i kontrollerklassen som oppdaterer bakgrunnsfargen til hvert av **Pane**-objektene som representerer rutene i brette med den fargen som **getTileColor** returnerer for den spesifikke ruten. **drawBoard** skal kalles etter **createBoard** i initialiseringsmetoden.

Testing

Ved riktig implementasjon skal kjøring av appen vise brettet med farger.

Oppgave 2

 5 min

Motivasjon

For at spilleren skal kunne skjønne hvor hodet til slangen er, må vi representere dette med en annen farge.

Oppgavetekst

Lag en funksjon **isSnakeHead** i **Game** som tar inn et **Tile**-objekt og sjekker om dette er hodet til slangen. Endre **getTileColor** til å bruke denne funksjonen til å sette en annen farge på hodet til slangen. Fargekoden "#1db121" en mørkere grønnfarge enn den som brukes for resten av slangen.

Testing

Ved riktig implementasjon skal kjøring av appen vise brettet slik som etter forrige oppgave, men nå med en annen farge på hodet til slangen.

Oppgave 3

 7 min

Motivasjon

Vi har nå en grei grafisk representasjon av brettet og kan begynne å se på bevegelse.

Oppgavetekst

Implementer funksjonene **handleUp**, **handleDown**, **handleLeft**, **handleRight**. Disse skal flytte slangen et hakk i den retningen som metodenavnet tilsier, og oppdatere grafikken.

Testing

Prøv å kalle metodene i initialiseringsmetoden til kontrollerklassen og se at grafikken blir oppdatert riktig.



Oppgave 4

 5 min

Motivasjon

Det eneste som mangler for å flytte slangen er at trykking på knappene kaller riktig metode.

Oppgavetekst

Endre FXMLen slik at når de fire knappene trykkes på, så kjøres den respektive **handle<direction>** metoden.

Testing

Kjør spillet og se at trykking på knappene flytter slangen.

Oppgave 5



10 min

Motivasjon

Vi har nå et fungerende spill, men hvis vi vinner eller taper virker det kun som om spillet slutter å fungere.

Oppgavetekst

Endre **drawBoard** klassen slik at hvis spillet er vunnet kommer det opp en grønn tekst som sier "You Won!" og hvis spillet er tapt kommer det opp en rød tekst som sier "You Lost!".

Testing

Prøv å tape og å vinne og verifiser at meldingen dukker riktig opp.



Resten av tiden



Vi har nå fullført implementasjonen vår av SnakeBird. Resten av tiden er satt av til at dere kan jobbe med det dere vil. Hvis du ønsker å jobbe videre med Snakebird er det satt opp noen idéer for hvordan du kan utvide spillet.

Mulige utvidelser av SnakeBird implementasjonen

- Ta vare på historikk over bevegelser, og implementere **undo** funksjonalitet.
- Mulighet for å starte spillet på nytt.
- Bruke bilder og teksturer til å gjøre spillet penere.
- Støtte for blokker (tiles) som kan drepe deg.
- Muligheten for å ha flere slanger du kan kontrollere.