

Christian Maurer

Das Spiel des Lebens

v. 17. September 2010

Dr. Christian Maurer
Keithstr. 16
10787 Berlin

<http://murus.org/>

Die Quelltexte des ^{Ula}Mod_{den}U^{Sums} sind mit größter Sorgfalt entwickelt und werden laufend gepflegt. Kein Programmsystem dürfte jedoch jemals frei von Fehlern sein; deshalb ist auch nicht damit zu rechnen, daß *dieses* System fehlerfrei ist: Es darf nur benutzt werden „wie es ist“.

Die Modula-2- und die Java-Quelltexte von Murus sind freie Software. Sie können sie unter den Bedingungen der GNU General Public License, wie von der Free Software Foundation veröffentlicht, weitergeben und/oder modifizieren, entweder gemäß Version 3 der Lizenz oder (nach Ihrer Wahl) jeder späteren Version. Ihre Veröffentlichung erfolgt in der Hoffnung, daß sie Ihnen von Nutzen sein könnten – aber *ohne irgendeine Garantie*, auch ohne die implizite Garantie der *Marktreife* oder der *Verwendbarkeit für einen bestimmten Zweck*.

Der Originaltext der GPL ist im weltweiten Netz unter der Adresse www.gnu.org/licenses/gpl.html zu finden; eine deutsche Übersetzung unter www.gnu.de/documents/gpl-3.0-de.html.

Die Quelltexte von Murus sind nur zu Lehrzwecken konstruiert und haben rein akademischen Wert. Ihre Verwendung in Programmen könnte zu *Schäden* führen, z. B. zur Inbrandsetzung von Rechnern, zur Entgleisung von Eisenbahnzügen, zum GAU in Atomkraftwerken oder zum Absturz des Mondes ...

Satz: Autor mit $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\mathcal{E}\mathcal{X}$

Leben ist lebensgefährlich.

Ulrich Scholtze

England, im Sommer 1967

Vorwort

Die vorliegende Ausarbeitung auf der Basis der Ergebnisse eines Lehrerweiterbildungskurses Informatik an der Freien Universität Berlin dokumentiert das Softwareprojekt **Spiel des Lebens**.

Es handelt sich um die Simulation eines einfachen *Räuber-Beute-Systems* und des *Game of Life* von JOHN CONWAY.

Dank gebührt den Kursteilnehmerinnen und -teilnehmern, die die Anforderungsdefinition entwickelten und an der Implementierung der Komponenten maßgeblich beteiligt waren.

Christian Maurer

Inhaltsverzeichnis

ALLGEMEINES

Arbeitsverzeichnis	1
--------------------------	---

SYSTEMANALYSE

Ökosystem	2
-----------------	---

Spiel des Lebens	3
------------------------	---

Optionen für Erweiterungen	3
----------------------------------	---

ANFORDERUNGSDEFINITION

Ein-/Ausgabeformate	4
---------------------------	---

Die Bedienung des Programms	5
-----------------------------------	---

SPEZIFIKATION UND IMPLEMENTIERUNG

Systemarchitektur	7
-------------------------	---

ALLGEMEINES

Spiel des Lebens ist als Bestandteil des $\text{ModulenUivers}^{\text{Ula}}^{\text{Sums}}$ nach dessen Installation oder Aktualisierung ebenfalls installiert.

Voraussetzung für die Installation des $\text{ModulenUivers}^{\text{Ula}}^{\text{Sums}}$ ist die Installation von **Mocka**, dem Modula-2-Compiler der GMD (Hinweise dazu finden sich unter lwb.mi.fu-berlin.de/inf/mocka).

Arbeitsverzeichnis

Das Arbeitsverzeichnis, aus dem heraus **Leben** aufgerufen wird und in dem ggf. Daten abgelegt werden, ist $\$HOME/.Leben$ (es sei denn, daß durch die Umgebungsvariable **Leben** ein anderes Arbeitsverzeichnis definiert ist). Damit ist sichergestellt, daß Benutzer/innen mit dem Aufruf von **Leben** ihre eigenen Daten verwalten.

Wenn dieses Verzeichnis noch nicht existiert, wird es beim ersten Aufruf von **Leben** selbsttätig angelegt und es werden Beispieldaten aus dem $\text{ModulenUivers}^{\text{Ula}}^{\text{Sum}}$ dorthinein kopiert, falls es welche gibt.

SYSTEMANALYSE

Das Öko-System aus Füchsen, Hasen und Pflanzen

Füchse fressen *Hasen*, *Hasen* fressen *Pflanzen*. Beide Gruppen von Lebewesen können nur überleben, wenn ihre Umgebung nicht mit der eigenen Art überbevölkert ist und sie deshalb nichts mehr zu fressen finden können.

Das *Ökosystem* wird als rechteckige Welt aus schachbrettförmig angeordneten Plätzen modelliert. Jeder Platz ist entweder von einer Pflanze, einem Hasen oder einem Fuchs besetzt; jeder Platz hat damit – sofern er nicht am Rand der Welt liegt – links, rechts, oben und unten einen Nachbarplatz.

Die *Überlebensregeln* sind sehr einfach:

- Der Platz einer Pflanze wird von einem Hasen eingenommen, wenn auf einem, zwei oder drei Nachbarplätzen ein Hase sitzt („Hasen fressen Pflanzen“).
- Ein Hase verliert seinen Platz an eine Pflanze, wenn es in der Nachbarschaft bereits vier Hasen gibt („Hase findet nichts zu fressen“).
- Der Platz eines Hasen fällt an einen Fuchs, wenn es auf mindestens einem Nachbarplatz einen Fuchs gibt („Füchse fressen Hasen“).
- Ein Fuchs muß seinen Platz an eine Pflanze abgeben, wenn es auf keinem Nachbarplatz einen Hasen gibt („Fuchs findet nichts zu fressen“).

Im Laufe der Simulation der „generationsweisen“ Entwicklung des Ökosystems ist folgendes möglich:

- Eine anfangs nur mit Pflanzen besetzte Welt wird vom Benutzer „geschaffen“ (d. h. ihre Plätze gezielt mit Hasen und Füchsen besetzt),
- die Entwicklung einer Welt nach den obigen Regeln wird schrittweise verfolgt (wobei in jedem Schritt einmal die Regeln angewandt werden),
- die Simulation kann jederzeit abgebrochen, der Zwischenstand archiviert und jederzeit wieder restauriert und weiterverfolgt und dazu
- jede Welt mit einem Namen versehen werden.

Der aktuelle Stand der Welt ist übersichtlich auf dem Bildschirm dargestellt.

Game of Life

Im *Game of Life* gibt es nur eine Art von Lebewesen: *Zellen*. Sie überleben nach den folgenden Regeln:

- Wenn sie mehr als drei Zellen in der Nachbarschaft haben, sterben sie an Streß.
- Wenn sie nicht wenigstens zwei Zellen in der Nachbarschaft haben, sterben sie an Vereinsamung.
- Auf einem freien Platz, in dessen Nachbarschaft sich drei Zellen befinden, entsteht eine neue Zelle.

Als *Nachbarschaft* gelten dabei die bis zu acht benachbarten Plätze, nicht nur horizontal und vertikal, sondern auch diagonal.

Optionen für Erweiterungen des Systems

Das Programm, das ja nur ein sehr einfaches (um nicht zu sagen: primitives) Ökosystem modelliert, könnte zu komplexeren Systemen mit

- mehreren Arten von Lebewesen,
 - Berücksichtigung verschiedener Generationen der Lebewesen,
 - komplizierteren Regeln,
 - Einfügung stochastischer Aspekte bei den Regeln zum Entstehen neuer Generationen,
 - mehr Komfort bei der Bedienung (z.B. Maussteuerung für den Welteditor)
- erweitert werden.

ANFORDERUNGSDEFINITION

Ein- und Ausgabeformate

Lebensarten

Lebensarten sind im Textmodus durch ihre großen Anfangsbuchstaben dargestellt

- **P**flanzen, **H**asen und **F**üchse im Ökosystem,
- **Z**elle im Game of Life, ein freier Platz durch ein Minuszeichen.

Zur besseren Übersichtlichkeit werden die Zeichen für verschiedene Lebensarten in unterschiedlichen Farben dargestellt: Füchse in *braun*, Hasen in *gelb* und Pflanzen in *grün*; auf dem aktuellen Platz in der jeweiligen Komplementärfarbe.

Bei der Eingabe werden auch die entsprechenden Kleinbuchstaben akzeptiert.

Im Graphikmodus werden die Lebensarten pixelweise durch kleine „Ikonen“ im 16x16-Raster dargestellt; Eingaben erfolgen jedoch auch mit den entsprechenden Anfangsbuchstaben.

Lebenswelten

Rechtecke aus Plätzen in 36 Zeilen und 64 Spalten, in dem jeder Platz ein Lebewesen einer Art aufnehmen kann.

Die erste Zeile bleibt dem Weltnamen und den Zahlenangaben der Lebensarten und die letzte Zeile Bedienhinweisen und Fehlermeldungen des Systems vorbehalten.

Bedienung des Programms

Nach dem Aufruf des Programms erscheint ein Menü, in dem zwischen

- dem *Ökosystem aus Füchsen, Hasen und Pflanzen* oder
- dem *Spiel des Lebens* ausgewählt werden kann.

Abschluß der Eingabe durch

- $\leftarrow \downarrow$: Das entsprechende System ist ausgewählt; weiter bei Festlegung der Welt.
- Esc: Das Programm ist beendet.

Festlegung der Welt

Der Cursor steht im Feld für den Namen der Welt; das Feld ist leer, der Name kann editiert werden. Abschluß der Eingabe durch

- $\leftarrow \downarrow$: Das Hinweisfeld ist gelöscht. Wenn der Name leer ist, ist das Programm beendet, sonst weiter beim Welteditor.
- Esc: Das Programm ist beendet.

Welteditor

Falls es eine Welt mit dem eingegebenen Namen schon gibt, ist sie geladen; eine neue Welt ist beim Ökosystem nur voller Pflanzen, beim Spiel des Lebens leer.

Der aktuelle Platz ist der beim Verlassen der Welt im letzten Programmlauf (bei einer neuen Welt in der Mitte); der Cursor steht auf diesem Platz. Es kann eins der Symbole für die Art eines Lebewesens oder ein Kommando eingegeben werden; im Hinweisfeld steht ein entsprechender Hinweis. Eingabe von

- einem Anfangsbuchstaben der Lebensarten (P, H oder F): Auf dem aktuellen Platz sitzt eine Pflanze, ein Hase bzw. ein Fuchs.
- Pfeil- und Positionstasten (auch in Kombination mit \uparrow oder Strg): Der aktuelle Platz ist entsprechend verschoben.
- linker Mausknopf: Im *Ökosystem* ist ein Hase, im *Game of Life* eine Zelle gesetzt.
- rechter Mausknopf: Im *Ökosystem* ist ein Fuchs gesetzt, im *Game of Life* ist nichts verändert.

- mittlerer Mausknopf: Im *Ökosystem* ist eine Pflanze gesetzt, im *Game of Life* ist, falls vorhanden, eine Zelle gelöscht.
- $\leftarrow \downarrow$: Ein Schritt der Simulation ist nach den Regeln durchgeführt; der Cursor steht auf dem gleichen Platz.
- Esc: Zurück zur Festlegung der Welt.
- Strg C: Das Programm ist abgebrochen.

In jedem Fall sind die Gesamtzahlen der Lebensarten in der Welt aktualisiert und ausgegeben.

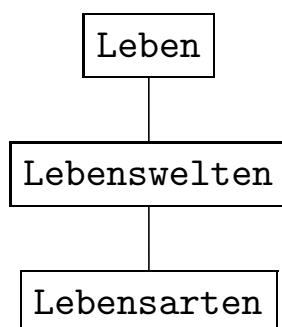
SPEZIFIKATION UND IMPLEMENTIERUNG

Systemarchitektur

Aus der Anforderungsdefinition lassen sich folgende Komponenten des Systems ableiten:

- die unterschiedlichen *Lebensarten* auf ihren Plätzen im betrachteten System und
- die *Welten*, in denen sie leben.

Die entsprechenden Moduln hängen untereinander wie folgt ab, wobei der jeweils tiefer liegende Modul vom darüber liegenden benutzt (importiert) wird:



An tieferen Stellen werden weitere projektunabhängige Moduln von $\text{dasModell}^{\text{Ua}}_{\text{Modell}}^{\text{Sum}}_{\text{Univer}}$ gebraucht, wie z. B. **Felder** zur Ein-/Ausgabe auf dem Bildschirm oder **PFolgen** (persistente Folgen = sequentielle Dateien).

Bemerkung zur Spezifikation und Implementierung

Ein interessanter Aspekt der frühen Phasen ist die Feststellung, daß beide Simulationen dem Wesen nach hochgradig verwandt sind (im Grunde handelt es sich um einen didaktisch reduzierten *zellulären Automaten*).

Konsequenz daraus ist die weitestgehende Bündelung der unterschiedlichen Ausprägungen bei der Repräsentation der Daten und der Konstruktion der Algorithmen – nicht zuletzt mit dem Blick auf die leichte Erweiterbarkeit; ferner die Möglichkeit der Auswahl zwischen den beiden Simulationen bei Programmbeginn.