

Aufgabe 01 - Game of Life

Einleitung

Game of Life implementiert einen zellulären Automaten und simuliert wie sich Zellen, ausgehend von einer initialen Konfiguration, über mehrere Generationen hinweg entwickeln. Die Entwicklung der Zellen erfolgt dabei automatisch abhängig eines definierten Regelwerks. Bei Game of Life handelt es sich um ein sogenanntes Zero-player Game, also ein Spiel bei dem keine aktiven Mitspieler notwendig sind und das Spiel automatisiert anhand des Regelwerks abläuft. Das Spiel wurde von John Horton Conway erdacht. Figure @ref(fig:GameOfLife) zeigt einen Ausschnitt aus einer Game Of Life Implementierung.

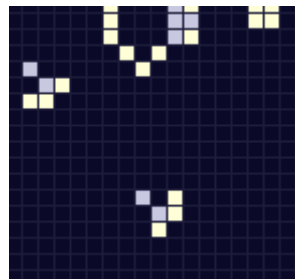


Figure 1: Ausschnitt einer Game Of Life Implementierung

Game Of Life wird auf einem zweidimensionalen Grid, bestehend aus Zellen, simuliert. Jede Zelle hat dabei acht Nachbarzellen und ist entweder lebendig oder tot. Abhängig von seinen Nachbarn entwickelt sich eine Zelle über die Generationen hinweg.

Es gibt für das Spiel viele verschiedene verschiedene Ausprägungen, aber in dieser Übung behandeln wir das Standard-Regelwerk. Dieses Regelwerk umfasst folgende Regeln:

- Reproduktion/Geburt einer Zelle: Hat eine tote Zelle exakt drei lebende Nachbarn, so ist die Zelle im nächsten Generationsschritt lebendig.
- Tot durch Überbevölkerung: Hat eine lebende Zelle mehr als drei Nachbarn, so stirbt die Zelle.
- Tot durch fehlende Nachbarn: Hat eine lebende Zelle weniger als zwei Nachbarn, so stirbt die Zelle.
- Überleben: Hat eine lebende Zelle zwei oder drei Nachbarn, so bleibt die Zelle am leben.

Die Simulation erfolgt anhand dieses Regelwerks und erfolgt in Zeitintervallen. In jedem Zeitschritt t_n werden die oben beschriebenen Regeln auf alle Zellen angewendet und dadurch die nächste Generation im Zeitschritt t_{n+1} erzeugt.

Aufgabe

Start: 2023-03-28 Ende: 2023-18-04

Implementierung von Game of Life in einer frei wählbaren Programmiersprache. Die Implementierung muss das oben beschriebene Regelwerk umsetzen. Eine graphische Visualisierung der einzelnen Simulationsschritte muss möglich sein. Die Ausarbeitung der Übung kann in 2er-Gruppen erfolgen.

Features

Zusätzlich zum Bereitstellen einer initialen Konfiguration soll die Simulation zu jedem Zeitpunkt angehalten werden können. Ist die Simulation angehalten, kann manuell eine Manipulation der Spielwelt erfolgen. Konkret sollen der Status einer Zelle (lebendig oder tot) geändert werden können. Alternativ kann dies auch direkt im Spiel (ohne Pausierung) erfolgen.

Zu jedem Zeitpunkt soll der Zustand der Spielwelt gespeichert und auch wieder geladen werden können. Überlegen Sie sich eine geeignete Datenstruktur, um den Zustand der Spielwelt zu serialisieren.

Ermöglichen Sie eine Steuerung der Geschwindigkeit der Simulation (um Eingriffe über die graphische Benutzeroberfläche zu ermöglichen).

Erstellen Sie zufällige, initiale Konfigurationen und prüfen Sie ob diese Konfiguration nach einer gewissen Zeit (nach einer vorgegeben Anzahl an Generationen) in einer statischen Welt (Zellen ändern sich nicht mehr), einer leeren Welt (alle Zellen sterben) oder einer wachsenden Welt (Zellen ändern sich) enden.

Abgabe

Die Abgabe erfolgt in Ilias bis 2023-18-04. Geben Sie den Source-Code und eine kurze Beschreibung der Funktionalität inkl. Screenshot Ihrer Implementierung ab.