

Programmering rapport

Autonomous agents

Eddie Arthur Knoll-Tech 2.T

Indhold

Introduktion og principper bag Game of Life	2
Krav til program	2
Design	2
Teori.....	2
Flowdiagram af programmet.....	4
Klassediagram.....	5
Implementering	5
Uddybende forklaring på udvalg af kode	5
Test	6
Opstillet test case	6
Video af test case	6
Konklusion	6
Hvad gik godt.....	6
Hvad gik skidt.....	6
Hvad kan man arbejde videre med	7
Bibliografi.....	7

Introduktion og principper bag Game of Life

Game of life er et spil skabt af en matematiker i 1970 og lavet til at simulere på simpel vis karakteristikkene fra rigtige biologiske miljøer og systemer. Game of life foregår ved nogle sætte principper der bestemmer hvornår en celle er "levende" eller "død" så tændt eller slukket.

Der er tre forskellige states. Den bliver "født" eller gjort levende når der 3 naboer. Dette er den regel der gør at det hele kan gro. Der er dødsreglen der gør at den dør hvis der er for mange eller for lidt naboer. Og til sidst er der stasis reglen der gør at hvis intet af de to andre reglers krav er opfyldt. Vi her i dette projekt arbejder med at lave vores eget game of life og så udviklet det til at ligne for eksempel et biome i dette tilfælde.

Krav til program

- Tegne celler med farver tilsvarende deres værdi
- Lave et program der simulerer game of life med specifikke regler
- Eksperimentere med regler for at ændre på spillet
- Lave regler der får spillet til at ligne mere en simulation af potentielt biome
- Skal gøres ved brug af classes

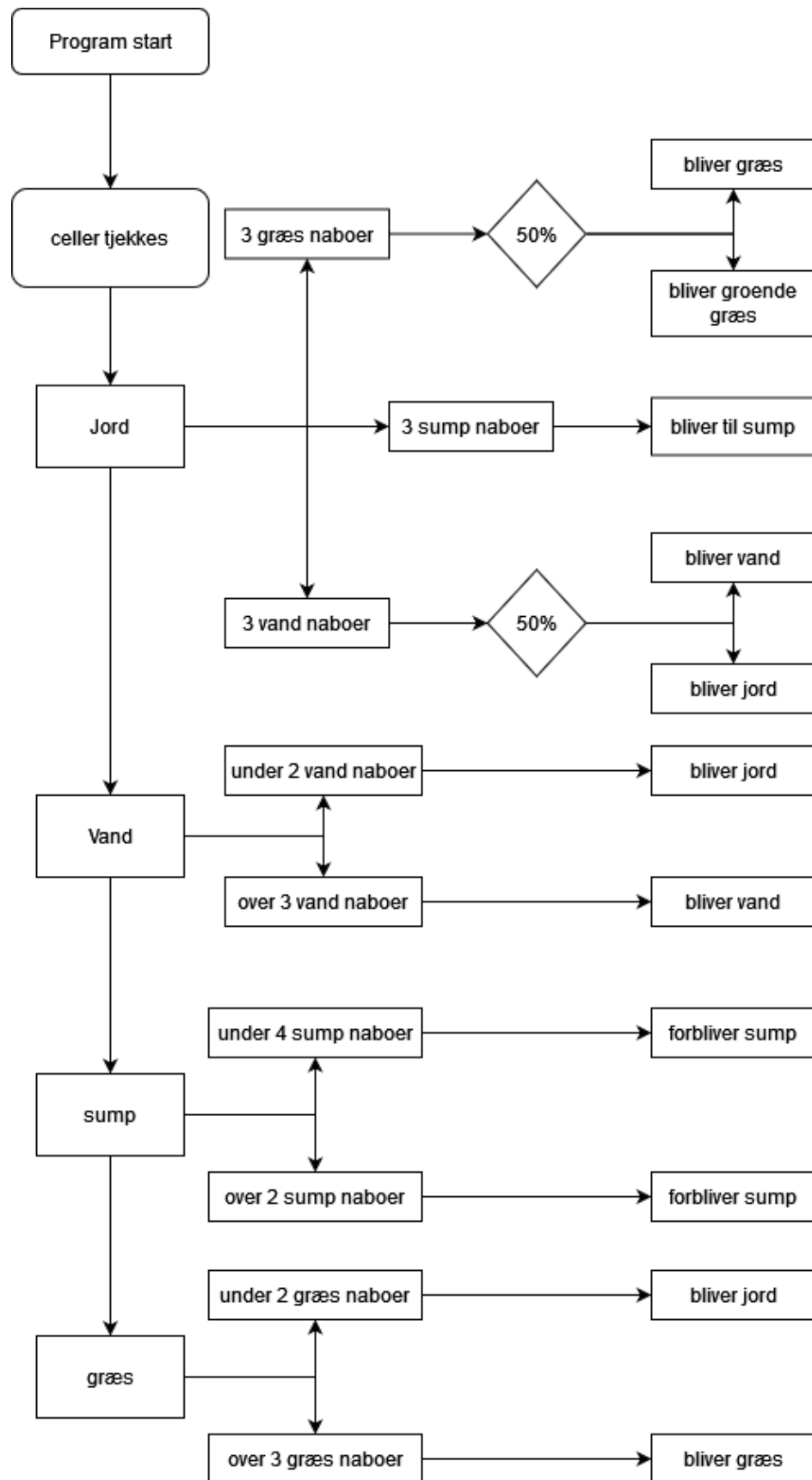
Design

Teori

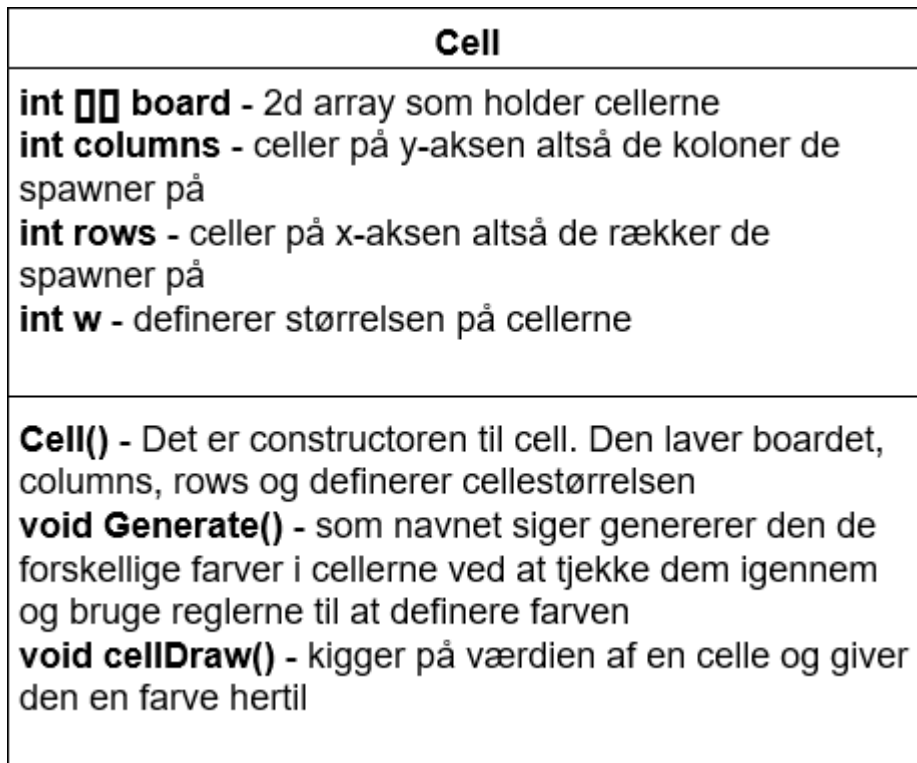
Cellular automata er et system med en celle objekt der følger nogle bestemte karakteristikkene. Den er på et grid der kan være i forskellige gyldige dimensioner. En celle har også et state. I simple eksempler har en celle to states 0 og 1 altså slukket og tændt. Dette kan også gøres større som brugt i mit program ved at lave biomes. De har også et nabolag som er alle celler rundt om den bestemte celle. Udviklingen er lavet af Stanisław Ulam og John von Neumann. Dette har givet ide til at kunne se ting i et grid af celler og lave regler til sådanne celler for at lave bestemte mønstre og ændre sig på et display. Dette har så senere ledt til iterationen som vi har arbejdet med nemlig game of life. Som også beskrevet i starten af rapporten så er Game of life er et spil skabt af en matematiker i 1970 og lavet til at simulere på simpel vis karakteristikkene fra rigtige biologiske miljøer og systemer. Det foregår helt konkret med bestemte parameter som er bygget videre fra i projektet. Der er Death stage når værdien er 0 og det kan ske på to måder. Overpopulation hvis den har 4 eller flere naboer og ensomhed hvis den har 1 eller mindre naboer. I mit videreudviklede

program er en celle jord når den har en værdi på 0 og er død og hvordan de forskellige celler dør på er forskellige men mange er enten overpopulation eller underpopulation. I de fleste tilfælde dør ting ikke men bliver til noget andet og oftest græs i stedet via over eller underpopulation. Der er også i game of life, birth, der bestemmer at hvis der er præcis 3 naboer bliver en celle levende dvs. får en værdi på 1. I mit program bruges der ikke rigtig samme funktion da der er mere end bare 2 værdier og det bevæger sig mere flydende imellem dem. Og sådan de opstår er forklaret senere men det er chance baseret og så udvikler alt sig derfra ved at gå til forskellige stadier. Der er til sidst stasis som foregår enten ved at forblive levende ved 2 eller 3 naboer og forblive død ved at have alt andet end 3 naboer. Her i mit tilfælde går alt andet end vand og jord hurtigt i en form for stasis da de har et større kriterie for at forblive i levende stasis og et meget lille kriterie for at blive jord så derfor udvikler landskabet sig som groende og sjældent aftagende.

Flowdiagram af programmet



Klassediagram



Implementering

Uddybende forklaring på udvalg af kode

```
board = new int[columns][rows];
for (int x = 0; x < columns; x++) {
    for (int y = 0; y < rows; y++) {
        int temp = int(random(100)); // gør :
        if (temp < 15) { //har 15% chance fo
            board[x][y] = 3;
        } else if (temp > 80) { // har 20% cl
            board[x][y] = 1;
        } else if (temp > 15 && temp < 25) {
            board[x][y] = 6;
        } else { // bliver til jord hvis den
            board[x][y] = 0;
        }
    }
}
```

Dette er board funktionen i celle constructoren der laver start boardet. Der er de to for funktioner der via x og y går igennem alle celler og giver dem en temp værdi. Den værdi bliver sat til et tilfældigt tal fra 1-100 og så bruger den if funktioner der hver definerer forskellige ting ud fra hvad temp værdien er i hver celle. Hvis det er mindre end 15 får den værdien 3 og bliver vand, hvis det

er over 80 bliver det græs med værdien 1 og hvis det er mellem 15 og 25 bliver det sump. Resten af værdierne bliver til jord hvis det ikke er noget andet med else funktionen til jord.

```
if ((board[x][y] > 0) && (neighbors < 2) && (board[x][y] < 3)) next[x][y] = 0;
else if ((board[x][y] > 0) && (neighbors > 3) && (board[x][y] < 3)) next[x][y] = 1;
else if ((board[x][y] == 0) && (neighbors == 3) && (board[x][y] < 3)) next[x][y] = int(random(1, 3));
```

Her er reglerne for hvordan cellerne udvikler sig og ændrer værdien til at blive andre værdier. De øverste er til græs der sørger for at hvis en celle er mere end 0 og mindre end 3 samt har mindre end 2 naboer dør cellen og bliver til jord. Hvis der er de samme regler som før men i stedet over 3 naboer bliver den til levende græs ligesom standard game of life regler. Hvis boardet er lig med 0 altså den er jord og den har 3 naboer bliver den et tilfældigt tal fra 1 til 3 så enten græs, groende græs eller vand

Test

Opstillet test case

Jeg laver en optagelse af mit game of life der bliver kørt to gange for at vise de forskellige situationer det kan ende i. Hver tur bliver kørt til at græsset har spredt sig.

Cellerne burde med forskellige chancer være forskellige ting fra vand og sump og resten burde være jord. Græs burde formes og udvikle sig fra starten af og sprede langsomt ud over hele skærmen imens sump og vand stabiliserer og bliver statiske. Derefter når græsset har fyldt det meste op er

Video af test case

<https://youtu.be/2c4GcAstem0>

Konklusion

Hvad gik godt

Jeg fik brugt classes meget fint og fået lært hvordan celler kan bruges til at tegne og automatiseres til at simulere bestemte ting via regler og states. Så fik generel ny viden om automatisering og simulering med kode via det simple koncept med celler på en ok let måde at forstå.

Hvad gik skidt

Det meste af læringen gik meget godt men dog var der gange hvor man bare fik bogen at støtte sig op af og skulle finde ud af det derfra. Det ville være dejligt med en slags fælles opsummering eller

lignende på de kapitler af bogen man har læst så det er sikkert at alle følger med og de spørgsmål der er kan blive besvaret i fællesskab så alle der måske har samme spørgsmål og undringer om ting i bogen kan finde ud af det uden at der skal nås ud til hver enkel elev.

Hvad kan man arbejde videre med

Jeg ville gerne i ferien eller når jeg har tid inden vi arbejder meget videre med også at blande det med autonomous agents lave noget skovareal der udvikler sig som græsset men i langsommere facon og der er også de jordpletter som kan fikses ved at sige at hvis der er 7-8 græs naboer bliver det også græs så der ikke kommer så mange små pletter.

Bibliografi

Shiffman, D. (2012). *The Nature of Code*.