Analyse «Game of Life»

Verteilte Systeme und Komponenten

Gruppe 5 (Bucher, Kiser, Meyer, Sägesser)

25.02.2018

Inhaltsverzeichnis

Allgemein

Bei der vorliegenden Applikation handelt es sich um eine Implementierung von John Horton Comways Simulation *Game of Life* als Java Applet.

Bei der Simulation gibt es ein Feld mit Zellen, die entweder den Satus aktiv oder inaktiv bzw. lebendig oder tot haben. Zu Beginn der Simulation werden einige lebendige Zellen vordefiniert. Mit jedem Zyklus entstehen, bleiben oder verschwinden die Zellen gemäss folgenden Regeln:

1. Eine lebendige Zelle mit weniger als zwei lebendigen Nachbarzellen stirbt (Unterbevölkerung).
2. Eine lebendige Zelle mit zwei oder drei lebendigen Nachbarzellen überlebt.
3. Eine lebendige Zelle mit mehr als drei lebendigen Nachbarzellen stirbt (Überbevölkerung).
4. Eine tote Zelle mit genau drei lebendigen Nachbarzellen wird lebendig (Fortpflanzung).

Dieses Regelwerk ist in der Methode `org.bitstorm.gameoflife.GameOfLifeGrid.next()` implementiert.

Als Nachbarn gelten nicht nur die Zellen oberhalb, unterhalb, rechts, links von einer Zelle, sondern auch die diagonal angrenzenden (8er-Nachbarschaft).

Es gibt auch abweichende Regelwerke (sogenannte *Welten*), die hier jedoch nicht von Interesse sind.

Packages

- `org.bitstorm`: Dies ist das Oberpackage, das die ganze Applikation beinhaltet.
- `org.bitstorm.gameoflife`: In diesem Package liegt der Cod zur Erzeugung der grafischen Benutzeroberfläche und die eigentliche Spiellogik.
- `org.bitstorm.util`: Dieses Package beinhaltet Hilfsklassen wie Dialoge und Utility-Klassen für allgemeine Aufgaben.

Klassen

- `Cell`
- ...

Aufbau und Zusammenspiel

TODO

Spezielles

Das Projekt enthält beispielsweise solchen Code:

```
System.out.println("Hello, world!\n");
```