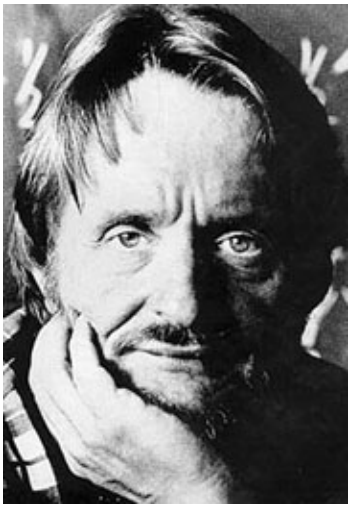

· John Horton Conway

Die von John von Neumann 1952 gestellte Frage, ob Automaten zur Selbstreproduktion fähig seien, kann als Ausgangspunkt für John Horton Conways 2- dimensionale zelluläre Automaten gesehen werden. Der Automat ist eine Ebene, die aus lauter einzelnen Zellen (Quadrate) aufgebaut ist. Jede dieser Zellen stellt einen endlichen Automaten dar, der 2 Zustände annehmen kann, lebend oder tot. Das Überleben der Zelle hängt entscheidend vom Zustand der Nachbarn ab. J. Conway stellt hier vergleichsweise einfache Regeln auf.

Eine Zelle stirbt an Vereinsamung wenn weniger als 2 Nachbarn besitzt, an Überbevölkerung, wenn Sie mehr als drei Nachbarn besitzt und wird geboren, wenn 3 Nachbarn leben. Mit jedem Schritt wird im gesamten zellulären Automaten über Leben und Tod entschieden und die Welt neu generiert.

Die dadurch entstehenden verschiedenste Gebilde (statisch, oszillierend oder zyklisch) beschäftigten Scharen von Informatikern. Eine Abwandlung der Regeln machte das Betätigungsfeld noch weiter und läßt auch Anwendungen außerhalb der Informatik zu. Eine Abfrage der Regeln läßt sich in Python relativ einfach realisieren. Eine mögliche Umsetzung in Python steht als Link zur Verfügung.

01 **def** go (self): 02 ....
[Source]

Das Spiel des Lebens wirft die Frage nach dem Verhalten einzelner Populationen auf, bzw. im Besonderen die Frage, ob es vorhersehbar ist, ob eine Population stirbt oder nicht. Mittlerweile wurde nachgewiesen, dass dieses Problem nicht entscheidbar ist. Eine vorhersehbare Entwicklung gibt es nicht. Die einzige Möglichkeit eine Entwicklung nachzuweisen ist die

Simulation durch ein Programm.
J. H. Cornway bot demjenigen 50 $, der nachweisen konnte, dass sogar unbegrenztes, geordnetes Wachstum möglich ist. Dieses Problem wurde durch die sogenannte Gleiterkanone gelöst. Simulationsprogramme sind im Internet verfügbar. Ein Beispiel ist Life 32 von Johan Bontes und die ebenfalls auf dieser Seite zu findende Kollektion von Objekten von Alan Hensel.
Die folgende Kundenklasse für obiges Programm basiert auf einer Koordinatenliste. Die Figuren können beliebig im Quelltext auf dieser Grundlage verändert werden. Das Programm wurde bewußt schmal gehalten um die Funktionalität auf Basis des Klassenkonzeptes zu verdeutlichen.

01 **def** figur (self, x, y, liste): 02 for i
[Source]