

Semesteroppgave: animasjon

DATS1600 oppgavesett 3.

I dette oppgavesettet skal dere muliggjøre animering av spillebrettet i GoL, slik at spillet kan endres over tid.

Animasjon i JavaFX

Animasjon, eller 'keyframe animation', på en dataskjerm muliggjøres ved å repetitivt tegne data (grafikk) til skjermen, der de underliggende dataene (modellene) endres mellom hvert bilde. Dette skaper en illusjon av jevn forflyttelse av visuelle objekter hvis antall bilder vist på skjermen per sekund er høyt nok.

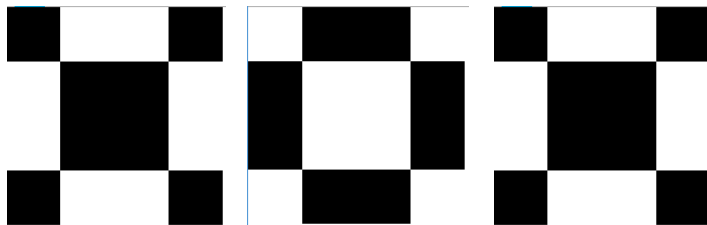
Hvis animasjon er et ukjent konsept, sett dere inn i dette konseptet. Animasjon uten digital teknologi kan for eksempel oppnås ved hjelp av såkalt '[flip book animation](#)'. Innen spilldesign, kalles muliggjøringen av animasjon og bevegelse for 'game loop'.

Denne oppgaven omhandler å sette seg inn i JavaFX sin støtte for animasjon og vurdere hvilken type animasjon som skal brukes i programmet.

Opprett følgende metode i en relevant klasse: `public void nextGeneration();`

Denne metoden skal oppdatere spillebrettet slik at vi 'beveger' oss til neste generasjon i spillet. For å teste animasjon, kan vi i denne øvelsen forenkle reglene til spillet. Implementer følgende regler for å bestemme om en celle er levende eller død: hvis en celle er i live, skal cellen dø og hvis en celle er død, skal cellen komme til live.

Resultat av tre repetitive kall til `nextGeneration()`:



Utfør `nextGeneration()` og `draw()` metodene for hvert 'key frame' (eller game loop) for å gjøre animasjon.

JavaFX støtter flere muligheter for animasjoner: mer avanserte animasjonssekvenser ved hjelp av såkalt 'timeline animation' eller mer primitiv animasjon ved bruk av timer. Les om disse alternativene og diskuter hvilken som skal brukes. Gjerne utforsk flere alternativer.

Etter at animasjonen fungerer, burde GUI funksjonalitet relatert til de frie variablene til animasjonen implementeres. Eksempler på frie variabler er antall bilder (key frames) per sekund (frames per second) og start/pause av animasjon.