

Game of Life

Pflichtenheft

SEW-Projekt

Name	Richard KRIKLER
Klasse	3BI
Projektbeginn	23.11.2020
Projektabschluss	11.01.2020
Auftraggeber	HOL / BRE
Thema der Übung	Game of Life

Inhaltsverzeichnis

Kurzbeschreibung	2
Funktionsumfang	2
Optional	2
Screenshots	3
Benötigte Ressourcen	4
Testfiles.....	4
Know-How	5
Grafische Oberfläche.....	5
Zeitplan / Meilensteine	5
27.11. Repo mit Pflichtenheft PDF.....	5
21.12. Abgabe lauffähiger Prototyp.....	5
11.01. Abnahme	5

Kurzbeschreibung

Das Spiel „[Conway's Game of Life](#)“, wurde von John Horton Conway 1970 entworfen.

Es gibt ein Spielfeld, welches in Zeilen und Spalten unterteilt wird. Somit ergeben sich Zellen, welche entweder tot oder lebendig sind. Vor dem Start des Spiels müssen einige Zellen vom Anwender lebendig gemacht werden.

Nach dem Start des Spiels gibt es in periodischen Zeitabständen, immer eine neue Generation, welche auf die Vorherige basiert. Um die neue Generation zu erhalten, wird jede Zelle des Spielfeldes mit den Spielregeln abgeglichen und entweder getötet, lebendig gemacht oder wieder zum Leben erweckt.

Bei den Spielregeln werden immer die 8 umliegenden Felder, des gerade betrachteten Feldes beachtet. Wenn diese Felder (= Nachbarfelder) außerhalb des Spielfeldes liegen, dann werden diese als tot angesehen.

Betrachtete Feld (C)	Nachbarfelder	Ergebnis für C
tot	3 lebende	lebend
lebend	< 2 lebende	tot
lebend	2 oder 3 lebende	lebend
lebend	> 3 lebende	Tot

	NW	N	NE	
	W	C	E	
	SW	S	SE	

Funktionsumfang

- ✓ Eingabe der Dimensionen vor dem Starten des Spiels (Anzahl der Felder in x- und y-Richtung).
- ✓ Zoom: In das Spielfeld soll hineingezoomt werden können
- ✓ Eingabe von Figuren mit der Maus (auch während des Spieles). Mausklick auf das gewünschte Feld.
- ✓ Reset: Spiel stoppen und das Spielfeld leeren
- ✓ Reset to Start: Spiel stoppen und das Spielfeld in den ursprünglichen Zustand zurückversetzen
- ✓ Geschwindigkeit: Einstellen der Geschwindigkeit (in Sekunden), in der von einer Generation zur nächsten gegangen wird.
- ✓ 1-Step-Forward: Das derzeitige Spielfeld in die nächste Generation bringen.
- ✓ 1-Step-Backward: Das derzeitige Spielfeld in die vorherige Generation bringen.
- ✓ Anpassbare Spielregeln: Ab wann wird das Feld getötet, wiederbelebt oder am Leben gelassen.
- ✓ Laden / Speichern:
 - ✓ Laden / Speichern des derzeitigen Spielfeldes von / in einer CSV-Datei (Datei-Auswahl Dialog; Standardmäßiger Ordner: „resources\PlayFieldPresets“)
 - ✓ Dropdown-Menü im GUI mit den Presets
- ✓ Derzeitig lebende Felder auflisten
- ✓ Go to: Zu einer bestimmten Generation wechseln

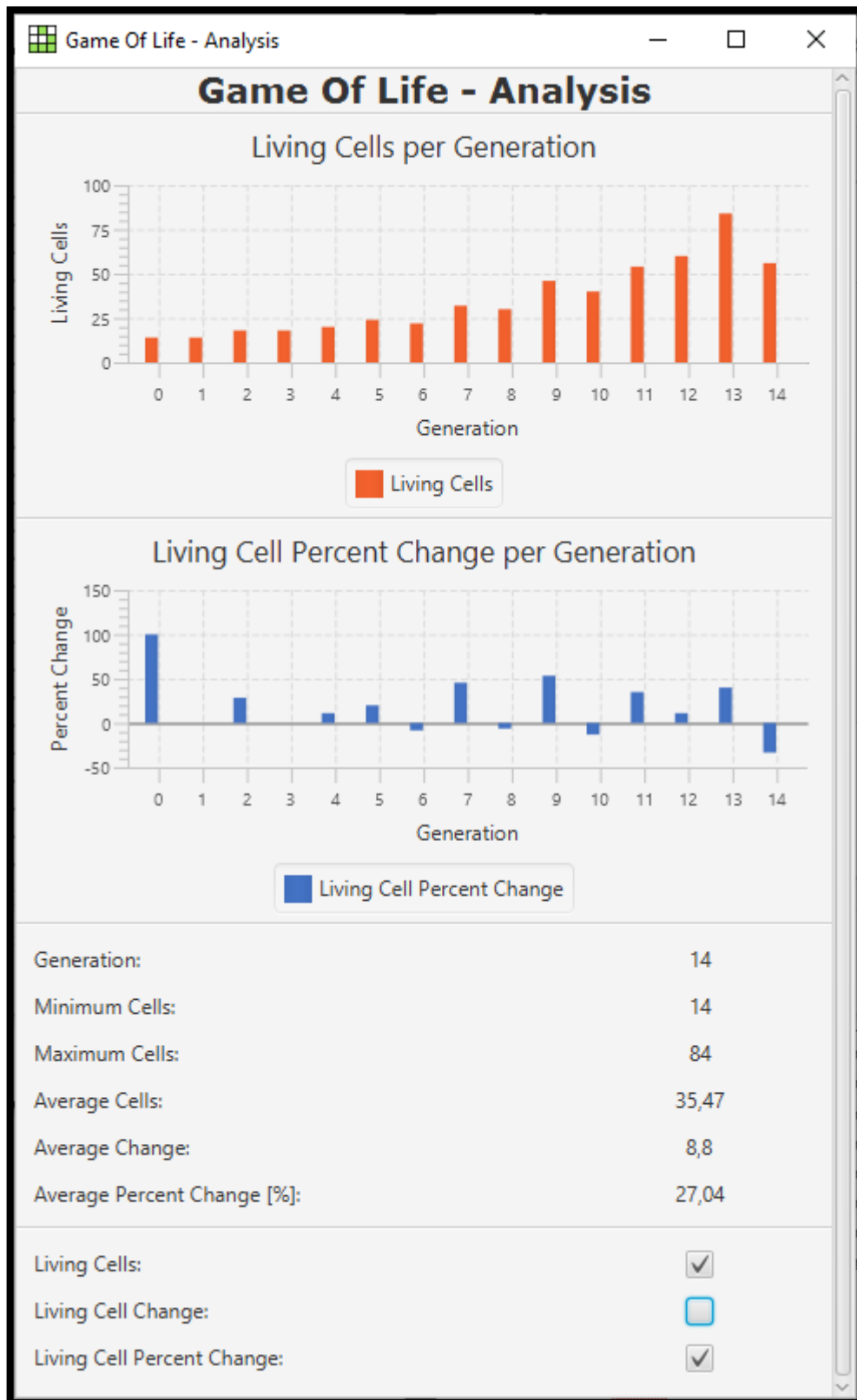
Optional

- ✓ Analysis: Bei jeder Generation speichern wie viele lebende Zellen es gab. Während dem Spiel auswerten → Graph
- ~~Farblich die lebenden Zellen markieren (Grün) und die Zellen markieren, die im nächsten Schritt sterben werden (Orange).~~
- ✓ Zufällige Platzierung der lebenden Zellen (Button)

Screenshots

The screenshot shows the 'Game Of Life' application window. The main area displays a 20x15 grid with a pattern of green cells. The right sidebar contains various settings and controls. Handwritten annotations in German provide explanations for the interface elements:

- Current Generation:** 10 (Annotation: *Current generation*)
- Living Cells:** 18 (Annotation: *Current amount of living cells*)
- Set Play Field Dimensions:** 20 x 15 (Annotation: *Dimensions*)
- Game Controls:** Play, Pause, 1-Step Backward, 1-Step Forward, Reset, Reset to Start (Annotations: *Start, Pause*, *1-Step Backward*, *Reset, Reset to Start, 1-Step Forward*)
- Go to:** 5 (Annotation: *Go to generation*)
- Game Speed (in s):** 1.0 (Annotation: *Geschwindigkeit*)
- Game Rules:** (Annotation: *Regeln einstellen*)
- Reanimate Rule:** 3 (Annotation: *Zahl zw. 0-8*)
- Keep Alive Rule:** 2,3 (Annotation: *(mehrere Zahlen mit „“)*)
- Presets:** (Annotation: *Dropdown-Liste der Presets*)
- Load / Save:** (Annotation: *Speichern und laden eines Spielfeldes*)
- Random Placement:** (Annotation: *Zufällige Platzierung*)
- Analysis:** Show Analysis (Annotation: *Auswertung der lebenden Zellen im Verlauf des Generations*)
- Stop game if minimized:** ☒ (Annotation: *Stop game if minimized*)
- Zoom:** (Annotation: *Zoom*)
- Scrollbar:** (Annotation: *Scrollbar*)



Benötigte Ressourcen

Testfiles

Die Testfiles werden in den JUnit-Tests verwendet.

Mit 1 und 0 wird der Status einer Zelle angegeben, 1 für lebend und 0 für tot.

In dem Testfile wird zuerst ein ganzes Spielfeld im CSV-Format, also mit Beistrichen getrennt, gespeichert.

In einem zweiten File wird dann das Ergebnis des Spielfeldes nach x Generationen gespeichert.

Für welche Generation das Ergebnis-File steht, wird im JUnit-Test angegeben.

Hier ein Beispiel für ein 8x8 Spielfeld.

Input-File	Output-File (nach der 2. Generation)
0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0	0,0,0,0,1,0,0,0
0,0,0,0,1,1,0,0	0,0,1,0,1,0,0,0
0,0,1,1,0,0,0,0	0,1,0,0,1,0,0,0
0,0,1,0,0,0,0,0	0,0,1,1,0,0,0,0
0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0

Know-How

- JUnit
- Objektorientiertes Programmieren
- JavaFX:
 - GUI-Positionierung
 - GUI-Elemente
 - Scrolling
 - Datei Auswahl
 - Graphen
 - Event Handling
- RegExp: User-Input
- Arrays: Speicherung des Spielfeldes
- Collections: Speicherung / Laden des Spielfeldes einer Datei
- Exceptions (bzw. Error Dialog): Files, Zellen außerhalb des Spielfeldes, falsche Input-Werte
- Datei Lesen / Schreiben, Dateien Auflisten

Grafische Oberfläche

- ✓ Alle Einstellungen sind auf der rechten Seite des Fensters
- ✓ Anpassbare Breite des Settings Bereiches
- ✓ Bei Minimieren (in der Taskleiste) des Java-FX Fensters wird die Simulation gestoppt
- ✓ Wenn das Spielfeld nicht auf das Fenster passt, soll es Bildlaufleisten (Scrollbars) geben.
- ✓ Wenn die Einstellungen nicht mehr auf das Fenster passen, soll es Bildlaufleisten (Scrollbars) geben.

Zeitplan / Meilensteine

27.11. Repo mit Pflichtenheft PDF

21.12. Abgabe lauffähiger Prototyp

11.01. Abnahme