

Pflichtenheft

Conways's Game of Life

„Eine universelle Software zur Simulation zellulärer Automaten“

Auftraggeber:

- Hochschule Bochum
- Ansprechpartner: Dipl.-Inform. Christian Düntgen
- Raum: D 3-30

Auftragnehmer:

- Die 5 Kranken Schwestern
- Weder krank noch Frauen
- Definitionsphasenmanager: Jörg Galilee Uwimana
- Architekt (Entwurfsbeauftragter): Felix Reinhardt
- Gruppenschuldiger, Spezifikationsbeauftragter: Alex Chojnatzki
 - Implementierungs-Beauftragter: Nicholas Schuran
- Kundenbetreuer, Außenminister, Abnahmebeauftragter: Diaa El Bathich

Stand: 28.10.2021

Contents

| | |
|---|-----------|
| 1 Zielbestimmung | 3 |
| 1.1 Musskriterien | 3 |
| 1.2 Wunschkriterien | 6 |
| 2 Produkt-Einsatz | 7 |
| 2.1 Anwendungsbereich | 7 |
| 2.2 Zielgruppen | 7 |
| 2.3 Produktumgebung | 7 |
| 2.3.1 Softwareanforderungen | 7 |
| 2.3.2 Hardwareanforderungen | 7 |
| 2.4 Betriebsbedingungen | 7 |
| 3 Produktfunktionen | 8 |
| 3.0.1 Benutzeroberfläche | 8 |
| 3.0.2 Datenverarbeitung | 8 |
| 3.0.3 Datenspeicherung | 8 |
| 3.1 Nichtfunktionale Anforderungen | 8 |
| 3.1.1 Performance | 8 |
| 3.1.2 Zuverlässigkeit | 8 |
| 4 Testszenarien | 9 |
| 4.1 UI | 9 |
| 4.2 Verarbeitung | 9 |
| 4.3 Speichern | 9 |
| 4.4 Performance | 9 |
| 4.5 Benutzbarkeit (Schimpanse benötigt) | 9 |
| 5 Entwicklungsumgebung | 10 |
| 5.1 Verwendete Software | 10 |
| 5.2 Verwendete Hardware | 10 |
| 5.3 verwendete Organisation | 10 |

1 Zielbestimmung

1.1 Musskriterien

Das Programm soll dazu dienen, Zelluläre Automaten auf einem 2-D orthogonalen Spielfeld darstellen zu können. Dazu werden als Beispiel die Regeln für Conway's Game of Life verwendet. Hierzu sind unbedingt die folgenden Features erforderlich:

| | | |
|-------|------------------------|--|
| M0001 | UI | Das Programm muss eine graphische Oberfläche haben. |
| M0002 | Scope | <ul style="list-style-type: none">• Es soll ein zellulärer Automat mit möglichst großer Freiheit definiert und simuliert werden können. |
| M0003 | Darstellung Spielfeld | <ul style="list-style-type: none">• Die Darstellung des Zellulären Automaten erfolgt über eine 2 Dimensionale Matrix aus Quadraten deren Farbe und Helligkeit den Zustand eines Feldes wiedergeben. |
| M0004 | Transitionsregeleditor | Die Transitionsregeln sollen über eine definierte und im Handbuch dokumentierte Syntax (invers Polnische Notation, ggf. auch mathematische Schreibweise) formuliert werden können. Der neue Zustand einer Zelle darf dabei von der Zelle selbst, sowie von den umliegenden acht benachbarten Zellen abhängen. Ihr Status wird in Variablen bereitgestellt. |
| M0005 | Spielfeldaufbau | Das Spielfeld soll als 2-D Array von Integerwerten ausgeführt sein, welche den Zellzustand repräsentieren. |
| 0006 | Spielfeldgröße | Die Spielfeldgröße soll vor Simulationsstart vom Benutzer über (Text-)Eingabefelder festgelegt werden können. |

| | | |
|-------|-------------------------|---|
| M0007 | Speichern | Spielfeldzustand und Transitionsregeln sollen separat gespeichert und geladen werden können. |
| M0008 | Einfügen | Es sollen Figuren in das Spielfeld eingefügt werden können. Dies soll so geschehen, dass Figuren als Spielstände mit kleinerer Feldgröße als ganzes geladen und eingefügt werden können. |
| M0009 | Navigation | es soll möglich sein, das Spielfeld mit Zoom und Pan verschieden zu betrachten. |
| M0010 | Spielfeldmanipulation | Der Zustand einer Zelle soll durch Mausklick darauf auf einen wählbaren Wert einstellbar sein. Das Wählen des Werts soll durch ein Texteingabefeld auf der Benutzeroberfläche erfolgen. Details in der Beschreibung der Benutzeroberfläche. |
| M0011 | Topologie | Das Randverhalten des Spielfelds soll zwischen begrenztem Rechteck und Torus (Zellen an den Kanten sind mit den ihnen gegenüberliegenden Zellen benachbart) wählbar sein. |
| M0012 | Automatische Simulation | Die Simulationsgeschwindigkeit soll über einen Slider einstellbar sein. Die Simulation soll über einen Button gestartet und unterbrochen werden können. |
| M0013 | Manuelle Simulation | Über einen Button soll die nächste Generation berechnet und angezeigt werden können. |

| | | |
|-------|---------------------------|---|
| M0013 | Zufälliger Anfangszustand | Der Spielfeldzustand soll zufällig generierbar sein. Dazu soll einem Zellzustand eine Wahrscheinlichkeit zugewiesen werden können, mit dem Default-Zustand 0, sodass jede Zelle genau einen Zustand erhält. |
| M0014 | Anzeige | Die Anzeige des Spielfeldzustands soll durch Farben erfolgen, wobei einem Zustand eine Farbe zugeordnet wird. |

1.2 Wunschkriterien

| | | |
|-------|-------------|--|
| W0001 | Undo | Es sollen Eingaben rückgängig gemacht werden können. |
| W0002 | Regeleditor | Eingabe der Regeln in für Menschen gut lesbare Mathematischer Schreibweise, mit Grundrechenarten und logischen Operationen |
| W0003 | Performance | Multithreading parallelisierbarer Prozesse |

- Es ist wünschenswert, einer Farbe einen Zustand zuzuordnen zu können. Ein einfaches Color ramp kann ggf. verwendet werden, alternativ ist es

2 Produkt-Einsatz

2.1 Anwendungsbereich

Das Programm soll dazu dienen, Zelluläre Automaten mit recht großer Freiheit bauen zu können.

2.2 Zielgruppen

Die Verwendung dieses Programms für Conway's Game of life ist einfach, da die Spielregeln mitgeliefert werden. Dies kann von allen interessierten ausprobiert werden, da die Manipulation des Spielfelds zum ausprobieren einlädt.

Leider ist es nicht möglich, den Regeleditor intuitiv bedienbar zu gestalten, da es für eine effiziente Verarbeitung notwendig ist, den Zustand einer Zelle in der nächsten Generation als Mathematische Funktion der Zustände der Nachbarzellen darzustellen. Aus diesem Grund gibt es zwar einen Leitfaden, um Mathematische Funktionen mit den Umliegenden Zellen als Ausgangsdaten zu erstellen, es ist jedoch nicht einfach, dies zu tun. Deal with it.

2.3 Produktumgebung

2.3.1 Softwareanforderungen

2.3.2 Hardwareanforderungen

2.4 Betriebsbedingungen

3 Produktfunktionen

3.0.1 Benutzeroberfläche

3.0.2 Datenverarbeitung

3.0.3 Datenspeicherung

3.1 Nichtfunktionale Anforderungen

3.1.1 Performance

3.1.2 Zuverlässigkeit

4 Testszenarien

4.1 UI

4.2 Verarbeitung

4.3 Speichern

4.4 Performance

4.5 Benutzbarkeit (Schimpanse benötigt)

5 Entwicklungsumgebung

5.1 Verwendete Software

5.2 Verwendete Hardware

5.3 verwendete Organisation