NTNU

Kunnskap for en bedre verden

Kollokvium 1 - Oppgaver

TDT4100 Objektorientert programmering

Mandag 10.02.2020

NTNU

Oppgaver

Vi skal fortsette på implementasjonen vår av en forenklet textbasert versjon av spillet Snakebird.

Koden som er utdelt består av en **Tile** klasse som representerer en rute på spillbrettet. I tillegg består den av en **Game** klasse skrevet i forrige øvingsforelesning. Denne representerer brettet i en 2D-array, og har metoder for utskrift av brettet og henting av ruter i brettet.

Spillregler

En oversikt over spillets regler finnes i oppgavene gitt i øvingsforelesning 5.

NTNU

Pakker

Utdelt kode og oppgavene finnes i mappen foreksempel/src/kollokvie1.

foreksempel/src/kollokvie1.kode

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

foreksempel/src/kollokvie1.underveis

Her blir kode lagt inn rett etter felles gjennomgang.

foreksempel/src/kollokvie1.tester

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

foreksempel/src/kollokvie1.lf

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

NTNU

Oppgave 1

7 min

Motivasjon

Det neste vi trenger er en måte å gjøre **Game** klassen klar over hvor slangen befinner seg.

Oppgavetekst

Lag en funksjon **createSnake** i **Game**, som tar inn en liste av **Tile**-objekter som representerer slangen (listen er satt opp slik: [hode, ..., hale]). Denne funksjonen skal lagre denne listen internt, og sette alle **Tile**-objektene i listen til å være av typen **Snake**. Du trenger ikke å sjekke at listen representerer en gyldig slange.

Testing

main-metoden i filen "oppgave1-main.txt" skal ved riktig implementasjon skrive ut brette som kan finnes i "oppgave1-result.txt".

Oppgave 2

5 min

Motivasjon

Vi har nå en indre representasjon av slangen, men siden hele slangen skrives ut med det samme tegnet er det vanskelig å vite hvor man vil bevege seg, da dette er avhengig hvilken del av slangen som er hodet.

Oppgavetekst

Endre **toString** funksjonen til **Game** til å skrive ut hodet til slangen med et annet symbol (f.eks. "8") enn resten av kroppen.

Testing

main-metoden fra forrige oppgave skal ved riktig implementasjon skrive ut samme brett som tidligere, men nå med den høyre delen av slangen representert med symbolet for hodet.

NTNU

Oppgave 3

7 min

Motivasjon

Vi kan ennå ikke flytte slangen, men for dette trenger vi en hjelpemetode for å sjekke om en bevegelse er gyldig.

Oppgavetekst

Lag en funksjon **canMove** i **Game** som tar inn to argumenter, dx og dy (endring i x- og y-koordinat). Metoden skal sjekke om det er en gyldig bevegelse for slangen. Du må sjekke at slangen ikke beveger seg for langt, ikke faller utenfor brettet, og at den nye posisjonen enten er luft, målet, frukt eller halen til slangen.

Testing

Med **main**-metoden fra forrige oppgave, burde **canMove(1, 0)** og **canMove(-1, 0)** være lov, mens **canMove(0, 1)** og **canMove(1, 1)** burde ikke være lov.

NTNU

Oppgave 4

7 min

Motivasjon

Vi har nå alt vi trenger for å kunne implementere bevegelse i slangen.

Oppgavetekst

Lag en funksjon **move** i **Game** som tar inn to argumenter, dx og dy (endring i x- og y- koordinat). Metoden skal flytte slangen i denne retningen hvis dette er et gyldig trekk, og ellers utløse et unntak. Du trenger ikke å ta hensyn til gravitasjon, frukt eller mål.

Testing

main-metoden i filen "oppgave4-main.txt" skal ved riktig implementasjon skrive ut bevegelse av slangen i en sirkel.

NTNU

Oppgave 5

7 min

Motivasjon

Det er tungvint å måtte gi inn endring i x og y koordinater alle plasser slangen skal flyttes.

Oppgavetekst

Lag fire funksjoner **moveLeft**, **moveRight**, **moveUp** og **moveDown**, som hver flytter slangen i retningen navnet tilsier, hvis dette er en lovlig bevegelse.

Testing

main-metoden i filen "oppgave5-main.txt" skal ved riktig implementasjon skrive ut bevegelse av slangen i en sirkel.

NTNU

Oppgave 6

7 min

Motivasjon

Koden i **move** funksjonen tar ikke hensyn til om slangen beveger seg over frukt, vi ønsker nå å implementere denne funksjonaliteten.

Oppgavetekst

Endre **move** funksjonen til å sjekke om slangen beveger seg oppå en bit med frukt. Hvis det er tilfelle, så skal slangen utvides med en blokk. Slangen utvides ved at halen ikke forsvinner i den bevegelsen hvor slangen beveger seg oppå en bit med frukt.

Testing

main-metoden i filen "oppgave6-main.txt" skal ved riktig implementasjon flytte slangen slik at den beveger seg over en frukt og blir 3 blokker lang.

Oppgave 7

7 min

Motivasjon

En viktig del av Snakebird er at slangen faller ned hvis den er i løse luften. For kunne implmentere gravitasjon trenger vi å kunne sjekke om slangen er i luften.

Oppgavetekst

Lag en funksjon **isInAir** som sjekker om slangen er i luften eller ikke.

Testing

main-metoden i filen "oppgave7-main.txt" sjekker om **isInAir** fungerer som den skal.

NTNU

Oppgave 8

7 min

Motivasjon

Vi har nå alt vi trenger for å implementere gravitasjon.

Oppgavetekst

Endre **move** funksjonen til å sjekke om slangen henger i løse luften etter en bevegelse og i så tilfelle flytte slangen helt til den ikke lengre er i luften, eller vil falle ut av brettet hvis den faller lengre.

Testing

main-metoden i filen "oppgave8-main.txt" skal ved riktig implementasjon flytte slangen utenfor "øyen" i brettet, og slangen skal falle helt ned til bunnen av brettet.

NTNU

Oppgave 9

5 min

Motivasjon

Hvis slangen egentlig hadde falt utenfor brettet grunnet gravitasjonen er spillet over, og det burde egentlig ikke være mulig å bevege seg videre.

Oppgavetekst

Endre funksjonene **move** og **canMove** sånn at hvis slangen kommer til å falle utenfor brettet er det ikke lengre mulig å bevege seg.

Testing

main-metoden i filen "oppgave9-main.txt" skal ved riktig implementasjon ende opp med slangen i samme posisjon som i forrige oppgave.

NTNU

Oppgave 10

5 min

Motivasjon

Hvis spillet er tapt, burde spilleren få vite om dette.

Oppgavetekst

Lag en hjelpefunksjon **isGameOver** i **Game** som sier om spillet er over eller ikke. Samtidig, endre **toString** metoden til **Game** slik at denne skriver ut at spillet er over, hvis spilleren har tapt.

Testing

main-metoden fra forrige oppgave skal ved riktig implementasjon sende slangen utenfor brettet og skrive ut at spillet er over.

NTNU

Oppgave 11

10 min

Motivasjon

Det eneste som mangler med spillet nå, er muligheten for å vinne.

Oppgavetekst

På samme måte som for at spillet er tapt, lag en hjelpemetode **isGameWon** og endre **move**, **canMove** og **toString** i **Game** slik at brukeren får en melding når spillet er vunnet, og ikke lengre kan bevege seg.

Testing

main-metoden i filen "oppgave11-main.txt" skal ved riktig implementasjon løse brettet og gi spilleren beskjed om at spillet er vunnet.