

Game of Life

in Java

Auteur: drs. M.S.L.F. Manssen

<http://www.manssen.eu>

Game of Life is een spel van een rooster van cellen, waarbij telkens een ronde gebeurt waarbij de nieuwe waarden van cellen gebeurt. Deze regels worden bekend bij het verklaren van de klassen van het spel, zoals ik het heb geïmplementeerd.

Om te beginnen maken we een aantal bestanden aan.

Zet de volgende listing in een bestand `Life.java`

```
import java.io.*;
import java.lang.Number.*;
import java.util.Random;

/**
 *
 * @author matheas
 */

public class Life {

    private int field[][];
    private int w, h;

    public Life( int width, int height) {

        field = new int [width][height];
        w = width;
        h = height;

    }

    public void live() {

        overgang();

    }

    public int getPoint( int r, int k ) {
        return ( field[r][k] );
    }

    public void overgang() {
        int [][] tmp = new int [w][h];
        int rij, kolom;

        for ( rij = 1; rij < h-1; rij++ ) {
            for ( kolom = 1; kolom < w-1; kolom++ ) {
                int burens = aantalBuren( rij, kolom );

                if ( field[rij][kolom]==1 )
                    if (burens <2 || burens > 3)
                        tmp[rij][kolom] = 0;
                    else
                        tmp[rij][kolom] = 1;
                else
                    if ( burens == 3 )
                        tmp[rij][kolom] = 1;
            }
        }

        field = tmp;

    }
}
```

```

private int aantalBuren(int rij, int kolom) {
    int aantal = 0;

    aantal += field[rij-1][kolom-1];
    aantal += field[rij-1][kolom];
    aantal += field[rij-1][kolom+1];

    aantal += field[rij][kolom-1];
    aantal += field[rij][kolom+1];

    aantal += field[rij+1][kolom-1];
    aantal += field[rij+1][kolom];
    aantal += field[rij+1][kolom+1];

    return aantal;
}

public int [][] getLife() {
    return field;
}

public void vulRandom()
{
    Random random = new Random();

    for ( int r=0; r< 100; r++ ) {
        int n = random.nextInt(50*50-1);
        field[n/50][n%50]=1;
    }
}

public void leesVanInvoer() {
    String regel;
    int regels = 0;
    try {
        BufferedReader in = new BufferedReader(new FileReader("GOL.invoer"));
        while ((regel = in.readLine()) != null) {
            String[] tokens = regel.split(",");
            if (tokens.length != 50) {
                System.out.println("Fout aantal kolommen: ");System.out.println(tokens.length);
                System.exit(-1);
            }
            for (int i =0; i<50; i++)
                field[regels][i] = Integer.parseInt(tokens[i]);
            regels++;
        }
        if ( regels!= 50 ) {
            System.out.println("fout aantal rijen");
            System.exit(-1);
        }
    }
    catch( Exception e ){
    }
}
}

```

Zet nu de volgende listing in een bestand met de naam GameOfLife.java

```

import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.Timer;

public class GameOfLife extends JPanel implements ActionListener {
    int C=50;

    Timer timer;
    Life life;

    public GameOfLife() {
        life = new Life(50,50);
    }
}

```



```
public void vulRandom()
```

De laatste methode in deze class betreft het lezen van het initiële veld van cellen, door `vulRandom()`. Er wordt uitgegaan van 50 niet-lege regels, die elk 50 waarden bevatten, welke gescheiden zijn door komma's. Deze invoer moet zich bevinden in een bestand `GOL.invoer`.

```
public void leesVanInvoer()
```

Verklaring GameOfLife.java

```
public class GameOfLife extends JPanel implements ActionListener
```

Deze class erft over van `JPanel`. Dit verzorg samen met `JFrame` uit `Main` het grafische gedeelte.

Doordat de class `ActionListener` implementeert, kan hij dienen als `Timer`. De waarde 1000 die meegegeven wordt aan `setDelay()`, geeft aan dat elke 1000 milliseconden de `actionPerformed` methode wordt aangeroepen.

`paint` verzorgt de vertaling van het cellen array naar het grafische gedeelte.

```
public void paint(Graphics g)
```