

Rin Räuber

**Artificial Art: Erzeugung künstlicher Bilder
mit evolutionären Algorithmen in Javascript**

Bachelorarbeit

Bibliografische Information der Deutschen Nationalbibliothek:

Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de/> abrufbar.

Dieses Werk sowie alle darin enthaltenen einzelnen Beiträge und Abbildungen sind urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsschutz zugelassen ist, bedarf der vorherigen Zustimmung des Verlanges. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen, Auswertungen durch Datenbanken und für die Einspeicherung und Verarbeitung in elektronische Systeme. Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

Copyright © 2013 Diplom.de
ISBN: 9783842815452

Rin Räuber

Artificial Art: Erzeugung künstlicher Bilder mit evolutionären Algorithmen in Javascript

Räuber, Rin: Artificial Art: Erzeugung künstlicher Bilder mit evolutionären Algorithmen in Javascript, Hamburg, Diplomica Verlag GmbH 2013

PDF-eBook-ISBN: 978-3-8428-1545-2

Herstellung: Diplomica Verlag GmbH, Hamburg, 2013

Zugl. Fachhochschule Dortmund, Dortmund, Deutschland, Bachelorarbeit, Februar 2013

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Bearbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Die Informationen in diesem Werk wurden mit Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig ausgeschlossen werden und die Diplomica Verlag GmbH, die Autoren oder Übersetzer übernehmen keine juristische Verantwortung oder irgendeine Haftung für evtl. verbliebene fehlerhafte Angaben und deren Folgen.

Alle Rechte vorbehalten

© *Diplom.de*, Imprint der Diplomica Verlag GmbH
Hermannstal 119k, 22119 Hamburg
<http://www.diplom.de>, Hamburg 2013
Printed in Germany

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Einleitung | 3 |
| 1.1 | Motivation | 5 |
| 1.2 | Zielsetzung | 7 |
| 1.3 | Gliederung | 8 |
| 2 | Einführung in Evolutionäre Algorithmen | 9 |
| 2.1 | Ursprung und Historie | 10 |
| 2.2 | Biologische Grundlagen und Terminologie | 11 |
| 2.3 | Ein einfacher evolutionärer Algorithmus | 11 |
| 2.4 | Arten Evolutionärer Algorithmen | 16 |
| 2.5 | Diskussion | 24 |
| 2.6 | Praktische Anwendung | 24 |
| 3 | Evolutionäre Kunst | 27 |
| 3.1 | Definition | 27 |
| 3.2 | Exploration versus Optimierung | 28 |
| 3.3 | Kunst mit interaktiver Evolution | 28 |
| 3.4 | Beispiele | 29 |
| 3.5 | Rezeption, Probleme und offene Fragen | 38 |
| 4 | Analyse | 41 |
| 4.1 | Evaluation von JS-Grafik-Frameworks | 41 |
| 4.2 | Existierende Projekte | 51 |
| 5 | Implementierung | 57 |
| 5.1 | Implementierung eines einfachen EA | 58 |
| 5.2 | Die grundlegende Implementierung | 59 |
| 5.3 | Die Fitnessfunktion | 61 |
| 5.4 | Die Repräsentation | 64 |
| 5.5 | Diversität | 68 |
| 5.6 | Selektion | 69 |
| 5.7 | Mutation | 72 |
| 5.8 | Rekombination | 81 |
| 5.9 | Startpopulation | 82 |
| 5.10 | Javascript-Performance | 84 |
| 6 | Ergebnisse | 87 |
| 6.1 | Einordnung des entwickelten Algorithmus | 87 |
| 6.2 | Entwickelte Bilder | 88 |
| 7 | Fazit und Ausblick | 95 |
| 7.1 | Fazit | 96 |
| 7.2 | Ausblick | 99 |

| | | |
|----------|---|------------|
| 8 | Anhang | 101 |
| 8.1 | Quellen | 101 |
| 8.2 | Kurz-Einführung Coffeescript | 111 |
| 8.3 | Liste aller gefundenen Projekte | 114 |
| 8.4 | Glossar | 116 |

Abstract

This thesis was aimed at developing an evolutionary algorithm in clientside Javascript which gradually evolves an artistic rendering for a given image by placing graphic primitives on a canvas.

As a prerequisite, evolutionary algorithms and their main paradigms and characteristics are discussed, followed by a introduction to evolutionary art and a presentation of some of its works. In addition, similar existing projects and candidates for the graphic framework are evaluated.

The implementation focuses on the choice of suitable methods and parameters like population size, selection method etc. and tests different combinations. For mutation operator probabilities, Niehaus' *Individual-Level Dynamic Probabilities* (IDP) were implemented successfully.

The resulting evolutionary artistic rendering technique, called the *Ballpit Effect*, is capable of producing interesting results, while the underlying implementation is readable and compact, and therefore might serve to spark the interest of developers outside of the EA community looking for a gentle introduction to the topic.

Zusammenfassung

Ziel dieser Arbeit war die Entwicklung eines evolutionären Algorithmus in clientseitigem Javascript, der ausgehend von einem gegebenen Bild ein sogenanntes *Artistic Rendering* entwickelt indem grafische Primitive auf einer Leinwand platziert werden.

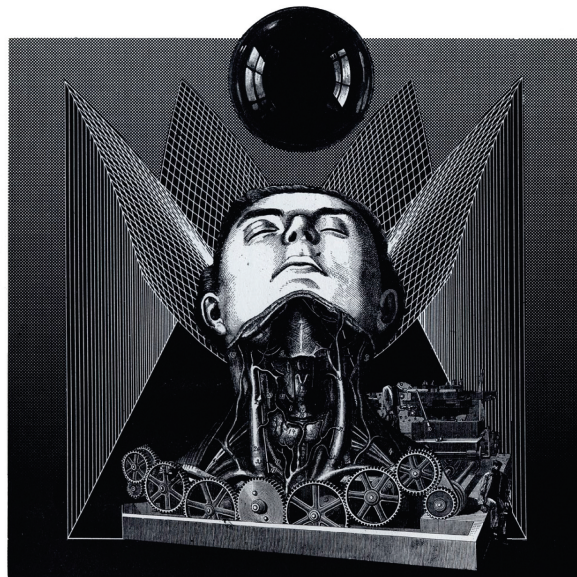
Einführend werden evolutionäre Algorithmen, ihre wichtigsten Paradigmen und Eigenschaften diskutiert. Anschließend folgt ein Überblick über evolutionäre Kunst und ihre wichtigsten Werke. Als Vorarbeit zur Implementierung werden ausserdem ähnliche, bereits existierende Projekte evaluiert und ein geeignetes Grafikframework ausgewählt.

Der Fokus der Implementierung liegt auf der Wahl geeigneter Methoden und Parameter wie Populationsgröße, Selektionsmethode, Mutationsrate etc., für die verschiedene Kombinationen erprobt werden. Zur dynamischen Anpassung der Wahrscheinlichkeit der Mutationsoperatoren wurden erfolgreich Niehaus' *Individual-Level Dynamic Probabilities* (IDP) implementiert.

Die entwickelte evolutionäre Rendering-Technik, der *Ballpit Effect*, ist in der Lage interessante Ergebnisse zu produzieren, während die zugrundeliegende Implementierung aufgrund ihrer Kompaktheit und Lesbarkeit dazu dienen kann, bei Entwicklern ausserhalb der EA-Community das Interesse an diesem Thema zu wecken.

1 Kapitel 1

1 Einleitung



1

Natural evolution is a very creative problem solver, and the solutions of nature are ever present to remind us just what evolution is capable of.

Can we make a machine 3 millimeters long that is capable of flying under its own power? What about giving it sight, or the ability to keep itself functioning by converting chemicals into energy, or even the ability to make copies of itself?

We simply do not have the know-how to achieve these marvels. But this is what nature has achieved in a creature as simple as a fly.

(Corne und Bentley, 2001b, S. 3)

¹Bild aus (Chamberlain und Etter, 1984, S. 23)

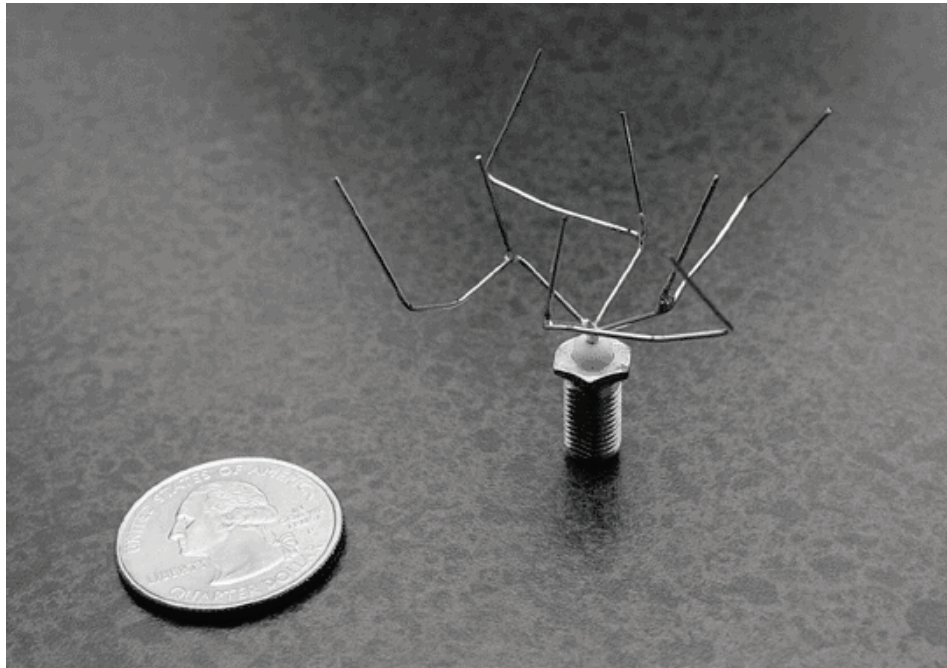


Abbildung 1.2: Prototyp der Antenne der ST5-Mission. (Lohn et al., 2005, S. 11).

Am 22. März 2006 starteten drei Mikrosatelliten ins Weltall um das Magnetfeld der Erde zu kartieren. An Bord dieser Satelliten befand sich eine Besonderheit: Die Antenne mit deren Hilfe Daten zur Erde gesendet wurden. Sie ist kaum drei Zentimeter groß, und sieht aus als hätte man einem gelangweilten Kind ein Stück Draht in die Hand gegeben (Abbildung 1.2). Diese Antenne ist nicht von einem Menschen konzipiert worden, sondern von einem Programm, das mit Hilfe eines evolutionären Verfahrens Millionen von Entwürfen entwickelte und schließlich einen als optimal auswählte (Bluck, 2004).

Über ein halbes Jahrhundert nachdem sich Informatiker zum ersten Mal damit beschäftigten, die natürliche Evolution als Vorbild für Problemlösungsverfahren heranzuziehen (Bremermann, 1958), flog damit zum ersten Mal ein mit künstlicher Evolution entwickeltes Objekt in den Weltraum (Bluck, 2006).

Die Antenne der ST5-Satelliten ist ein Beispiel für die praktische Anwendung einer Gruppe von Optimierungsverfahren, die als *Evolutionäre Algorithmen* bezeichnet werden. Evolutionäre Algorithmen bilden das Prinzip der natürlichen Evolution nach; sie benutzen Mutation, Rekombination und Selektion um aus einer Anfangsmenge zufälliger Lösungen die besten auszuwählen, deren Merkmale zu kombinieren und weiterzuentwickeln. Evolutionäre Algorithmen finden in zahlreichen Gebieten Anwendung, beispielsweise in den Ingenieurwissenschaften und in der Medizin.

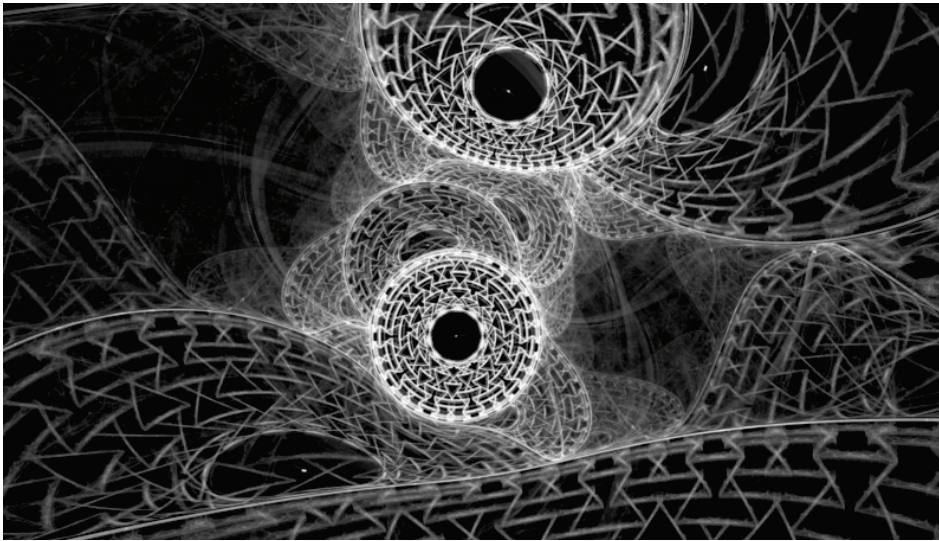


Abbildung 1.3: *Beispiel für ein Electric Sheep*³.

Ein weniger bekanntes, aber nicht weniger spannendes Einsatzgebiet für evolutionäre Algorithmen liegt in der sogenannten *generativen Kunst*. Generative Kunst ist Kunst, die als Ergebnis eines autonomen Prozesses, beispielsweise eines Programms, entsteht. Evolutionäre Kunst ist dementsprechend Kunst, die als Ergebnis eines evolutionären Prozesses entsteht. Ein Beispiel für evolutionär entwickelte Kunst ist Al Biles' *GenJam*, ein genetischer Algorithmus der Jazz-Soli erzeugt (Biles, 1994)². Ein anderes prominentes Beispiel ist der *Mondriaan Evolver*, ein Programm, das evolutionär Bilder ähnlich der streng geometrischen Gemälde des niederländischen Malers Piet Mondriaan entwickelt (Eiben, 2008). Ein zeitgemäßes Beispiel ist das Programm *Electric Sheep*, das als Bildschirmschoner funktioniert und so sich im Leerlauf bedingliche Rechner als Renderfarm benutzt um mit genetischen Algorithmen fraktale Animationen zu entwickeln und diese von Benutzern bewerten lässt (Draves, 2005) (siehe Beispiel in Abbildung 1.3).

1.1 Motivation

Für gewöhnlich wird ein Computerprogramm definiert als eine Menge geordneter Instruktionen, die einen Computer befähigen eine bestimmte Aufgabe automatisch auszuführen⁴. Bei unerwartetem Verhalten eines Programms handelt es sich in diesem Zusammenhang meist um einen Fehler. Die Instruktion `BE CREATIVE AND INVENT STUFF` gehört typischerwei-

²Ein aktueller Vortrag mit Demonstration findet sich unter <http://www.youtube.com/watch?v=rFBhwQUZGxg> (Stand 18.02.2013)

³Quelle: <http://v2d7c.sheepserver.net/cgi/dead.cgi?id=59383> (Stand 18.02.2013)

⁴aus *Britannica Concise Encyclopedia*, siehe <http://www.answers.com/topic/computer-program> (Stand 18.02.2013); interessanterweise sind die meisten aktuellen Definitionen sehr viel weiter gefasst.

se nicht zum Repertoire der Instruktionen; für die meisten Erfinder und Künstler dürfte der Computer ein Werkzeug wie Reißbrett oder Pinsel sein, aber kein mit eigener Kreativität ausgestatteter Gehilfe, geschweige denn Konkurrent. Für andere ist er jedoch genau das – er ermöglicht, aus unzähligen möglichen Kombinationen von Farben, Formen, Materialien, bestimmte auszuwählen und weiterzuentwickeln. Unerwartete Entwicklungen – Innovation und Originalität – sind hier erwünscht.

Besonders für interessierte Laien dürfte evolutionäre Kunst ein willkommener Farbkleck in der Landschaft der Informatik sein. Trotzdem existieren bisher nur wenige Beispiele⁵ evolutionärer Kunst, die zeitgemäße Web-Technologien nutzen und damit dieses immer noch etwas exotische Gebiet in nutzerfreundlicher Form einer größeren Öffentlichkeit zugänglich machen. Mit der vorliegenden Arbeit soll diese Lücke gefüllt werden.

1.1.1 Exkurs: Warum Javascript/Coffeescript?

Zum ersten Mal veröffentlicht wurde Javascript 1995 als Teil der *Netscape Navigator 2.0 Beta*. Die erste Anwendung war die Validierung von Formularen. Neun Jahre später verwendete Google in seinem Webmail-Dienst *Google Mail* als einer der ersten Javascript um Daten mit dem Server auszutauschen, ohne dass die Webseite neu geladen werden musste.⁶ Die als *Ajax (Asynchronous Javascript and XML)* bekannt gewordene Technologie ist maßgeblich für die heutige Popularität von Javascript verantwortlich. (Swartz, 2005) Auf Ajax folgten Javascript-Frameworks wie *Prototype* (Stephenson), *JQuery* (Resig) und *MooTools* (Proietti), die die Entwicklung vollwertiger Applikationen ermöglichten, welche sich für den Nutzer wie Desktop-Anwendungen anfühlen. Auch hier stammt mit *Google Docs* das prominenteste Beispiel von Google. Aus dem Browser, der ursprünglich nur ein Betrachtungswerkzeug für strukturierten Text war, ist eine Anwendungsplattform geworden – und zwar eine der am weitesten verbreiteten, die sowohl auf mobilen Geräte als auch auf Desktop-Rechnern verfügbar ist. Damit eignet sich Javascript besonders um die Idee der Erzeugung künstlicher Bilder mit evolutionären Algorithmen einer größeren Öffentlichkeit bekannt zu machen; zur Ausführung wird lediglich ein aktueller Browser benötigt. Technologien wie die HTML5 Canvas, SVG und die verfügbaren Grafikframeworks für Javascript machen die Bilderzeugung ausserdem relativ entwicklerfreundlich.

Im akademischen Kontext hat Javascript immer noch einen gewissen Exotenstatus⁷; das Thema der Arbeit ist in der Hoffnung gewählt, dass die

⁵Für existierende Projekte, siehe Kapitel 5.

⁶Die *XMLHttpRequest*-API (kurz *XHR*), die dies ermöglicht, war bereits seit 1999 Teil des Internet Explorers, wurde aber bis dahin kaum verwendet.

⁷Als Indiz kann man beispielsweise eine Suche in der Zitationsdatenbank *CiteSeer* heranziehen: Für den Zeitraum von 2010 bis heute finden sich knapp 300 Veröffentlichungen mit dem Wort "JavaScript", während es zu "Java" über 1.600 sind.