

ChaosPro

Martin Pfingstl

COLLABORATORS

	<i>TITLE :</i> ChaosPro		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Martin Pfingstl	January 7, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ChaosPro	1
1.1	Inhaltsverzeichnis	1
1.2	Vorwort	4
1.3	Warum sollte ich dieses Programm benutzen?	5
1.4	Systemvoraussetzungen	7
1.5	Installation	8
1.6	Autor	9
1.7	Konzept	9
1.8	PicTask	9
1.9	Farbpaletten	12
1.10	Palettenbearbeitung	14
1.11	Animationswindows	15
1.12	CycleControl-Window	19
1.13	Benutzerdefinierte Windows	20
1.14	Dockwindows	21
1.15	Formeleditor für Julia/Mandel	21
1.16	Formeleditor für IFS	29
1.17	Formeleditor für L-Systeme	30
1.18	Outputwindow	32
1.19	Networkwindow	32
1.20	2D/3D-Fraktalwindows	33
1.21	Juliamengen: Theorie	34
1.22	Mandelbrotmengen: Theorie	37
1.23	2.3 Fraktale --- 2.3.2 Julia- und Mandelbrotmengen	38
1.24	2.3 Fraktale --- 2.3.2 Julia- und Mandelbrotmengen	41
1.25	2.3 Fraktale --- 2.3.2 Julia- und Mandelbrotmengen	45
1.26	2.3 Fraktale --- 2.3.2 Julia- und Mandelbrotmengen	46
1.27	2.3 Fraktale --- 2.3.2 Julia- und Mandelbrotmengen	47
1.28	2.3 Fraktale --- 2.3.2 Julia- und Mandelbrotmengen	47
1.29	2.3 Fraktale --- 2.3.3 Bifurkationsdiagramme	49

1.30	2.3 Fraktale --- 2.3.3 Bifurkationsdiagramme	51
1.31	2.3 Fraktale --- 2.3.3 Bifurkationsdiagramme	52
1.32	2.3 Fraktale --- 2.3.4 Dynamische Systeme	52
1.33	2.3 Fraktale --- 2.3.4 Dynamische Systeme	53
1.34	2.3 Fraktale --- 2.3.4 Dynamische Systeme	54
1.35	2.3 Fraktale --- 2.3.5 Plasma	56
1.36	2.3 Fraktale --- 2.3.5 Plasma	57
1.37	2.3 Fraktale --- 2.3.6 Lyapunov-Raum	58
1.38	2.3 Fraktale --- 2.3.6 Lyapunov-Raum	59
1.39	2.3 Fraktale --- 2.3.6 Lyapunov-Raum	60
1.40	2.3 Fraktale --- 2.3.7 IFS	61
1.41	2.3 Fraktale --- 2.3.7 IFS	63
1.42	2.3 Fraktale --- 2.3.8 L-System	64
1.43	2.3 Fraktale --- 2.3.8 L-System	65
1.44	2.3 Fraktale --- 2.3.9 Diffusion	66
1.45	2.3 Fraktale --- 2.3.9 Diffusion	66
1.46	2.3 Fraktale --- 2.3.10 Brown	68
1.47	2.3 Fraktale --- 2.3.10 Brown	68
1.48	2.3 Fraktale --- 2.3.11 3D-Ansichten	69
1.49	2.3 Fraktale --- 2.3.11 3D-Ansichten	69
1.50	2.3 Fraktale --- 2.3.11 3D-Ansichten	71
1.51	2.3 Fraktale --- 2.3.11 3D-Ansichten	72
1.52	2.3 Fraktale --- 2.3.12 Das Wizardwindow	74
1.53	2.3 Fraktale --- 2.3.13 Das Kommentarwindow	75
1.54	2.4 Menüs	75
1.55	2.4 Menüs	78
1.56	2.4 Menüs	80
1.57	2.4 Menüs	81
1.58	2.4 Menüs	82
1.59	2.4 Menüs	86
1.60	2.5 Programmverzeichnisse	86
1.61	2.6 Preferencesprogramm	87
1.62	2.7 Problemecke	90
1.63	2.8 Sonstiges Erwähnenswertes	91
1.64	2.9 Tooltypes	92
1.65	2.10 Rechtliches	96
1.66	2.11 Gesucht...	97
1.67	2.12 Zur Geschwindigkeit	98
1.68	2.13 Geschichtliches & Änderungen seit V1.0	99
1.69	Humor, die zweite	108
1.70	2.14 Danksagungen & liebe Grüße...	110
1.71	2.15 Fähigkeiten von ChaosPro	112
1.72	Index	120

Chapter 1

ChaosPro

1.1 Inhaltsverzeichnis

Ich bitte darum, das Kapitel 2.15 zu lesen. In diesem Kapitel sind ←
die besonderen Fähigkeiten
von ChaosPro gesondert erläutert. In der Vergangenheit passierte es des öfteren, ←
daß sich Leute
beschwerten, daß z.B. das Abspeichern von 24 Bit Bildern nicht richtig ←
funktioniert, oder aber
keine vernünftige 3D-Transformation möglich war, etc. Dabei wurde nur nicht die ←
richtige Stelle in der Dokumentation gefunden, die das Verhalten
von ChaosPro erklärt.

Inhaltsverzeichnis

I.	Einleitung	
	1.1
	Vorwort	
	1.2
	Warum sollte ich dieses Programm benutzen?	
	1.3
	Systemvoraussetzungen	
	1.4
	Installation und Deinstallation	
	1.5
	Autor	
II.	Programmbeschreibung	
	2.1
	Das Programmkonzept	
	2.2
	Die verschiedenen Windows	
	2.2.1
	PicTask-Window	
	2.2.2
	Farbpalettenwindow	
	2.2.3
	Palettenbearbeitungswindow	
	2.2.4

Animationswindow	
.....	2.2.5
CycleControl-Window	
.....	2.2.6
Benutzerdefinierte Windows	
.....	2.2.7
Dockwindows	
.....	2.2.8
Formeleditor für Julia/Mandel	
.....	2.2.9
Formeleditor für IFS	
.....	2.2.10
Formeleditor für L-Systeme	
.....	2.2.11
Outputwindow	
.....	2.2.12
Networkwindow	
.....	2.3
Fraktale	
.....	2.3.1
Die 2D/3D-Fraktalwindows	
.....	2.3.2
Julia- und Mandelbrotmengen	
.....	2.3.2.1
Theorie: Juliamengen	
.....	2.3.2.2
Theorie: Mandelbrotmengen	
.....	2.3.2.3
Parameterwindow 1	
.....	2.3.2.4
Parameterwindow 2	
.....	2.3.2.5
Parameterwindow 3	
.....	2.3.2.6
Das Datenwindow	
.....	2.3.2.7
Das Formelwindow	
.....	2.3.2.8
Das Farbverteilungswindow	
.....	2.3.3
Bifurkationsdiagramme	
.....	2.3.3.1
Theorie	
.....	2.3.3.2
Parameterwindow 1	
.....	2.3.3.3
Datenwindow	
.....	2.3.4
Dynamische Systeme	
.....	2.3.4.1
Theorie	
.....	2.3.4.2
Parameterwindow 1	
.....	2.3.4.3
Parameterwindow 2	
.....	2.3.5
Plasma	

.....	2.3.5.1
Theorie	
.....	2.3.5.2
Parameterwindow 1	
.....	2.3.6
Lyapunov-Raum	
.....	2.3.6.1
Theorie	
.....	2.3.6.2
Parameterwindow 1	
.....	2.3.6.3
Datenwindow	
.....	2.3.7
IFS	
.....	2.3.7.1
Theorie	
.....	2.3.7.2
Parameterwindow 1	
.....	2.3.8
L-System	
.....	2.3.8.1
Theorie	
.....	2.3.8.2
Parameterwindow 1	
.....	2.3.9
Diffusion	
.....	2.3.9.1
Theorie	
.....	2.3.9.2
Parameterwindow 1	
.....	2.3.10
Brownsche Bewegung	
.....	2.3.10.1
Theorie	
.....	2.3.10.2
Parameterwindow 1	
.....	2.3.11
3D-Ansichten	
.....	2.3.11.1
3D-Einführung	
.....	2.3.11.2
3D-Parameterwindow 1	
.....	2.3.11.3
3D-Parameterwindow 2	
.....	2.3.11.4
3D-Parameterwindow 3	
.....	2.3.12
Das Wizardwindow	
.....	2.3.13
Das Kommentarwindow	
.....	2.4
Die Menüs	
.....	2.4.1
Systemmenü	
.....	2.4.2
Fraktalmenü	
.....	2.4.3

Fraktalwindows	
.....	2.4.4
Windows	
.....	2.4.5
Extras	
.....	2.4.6
Benutzerdefiniertes Menü	
.....	2.5
Programmverzeichnisse	
.....	2.6
Preferencesprogramm	
.....	2.7
Problemecke	
.....	2.8
Sonstiges Erwähnenswertes	
.....	2.9
Tooltypes, Startup-Parameter, Start vom CLI	
.....	2.10
Rechtliches	
.....	2.11
Gesucht...	
.....	2.12
Zur Geschwindigkeit...	
.....	2.13
Geschichtliches & Änderungen seit V1.0	
.....	2.14
Danksagungen & liebe Grüße...	
.....	2.15
Fähigkeiten von ChaosPro	
III.	
Index	

1.2 Vorwort

I. Einleitung

1.1 Vorwort

Was? Noch ein Programm zur Fraktalerzeugung?

Eigentlich könnte man meinen, daß es schon genug Fraktal\generatoren auf dem Amiga gibt. Doch riskiert man einen Blick auf andere Computersysteme und testet z.B. FractInt auf den PC's (nun auch auf dem Amiga.), so erkennt man schnell, daß es ← nicht recht weit her ist mit den vorhan\denen Fraktal\programmen: Keines von ihnen besitzt so viele Fraktal\typen und keines von ihnen erlaubt die Ein\stellung von so vielen Fraktal\parametern. Nun, sicher ist auch FractInt keineswegs perfekt, aber als Anregung allemal zu gebrauchen...

Wenn man dieses Programm betrachtet, so wird man etwas erstaunt feststellen, daß ← es doch recht umfangreich und leistungsfähig ist. Wiederholt haben sich die Benutzer ← von ChaosPro

gewundert, wieso dieses Programm denn Public Domain ist, werden doch für viele andere, weitaus schlechtere Programme schon Sharewaregebühren verlangt. Ich selber sehe die ganze Sache jedoch anders:

1. Wenn ich Geld mit Programmieren verdienen wollte, hätte ich mit Sicherheit kein Fraktalprogramm für den Amiga geschrieben.
2. Was würde es bringen, wenn ChaosPro Shareware wäre? Andere Autoren haben so etwas ja bereits versucht und waren dann sehr verärgert, daß sich kaum jemand für ihr ach so tolles Programm registrieren ließ, worauf sie die Entwicklung einstellten. Nun, ich frage mich, mit welchem IQ diese Leute ausgestattet sind. Wer sollte sich denn für ein Fraktalprogramm registrieren lassen? So ein Programm benutzt man nicht regelmäßig, es ist zu nichts nütze, man startet es nur, weil man zuviel Zeit hat und mal ein paar hübsche Bildchen sehen will, und das war es schon.
3. Also habe ich beschlossen, auf die Sharewaregebühr von vielleicht 5 Leuten zu verzichten, das bisschen Geld, das mir damit durch die Lappen geht, ist nicht der Rede wert. Stattdessen ist ChaosPro weiterhin Public Domain. So fühlen sich wenigstens mehr Leute berufen, mir zu schreiben und Vorschläge einzubringen. Somit fühle ich, daß ChaosPro benutzt wird, wodurch es vermehrt Spaß macht, daran herumzuprogrammieren.

Resümee: ChaosPro wird weiterentwickelt, sofern es Leute gibt, die Vorschläge machen, da meine eigenen Ideen beinahe restlos verwirklicht wurden.

1.3 Warum sollte ich dieses Programm benutzen?

1.2 Warum sollte ich dieses Programm benutzen?

Oder anderes gefragt: Was hat dieses Programm denn besonderes zu bieten gegenüber anderen Fraktalprogrammen? Nun, wenn Sie mit den bisherigen Fraktalprogrammen zufrieden waren und Sie niemals an deren Leistungsfähigkeit gestoßen sind, dann halte ich es für wahrscheinlich, daß Sie mit einem dieser anderen Fraktalprogramme vielleicht besser bedient sind. Dieses Programm erscheint Neulingen durchaus etwas verwirrend, da es relativ leistungsfähig ist. Wem es einfach nur darum geht, ein paar Fraktalbilder auf den Monitor zu bringen, für den ist dieses Programm schlichtweg etwas überdimensioniert. Man kauft ja Brilliance und DPaintIV AGA auch nicht nur, um Icons zu malen...

Folgend ein paar Features des Programms:

(Inspiziert von Mand2000Demo, FractInt, MisterM, MandelMania, Fractal Dynamics, Slicer, MultiFractals, MandelMountains, Fractal V1.3, MandelPlot 24, Mandelsquare, SmartFractal, LyapunoviaV1.5, CloudsAGA, KFP und neuerdings FractalUniverse)

- Multiwindowing

Alle Fraktale werden in Windows gezeichnet, die man einfach mit dem Sizingadget ↔ vergrößern kann.

- Multitasking

Für die Berechnung eines jeden Fraktals wird ein Task geschaffen, d.h. es können ↔ mehrere Fraktale gleichzeitig berechnet werden.

- Echtzeiteffekte

Die Änderung eines Parameters wirkt sich sofort aus.

- Click and Zoom

Durch Doppelklick auf eine Stelle wird hineingezoomt.

-Ausschnittverschiebung

Der Berechnungsausschnitt kann während der Berechnung verschoben werden, einfach ↔ mit der Maus das Fraktal anklicken, oder Cursortasten benutzen, oder den Joystick in Port 2...

- Systemkonformität

Programm läuft nach Aussage meiner Betatester bisweilen auf:

- Picasso
- Piccolo
- GVP EGS110/24
- GVP Spectrum
- ECS/OCS/AGA
- Merlin

Es läuft ab OS2.0 bis OS3.1, mit einem Screenmoderequester kann man die Auflösung wählen.

- Formeleditor

Für alle, die mal ein bisschen rumprobieren wollen - in meinem Programm allerdings ↔ steht dafür ein Stringgadget bereit, in das man die Formel einfach eintippt, nicht wie ↔ bei FractInt, wo man ein ASCII-File ändern und das Programm neu starten muß.

- mehrere Fraktaltypen

- Juliamenge
- Mandelbrotmenge
- Bifurkationsdiagramme (Verhulst)
- Dynamische Systeme
- Plasma
- Lyapunov-Räume

- Parameter

Es stehen je nach Fraktaltyp bis zu 3 Parameterwindows bereit.

- logischer Programmaufbau

Schlimmstes Beispiel eines Programms, das so eigentlich nicht existieren dürfte:
FractInt...

- 3D-Transformationen

In 3 weiteren Parameterwindows können allerlei Parameter eingestellt werden. Es ←
muß natürlich gesagt werden, daß
die Wahl der richtigen Parameter etwas Geduld erfordert. Doch das Multitasking des ←
Programms (die sofortige Auswirkung
der Änderung eines Parameters) hilft hierbei enorm. Man sieht sofort, ob nun der ←
Wert paßt
oder nicht (einen schnellen Amiga vorausgesetzt...:-)).

- Animationen

Nicht bloß simple Zoom-in Movies, sondern auch Zoom-out-Movies, oder irgendwelche
Animationen bzgl. irgendeines Parameters, der sich dann kontinuierlich ändert. ←

Natürlich

können sich auch mehrere Parameter ändern. Wie wäre es mit einer 3D-Animation in ←
eine

Juliamenge hinein, deren Parameterwert c sich dabei ändert? Nichts ist unmöglich, ←
sofern genug

Fantasie vorhanden ist.

- 24 Bit

Die Fraktale können in 24 Bit Farbtiefe abgespeichert werden.

- Online-Hilfe

Natürlich kontextsensitiv ;-)

- Locale-Unterstützung

Warum auch nicht?

- Arexx-Interface

Ich konnte es einfach nicht lassen ;-)

- und noch ein paar Kleinigkeiten, die mir jetzt so kurz einfallen:

- + Filename-Multiselect (wirklich eine Kleinigkeit)
- + Menu-Multiselect (sollte eigentlich selbstverständlich sein)
- + Farbrad unter OS3.0 beim Palettenbearbeiten
- + Abspeichern von Bildern ins Clipboard
- + Fontsensitivität

usw.

1.4 Systemvoraussetzungen

1.3 Systemvoraussetzungen

Als ich dieses Programm geschrieben habe, stand ich oft vor der Entscheidung, ob
ich ein Feature weglassen sollte, damit das Programm auch auf kleinen Amigas läuft ←

oder ob das Programm eben höhere Systemanforderungen stellen sollte. Ich habe mich
für letzteres entschieden, da ich finde, daß es im Jahre 1995 an der Zeit ist,

daß die Amigauser einsehen, daß man mit einem 7.09 Mhz-68000 Amiga mit 1 MB RAM und einem Diskettenlaufwerk nix mehr anfangen kann...
 Wenn ich das Programm so geschrieben hätte, daß es auch auf kleineren Amigas lief ←
 , dann
 würde es in dieser Form einfach nicht existieren, denn die Anpassung hätte viel zu ←
 viel Zeit
 gekostet. Herausgekommen wäre irgendein anderes Fraktalprogramm, das sich kaum von ←
 den
 existierenden unterschieden hätte. So hatte ich Zeit, um endlich einmal 24 Bit, ←
 Animationen,
 3D-Transformationen, etc. in ein Fraktalprogramm einzubauen.

Das Programm benötigt mindestens einen 68020 mit mathematischem Koprozessor. ←
 Aufgrund des
 internen Multitaskings kommt es auch zu einer ziemlichen Überfüllung des ←
 Bildschirms mit Windows,
 was sich leider nicht vermeiden läßt (oder doch?). Aus diesem Grund ist eine ←
 höhere
 Auflösung mit mehr Platz empfehlenswert.
 Die vielen Windows tragen natürlich auch nicht gerade zur Beschleunigung des ←
 Systems
 bei, so daß hier ein schneller Amiga doch auch eine Daseinsberechtigung hat.
 Einhergehend mit der Leistung ist auch eine ziemlich hohe Anforderung an den ←
 vorhandenen
 RAM-Speicher. Ab ca. 2 MB kann man damit arbeiten, doch mehr auch nicht...
 Auch hier gilt wieder: Je mehr, desto besser.
 Für die Betriebssystemversion gilt natürlich: Ab OS2.0 läuft das Programm, eher
 natürlich nicht.

1.5 Installation

1.4 Installation und Deinstallation

Die Installation wird mittels des Installers von Commodore erledigt.
 Wer es per Hand machen will, bitte sehr:
 1. Kopieren Sie die reqtools.library ins libs: Verzeichnis, falls sie dort noch
 nicht vorhanden sein sollte. Benötigt wird V38 oder höher.
 2. Kopieren Sie das Verzeichnis ChaosPro/ mit allen seinen Unterverzeichnissen
 irgendwohin. Lassen Sie alles beisammen, kopieren Sie nichts herum.
 3. Kopieren Sie den Inhalt des Fonts-Verzeichnisses in ihr FONTS: Verzeichnis..
 .

Damit ist die Installation bereits beendet. Zur Anpassung an Ihr System starten ←
 Sie bitte
 das

Preferences-Programm

Falls Sie gerade die Datei 'deutsch.guide' betrachten, wundern Sie sich nicht, ←
 wenn manche "Nodes"
 nicht gefunden werden und AmigaGuide deswegen Fehlermeldungen bringt. Diese Datei ←
 wird
 vom Preferences-Programm in die Datei ChaosPro.guide konvertiert. Hierbei werden ←
 Links zu unbekanntem Nodes in einer "magischen" Weise gelöst (Eigen\ ←
 implementation des

fehlenden Lokali\sierbar\keits\features).

Zur Deinstallation löschen Sie einfach das komplette ChaosPro-Verzeichnis. Schauen Sie auch
in libs: nach, vielleicht befindet sich dort aus irgendeinem Grund die Bibliothek ChaosPro.library, die sollten Sie dann auch löschen. ChaosPro kopiert von Haus aus nichts
auf Ihrer Festplatte herum...

1.6 Autor

1.5 Autor

Adresse:

Martin Pfingstl

Dorfen 16 1/5

84508 Burgkirchen

Tel.: Semesterferien: 0 86 79 / 62 41 Semester: 089 / 28 44 91

(Deutschland)

E-Mail: pfangstl@informatik.tu-muenchen.de

Bin sehr dankbar für:

- Bugreports
- Verbesserungsvorschläge
- Kommentare
- Registrierungen (quatsch! Untersteht euch ;-)

1.7 Konzept

II. Programmbeschreibung

2.1 Das Programmkonzept

...ist nicht neu, aber äußerst praktisch: Multitasking und Multiwindowing überall. Das heißt, es können so viele Fraktale gleichzeitig berechnet werden wie der Benutzer wünscht. Es kann gleichzeitig die Farbpalette bearbeitet, eine Animation berechnet werden, während ein Arexx-Programm irgendwas anderes im Programm steuert. Es kann die Online-Hilfe angezeigt werden und während man in ihr was liest, kann das Gelesene sofort praktisch getestet werden.

1.8 PicTask

2.2 Die verschiedenen Windows

2.2.1 PicTask-Window

Liste

- ChaosPro bietet die Möglichkeit, 5 verschiedene Listen von Fraktalbildern zu handhaben. Mit diesem Gadget wird die Liste ausgewählt. Bei einem Listenwechsel wird jedes Window, das zu einem Fraktal gehört, geschlossen, jede Fraktalberechnung abgebrochen und gelöscht. Der Sinn dieses Gadgets liegt darin, daß man eine Menge von Fraktalen erzeugen bzw. bearbeiten kann und dabei nur diese zu sehen bekommt, ohne daß man die normalen Fraktale dabei verliert. Z.B. existiert im Benutzerwindow 1 ein Gadget, das alle Frames, die durch eine Animation erzeugt werden würden, in Fraktalbilder verwandelt und als Fraktalbilder in die Fraktalliste einfügt. Hier ist es sinnvoll, vor der Benutzung des Gadgets auf eine leere Liste zu wechseln.

Fraktalbilder

- Im Sichtfenster mit der Überschrift Fraktalbilder werden sämtliche dem Programm augenblicklich bekannten Fraktalbilder angezeigt. Zu jedem dort vorhandenen Bild existiert eine Datenstruktur, die sämtliche Parameter, die zur Berechnung des Fraktalbildes nötig sind, enthält. Zu Beginn des Programms werden alle Fraktale, die im Verzeichnis 'FractPic' zu finden sind, eingeladen und hier angezeigt. Siehe dazu auch Kapitel 2.6

Programmverzeichnisse

.

Bildname

- Direkt darunter befindet sich ein String-Gadget, in dem der Name des Bildes geändert werden kann. Damit die Änderung wirksam wird, muß das Gadget durch Drücken der Return-Taste verlassen werden.

Bild löschen

- Durch dieses Gadget wird der angewählte Eintrag gelöscht.

Bild berechnen

- Etwas mehr tut sich, wenn man auf den Knopf 'Bild berechnen' klickt. In diesem Fall wird das Bild in die Taskliste aufgenommen, ein Window geöffnet und darin das Fraktal gezeichnet. Wie der Name des Sichtfensters vermuten läßt, wird dazu ein separater Task geschaffen, der die Berechnung komplett übernimmt. Dieser läuft mit einer um 1 niedrigeren Priorität, so daß die Berechnung die Programmsteuerung nicht beeinflußt, wohl aber die Steuerung die Berechnung.

Bild duplizieren

- Unter Umständen will man, ausgehend von einem Fraktalbild und seinen Daten, ein paar Parameter verändern ohne sich die alten Werte merken zu müssen. Aus diesem Grund existiert durch den Knopf 'Bild duplizieren' eine Möglichkeit, ein Bild zu klonen, also einen weiteren Eintrag in der Liste einzufügen, dessen Datenstruktur eine Kopie des gewählten Eintrages ist.

Windows schließen

- Dieses Gadget schließt alle zum aktuellen Fraktal gehörigen Windows, löscht also den Task, das Fraktalwindow und alle Parameterwindows.

Previewbreite/-höhe einstellen

- Hiermit wird die Größe des Ausschnitts eingestellt, der vorab berechnet wird. Der Ausschnitt wird mittig plaziert, eben da, wo mit der größten Wahrscheinlichkeit die interessanten Teile des Fraktals versteckt sind. Extreme Werte, seien sie klein oder groß, führen dazu, daß kein Preview berechnet wird. Bei manchen Fraktaltypen ist die Previeweinstellung wirkungslos, da das Zeichnen eines kleinen Ausschnitts entweder keinen Sinn macht oder gar nicht möglich ist.

Bildeinstellungen

3D-Puffertyp

Für Julia- und Mandelbrotmengen kann eine 3D-Transformation gewählt werden. Um nun gutaussehende Bilder zu ermöglichen, ist es möglich, hier einen Puffer für die 3D-Darstellung zur Verfügung zu stellen, so daß die 3D-Bilder nach der Berechnung in 24 Bit Farbtiefe abgespeichert werden können. Zur Beeinflussung des Aussehens der 3D-Bilder existieren im 3D-Parameterwindow Nr. 3 noch 2 Gadgets. Diese bestimmen, wie sich das einfallende Licht auf die Farben des Fraktals auswirken.

Puffertyp

- Es stehen 3 verschiedene Puffertypen zur Auswahl:

1. Gar kein Puffer: Diese Methode ist speichersparend. Allerdings können dann keine 3D-Darstellungen berechnet werden, da die Routinen dafür einen Puffer voraussetzen. Auch das Abspeichern als IFF-ILBM-Bild ist nur in der aktuellen Screentiefe möglich.
2. 16Bit-Int: Hier wird für jeden Punkt ein Wort (16Bit) reserviert, in dem der berechnete Wert abgelegt wird. Hier kann dann eine 3D-Ansicht gewählt werden. Außerdem ist es möglich, das Fraktal in einer beliebigen Farbtiefe zu speichern, z.B. in 24 Bit.
3. 32 Bit IEEE Single Precision-Puffer (für Speichermillionäre): Hier wird für jeden Punkt ein ganzes Langwort reserviert, in dem der genaue Wert des jeweiligen Punktes abgelegt wird. Hiermit ist es möglich, auch das Innengebiet der Julia- / Mandelbrotmenge als wahre 24Bit-Bilder zu speichern.

Windows

2 Wahlmöglichkeiten:

- 1 Window: Falls eine 3D-Ansicht des Fraktals gezeichnet werden soll, wird diese im selben Window wie das 2D-Fraktal gezeichnet. Das 2D-Fraktal wird einfach übermalt.

- 2 Windows: Falls eine 3D-Ansicht des Fraktals gezeichnet werden soll, wird ein

zweites Window hierfür geöffnet.

3D

Und wieder 2 Möglichkeiten:

- Kein 3D-Bild: Nur das Fraktal selbst wird gezeichnet. Keine 3D-Ansicht davon ←
wird

berechnet.

- 3D-Bild: Nach dem Zeichnen des 2D-Fraktals werden die Daten als Höhen ←
interpretiert und eine
3D Ansicht gezeichnet.

Palettenwahl

Wird ein neues Window aktiviert, versucht das Programm herauszufinden, welche ←
Palette
nun eingestellt werden soll. Dazu untersucht es das Window daraufhin, zu welchem ←
Fraktal

es gehört und stellt die entsprechende Farbpalette ein.

Das Programm hat stets eine globale Farbpalette. Zusätzlich existieren in jeder ←
Fraktalstruktur

noch 2 Einträge für Palettennamen. Der eine Palettenname ist für die Palette ←
gedacht, die für das 2D-Fraktal verwendet werden
soll, der andere Name für das 3D-Fraktal.

Zur Kontrolle über das mögliche Verhalten des Programmes bei der Palettenwahl ←
existieren 2

Gadgets. Falls das Checkboxgadget angewählt ist, wird stets die globale ←
Farbpalette

angezeigt, völlig unabhängig vom Fraktal und den beiden Palettennamen in der ←
Fraktalstruktur.

Dieser Modus ist vor allem für Leute wie mich gedacht, die verwirrt werden, wenn ←
bei einer plötzlichen Windowaktivierung

die Farbpalette geändert wird. (Ich benutze 'SunWindow' - kreierte von Bernhard ←
Scholz - Werbung! - fürs

automatische Aktivieren meiner Window...)

Falls das Checkboxgadget nicht angewählt ist, bestimmt das Cyclegadget daneben, ←
welche

Palette für das Fraktal eingestellt werden soll. 'Eigene Palette' stellt die ←
Palette

ein, die durch die Angaben in der Fraktalstruktur definiert ist. 'Globale Palette' ←
stellt

die globale Palette ein, wenn immer ein Window aktiviert wird, das zu diesem ←
Fraktal

gehört.

Änderungen der Palette wirken sich immer auf die eingestellte Palette aus, das ←
heißt,

wenn gerade die globale Palette eingestellt ist und im Palettenwindow eine andere ←
Palette

angewählt wird, so wird damit die globale Palette umdefiniert. Soll heißen, es ←
passiert eben genau das,

was man erwartet.

1.9 Farbpaletten

2.2.2 Farbpalettenwindow

Das Farbpalettenwindow enthält sämtliche Paletten, die das Programm beim Start im Verzeichnis 'Palette' gefunden hat. Dort können sich auch ganze Bilder befinden ↔

In diesem Fall wird aus dem Bild der Farbchunk mit den Farbinformationen ↔ herausgefilert, also die Palette des Bildes in die Liste aufgenommen.

Zu Beginn des Programms ist die Palette mit dem Namen 'Default' eingestellt. Bevorzugt irgendjemand also eine andere Farbpalette, so muß er sie nur unter dem Namen 'Default' im Palette-Verzeichnis ablegen. Existiert keine Palette mit diesem Namen, so wird die erste gefundene Palette eingestellt.

Will man nun eine andere Palette einstellen, so muß nur auf den entsprechenden Eintrag geklickt werden. Die Änderung erfolgt sofort. Damit die Bilddarstellung nicht leidet, können durch Paletten die Farben 0 bis 3 nicht beeinflußt werden. Sie werden benötigt, um den 3D-Effekt der graphischen Elemente aufrechtzuhalten und die Schrift lesbar zu halten. Andere Fraktalprogramme gingen hier ja mit ↔ schlechtem Beispiel voraus...

Palettenname

- Um den Namen einer Palette zu ändern, ist nur ein neuer Name in das ↔ Stringgadget einzutragen und Return zu drücken.

Palette bearbeiten

- Will man die Farben einer Farbpalette ändern, so klickt man einfach auf den Knopf mit der Aufschrift 'Bearbeiten'. Es öffnen sich 2 weitere Windows, eines in dem die Werte der aktuellen Farbe dargestellt werden und eines, in der ein Palettengadget die Farben anzeigt. Abhängig davon, ob auf dem aktuellen Screen 256 Farben zur Verfügung stehen, wird evtl. ein neuer Screen anhand der im Preferences-Programm zu ChaosPro eingestellten Daten geöffnet.

Siehe Kapitel 2.1.3

Palettenbearbeitungswindow

Palette laden und Palette speichern

- Mit diesem Knöpfen können Paletten eingeladen und abgespeichert werden. Wird beim Abspeichern einer Palette festgestellt, daß eine Datei mit dem gleichen Namen überschrieben würde, so erfolgt zur Sicherheit eine Abfrage. Beim Einladen wird ↔ natürlich

Multiselect unterstützt. Desweiteren lädt ChaosPro seit V2.0 auch Paletten des ↔ Programms

FractInt. Diese haben normalerweise die Endung '.map'.

Palette löschen und Palette duplizieren

- Die Namen sagen alles...

Farboffset

- Mit diesem Gadget stellt man ein, ab welcher Farbe die Farbpalette benutzt ↔ werden

soll. Beispiel: Jemand hat einen Screen mit 32 Farben, er stellt hier einen Wert ↔ von

30 ein. Der Screen erhält nun die Farben der Farbpalette ab Farbe 30 (wird zu ↔ Screenfarbe Nr. 4)

bis zu Farbe 57 (Screenfarbe 31). Ändert man den Wert kontinuierlich, so erhält man natürlich eine Art Colorcycling-Effekt. ↔

Überspringen

- Ist hier z.B. der Wert 2 eingestellt, so wird bloß jede 2. Farbe der Farbpalette hergenommen. Dies ist sinnvoll z.B. für Farbpaletten, die eigentlich für 256-Farb-Screens ausgelegt sind und die nun für z.B. 32-Farb-Screens hergenommen werden. Hier stellt man diesen Wert einfach auf 8, nimmt somit nur jede 8. Farbe der Palette her und schon hat man eine Vorstellung davon, wie es wohl auf einem 256-Farb-Screen aussieht. ↔

1.10 Palettenbearbeitung

2.2.3 Palettenbearbeitungswindow

Eigentlich sind es ja 3 Windows. Das eine ist für die Steuerung vorhanden, das andere wird zur Anzeige der Farben verwendet. Das 3. evtl. für das Colorwheel und den Gradientslider. ↔

Farbbereich

- Nicht jeder wird die Möglichkeit haben, sich alle 252 Farben - die ersten 4 Farben werden ja vom Programm für sich reserviert - gleichzeitig anzeigen zu lassen. ↔

Aus diesem Grund existiert dieses Gadget. Es zeigt mit der Größe und der Position seines Rollbalkens, wie viele Farben aus welchem Bereich der Palette im anderen Window angezeigt werden. Wird er verschoben, so verschieben sich gleichzeitig auch die angezeigten Farben im anderen Window entsprechend der neuen Position des Rollbalkens. ↔

Farbnummer

- Mit diesem Gadget läßt sich das Farbregister einstellen, gleichzeitig werden auch alle anderen Werte aktualisiert, wie z.B. der Rollbalken des Farbbereich-Gadgets bzw. die RGB- und FSH- Werte. ↔

Die RGB- und FSH- Regler

- Mit diesen Reglern kann man die Farbwerte ändern. Gleichzeitig mit einer Änderung der RGB- Werte werden auch die FSH- Werte aktualisiert und umgekehrt. Wer also bisher noch nicht so recht den Zusammenhang zwischen RGB und FSH erkannte, der kann diese Wissenslücke nun schließen. ↔

Kopieren, Austauschen, Farbnuancen

- 'Kopieren' kopiert die aktuelle Farbe an die Stelle, die der Benutzer im Palettengadget als nächstes anklickt, 'Austauschen' tauscht sie aus und 'Farbnuancen' schafft einen fließenden Übergang zwischen den beiden Farbregistern und ihren Farben. ↔

Cycling-Modus

- Etwas unkonventionell ist dieser Modus: klickt man das Gadget an, so befindet man sich im Modus, in dem man genau festlegen kann, welche Farben beim Colorcycling mitmachen sollen und welche nicht. Sinnvoll z.B. für die Mandelbrotmenge, wenn man nur das Außengebiet cyclen lassen will und nicht das (evtl. einfarbige) Innengebiet. Sämtliche Farben, die sichtbar sind, nehmen am Colorcycling teil, sämtliche Farben die in Grautönen blinken, nehmen nicht teil. Anklicken einer Farbe im Palettengadget ändert den Zustand desselbigen. Die 3 Gadgets 'Alle', 'Keines' und 'Invertieren' machen genau das, was man erwartet: 'Alle' läßt alle Farben am Colorcycling teilnehmen, 'Keines' keine Farbe und 'Invertieren' invertiert den Zustand jedes Farbregisters.

Bereichsfunktionen

- 'Stauchen zu' und 'Stauchen'
Mit diesen beiden Gadgets ist es möglich, die Palette auf weniger Farben zusammenzustauchen. Dazu wählt man mit dem Schieberegler die Anzahl an Planes, und mit 'Stauchen' löst man die Aktion aus.
- 'Bereich invertieren'
Damit lassen sich die Farben in einem Bereich umdrehen. Dazu klickt man auf das Gadget, dann auf die erste Farbe des Bereichs, dann auf die letzte Farbe des Bereichs.
- 'Bereich kopieren'
Dazu klickt man auf das Gadget, dann auf die erste Farbe des Bereichs, dann auf die letzte Farbe des Bereichs, und nun noch auf die erste Farbe des Zielbereichs. Überlappende, überlaufende, etc. Bereiche werden korrekt behandelt...

Farben einer Farbpalette

- Dieses Window zeigt die Farben der Farbpalette an. Es besitzt ein Sizegadget und kann daher in der Größe verändert werden.

Farbrad

Leute, die OS3.0 benutzen, können hier ein Farbrad haben, in dem auf recht intuitive Art und Weise die Farben gewählt werden können. Um das Farbrad zu erhalten, muß das ToolType COLORWHEEL angegeben werden. Da es die Hälfte der zur Verfügung stehenden Farben für seine Darstellung benötigt, kann man es auf diese Art und Weise an- und ausschalten.

1.11 Animationswindows

2.2.4 Animationswindows

Mit diesen Windows werden Fraktalanimationen berechnet. Dazu werden Fraktale,

die sich nur in kontinuierlich veränderlichen Parameterwerten unterscheiden dürfen ←
 ,
 als Keys definiert. Bei der Berechnung werden dann die Zwischenwerte zwischen den
 sich unterscheidenden Parametern berechnet (deshalb dieses 'kontinuierlich ←
 veränderlich')
 und einem Fraktal als Parameter übergeben, das dann berechnet wird. Dieses System ←
 ist
 recht leistungsfähig, so kann z.B. der Parameterwert c der Standardjuliamege ←
 kontinuierlich
 verändert werden, gleichzeitig können sich die Ausschnittwerte des Fraktals und ←
 die
 Iterationstiefe ändern.
 Der Effekt: z.B. ein Hineinzoomen in eine sich verändernde Juliamege, was einem ←
 effektvollen
 Morphing gleichkommt...

Bevor ich es vergesse, Information über Palettemorphing findet sich hier
 .

Doch nun zur Beschreibung der Gadgets:

Fraktalbilder

- Hier werden nochmal die Fraktale aus dem
 PicTask-Window
 angezeigt.

AnimKeys

- Das sind die Schlüsselpositionen. Eine Animation wird als kontinuierlicher
 Übergang von einem Key zum nächsten berechnet, bis hin
 zum letzten, berechnet.

Aktionen

- neuer Key / als erster

Damit wird das im Fraktalbilder-Gadget des Animationswindows angewählte Fraktal
 als neuer Key definiert. Der Key wird im einen Fall hinter dem angewählten Key
 in die Liste eingefügt, im anderen Fall an die erste Stelle. Beim Einfügen wird
 überprüft, ob dieser Key mit den in der Liste befindlichen Keys verträglich ist,
 d.h. ob es sich auch um denselben Fraktaltyp, um denselben Subtyp handelt und
 ob er sich von den anderen nur in kontinuierlich veränderbaren Parametern ←
 unterscheidet.

Andernfalls wird eine Fehlermeldung gebracht, evtl. mit dem Hinweis, welcher ←
 Parameter

nicht zu den anderen paßt, verbunden mit dem Angebot, den Parameterwert des neuen ←
 Fraktals

entsprechend anzugleichen. Zu bemerken ist noch, daß der neue Key den Namen des ←
 Ursprungsfraktals trägt und beim

Einfügen kopiert und NICHT referenziert wird.

Das heißt insbesondere, daß die Änderung eines Parameterwertes des Fraktals sich
 nicht auf den gleichnamigen Key auswirkt. Das schafft eine sehr schnelle ←
 Möglichkeit,

Animationen zu generieren. Dazu wird das Fraktal berechnet, als Key eingefügt, ←
 dann

zoomt man in das Fraktal hinein (oder ändert irgendeinen anderen Wert des Fraktals ←
) und fügt es erneut in die Key-Liste nach dem einen ein.

Fertig ist ein Zoom-In-Movie...

Der erste Key wurde ja kopiert, somit stehen seine Daten unwiderruflich fest und ←
 lassen

sich nicht mehr ändern.

- lösche Key

Ich spare mir eine Erklärung...

- Key aufwärts / Key abwärts

Damit läßt sich die Reihenfolge der Keys ändern...

- Key nach Bild

Ein Nachteil des Kopiertwerden der Keys ist, daß auf den Key nicht mehr ←
zugegriffen und nachträglich

Parameter geändert werden können. Es geht sogar so weit, daß überhaupt keine ←
Parameter mehr

betrachtet werden können. Ein Key ist kein Bild, also kann er nicht als Bild ←
berechnet werden, folglich können keine Parameterwindows

geöffnet werden. Wenn man also vom Freund eine AnimData-Datei bekommt, so weiß man ←
nicht einmal, welcher Fraktaltyp hierbei verwendet wird.

Mit diesem Gadget nun ist es möglich, einen Key in ein Bild zurückzuverwandeln, ←
sich die Parameter anzusehen, zu verändern, und dann den alten Key

zu löschen und dieses neue Bild als Key an die alte Position einzufügen. Perfekt ←
ist diese Lösung nicht, aber sie war einfach

... sorry.

- Start / Abbruch

Bei Betätigen des Gadgets 'Start' wird die Animationsberechnung gestartet. Dazu ←
wird

ein neues Fraktalwindow geöffnet, in dem das Fraktal berechnet wird. Natürlich ←
kann dabei

weitergearbeitet werden. Lediglich eine 2. Animation läßt sich nicht starten. Um ←
dies

dem Benutzer klarzumachen, werden alle Gadgets in diesem Window unwählbar, nur das ←
Abbruchgadget

bleibt anwählbar. Was es tut, ist wohl jedem klar...

- Laden/Speichern

Mit diesen Gadgets wird die Liste der Keys abgespeichert (kann dann z.B. bei einem ←
Freund mit einem schnelleren Amiga geladen und
dort berechnet werden) oder geladen.

Zeiteinstellungen

Das Animationssystem ist nun zeitorientierter als vorher. Es existiert eine ←
einstellbare

Zeiteinheit, die als Verweildauer eines einzelnen Frames angesehen werden kann. ←
Man sieht nun

gleich, wie lange eine Animation dauern wird, wann ein bestimmter Frame angezeigt ←
werden wird,

etc.

Moment

Diese beiden Gadgets, die nur lesbar sind, zeigen, zu welchem Zeitpunkt der gerade
aktuelle Keyframe gezeigt wird. Die Angabe erfolgt sowohl als Zeit in Sekunden vom ←
Start an als auch

als Frameanzahl vom Start der Animation an.

relativ zum letzten

Das obere Gadget zeigt die Zeitdifferenz zum letzten Key, d.h. wieviel Zeit ←
zwischen der Anzeige des

letzten Keys und des aktuellen Keys liegt. Das untere der beiden Gadgets ist einstellbar und bezeichnet die Anzahl der zu berechnenden Frames zwischen dem letzten Key und dem aktuellen.

total

Die beiden Gadgets, beide nur lesbar, zeigen, wie lange die Animation dauern wird, und aus wie vielen Frames sie derzeit besteht. Die Zeitdauer ist natürlich die Anzahl an Frames multipliziert mit der Zeiteinheit.

Zeiteinheit

Hier kann die Zeiteinheit eingestellt werden. Normal sollte ein Wert von 0.05 sein, d.h. 0.05 Sekunden pro Frame, also 20 Frames pro Sekunde. Ändert man diesen Wert, so ändern sich alle Zeiten, allerdings keine einzige Framezahl. Um die Framezahlen, aber nicht die Zeiten zu ändern, muß auf 'Zeit normalisieren' geklickt werden.

Zeit normalisieren

Dieses Gadget setzt die Zeiteinheit auf 0.05, läßt die Zeitpunkte jedes einzelnen Keys gleich und berechnet die Frameanzahlen neu. Angenommen, die Zeiteinheit ist 0.05 und man will die Frameanzahlen erhöhen, die Animation also quasi 'smoothen', so stellt man als erstes die Zeiteinheit auf einen Wert größer 0.05, z.B. 0.1, was die Animationsdauer erhöht (um den Faktor 2), aber die Framezahlen gleich läßt. Anschließend betätigt man 'Zeit normalisieren', was die Zeitdauer wieder auf 0.05 setzt, die Zeitpunkte jedes einzelnen Frames aber gleich läßt, also auch die Gesamtdauer der Animation. Das erhöht die Frameanzahl jeweils um den Faktor 2. Ergebnis: Die Animation ist dieselbe, aber sie besteht aus doppelt so vielen Frames. Entsprechendes gilt, falls man weniger Frames haben will.

Zeit berechnen

Dieses Gadget versucht, eine ideale Anzahl an zu berechnenden Frames zwischen dem letzten Key und dem aktuellen zu berechnen. Dazu ermittelt es alle Parameter, in denen sich die beiden Keys unterscheiden und berechnet anhand der Größenordnungen der jeweiligen Unterschiede eine Zahl.

Alle berechnen

Macht dasselbe wie 'Zeit berechnen', aber für alle Frames.

Sonstiges

Puffer

Hier wählt man den Puffer, der für die Berechnung gewünscht wird. Für 3D-Animationen oder 24 Bit muß ein Puffer vorhanden sein. Für 3D-Animationen empfiehlt es sich außerdem, einen IEEEFP-Puffer zu benutzen, sonst kann es zu unschönen Effekten kommen.

Interpolation

Hier kann man wählen, ob die einzelnen Keys linear interpoliert werden sollen oder

aber ein Spline (zur Zeit nur ein kubischer Spline) durchgelegt werden soll. Im Falle einer linearen Interpolation kommt es zu gewissen Ruckeffekten, besonders beim Hineinzoomen, was aber meines Erachtens gar nicht so schlecht aussieht. Dies wird bei der Splineinterpolation naturgemäß vermieden.

Speichermodus

Hier kann man wählen, ob die Animation im AnimOpt5-Format abgespeichert werden soll, so daß sie ohne Probleme mit allen gängigen Animationsplayern abgespielt werden kann, oder ob jedes einzelne Bild als Bild gespeichert werden soll. Im letzteren Fall kann man nach dem Start der Animation im Filerequester den Basisnamen angeben, die Einzelbilder bekommen dann noch Nummern angehängt. Diese Wahl ist nötig, da das AnimOpt5-Format keine 24Bit-Animationen zuläßt. Im Speichermodus=Bilder kann man bei der Planetiefe auch auf 24 gehen.

Breite / Höhe

- Hier wird die Größe der Animation eingestellt. Keine Angst, Falschmachen kann keiner was. Die Werte werden auf Korrektheit überprüft.

Planes

- Bei manchen Fraktaltypen ist es möglich, eine Animation mit bis zu 256 Farben zu berechnen, obwohl das Programm nur auf einem 16-Farb-Screen läuft bzw. die Hardware dazu überhaupt nicht in der Lage ist. Mit diesem Gadget wird eingestellt, in wievielen Planes Sie die Animation berechnen lassen wollen. Evtl. kann sie dannach ja mit z. B. Clarissa nachbearbeitet, konvertiert, etc. werden. Der Wertebereich dieses Gadgets reicht normalerweise von 3 bis 8, falls der Speichermodus 'Bilder' ist, enthält er alle Werte von 3 bis 8 und den Wert 24.

Startframe&Endframe

Hiermit kann man den Start- und den Endframe angeben. Falls ihr Computer während einer Berechnung plötzlich versagen sollte, können sie nun auch die Restanimation berechnen lassen. Falls für EndFrame 0 eingetragen wird, heißt das, daß die Animation bis zum maximalen Frame berechnet wird.

3D-Animation

- Manch ein Fraktaltyp kann in 3D dargestellt werden. Mit Hilfe dieses Gadgets wird bestimmt, ob eine 3D-Animation gewünscht wird. Hierzu werden bei der Berechnung automatisch beide Windows, das 2D- und das 3D-Window geöffnet und der Inhalt des 3D-Windows jeweils als Animationsframe abgespeichert.

1.12 CycleControl-Window

2.2.5 CycleControl-Window

Dieses Window ist dazu gedacht, die Kontrolle über das Colorcycling leichter zu machen. ↔

Dazu existieren 3 Gadgets:

Colorcycling

- Schaltet Colorcycling an und aus

Speed

- Stellt die Geschwindigkeit ein, Wertebereich von 20 bis 999.

Richtung

- Soll aufwärts oder abwärts gecycled werden?

1.13 Benutzerdefinierte Windows

2.2.6 Benutzerdefinierte Windows

Es können eine beliebige Anzahl an Windows vom Benutzer definiert werden. Alle Windows ↔

bestehen aus einer vertikalen Knopfleiste. Wenn ein Knopf betätigt wird, so wird das dem Knopf zugeordnete Arexx-Script ausgeführt. Auf diese Art können einzelne nicht ↔

implementierte Funktionen dem Programm hinzugefügt werden.

Der Aufbau der Windows steht in der ASCII-Datei 'Windows.asc', die sich in ChaosPro/Prefs ↔

befinden muß.

Aufgebaut ist sie wie folgt:

```
WINDOW <Windowtitel> <Arexx-Scripts>
GADGET <Gadgetname> <Arexx-Script>
...
GADGET <Gadgetname> <Arexx-Script>
WINDOW <Windowtitel> <Arexx-Scripts>
GADGET <Gadgetname> <Arexx-Script>
...
GADGET <Gadgetname> <Arexx-Script>
WINDOW <Windowtitel> <Arexx-Scripts>
...
END
```

Das Arexx-Script, das in der Zeile mit dem WINDOW-Schlüsselwort steht, wird jedesmal ↔

beim Öffnen dieses Windows ausgeführt. Zu beachten ist, daß ein Fehler in dieser Datei bis zum Absturz führen kann, also Achtung.

Ist die Datei geschrieben, so muß sie vom

Preferencesprogramm
aus übersetzt werden.

Ich hoffe, daß dieses Programm möglichst alle Fehler abfängt, so daß es zu keinem Absturz bei Fehleingaben kommt...

Die übersetzte Datei wird automatisch in ChaosPro/Prefs/ unter dem Namen Windows.prefs ↔

abgespeichert.

1.14 Dockwindows

2.2.7 Dockwindows

Derzeit besitzt das Programm 3 Dockwindows.

Dock1

Nr.1 kontrolliert alle Windows, die für jedes Fraktal getrennt geöffnet werden können.

Welches Gadget welches Window öffnet, kann man, sofern man es nicht sowieso gleich sieht, ja einfach durch ausprobieren erfahren.

Dock2

Nr.2 kontrolliert die Windows, die im Programm nur einmal geöffnet werden können.

Dock3

Nr.3 bietet ein paar Aktionen, die auf Fraktale angewendet werden können. Diese Aktionen sind im

Menü auch zu finden, aber Anklicken ist einfacher. Die ersten beiden Gadgets führen ein 'UnDo' und ein 'ReDo' aus, symbolisiert durch 2 kleine unterschiedliche Fraktale, die in einer Liste vorhanden sind.

Mit 'Undo' bewegt man sich in der Liste zurück (Pfeile nach links), mit 'ReDo' bewegt man sich

in der Liste vorwärts (Pfeile nach rechts). Darunter finden sich die Aktionen 'Box Zoom hinein' und 'Box Zoom heraus', darunter noch 'Bild speichern als IFF'.

1.15 Formeleditor für Julia/Mandel

2.2.8 Formeleditor für Julia/Mandel

1. Beschreibung der Formeln
2. Beschreibung der Gadgets
3. Der Parser und seine Funktionen
4. Fehlermeldungen
5. Beschreibung des ASCII-Fileformats
6. Einfache Erzeugung eigener Formeln

ChaosPro verfügt über einen relativ leistungsfähigen Formeleditor. Mit dessen Hilfe

ist es möglich, eigene Formeln zu kreieren und die Julia- bzw. Mandelbrotmengen davon zu berechnen.

Es ist sinnvoll, sich zuerst mit der Theorie der Julia- und Mandelbrotmengen vertraut zu machen. Andernfalls wird man im folgenden Verständnis\schwierigkeiten

haben, da der Formeleditor, soweit ich weiß, seit ChaosProV2.0 der leistungsfähigste überhaupt ist. Leider ist er damit auch der am schwersten Verständliche...

Es folgt nun eine Beschreibung der Fähigkeiten. Anschließend werden die Gadgets

genauer erklärt:

ChaosPro unterscheidet bei jeder Formel mehrere Teile. Der erste Teil ist der Initialisierungsteil. Alle mir bekannten Formeleditoren besitzen hier nur die Möglichkeit, eine einzige Initialisierung anzugeben. Doch ist es z.B. nötig, bei einer Mandelbrotmenge die Bahnen sämtlicher kritischen Punkte zu verfolgen. Bei der Standardmandelbrotmenge z^2+c gibt es nur einen kritischen Punkt (Nullstelle der ersten Ableitung), nämlich die 0. Um diese Formel einzugeben, wäre hier also im Initialisierungsteil nur ein Element einzutragen. Doch andere Formeln haben unter Umständen mehrere kritische Punkte, die allesamt untersucht werden müssen, um die Mandelbrot- bzw. Juliamenge zu berechnen

Mit ChaosPro ist dies nun möglich.

Der zweite Teil einer Formel ist der Iterationsteil. Auch hier sind mehrere Elemente möglich. Für die Standardmandelbrotmenge wird auch hier nur 1 Element eingetragen, nämlich einfach $z^2+pixel$. Jedes Iterationselement besteht seinerseits aus 2 Teilen, einer Bedingung für die Iteration und der Iterationsformel selbst. Um die Sache zu verstehen, ist es nötig, das ChaosPro zu kennen: Zuerst wird die Bedingung ausgewertet. Kommt dabei der Wahrheitswert 'wahr' heraus, so wird die zugehörige Iterationsformel auf z angewandt. Ansonsten wird die Bedingung des nächsten Elements ausgewertet, dann die Bedingung des übernächsten Elements, bis eben die Bedingung eines Elements 'wahr' ist. Ist keine 'wahr', so liegt ein Fehler in der Formel vor. ChaosPro führt dann automatisch die Iterationsformel des letzten Elements aus. Warum dieses Vorgehen? Barnsley, ein Fraktalspezialist, hat eine besondere Klasse von Fraktalen untersucht. Dabei wird eben die Iterationsformel während der Iteration geändert. Ist der Realteil von z kleiner 0, so wird die Iterationsformel $(z+1)*c$ ausgeführt, ansonsten eben $(z-1)*c$. Mit ChaosPro können nun auch solche Fraktale berechnet werden.

Der dritte Teil einer Formel ist der Abbruchteil. Hier kann die Abbruchbedingung angegeben werden. Die 'normalen' Abbruchbedingungen wie 'Test auf unendlichen Attraktor', 'Test auf endlichen Attraktor' stehen nach wie vor zur Verfügung. Zusätzlich kann man im Parameterwindow Nr.2 noch das Checkboxgadget 'Formeldefiniert' anwählen, wodurch abgebrochen wird, sobald eben die definierte Abbruchbedingung erfüllt ist. Damit kann man dann beliebige Bereiche definieren, wie z.B. Abbruch, falls Realteil von z kleiner als -10 oder ähnlich.

Wie geht ChaosPro nun bei der Berechnung eines Fraktals, für das eine Formel erzeugt wurde, vor?

1. hole erstes Initialisierungselement
2. führe die Initialisierung aus, d.h. z = Ergebnis der Formelauswertung

3. hole erstes Iterationselement
4. werte die Bedingung aus
- 4a. falls Bedingung='wahr', dann führe die Iterationsberechnung aus und gehe zu ↔ Schritt 5
- 4b. falls Bedingung='falsch', dann hole nächstes Iterationselement und gehe zu ↔ Schritt 4
- 4c. falls keines mehr da ist, führe die letzte Iterationsberechnung aus
5. teste der Reihe nach auf die verschiedenen Abbruchkriterien, ist MaxIt erreicht ↔ ?
- 6a. falls kein Abbruch, so gehe zu Schritt 3
- 6b. Abbruch, hole nächstes Initialisierungselement, sofern noch eines da ist, und ↔ gehe zu Schritt 2. Falls keines mehr da ist, dann Punkt berechnet.
7. Zu allen Initialisierungselementen wurden Werte berechnet. Bestimme das Minimum ↔ aus dieser Menge. Der Punkt bekommt diesen Wert zugewiesen.

Beschreibung der Gadgets:

Modus

Dieses Gadget bestimmt den Inhalt des Listviews.

Es gibt 3 Möglichkeiten:

1. Alle verfügbaren Formeln werden durch ihren Namen angezeigt.
2. Alle Initialisierer der gerade gewählten Formel werden angezeigt.
3. Alle Iterationselemente der gerade gewählten Formel werden angezeigt.

Rechts daneben steht der Name der aktuellen Formel.

4 Gadgets rechts neben dem ListView

Mit diesen 4 Gadgets werden neue Formeln hinzugefügt oder geladen, alte Formeln ↔ gelöscht oder gespeichert.

Gadgets unter dem ListView

Hier wird die Formel geändert. Langatmige Erklärungen kann ich mir, denke ich, ↔ sparen. Entweder man hat das

Prinzip der Formelerzeugung verstanden, dann versteht man diese Gadgets, oder man ↔ hat es nicht verstanden, dann

sollte man mich informieren, damit ich dieses Kapitel überarbeiten kann...

Bem.:

Die ersten 6 Formeln sind im Programm eingebaut, daher schneller, als wenn sie mittels des Formeleditors erstellt würden.

Der Parser und seine Funktionen

Der Parser unterscheidet nicht zwischen Groß- und Kleinschreibung. Mehr als 2 ↔ Parameter

sind nicht erlaubt - sie haben keinen Platz im Window für die Parameter von Julia-

und Mandelbrotmengen. Mit Parameter 1 ist dort stets der 1. im Alphabet \leftrightarrow
vorkommende
Parameter gemeint.

Der Parser versteht folgende Funktionen:

+ - * / ^	- Plus, Minus, Mal, Geteilt, Hoch
sin	- die Sinusfunktion
cos	- die Cosinusfunktion
tan	- die Tangensfunktion
asin	- die Arcussinusfunktion
acos	- die Arcuscosinusfunktion
atan	- die Arcustangensfunktion
abs	- die Betragsfunktion (Absolutwert)
ln	- der natürliche Logarithmus
exp	- die Exponentialfunktion
log	- der Logarithmus zur Basis 10
sinh	- der Sinus Hyperbolicus
sqrt	- Square Root, also die Wurzel
tanh	- Tangens Hyperbolicus
cosh	- Cosinus Hyperbolicus
cotan	- Cotangens
cotanh	- Cotangens Hyperbolicus
conj	- das konjugiert komplexe einer Zahl
real	- der Realteil einer komplexen Zahl
imag	- der Imaginärteil einer komplexen Zahl
acot	- der Arcuscotangens
asinh	- Area Sinus Hyperbolicus
acosh	- Area Cosinus Hyperbolicus
atanh	- Area Tangens Hyperbolicus
acoth	- Area Cotangens Hyperbolicus
arg	- das Argument einer Zahl (der Winkel von 0 bis 2*pi)
e	- Konstante: die Eulersche Zahl
i	- Konstante: die Imaginäre Einheit
p	- Konstante: Pi - die Kreiszahl
z	- dieser Wert ist die Variable der Formel und sollte deshalb sinnvollerweise vorkommen
pixel	- komplexe Zahl, die der Bildschirmposition entspricht.
12.44	- eine Zahl - oder auch irgendeine andere, wird als Konstante behandelt
&&	- logisches AND
	- logisches OR
!	- logisches NOT
==	- logisches EQUAL
>=	- größer oder gleich
<=	- kleiner oder gleich
!=	- ungleich
>	- größer
<	- kleiner
t	- TRUE
f	- FALSE

Fehlermeldungen

- "Fehler in der Formel entdeckt. Wahrscheinlich Klammersetzung fehlerhaft. \leftrightarrow
Übersetzung

abgebrochen..."

Hier sollte die Klammersetzung überprüft werden. Der Parser stieß in diesem Fall bei offener

Klammer auf das Ende der Formel oder aber auf eine schließende Klammer, ohne noch eine öffnende Klammer zur Verfügung zu haben.

- "Fehler in der Formel entdeckt. Ein unbekanntes Zeichen angetroffen. Übersetzung abgebrochen..."

Hier stieß der Parser zu Beginn seiner Übersetzung auf ein unbekanntes Zeichen und hat

sofort abgebrochen.

- "Fehler in der Formel entdeckt. Irgendetwas stimmt mit den Operatoren nicht... Übersetzung abgebrochen..."

Dieser Fehler darf nicht auftreten, sonst ist ein Fehler im Parser...

- "Formel zu komplex. Es sind maximal 2 Parameter erlaubt. Übersetzung abgebrochen ..."

Diese Fehlermeldung dürfte klar sein.

Außer z als Variable dürfen nur 2 Parameter vorkommen, bestehend aus einem Buchstaben ungleich

e, i, p, z...

- "Fehler in der Formel. Anzahl an Operatoren paßt nicht zur Anzahl an Operanden, kurz: entweder zu viele Operatoren oder zu viele Operanden. Übersetzung abgebrochen..."

Diese Fehlermeldung zeigt an, daß bei einem Testdurchlauf der Formel zum Schluß noch Operanden

übrig waren oder aber beim Auswerten plötzlich kein Operand mehr da war, mit dem die durch den

Operator bestimmte Aktion ausgeführt werden könnte. Klingt kompliziert, also ein Beispiel, das diese

Fehlermeldung provoziert:

a) a**b

Hier versucht er die Formel auszuwerten. Es existieren 2 Operatoren, nämlich 2 mal ein Multiplikationszeichen. Multipliziert werden

immer 2 Operanden, man sagt, die Multiplikation ist eine dyadische Operation. Um die Formel korrekt auszuwerten, wären aber nun 3

Operanden nötig, es sind aber nur 2 da, nämlich 'a' und 'b'. Hier sind also zu viele Operatoren im Vergleich zu der Anzahl an Operanden

vorhanden.

b) b b

Hier ist kein Operator da, aber 2 Operanden. Der Parser startet also einen Testdurchlauf, ist sofort fertig und merkt, daß nicht einfach ein

Operand übrigbleibt, den er als Ergebnis interpretieren würde, sondern 2. Also sind nun zuviele Operanden

vorhanden...

Beschreibung des ASCII-Fileformats

Im folgenden findet sich ein Beispielfile, das eine Formel erzeugt. Dieses File kann unverändert von ChaosPro geladen und verarbeitet werden, ergibt aber wenig Sinn, da es als Beispiel für alle Möglichkeiten dienen soll. Es kann als Vorlage für eigene Experimente dienen.

Die ersten 4 Zeichen müssen CPFR lauten, was für "ChaosPro FoRmula" stehen sollte...

```

-----
CPFR
Easy_Formula Das ist der Name der Formel, einfach ein String, er darf keine
Spaces
enthalten.
{
Hier darf wieder Kommentar stehen. ChaosPro liest einfach bis zur nächsten
geschweiften
Klammer.
{ 0          z*z+c } Hier werden die kritischen Punkte definiert. Meist wird man
hier nur ein Element angeben. Im Beispiel sind 2
kritische Punkte angegeben, 0 und z*z+c...
{
    Real(z)<0    z^2+c
    z<0         3^z-c
} Das hier sind die verschiedenen Iterationsregeln, also jeweils Paare von Strings
, der erste String ist die Bedingung, der 2. String ist die Iterationsregel

{ z*z<8 } Zum Schluß noch das Abbruchkriterium. Nach der nächsten schließenden
geschweiften Klammer
kann die nächste Formel folgen mit dem Formelnamen etc., eben genau wie diese
Formel
hier.
}
----- end of file -----

```

Die Routine in ChaosPro, die dieses Format liest, liest immer nur ganze Strings als Elemente ein.

Will sie keinen String mehr, so überliest sie alles, bis sie auf die nächste geschweifte Klammer '}' bzw. '{', je nach Zusammenhang trifft.

Deshalb braucht z.B. zwischen dem Formelnamen und dem evtl. vorhandenen Kommentar dannach nur ein Leerzeichen

stehen. ChaosPro liest einen String, interpretiert diesen als Namen und liest dann weiter bis zur nächsten öffnenden geschweiften Klammer '{'.

Ein Problem ergibt sich noch: Ist ein Fehler in diesem File, so meldet ChaosPro den Fehler. Allerdings sagt ChaosPro im wesentlichen nur, daß ein Fehler vorhanden ist und meldet nicht wo...

Einfache Erzeugung eigener Formeln

Es scheint so, als ob es sehr wenig bis gar keine Leute gibt, die mit dem Formelgenerator von ChaosPro

zurechtkommen. Daher versuche ich nun, ein Beispielfile sowie ein paar Anregungen für eigene Experimente anzugeben.

Am besten und verständlichsten ist es, wenn per ASCII-Editor ein Formelfile erzeugt wird. Folgendes ASCII-File berechnet z.B. die eingebaute Standard\mandel\brot\menge:

```
---- Start of file ----
CPFR
Mandelbrot
{
  { 0 }
  { T      z*z+pixel }
  { abs(z)>16 }
}
---- end of file ----
```

Was passiert hier? Nun, der Name wird auf 'Mandelbrot' gesetzt, der kritische Punkt (wer nicht weiß, was das ist, soll sich nicht den Kopf zerbrechen, notfalls einfach auf 0 setzen) wird auf 0 gesetzt. Dies liefert die mathematisch korrekte Mandelbrotmenge. Wer an hübschen Bildern interessiert ist, kann diesen Wert z.B. auch anders setzen, etwa auf 1 setzen, oder aber auf $\sin(\text{pixel})/2$ oder irgendwas, was einem gerade einfällt. 'z' soll darin allerdings nicht vorkommen, denn z soll dadurch ja gerade definiert werden, d.h. 'z' ist undefiniert und wird auf diesen Wert gesetzt, also $z:=0$ oder $z:=1$ oder $z:=\sin(\text{pixel})/2$. So, dannach folgt der Iterationsteil. Entweder man hat den Teil ganz weit oben verstanden, dann weiß man, was dieses 'T' zu bedeuten hat, oder aber man hat ihn nicht verstanden, dann einfach merken: 'T' hinschreiben (steht für den Wahrheitswert TRUE), Space dannach, dann die Iterationsvorschrift, in diesem Fall 'z*z+pixel'. Hier kann man natürlich auch andere Iterationsvorschriften eintragen, z.B. $z^3+\text{pixel}$, $z^4+\text{pixel}$, $\exp(z)-z+\text{pixel}$, oder sonstwas, was lustig aussieht. Zuletzt kommt noch das Abbruchkriterium, das frei definiert werden kann. Im Falle der Standardmandelbrotmenge nimmt man einfach $\text{abs}(z)>16$ als Kriterium für den Abbruch der Iteration. Man beachte allerdings, daß dieses Abbruchkriterium im Parameterwindow 2 noch extra ausgewählt werden muß. Dort muß bei den Abbruchbedingungen ein Haken bei 'Formeldefiniert' angebracht werden (am besten dort und nur dort! Es sei denn, der Benutzer weiß, was er tut, oder aber es kommen trotzdem hübsche Bilder heraus...). Für das Abbruchkriterium kann natürlich auch wieder etwas x-beliebiges gewählt werden, z.B. finde ich auch $\exp(z)*\sin(z)>1$ recht lustig, obwohl dies natürlich aus mathematischer Sicht recht wenig Sinn macht, aber wen stört es?

Das wichtigste an einer Formel ist der Iterationsteil. Hier ist es unter Umständen schwierig, Formeln zu finden, die schöne Bilder ergeben. Aus diesem Grund möchte ich hier nun eine große Klasse von sinnvollen Iterationsformeln angeben. Mit Fraktalprogrammen im allgemeinen und mit ChaosPro im speziellen kann man recht schön das Newton'sche Verfahren zur Nullstellenbestimmung anwenden. Dazu ist lediglich Kenntnis über das Ableiten von Funktionen nötig. Dieses Verfahren wird folgendermaßen implementiert:

---- Start of file ----

CPFR

Newton_Test

```
{
  { pixel }
  { T      z-f(z)/f'(z) }
  { abs(z)>16 }
}
```

---- end of file ----

Das Abbruchkriterium ist nicht von Bedeutung. Abgebrochen wird bei der Newton-Iteration dann, wenn

ein Fixpunkt der Iterationsvorschrift erreicht ist, was einer Nullstelle der gewählten Funktion

entspricht. Dazu ist im Parameterwindow 2 als Abbruchbedingung 'endlich' anzuwählen. Sonst macht die Newton-Iteration

keinen Sinn.

Der Initialisierer wird auf 'pixel' gesetzt, das ist Absicht. Ich will hier nicht auf die

mathematische Bedeutung eingehen, jeder mit hinreichend genauen mathematischen Kenntnissen wird hier den

absoluten Durchblick haben, jeder andere bräuchte vermutlich einige Semester Mathematikstudium. Für

$f(z)$ muß natürlich eine Funktion eingesetzt werden. Von dieser Funktion muß dann noch die Ableitung berechnet werden.

Dannach kann man den Ausdruck $z-f(z)/f'(z)$ noch vereinfachen, was eine Steigerung der Geschwindigkeit von ChaosPro bei

der Formelauswertung zur Folge hat.

Für $f(z)$ kann man z.B. z^3+2 eingeben, dann ist $f'(z)=3*z^2$, somit lautet dann die Iterationsvorschrift

$$z - (z^3+2)/(3*z^2)$$

Oder aber $f(z)=\sin(z) \implies f'(z)=\cos(z)$, also ist als Iterationsvorschrift $z-\sin(z)/\cos(z)$

einzutragen.

Oder aber $f(z)=\sin(z)*z \implies f'(z)=\sin(z)+\cos(z)*z$, also ist $z-\sin(z)*z/(\sin(z)+\cos(z)*z)$

einzutragen.

All dies wird am besten deutlich, wenn man eigene Formeln nach dem obigen Schema erzeugt. Hat man erst einmal die Formeln im ASCII-File stehen, so speichert man

es einfach im Verzeichnis 'ChaosPro:Formula' ab oder lädt die Formel in ChaosPro ein. Dannach kann man sie ja innerhalb von ChaosPro mit dem

Formeleditor bearbeiten. Da man die Formeln selbst erzeugt hat, wird man sich hier nun

wesentlich leichter beim Ändern tun.

Eine Anmerkung zum Schluß: Die Laufvariable 'pixel' muß vorkommen, sei es im Initialisierungsteil, oder sei

es im Iterationsteil. Dies ist klar, denn 'pixel' ist der einzige Wert, der sich von

Punkt zu Punkt ändert. Kommt 'pixel' nicht vor, so wird für jeden Pixel im Window exakt

dasselbe berechnet, was natürlich zu einer einfarbigen Fläche führt. Als Faustregel kann man sich merken: Kommt 'pixel' im Initialisierungsteil vor, so

erhält man ein

Fraktal vom Typ 'Juliamenge', kommt 'pixel' in der Iterationsvorschrift vor, so erhält man einen Abkömmling einer

Mandelbrotmenge. Wie gesagt, das ist nur eine Faustregel. 'pixel' darf überall vorkommen, und was dann herauskommt, ist schön anzusehen, aber nicht unbedingt von mathematischem Interesse.

1.16 Formeleditor für IFS

2.2.9 Formeleditor für IFS

Ich gehe wieder davon aus, daß die theoretischen Grundlagen für iterierte Funktionsschemata bekannt sind. Ansonsten verweise ich auf das entsprechende Kapitel.

Es geht also darum, ein System affiner Transformationen zu definieren, wobei jeder Transformation eine Wahrscheinlichkeit für die Ausführung eben dieser Transformation zugeordnet ist. ChaosPro ist dann in der Lage, den Attraktor des IFS zu berechnen.

Beschreibung der Gadgets

Im Listview sind natürlich alle verschiedenen dem Programm bekannten IFS aufgeführt. Darüber befindet sich ein Gadget, in dem man den Namen des IFS ändern kann.

Rechts daneben befinden sich 5 Knöpfe, mit denen man:

1. ein neues IFS hinzufügen kann. Man wird dabei nach der Anzahl der Transformationen gefragt, die das IFS haben soll. Die Anzahl an Transformationen läßt sich nachher nur noch indirekt ändern, nämlich über das Klonen eines IFS.
2. das gerade aktive IFS klonen kann. Man wird dabei nach der Anzahl an Transformationen gefragt, die das zu erzeugende IFS haben soll. Die Werte des gerade aktiven IFS werden dann in die neue Formel hineinkopiert, zumindest so viele Werte, wie eben möglich sind.
3. ein IFS von einem Datenträger laden kann.
4. das gerade aktive IFS speichern kann.
5. das gerade aktive IFS löschen kann.

Wenn man sich beim Hinzufügen eines IFS 'vertan' hat und zu wenig Transformationen angefordert hat, so bleibt einem also nur noch die Wahl, das IFS zu klonen und dabei dann eben mehr Transformationen anzufordern.

Unter den 5 Knöpfen befindet sich der Schieberegler, der die Nummer der anzuzeigenden und somit veränderbaren Transformation angibt. Unter dem Schieberegler befindet sich noch das

Gadget, mit dem man die Wahrscheinlichkeit der Ausführung eben dieser Transformation bestimmen kann.

Im unteren Teil des Windows befindet sich eine 3x3-Matrix, die eben für die Rotation und Skalierung verantwortlich ist. Beschränkt man sich auf 2D-IFS, so sind natürlich nur die linken oberen 2x2 Elemente auszufüllen, die 3. Zeile und 3. Spalte darf dann nur Nullen aufweisen. Neben der Matrix ist noch ein Vektor mit 3 Elementen, der für die Verschiebung zuständig ist. Auch hier gilt: Im zweidimensionalen Fall ist die dritte Komponente auf 0 zu setzen.

Es besteht die Möglichkeit, den Elementen der Matrix, des Vektors oder der Wahrscheinlichkeit keine feste Zahl, sondern eine Variable zuzuordnen. Dies macht durchaus Sinn, da ja im Parameterwindow Nr. 1 für IFS 4 Variablen a, b, c und d zur Verfügung stehen. Will man also ausprobieren, was passiert, wenn man die erste Komponente des Vektors ändert, so bietet es sich an, eine Animation darüber zu berechnen. Das Animationssystem bekommt jedoch von den Formeln gar nichts mit, es arbeitet nur auf Basis der Parameter eines Fraktals, also nur auf Basis der Werte, die im Parameterwindow Nr.1 stehen. Also trägt man als erstes Vektorelement einfach die Variable 'a' ein und ändert im Parameterwindow Nr. 1 die Zahl a, bzw. läßt eine Animation berechnen.

Leider ist es so, daß man ja nur Gleitkommazahlen in die Elemente des Formeleditors eingeben kann. Daher kann man kein 'a' hineinschreiben. Deshalb wurden die Buchstaben a, b, c und d in Zahlen kodiert. Statt 'a' trägt man '100' ein, statt 'b' '101', statt 'c' '102' und statt 'd' '103'.

1.17 Formeleditor für L-Systeme

2.2.10 Formeleditor für L-Systeme

Wieder findet sich hier ein Listview, das alle Formeln des Typs LSystem anzeigt und darüber ein Gadget, in dem der Name der Formel angezeigt wird und den man verändern kann.

Darunter befinden sich wieder 5 Knöpfe:

1. 'Hinzu': Damit fügt man eine L-System-Formel hinzu. Man wird nach der Anzahl der Regeln gefragt, die die Formel haben soll. Dannach wird man nach der maximalen Eintragsgröße pro Regel gefragt.

Das bestimmt natürlich die Größe der Formel, diesen Wert sollte man also nicht blindlings groß wählen. Lästig ist es aber auch, wenn er zu klein ist und man dies erst später beim Eintippen der Formel merkt.

2. 'Klonen': Damit klont man die gerade aktive Formel. Wieder wird man nach der Anzahl an Regeln und der maximalen Größe pro Regel gefragt. Die aktive Formel wird dannach

automatisch in die neu erzeugte Formel kopiert, wobei natürlich darauf geachtet wird, daß auch genug Platz vorhanden ist. Hat man sich beim Hinzufügen einer neuen Formel leicht vertan, also die Anzahl an Regeln oder die GröÙer einer Regel zu niedrig gewählt, so klont man die Formel einfach und wählt größere Werte.

Löschen, Laden und Speichern sollte wohl klar sein...

Im unteren Bereich des Windows kann man dann die eigentliche Formel angeben. Dies ist zuerst das Axiom, dannach noch der Winkel, den ein '+' oder '-' bewirken soll, darunter befindet sich wieder ein Schieberegler, der die Regel auswählt, die dann darunter angezeigt wird. Eine Regel hat immer die Form '<Zeichen> = <Zeichenkette>', keine Spaces einfügen...

So, nun zu den Kommandos, die das Programm versteht:

Man stelle sich immer eine Schildkröte vor, die sich drehen, die Zeichenfarbe ändern, vorwärts gehen und solche Sachen machen kann.

F: Zeichne eine Linie in die aktuelle Richtung mit der aktuellen Farbe in der aktuellen Länge.

f: Selbes wie 'F', bloß wird nichts gezeichnet...

+: Dreht die Schildkröte um den gerade aktuellen Winkel im Gegenuhrzeigersinn. Zu Beginn ist dies der im Formeleditorwindow eingestellte Winkel.

-: Macht dasselbe wie '+', Schildkröte wird aber im Uhrzeigersinn gedreht

.

[: Schiebt alle änderbaren Daten, also die Länge eines Schrittes, die Farbe, die Position, die Richtung, etc. auf einen Stack

]: Gegenstück zu '[', holt sich einen Zustand der Schildkröte vom Stack, d.h. die Schildkröte bekommt eine neue Position, eine neue Blickrichtung, eine neue Farbe, etc.

|: Schildkröte dreht sich um 180 Grad...

Besondere Kommandos:

'a', 'l' und 'c': Diese Kommandos beeinflussen den Winkel ('a' für 'Angle'), die Länge eines Schrittes der Schildkröte ('l') und die Farbe ('c' für 'Color').

In dieser Form ist die Sache natürlich sinnlos, es muß noch eine Zahl oder etwas ähnliches

dannach kommen, um die Werte zu ändern. Um eine schnelle Abarbeitung einer Formel zu ermöglichen, ist das Zahlenformat

sehr streng festgelegt. Es kostet immens viel Zeit, jede Zahl unter Berücksichtigung jedes Formats zu konvertieren, so daß man damit wirklich rechnen

kann.

Daher gilt:

- Entweder folgt eine Zahl der Form xx, dann wirkt das wie <Buchstabe> = xx

Beispiel: a12 oder a03 ==> a=12 oder a=3

- Oder es folgt ein '+' oder '-' und dannach noch eine Zahl der Form xx, dann \leftrightarrow
wirkt das wie $\langle \text{Buchstabe} \rangle = \langle \text{Buchstabe} \rangle + \langle \text{xx} \rangle$
- Beispiel: $a+02$ oder $a-13 \implies a=a+3$ oder $a=a-13$
- Oder es folgt ein '*' oder '/' und dannach eine Zahl der Form $\langle x.x \rangle$ dann wirkt \leftrightarrow
das wie $\langle \text{Buchstabe} \rangle = \langle \text{Buchstabe} \rangle * \langle x.x \rangle$
- Beispiel: $c*1.3$ oder $c*0.1 \implies c=c*1.3$ oder $c=c*0.1$

Anmerkung:

1. Das Kommando 'a' bewirkt keine Drehung. Es stellt lediglich den Drehwinkel ein. \leftrightarrow
Die Schildkröte kann man nur mit den Kommandos '+' und '-' drehen, dabei wird \leftrightarrow
der gerade aktuelle Drehwinkel
benutzt.
2. Statt 'a', 'l' und 'c' darf man auch 'A', 'L' und 'C' schreiben.

1.18 Outputwindow

2.2.11 Outputwindow

Dieses Window zeigt gelegentlich zusätzliche Informationen an. Die Absicht war, \leftrightarrow
daß ChaosPro manchmal
Dinge weiß, die es dem Benutzer gern mitteilen würde, z.B. falls der Benutzer ein \leftrightarrow
Fraktal in einer Tiefe
von 24 Bit abspeichert, aber als Außenfärbung nur 'Iteration' gewählt hat. Einen \leftrightarrow
Requester zu öffnen ist eine
schlechte Idee, da hierbei der Benutzer gezwungen ist, auf 'Ok' zu klicken, was \leftrightarrow
den Arbeitsfluß doch enorm
bremst. Nun erscheint einfach eine Meldung in diesem Window, wenn ChaosPro meint, \leftrightarrow
daß irgendeine Aktion des
Benutzers eines Kommentars würdig ist oder sonst etwas passiert ist, was dem \leftrightarrow
Benutzer mitgeteilt werden
sollte.

1.19 Networkwindow

2.2.12 Networkwindow

Im Listview werden die ChaosPro bekannten Tasks angezeigt, die zur Berechnung von \leftrightarrow
Fraktalen
über das Netz zur Verfügung stehen.
Jeden einzelnen kann man nach Belieben aktivieren bzw. deaktivieren. Allerdings \leftrightarrow
wirkt sich eine Änderung nur auf Fraktale aus,
die dannach berechnet werden.
In den Einträgen finden sich andere Informationen wie die Laufzeit (Runtime), die \leftrightarrow
ein Kommando von ChaosPro aus über
das Netz zu diesem Task braucht, eine Einschätzung der Geschwindigkeit des \leftrightarrow
Rechners, auf dem der Task läuft
(ein A4000/040 mit 25 Mhz hat rund 5800 Einheiten), die CPU, die FPU und der freie \leftrightarrow
Speicher. Im letzten Eintrag
steht zuerst die Gesamtanzahl berechneter Pixel, dannach die Anzahl Pixel, die \leftrightarrow
speziell von diesem Task aus berechnet wurden,
und dannach noch diese Angabe in Prozent.
Näheres zum Netzwerk findet sich in Kapitel 2.15, Unterkapitel über die \leftrightarrow
Netzwerkfähigkeiten von

ChaosPro.

1.20 2D/3D-Fraktalwindows

2.3 Fraktale

2.3.1 Die 2D/3D-Fraktalwindows

Im 2D-Fraktalwindow wird stets das den Parametern entsprechende 2D-Fraktal - bei den beiden dynamischen Systemen ist es bereits eine Projektion eines dreidimensionalen Fraktals - angezeigt. Ändert sich ein Parameter, so wird sofort das Fraktal neu berechnet.

Folgende Aktionen sind im Window möglich (bei Typ Plasma nicht):

1. Cursor-Tasten bzw. Joystick in Port 2
Damit verschiebt man das Fraktal um 8 Bildschirmpixel in die entsprechende Richtung.
2. Leertaste bzw. Feuerknopf Joystick in Port 2
Hineinzoomen in das Fraktal, je nach vorhandenem Speicher werden Zwischenzooms dadurch "berechnet", daß das ursprüngliche Bild durch Skalierung vergrößert wird.
3. Fraktal anklicken und Maus bewegen
"Packt" das Fraktal und verschiebt es. Hat man einen schnellen (einen sehr schnellen) Rechner, so kann man auf diese Weise über das Fraktal in Echtzeit hinwegfliegen.
4. Doppelklick auf eine Stelle
Zoomt um den Faktor 2 in das Fraktal hinein und plazierte dabei die Stelle, auf die gedoppelklickt (wurks) wurde, mittig im Window.

Im 3D-Fraktalwindow wird stets das den Parametern entsprechende 3D-Fraktal angezeigt.

Eine 3-dimensionale Ansicht ist nur bei Julia-/Mandelbrotmengen möglich. Prinzipbedingt ist es bei den dynamischen Systemen (das sind ja bereits 3D-Fraktale) und bei Bifurkationsdiagrammen nicht möglich, ein 3D-Fraktalwindow zu öffnen. (Was sollte man auch bei den Bifurkationsdiagrammen zeichnen?)

Alle anderen nicht unterstützten Tastendrücke werden vom KeyboardControl-Modul zuerst an das Datenwindow zum Fraktal weitergeleitet, werden sie auch von ihm nicht verstanden, dann an das Parameterwindow 1, dann 2, dann 3. Somit ist es möglich, im Fraktalmenü den Tastaturshortcut für das Erhöhen der Iterationstiefe zu drücken

und die Iterationstiefe wird tatsächlich erhöht, als ob die Taste im \leftrightarrow
entsprechenden
Parameterwindow gedrückt wurde. Ich hielt das für sinnvoll, da bei mir immer das \leftrightarrow
falsche Window aktiv ist.
(Murphys Gesetz, die Nummer hab' ich vergessen)

1.21 Juliamengen: Theorie

2.3.2 Julia- und Mandelbrotmengen

2.3.2.1 Theorie: Juliamengen

weiterhin:

2.3.2.2 Theorie: Mandelbrotmengen

Ich beziehe mich im folgenden auf die 'Standardformel' $f(z)=z^2+c$.
Bei der Erzeugung von Juliamengen ist die komplexe Zahl c beliebig zu Beginn \leftrightarrow
wählbar,
aber konstant während der Berechnung. Jeder Punkt im Window entspricht einer \leftrightarrow
komplexen
Zahl. Welcher, ergibt sich aus den Ausschnittwerten im Parameterwindow Nr. 1. Die
Frage ist nun, was passiert, wenn man z mit dem dem Bildschirmpunkt entsprechenden \leftrightarrow
komplexen Zahlenwert initialisiert
und dann die Formel iterativ anwendet. Also:
 z =dem Bildschirmpixel entsprechende komplexe Zahl
 $z_1 = f(z) = z^2+c$
 $z_2 = f(f(z)) = f(z_1) = z_1^2+c$
 $z_3 = f(f(f(z))) = f(z_2) = z_2^2+c$
...
Die Juliamenge besteht nun aus allen Punkten z , für die die Punktbahnen \leftrightarrow
nirgendwohin
führen, oder, anderes gesagt (kontrapositiv), alle Punkte, die NICHT zur \leftrightarrow
Juliamenge
gehören, haben eine Bahn, die irgendwohin führt, zu einem sogenannten anziehenden \leftrightarrow
Punkt,
einem Attraktor, bzw. allgemeiner gesagt, zu einer anziehenden Menge, die aus \leftrightarrow
mehreren
Punkten bestehen kann. Das heißt nun, daß die Juliamenge nicht dieses tolle bunte
Gebilde ist, sondern die - nicht unbedingt sichtbare - schwarze Fläche. Das, was
so bunt eingefärbt ist, sind die Punkte, deren Bahnen zu einem anziehenden Punkt
hinführen und die eben nicht zur Juliamenge gehören.

kritischer Punkt: $f'(z)=0$, also $2z=0$, also $z=0$

Fixpunkte: $z=f(z)$, $0.5 \pm \sqrt{0.25-c}$ und $z=\text{unendlich}$

- Fixpunkte und Eigenwerte

Wie der aufmerksame Leser bereits bemerkt haben wird, wird die Juliamenge \leftrightarrow
weitgehend
von den anziehenden Punkten bestimmt, Punkten also, für die $z=f(z)$ gilt, den \leftrightarrow
anziehenden
Fixpunkten. Es ist durchaus möglich, daß ein Punkt z_0 ein Fixpunkt ist, aber nicht \leftrightarrow
anziehend,
d.h. ist z sehr nahe bei z_0 , so 'flüchtet' er unter Umständen geradezu von z_0 weg. \leftrightarrow
Die

Frage ist nun, wann ist ein Fixpunkt z_0 anziehend und wann abstoßend. Dazu braucht man nur die Ableitung der Formel $f(z)$ zu berechnen, in diesem Fall also $f'(z)=2*z$. Nimmt man z aus einer sehr kleinen Umgebung (eigentlich einer unendlich kleinen) von z_0 und berechnet $f(z)$, so erkennt man, daß ungefähr gilt: $|f(z)-z_0|=|f'(z_0)*(z - z_0)|$

(vgl. Tangente).

Man erkennt also, daß Anziehung oder Abstoßung von $f'(z_0)$, dem sogenannten Eigenwert

des Fixpunktes z_0 , abhängt. Ist $f'(z_0)<1$, so ist der Abstand von $f(z)$ und z_0 kleiner als

der Abstand von z und z_0 , also ist z_0 anziehend. Ist $f'(z)>1$, so vergrößert sich der Abstand, der

Fixpunkt z_0 ist abstoßend Ist $f'(z_0)=1$, so ist der Fixpunkt neutral. Hier können dann viele interessante

Erscheinungen auftreten.

In der Standardformel $f(z)=z^2+c$ sind die Fixpunkte durch Lösen der Gleichung $f(z)=z$ herausfindbar.

Die Lösungen dieser Gleichung sind:

$$z_1=0.5+\sqrt{0.25-c}$$

$$z_2=0.5-\sqrt{0.25-c}$$

Will man also eine interessante Juliamenge berechnen lassen, so muß man c so wählen, daß die Beträge der Eigenwerte der Fixpunkte z_1 und z_2 kleiner als 1 sind.

Außerdem muß man aus theoretischen Gründen den Punkt 'unendlich' auch als Fixpunkt ansehen, obwohl klar ist, daß dieser 'Punkt' in der Praxis alles andere als anziehend ist.

Dieser Punkt ist in jedem Fall als anziehend einzustufen. Das Berechnen des Eigenwertes ist

hier ziemlich sinnlos. Da der Punkt 'unendlich' immer als anziehend einzuordnen ist, haben

es sich so ziemlich alle Fraktalgeneratoren bisher sehr einfach gemacht, indem sie nur den unendlich fernen Punkt als einzigen Fixpunkt annahmen. Na ja, richtig ist es nicht, aber

die charakteristischen Bilder der Juliamengen in den verschiedenen Publikationen lassen sich so berechnen.

Die ganze Sache von den Fixpunkten kann noch ein bisschen komplizierter werden.

Zu Beginn habe ich schon erwähnt, daß es nicht nur anziehende Punkte, sondern sogar

anziehende Mengen geben kann, sogenannte Zyklen mit einer Zykluslänge. Dies ist dann z.B. eine

Menge aus 3 Punkten (Zykluslänge 3) z_1, z_2, z_3 , für die gilt:

$$f(z_1)=z_2, f(z_2)=z_3, f(z_3)=z_1$$

Wendet man also f 3 mal auf z_1 oder z_2 oder z_3 an, so kommt wieder z_1, z_2 oder z_3 heraus. Mit meinem Programm kann man sich auch auf Zyklensuche begeben. Die

Theorie verstehe ich allerdings

selbst nicht ganz, also lass ich es lieber...

Weiteres zum Thema Juliamengen findet sich in den folgenden Kapiteln zur Beschreibung der

einzelnen Parameter. Dort wird (hoffentlich) klar, wie mein Programm bei der Berechnung

vorgeht, was die einzelnen Parameter bewirken.

Wenn Sie dieses Kapitel nicht verstanden haben, so kann es durchaus vorkommen,

daß Sie bei der Änderung von Parametern plötzlich (oder sogar recht oft) vor
 einem
 einfarbigen Fraktal oder irgendeinem anderen langweiligen Bild sitzen und nicht
 wissen,
 wie Sie dieses Aussehen interpretieren sollen...

So, und zum Schluß noch etwas vor allem für Mathematiker Interessantes:
 Das Newton'sche Verfahren zur Nullstellenbestimmung

Newton (wer auch sonst ?) hat sich stark mit der näherungsweise Bestimmung der
 Wurzeln

(Nullstellen) von Polynomen $P(z)$ beschäftigt. Die Formel, die er herausfand und
 deren iterative

Anwendung eine Näherung für eine Nullstelle ergibt, lautet allgemein:

$$f(z) = z - P(z)/P'(z)$$

z wird zu Beginn initialisiert, anschließend die Funktion iterativ jeweils auf das
 Ergebnis

angewendet, bis sich der Funktionswert mehr oder weniger bei einem bestimmten Wert
 einpendelt.

Das Problem, mit dem Newton nicht ganz fertig wurde und mit dem man heute noch
 nicht einmal fertig

wird, ist, für welche Startwerte von z sich eine, und wenn ja, welche Nullstelle
 ergibt bzw. wie

bekommt man alle Nullstellen des Polynoms $P(z)$, bzw. bekommt man so überhaupt
 irgendwie

alle?

Hier kommt man nicht weiter. Um sich ein Bild von der Sache zu machen, benutzt man
 den

Computer. Es handelt sich bei näherer Betrachtung nämlich um simple Juliamengen
 mit einer

benutzerdefinierten Formel!

Sei nun $P(z) = z^3 + 2$, also $P'(z) = 3z^2$

Als $f(z)$ ist also im Programm $z - (z^3 + 2)/(3z^2)$ einzutragen und die Juliamenge
 hiervon zu berechnen.

Im Parameterwindow 2 muß noch auf endliche Attraktoren (eben die Nullstellen von P
 (z)) geprüft werden, dieser

Punkt ist also dort anzuwählen. Das sich ergebende Bild ist das sogenannte 'Basin
 of Attraction' der Funktion.

Öffnet man nun noch das Datenwindow hierzu, so findet man unter dem Punkt 'Ende:'
 den Endpunkt der Berechnung, also den

endlichen Attraktor, eine Näherung für eine Nullstelle von $P(z)$. Fährt man über
 das Fraktal mit dem Mauszeiger hinweg, so erkennt man sehr schön, welcher

Wert von z zu welcher Nullstelle führt. Bei einem Polynom dritten Grades sind
 stets nur 2 Fälle möglich: Entweder sind alle 3 Nullstellen

reell, oder aber eine reelle Nullstelle und 2 komplexe, die zueinander konjugiert
 komplex sind, existieren. Letzteres

ist hier der Fall, wie leicht nachgeprüft werden kann.

Aus der Komplexität des Fraktals erkennt man, wieso sich die heutige Mathematik
 bei eigentlich so einfachen

auszudrückenden Problemen so schwer tut.

Für die Transformation in die 3. Dimension eignet sich all dies nicht, da hier die
 Continuous Potential

Method in meiner Implementation versagt, was zu Treppenstufen führt.

1.22 Mandelbrotmengen: Theorie

2.3.2.2 Theorie: Mandelbrotmengen

weiterhin:

2.3.2.1 Theorie: Juliamengen

Grundsätzlich gibt es 2 verschiedene Haupttypen von Juliamengen:

Typ A) Die Juliamenge ist staubartig, d.h. sie besteht aus einer unendlich großen Anzahl unzu\ammen\hängen\der Punkte

Typ B) Die Juliamenge ist zusammenhängend, d.h. sie besteht aus einer Vielzahl von Linien, einer Fläche, einer Schleife, oder etwas ähnlichem.

Der Typ einer Juliamenge hängt von ihrem Parameter ab, bei der Standardformel $f(z)=z^2+c$ also von c .

Die Mandelbrotmenge stellt nun graphisch dar, für welche Werte von c die zugehörige

Juliamenge vom Typ A oder vom Typ B ist.

Julia, der Erfinder der Juliamengen, hat sich einen Trick ausgedacht, mit dem es möglich ist zu entscheiden, zu welchem Typ eine Juliamenge gehört, ohne sie selbst zu konstruieren. Dazu sind nämlich lediglich die Bahnen der kritischen Punkte zu verfolgen. Die kritischen Punkte einer Formel sind die Punkte, für die die Ableitung verschwindet, bei der Standardformel ist also nur der Punkt 0 kritischer Punkt.

Zur Konstruktion der Mandelbrotmenge müssen sämtliche kritischen Punkte beachtet werden.

Die vorliegende Programmversion ist dazu nicht in der Lage. Sie kann nur einen kritischen Punkt verfolgen. Folglich ist das Ergebnis bei Vorhandensein von mehreren

kritischen Punkten nicht korrekt. Man kann sich damit behelfen, daß man nacheinander

die Bahnen der kritischen Punkte verfolgen läßt, die Bilder jeweils abspeichert und dann in einem Malprogramm übereinanderlegt. Das Ergebnis ist die korrekte Mandelbrotmenge.

Eine Mandelbrotmenge wird nun also folgendermaßen konstruiert:

Abhängig vom Ausschnitt aus der komplexen Zahlenebene initialisiert man c mit dem dem Bildschirmpixel entsprechenden komplexen Zahlenwert, z initialisiert man mit dem kritischen Punkt, also mit 0.

Anschließend iteriert man das Ganze, d.h. man berechnet $f(z)$, dann $f(f(z))$, etc. Führt die Bahn nach unendlich, so ist die zum Punkt c gehörige Juliamenge staubartig.

Führt die Bahn zu einem endlichen Attraktor – einem Punkt oder einem Zyklus – so ist die zugehörige Juliamenge zusammenhängend.

Somit ist die Mandelbrotmenge ein Atlas, eine Art grobe Landkarte, für sämtliche Juliamengen zu einer Formel. Sehr oft steht man vor dem Problem, daß sich einfach kein vernünftiger Parameterwert c für die Juliamenge finden läßt. In diesem Fall muß man nur in der zu derselben Formel gehörigen Mandelbrotmenge nachsehen, wo denn dieses c liegt, schon hat man die Erklärung für das langweilige Aussehen der Juliamenge. Eine Lösung ist nun, den Parameterwert in der Nähe des Randes der Mandelbrotmenge zu wählen. Dort ist nämlich zu erwarten, daß die Juliamenge noch nicht so recht weiss, ob sie nun staubartig oder zusammenhängend sein soll (Natürlich

weiß sie es ganz genau, aber der Computer nicht...).

Im Inneren der Mandelbrotmenge ist die Juliamenge größtenteils eine unattraktive Fläche. Weit außerhalb der Mandelbrotmenge ist die Juliamenge staubartig, beinahe alle Punkte

gehören nicht zur Juliamenge, sondern verlaufen recht geordnet zu einem Attraktor hin. Das heißt, innerhalb kurzer Zeit hat das mein Programm erkannt und färbt den Punkt ein. Resultat: Langweiliges Bild...

Zur einfachen Änderung des Parameters einer Juliamenge siehe den Menüpunkt `←`
`Juliaparameter setzen`

1.23 2.3 Fraktale --- 2.3.2 Julia- und Mandelbrotmengen

2.3.2.3 Parameterwindow 1

DeepZoom

Das Wichtigste gleich vorweg: Seit V3.0 bietet ChaosPro die Fähigkeit, öfter als `←`
 bisher

hineinzuzoomen. Mit V2.0 konnte man ungefähr 50 mal hineinzuzoomen, dann war die `←`
 Genauigkeit

des mathematischen Koprozessors erreicht. Von da an waren die arithmetischen `←`
 Operationen

so ungenau, daß vernünftiges Arbeiten nicht mehr möglich war. Nun sind weitere `←`
 mathematische

Routinen mitgeliefert, die es erlauben, ungefähr 130 mal hineinzuzoomen. Wer öfter `←`
 hineinzuzoomen

will, soll sich bei mir melden. Es muß lediglich eine Iterationsroutine `←`
 geschrieben werden, die

Zahlen werden intern bereits mit rund 240 Dezimalstellen gehandhabt, was auch für `←`
 die Zukunft

genug sein sollte.

Die DeepZoom-Fähigkeiten sind allerdings gravierenden Einschränkungen ausgesetzt: Sie stehen nur zur Verfügung, wenn die Standard Julia- oder Mandelbrotmenge `←`
 benutzt wird.

Auch die restlichen Parameter dürfen nicht verändert werden, bzw. sie dürfen, `←`
 haben aber keine

Funktion mehr und sollten nicht verändert werden. Die folgenden Parameter dürfen verändert werden:

- Parameterwindow 1: alle Parameter. Die Ausschnittswerte und der 1. Parameter, bei `←`
 der Standard\juliamenge der Parameter 'c', bei der Standard\mandelbrot\menge `←`
 der Anfangswert,
 sind intern im Deep-Format abgelegt, d.h. hier dürfen Zahlen mit rund 240 `←`
 Nachkommastellen eingegeben werden.

- Parameterwindow 2: Außenfärbung muß 'Farbe', 'Iteration' oder 'CPM' sein, `←`
 Innenfärbung ist stets
 'Farbe'. Bei Abbruchbedingung muß genau nur 'unendlich' angewählt sein, Bailout `←`
 ist fest auf 3.0 eingestellt, obwohl im Gadget alles mögliche stehen kann. `←`
 Bailin hat
 natürlich keine Funktion.

- Parameterwindow 3: Kreisinverson, Biomorphie und Dekomposition nicht möglich. `←`
 Virtuelles Zeichnen
 könnte möglich sein (soll heißen, wird noch implementiert, sollte implementiert `←`
 sein, wenn sie

dies lesen...)

Die 3D-Parameter sind natürlich von dem ganzen DeepZoom-Teil unabhängig und funktionieren weiterhin.

Eine Anmerkung zur (ungewollten) falschen Benutzung von DeepZoom: Angenommen, Sie berechnen eine Zoom-In-Animation und geraten dabei in den Bereich, in dem ChaosPro dann automatisch die genauere (und wesentlich langsamere) Arithmetik benutzt. Wahrscheinlich werden Sie am Bailout-Wert nichts geändert haben, er steht also auf 16. Die Animation wird nun jedoch eine 'Unstetigkeit' aufweisen, in dem Augenblick, in dem ChaosPro plötzlich die genauere Arithmetik verwendet. Klar, denn hierbei wird der Bailout-Wert automatisch sprunghaft auf 3 gesetzt, was man natürlich sieht. Andere Programme wie Mand2000 beheben dieses Problem dadurch, daß der Bailout-Wert dort gar nicht wählbar ist und außerdem fest bei 2 liegt. Es mag schon seltsam erscheinen, daß ich nicht in der Lage bin, DeepZoom durchgehend ohne Einschränkung zu implementieren, so wie es Mand2000 schafft, aber man muß sich immer vor Augen halten, daß ChaosPro wesentlich mehr Parameter ändern läßt und ich hierbei ganz schön viel Code neu schreiben müßte, wozu ich keine Zeit & Lust habe.

Man muß sich nicht wundern, wenn ChaosPro bei tiefen Zooms unsagbar langsam wird. Dies ist normal. Das ist der Preis, den man zahlen muß, wenn man mit genaueren Zahlen rechnen will. Doch das geht anderen Programmen auch nicht anders.

Eine Anmerkung: Es können nur noch Zahlen zwischen -4 Millionen bis +4 Millionen dargestellt werden. Um genauer zu sein, ist das Zahlenformat folgendermaßen aufgebaut: 1 Bit Vorzeichen, 23 Bit Vorkommastellen, 97 Bytes Nachkommastellen, insgesamt 100 Bytes für eine einzige Zahl...

Parameter

- Je nach Formel sind 1 oder 2 komplexwertige Parameter wählbar. Diese sind hier einzutragen. Falls 2 Parameter wählbar sind, so ist der obere Parameter dem im Alphabet als erstem vorkommendem Parameter zugeordnet.

Der Parameter ist für Juliamengen entscheidend, da hierdurch die Lage der Fixpunkte

und deren Eigenwerte bestimmt werden. Sie sollten deshalb entsprechend gewählt werden. Zum einfacheren Wählen der Parameter bietet es sich an, sich diesen innerhalb

der Mandelbrotmenge anzeigen zu lassen. Genau dies kann man durch Anwahl des Menüpunktes

Juliaparameter setzen erreichen.

Die Mandelbrotmenge für z.B. z^2+c ist ein 'Atlas' für alle Juliamengen z^2+c ! Interessante Juliamengen finden sich am Rand der Mandelbrotmenge.

Der erste Parameter bei Mandelbrotmengen war bis V2.0 nicht wählbar. Normalerweise wird dieser Wert auf den kritischen Wert der Funktion eingestellt, bei der Standardmandelbrotmenge eben auf 0. Allerdings erzeugen auch andere Werte als der aus mathematischer Sicht wahre kritische Wert sinnvolle, sprich hübsche Bilder. Mandelbrotmengen, die durch einen nicht korrekten kritischen Wert entstehen, werden auch Pseudo-Mandelbrotmengen genannt. Wird hier der Wert 10 eingetragen, kümmert sich ChaosPro automatisch darum, daß der mathematisch korrekte Wert eingetragen wird. Eine etwas seltsame Methode, aber wie soll man das sonst machen?

Iterationen

- Die Qualität einer Julia- /Mandelbrotmenge hängt stark vom Iterationswert ab. Je höher, desto besser, aber auch desto langsamer die Berechnung. Mit dem Slider kann der Iterationswert einfach ohne Tastaturbedienung verändert werden. Klickt man ihn an, so addiert sich sein Wert zum vorhandenen Iterationswert. Läßt man ihn wieder los, so schnappt er in die Ausgangslage, also in die 0-Position zurück. Alternativ läßt sich eine größere Änderung des Iterationswertes auch direkt in das Gadget eintragen. Es ist allerdings dannach die Return-Taste oder die Tab-Taste zu drücken. Wie in den theoretischen Kapiteln dargestellt, müssen sämtliche Bahnen von Punkten die geordnet verlaufen, also von einer Menge von Punkten angezogen werden, (Julia- bzw. sämtliche Bahnen, die von der Unendlichkeit angezogen werden (Mandelbrotmenge herausgefiltert werden. Diese 'geordnet' verlaufenden Bahnen werden der Optik wegen eingefärbt und zwar entsprechend der Anzahl an Iterationen, die es gedauert hat, bis meinem Program klar war, daß diese Punktbahn angezogen wird. All diese Punkte gehören zum Aussengebiet der Julia- /Mandelbrotmenge. Verläuft nach der eingestellten Anzahl an Iterationen eine Bahn eines Punktes immer noch nicht erkennbar zu einer anziehenden Menge, so wird der Punkt als zur Julia- /Mandelbrotmenge gehörig eingestuft und gehört somit zum Innengebiet.

Passes

- Hiermit läßt sich die Anzahl der Zeichendurchgänge festlegen. Wegen der Beschaffenheit der Julia- und Mandelbrotmengen kann man aus gleichen Iterationswerten am Rande eines z.B. 4x4 Rechtecks darauf schließen, daß sich innerhalb des Rechtecks auch nur Punkte mit derselben Iterationstiefe befinden. Nun, dieser Schluß ist nicht ganz richtig, er ist völlig falsch für eine staubartige Juliamenge, aber er hilft, die Zeichengeschwindigkeit erheblich zu steigern. Und bei staubartigen Juliamengen sieht man auch bei 1-Pass-Zeichnung nichts vom Staub, da es äußerst unwahrscheinlich ist, daß von der begrenzten Anzahl der Punkte, die hier gezeichnet werden, auch nur ein einziger exakt in die Juliamenge hineinfällt.

Sie fallen so gut wie immer ein ganz klein bisschen daneben, und dieser Punkt gehört dann ja schon nicht mehr zur Juliamenge. Hier wären andere Berechnungsmethoden wie z.B. die Abstandsmethode mittels Continuous Potential nötig, die mein Programm jedoch nicht bietet.

Ausschnitt

- Die Juliamenge zeigt, was mit den Punkten der komplexen Zahlenebene passiert, wenn man in iterativer Weise die Formel auf jeden einzelnen Punkt der Ebene anwendet.

Hier nun kann man den Ausschnitt aus der komplexen Zahlenebene wählen.

- Bei der Mandelbrotmenge wird hierdurch der Ausschnitt aus der komplexen Zahlenebene definiert, der dann für den Parameterwert c der Formel benutzt wird.

Eliminieren

Ist dieses Gadget angewählt, so markiert das Programm nach jedem Zeichendurchgang alle Bereiche, deren Eckpunkte jeweils dieselbe Iterationstiefe haben, d.h. alle inneren Punkte des Bereiches müssen nicht mehr berechnet werden. Dies bringt einen Geschwindigkeitsgewinn von etwa 50 Prozent. Der Nachteil ist natürlich, daß auf diese Art auch Bereiche gar nicht berechnet werden, die eigentlich doch berechnet werden müßten. Oftmals sind solche Bereiche selten, doch sie kommen auch vor.

Winkel

Um diesen Winkel wird das 2D-Fraktal noch gedreht. Insbesondere in Verbindung mit dem Animationssystem interessant. Es sind hier 2 Gadgets vorhanden, mit denen der Drehwinkel leicht eingestellt werden kann. Der Wertebereich reicht von -30000 bis 30000.

Seit V3.0 bietet ChaosPro die Möglichkeit, den Drehpunkt zu verändern. In V2.0 war er fest in der Mitte des Windows gelegen. Die Änderung erzeugt nette Effekte vor allem in Verbindung mit Animationen.

Theoretisches hierzu:

2.3.2.1 Theorie: Juliamengen

2.3.2.2 Theorie: Mandelbrotmengen

1.24 2.3 Fraktale --- 2.3.2 Julia- und Mandelbrotmengen

2.3.2.4 Parameterwindow 2

Aussenfärbung

- Hier gibt es einige Möglichkeiten:

1. Farbe

Hierdurch wird das komplette Aussengebiet mit der direkt darunter eingestellten Farbe gezeichnet. Da das Aussengebiet in der Regel für das schöne Erscheinungsbild des Fraktals verantwortlich ist, ist der Sinn dieser Wahl auf den ersten Blick fraglich. Doch damit läßt sich das staubartige Aussehen z.B. der Juliamenge recht schön darstellen, falls es überhaupt möglich ist, denn nun verwirren die vielen Farben den Blick nicht mehr.

2. Iteration

Damit wird jedem Punkt, dessen Bahn geordnet verläuft, also von einem Attraktor angezogen wird, die Anzahl an Iterationen zugeordnet, die es gedauert hat, bis mein Programm die Ordnung der Bahn bemerkt hat. Diesem Zahlenwert, der von 0 bis zur eingestellten Maximalzahl an Iterationen-1 gehen kann, wird dann eine Farbe zugeordnet.

3. CPM - ein Akronym für Continuous Potential Method

Wer den 2. Punkt betrachtet, dem wird auffallen, daß jedem Punkt ganz offensichtlich nur eine ganze Zahl zugeordnet werden kann. Das ist insbesondere bei Berechnung von 3D-Ansichten der Fraktale nachteilhaft, da es hier zu Treppen in der Darstellung kommt. Das Bild sieht aus wie chinesische Reisterassen. Die Höhenwerte der Punkte springen von einem Wert ganz plötzlich zum nächsten. Das ändert sich bei Anwahl dieses Modus. Mittels einer recht einfachen Funktion können die Außengebiete von Julia- und Mandelbrotmengen in das Innengebiet eines Einheitskreises transformiert werden. Das besondere dabei ist, daß die Sprünge von einer Iterationsstufe zur nächsten (also die Grenzen der farbigen Bänder), diese unheimlich komplizierten Kurven, in perfekte konzentrische Kreise mit Mittelpunkt 0 transformiert werden! Diese Kreise haben nun Radien und diesen Radius kann man anstelle der Iteration verwenden. Der Vorteil ist nun, daß ein mitten in einem einfarbigen Band liegender Punkt in den Kreis transformiert wird, und zwar zwischen 2 solcher Kreise. Bildet man nun einen Kreis, auf dem dieser Punkt liegt und dreht die Funktion um, so erhält man ein perfektes Zwischenband! Diese Methode wird also dazu benutzt, jedem Punkt eine beliebige reelle Zahl zuzuordnen, so daß Treppenstufen in der 3D-Darstellung der Vergangenheit angehören. Mit 'Mult.' kann hierbei eingestellt werden, mit welcher Zahl diese reelle Zahl multipliziert wird. Mein Programm merkt sich keine reellen Zahlen, da dies einfach zu viel Speicher kosten würde. Also merkt es sich hier ganze Zahlen, aber eben mit 'Mult' multipliziert. So heißt das, daß ein 'Mult'-Wert von 100 100 Zwischenstufen zwischen 2 Iterationen zuläßt, was ausreicht, um den Treppeneffekt zu umgehen. Beim Abspeichern als 24 Bit-Fraktal werden diese 100 Zwischenstufen in Zwischenfarben umgerechnet, was sanfte Farbübergänge zuläßt.

4. DEM - ein Akronym für Distance Estimator Method

Dieser Algorithmus ist im Grunde genommen eine Weiterentwicklung von CPM. Mit Hilfe dieser Einfärbung wird jedem Punkt eine Schätzung für seinen Abstand vom Rand der Mandelbrotmenge zugeordnet. Das ist dann natürlich wieder eine reelle Zahl und keine Integerzahl mehr, was zur Folge hat, daß dieser Algorithmus sich sowohl dazu eignet, das Fraktal in 24 Bit abzuspeichern, als auch eine 3D-Ansicht zu berechnen.

5. DEM Border

Diese Methode macht dasselbe wie DEM, bloß spielt hier der 'Bailin'-Wert noch eine besondere Rolle.

Manchmal ist es wünschenswert, nur den Rand der Julia- bzw. Mandelbrotmengen zu zeichnen, d.h. es macht hier Sinn, nur die Punkte zu zeichnen, deren Abstand vom Rand des Fraktals kleiner als ein bestimmter Wert 'l' ist. Das macht dieser Algorithmus. 'l' bestimmt man mittels 'Bailin'...

Bemerkung: Es war bisher ein großes Problem, gewisse Mandelbrot/Juliamengen in 3D darzustellen. Stellte man CPM ein, so traten sehr starke Sprünge auf. Dies passiert mit DEM nicht mehr in dieser Weise. 2 Punkte, die dicht beieinanderliegen, haben bei Färbung=CPM einen beliebig stark unterschiedlichen (Potential-)Wert, bei Färbung=DEM jedoch nicht. Falls es also einfach nicht gelungen ist, auftretende Zacken und Spitzen zu eliminieren, so sollte man es nun mit DEM versuchen...

Innenfärbung

- Nun gibt es sogar 6 Möglichkeiten:

Farbe - Infimum - Infimumsindex - Supremum - Supremumsindex - Betrag von z

Das Innengebiet ist für gewöhnlich recht einfarbig. Doch das muß nicht sein. Es besteht die Möglichkeit, auch das Innengebiet farbig zu gestalten. Sei $(z, z_1, z_2, z_3, z_4, \dots, z_n)$

die Bahn des Punktes z, n ist also die Maximalzahl an Iterationen.

Infimum

Hier wird das Infimum (bei Computern ist's das Minimum...) eines Punktes vom Startwert

berechnet, also das Minimum von $|z_1-z|, |z_2-z|, |z_3-z|, \dots, |z_n-z|$. Der erhaltende Wert wird

mit 'Multiplikator' multipliziert und als ganze Zahl abgespeichert.

Infimumsindex

Hier wird der Index des Infimums berechnet, also die Anzahl an Iterationen, bei der das Minimum aufgetreten ist. Ist das Minimum von $|z_1-z|, |z_2-z|, |z_3-z|, \dots, |z_n-z|$

gleich $|z_3-z|$, so ist der Index 3

Supremum

Hier nun wird das Supremum (bei Computern das Maximum...) eines Punktes vom Startwert

berechnet, also das Maximum von $|z_1-z|, |z_2-z|, |z_3-z|, \dots, |z_n-z|$. Der erhaltende Wert

wird mit 'Multiplikator' multipliziert und als ganze Zahl abgespeichert.

Supremumsindex

Hier wird der Index des Supremums berechnet, also die Anzahl an Iterationen, bei der das Maximum aufgetreten ist.

Betrag von z

Nach der Maximalzahl an Iterationen wird die Länge von z berechnet, mit '↔ Multiplikator' multipliziert und abgespeichert.

Angeregt worden bin ich hierbei von 'The Beauty of Fractals', Seite 62

Abbruchbedingungen

Hier kann festgelegt werden, welche Attraktoren von welcher Art als '↔ Abbruchbedingungen' zugelassen werden.

1. unendlich

Falls angewählt, wird jeder Punkt untersucht, ob er vom unendlich fernen Punkt angezogen wird.

2. endlich

Falls angewählt, wird jeder Punkt daraufhin untersucht, ob er von irgendeinem '↔ endlichen Fixpunkt angezogen wird

3. Zyklussuche

Falls angewählt, wird jeder Punkt daraufhin untersucht, ob er von irgendeinem '↔ Zyklus angezogen wird. Dazu muß man noch in 'Start' eine Iterationszahl eintragen, ab der mit der Suche begonnen werden soll. Sie sollte in etwa die Hälfte der Iterationen '↔ sein.

Dies ist nötig, da ein Punkt auf seiner Bahn normalerweise erst einmal ziemlich kopflos herumwandert, bis er sich einmal entschließt, zu einem Attraktor '↔ hinzuwandern.

Man sollte solchen Punkten also Gelegenheit zu einem Entschluß geben.

4. Formeldefiniert

Falls angewählt, wird jeder Punkt daraufhin untersucht, ob die in der Formel '↔ angegebene

Abbruchbedingung erfüllt ist und ggf. abgebrochen. Das funktioniert natürlich dann '↔ nur mit selbstgebastelten Formeln und nicht

mit den eingebauten Formeln, da dort keine zusätzliche Abbruchbedingung angegeben '↔ ist und die eingebauten Formeln naturgemäß nicht verändert werden können.

Bailout

Falls jeder Punkt daraufhin geprüft wird, ob er von dem unendlich fernen Punkt angezogen wird, so ist natürlich die Frage, wie man das am geschicktesten prüft.

Folgende Methode wird eigentlich überall angewandt (Ausnahme: {"Biomorphie" LINK '↔ ExpertJM_Bio}):

Man definiert einen Kreis um den Ursprung mit Radius 'Bailout'. Fällt ein Punkt auf '↔ seiner Bahn außerhalb des Kreises, so schätzt man, daß er ins unendliche flieht.

Bailin

Falls jeder Punkt daraufhin untersucht wird, ob er von einem endlichen Punkt oder

Zyklus angezogen wird, so definiert man einen Kreis um diesen endlichen Punkt mit dem Radius 'Bailin' und sagt, der Punkt wird von ihm angezogen, wenn er auf seiner Bahn einmal in den Kreis fällt.

Theoretisches hierzu:

2.3.2.1 Theorie: Juliamengen

2.3.2.2 Theorie: Mandelbrotmengen

1.25 2.3 Fraktale --- 2.3.2 Julia- und Mandelbrotmengen

2.3.2.5 Parameterwindow 3

Kreisinverson

- Dies ist eine geometrische Transformation. Hierbei wird alles, was außerhalb des mit 'Mittelpunkt' und 'Radius' definierten Kreises liegt, ins Innengebiet des Kreises transformiert und alles, was innerhalb liegt, nach außen.

Biomorphie

- Normalerweise werden mittels Bailout und Bailin Kreise definiert. Wenn Punkte auf ihren Bahnen außerhalb bzw. innerhalb dieser Kreise zu liegen kommen, so wird abgebrochen. Nun könnte man auf die Idee kommen, keine Kreise, sondern Rechtecke oder sternförmige Gebiete zu definieren und zu testen, ob und wann ein Punkt \leftrightarrow außerhalb bzw. innerhalb dieser Gebiete zu liegen kommt. Das Ergebnis sind Fraktale, die ein bisschen an Mikroorganismen erinnern, daher der Name. Die genauen \leftrightarrow

Abbruchbedingungen sind

mathematisch nun wie folgt definiert:

(x ist der Realteil von z, y der Imaginärteil)

$\text{abs}(x) + d \cdot \text{abs}(y) > \text{Bailout}$
und/oder
 $d \cdot \text{abs}(x) + \text{abs}(y) > \text{Bailout}$

bzw.

$\text{abs}(x) + d \cdot \text{abs}(y) < \text{Bailin}$
und/oder
 $d \cdot \text{abs}(x) + \text{abs}(y) < \text{Bailin}$

Ob diese beiden Ungleichungen mittels 'und' oder 'oder' verknüpft wird, kann man einstellen, die Variable d bezeichne ich im Programm mit Biomorphievariable.

Setzt man $d=0$, so ergibt sich:

$\text{abs}(x) < \text{Bailin}$
und/oder
 $\text{abs}(y) < \text{Bailin}$

Im Falle von 'und' also ein Quadrat, im Falle von 'oder' ein Kreuz.

Für $d=1/10$ ergibt sich ein sternförmiges Gebiet, usw.

Dekomposition

- Hier wird das Außengebiet in Winkelfelder eingeteilt. Die Anzahl der Winkelfelder entspricht dem Wert in 'Kodierung'. Zu jedem Bahnendwert wird nun berechnet, in welchem Winkelfeld von 0 bis 'Kodierung'-1 er zu liegen gekommen ist und so eingefärbt.

Virtuelles Berechnen

Seit V2.0 besitzt ChaosPro die Fähigkeit, Fraktale in einem virtuellen Modus zu berechnen. Das heißt, es ist möglich, riesige Fraktale zu berechnen, ohne viel Speicher zu benötigen. Bisher war es mit meinem Amiga z.B. nicht möglich, ein 1024x768 Punkte großes 3D-Fraktal zu berechnen, da ich nicht genügend Speicher für den nötigen Puffer hatte. Um nun ein Fraktal im virtuellen Modus zu berechnen, muß man in den 4 Gadgets die entsprechenden Daten angeben, also die Breite, die Höhe, die Tiefe in Planes und ob nach dem Berechnen des Fraktals eine 3D-Version davon berechnet werden soll. Sind die Daten eingestellt, so muß man nur noch den Menüpunkt 'Fraktal/Starte virtuell' anwählen, schon beginnt ChaosPro mit der Berechnung. Als Puffer wird nun nicht RAM-Speicher, sondern Festplattenplatz benutzt. Es wird dabei eine Datei erzeugt. Normalerweise wird diese Datei im Verzeichnis 'ChaosPro:' erzeugt, was man jedoch mit dem Tooltype 'VIRTUAL =<Dir>' ändern kann. Nachdem das Fraktal berechnet wurde, erscheint ein Filerequester, der nach dem Namen der zu speichernden IFF-ILBM-Datei fragt. Dieses virtuelle Berechnen ist insbesondere nützlich in Verbindung mit dem Druckertool Studio, das den Ausdruck von IFF-ILBM-Bildern in beliebiger Größe erlaubt, was es ermöglicht, große Fraktale auszudrucken, die wirklich exzellent aussehen. Normalerweise kommt beim Ausdruck eines Fraktals in 'normaler' Größe ein ziemliches Klötzchenbild heraus, was nicht unbedingt professionell aussieht.

Theoretisches hierzu:

2.3.2.1 Theorie: Juliamengen

2.3.2.2 Theorie: Mandelbrotmengen

1.26 2.3 Fraktale --- 2.3.2 Julia- und Mandelbrotmengen

2.3.2.6 Das Datenwindow

Mit dem Infowindow können die wichtigsten Daten zu den Julia- /Mandelbrot\mengen ausgegeben werden. Das Window wertet die Mauszeigerposition aus und errechnet nochmal

alle Werte an der betreffenden Stelle.

- Im Feld 'Grund' wird angezeigt, warum die Berechnung abgebrochen wurde.
- Im Feld Iterationen, wann das der Fall war.
- Im Feld 'Entfernung' wird angezeigt, wie weit der Punkt in etwa vom Rand des Fraktals entfernt ist.
- Punkt und Start enthalten bei Juliamengen identische Werte. Im Punkt-Feld ist

der dem Bildschirmpixel (Felder Pixel x und Pixel y) entsprechende komplexe Zahlenwert angegeben. In Start ist die Initialisierung von z angegeben. Bei Juliamengen eben dasselbe wie bei 'Punkt'. Bei Mandelbrotmengen ist das der kritische Wert, der durchaus aus der Auswertung einer Formel erhalten worden sein kann, z.B. bei der 2. Formel durch Auswertung der Formel $m/(2m-2)$

- Infimum und Supremum

Hier wird das Minimum bzw. das Maximum der Entfernung des Orbits des Punktes vom Ursprung und der Index, d.h. also die Iteration, in der dieses Maximum bzw. Minimum aufgetreten ist, angezeigt. Siehe auch die Einfärbungsalgorithmen Infimum/ Supremum/ Inf.index/ Sup.index.

Theoretisches hierzu:

2.3.2.1 Theorie: Juliamengen

2.3.2.2 Theorie: Mandelbrotmengen

1.27 2.3 Fraktale --- 2.3.2 Julia- und Mandelbrotmengen

2.3.2.7 Das Formelwindow

ChaosPro verfügt über einen relativ leistungsfähigen Formeleditor. Mit dessen Hilfe ist es möglich, eigene Formeln zu kreieren und die Julia- bzw. Mandelbrotmengen davon zu untersuchen.

In diesem Window kann die zu benutzende Formel ausgewählt werden. Man beachte, daß die verschiedenen Einfärbungsalgorithmen wie CPM, DEM, etc. nur für die erste Formel möglich sind und somit nicht existieren, wenn eine andere Formel angewählt ist.

1.28 2.3 Fraktale --- 2.3.2 Julia- und Mandelbrotmengen

2.3.2.8 Das Farbverteilungswindow

Dieses Window erlaubt es, die Verteilung der Farben über die verschiedenen Iterationswerte genauer zu kontrollieren. ChaosPro benutzt in jedem Fall fest eingebaute Funktionen, die dann auf die Iterationswerte angewendet werden. Das Ergebnis der Funktionsauswertung bestimmt dann die Farbe.

Zur Bestimmung der anzuwendenden Funktion stehen die Gadgets Typ, Faktor, Max und Min zur Verfügung.

'Min' bestimmt den minimalen Iterationswert. Hat ein Punkt einen Wert kleiner als 'Min', so wird er stets auf Farbe 4 abgebildet.

'Max' und 'Faktor' beeinflussen sich gegenseitig. Grundsätzlich ist es so, daß der gesamte Bereich zwischen 'Min' und 'Max' auf die gesamte Palette verteilt wird, d. h.

kommen Iterationswerte aufsteigend von 'Min' bis 'Max' vor, so wird die Farbpalette

genau 1 mal durchlaufen. 'Faktor' wird bei einer Änderung von 'Max' automatisch aktualisiert und umgekehrt. 'Faktor' ist der Wert, den das Programm zur Funktionsauswertung

direkt benutzt, 'Max' dagegen ist mehr intuitiv.

Sei ItWert der Wert, den das Programm an der Stelle berechnet hat, AnzFa ist gleich der Anzahl

der wirklich zur Verfügung stehenden Farben, also auf einem 8 Bit Screen eben 256-4=252

(die ersten 4 Farben werden ja nicht zum Zeichnen benutzt)

Folgende Funktionen stehen nun zur Verfügung:

Linear

Die Funktion lautet hier:

Farbe = (ItWert-Min) * Faktor/100 Modulo AnzFa + 4

Diese Funktion ist die primitivste und schlechteste und wird normalerweise von jedem

Fraktalprogramm benutzt, weil sie so schön einfach ist.

Sin

Farbe=abs(sin((ItWert-Min) * Faktor/10000)) * (AnzFa-1) + 4

Diese Funktion erzeugt interessante Effekte beim Colorcycling. Durch das ständige Durchlaufen der Farbpalette in beide Richtungen ist diese Funktion recht gut für beinahe jedes Fraktal.

0.75

Farbe=((ItWert-Min) * Faktor / 100) ^ 0.75 Modulo AnzFa + 4

Log

Farbe=log((ItWert-Min) * Faktor/100 +1) * AnzFa/6 Modulo AnzFa + 4

Gut, wenn man etwas weiter hineingezoomt ist und viele stark unterschiedliche Iterationswerte

auftreten.

ArcTan{ub}

Farbe=abs(arctan((ItWert-Min) * Faktor/10000)) * 2/PI * (AnzFa-1) + 4

Recht gut, prinzipbedingt kommt es hier nie zu einem mehrfachen Durchlauf der Farbpalette, was

sie vor allem für tiefe Zooms geeignet macht.

Sqrt

Farbe=sqrt((ItWert-Min) * Faktor/100) * Faktor/20 Modulo AnzFa + 4

Um einen guten Eindruck von der Art der gewählten Farbverteilung und den Parametern zu bekommen, wird darunter die aktuelle Funktion graphisch dargestellt.

Klickt man auf das Gadget 'Vorschlag', so durchsucht das Programm den Puffer (falls vorhanden) und setzt 'Min' und 'Max' auf den minimalen und maximalen Wert, der im Puffer vorhanden ist, was natürlich dazu führt, daß die Palette dann nur noch einmal durchlaufen wird, was meistens wünschenswert ist.

1.29 2.3 Fraktale --- 2.3.3 Bifurkationsdiagramme

2.3.3 Bifurkationsdiagramme

2.3.3.1 Theorie

Betrachtet wird das Verhulst-Modell $x \rightarrow a \cdot x \cdot (1-x)$

Dieses Modell läßt sich praktisch wie folgt veranschaulichen:

x ist die Population einer Rasse, z.B. von Hasen. Sie ist so skaliert, daß x stets zwischen 0 und 1 liegt, 0 heißt, es ist kein Hase vorhanden, 1 heißt, es ist alles voll von Hasen, so voll, daß kein einziger mehr Platz hat. a ist dann die Vermehrungsrate,

$a=1$ würde dann heißen, daß sich die Hasen nicht vermehren. Bleibt noch der Faktor $(1-x)$ zu erklären. Er ist ein Maß für den Freiraum, der den Hasen bleibt und kann als Menge des vorhandenen Futters erklärt werden, die auch wieder zwischen 0 und 1 liegt.

Von einem zum nächsten Jahr berechnet sich die Population der Hasen, indem einfach aus dem

alten x_0 ein neues $x_1 = a \cdot x_0 \cdot (1-x_0)$ errechnet wird. Betrachtet man das Modell, so stellt

man fest:

Sei nun $a=2$ (durchaus denkbar).

Ist in einem Jahr die Population gering, so ist viel Futter vorhanden, was die Population

ansteigen läßt. Ist die Population groß, so ist wenig Futter vorhanden, was die Hasen

sterben läßt.

Die Frage ist nun: Bei welchem Gleichgewicht stellt sich die Population ein?

Ist $x_0=0.1$, $a=2$, so ergibt sich:

$$x_1 = 2 \cdot 0.1 \cdot 0.9 = 0.18$$

$$x_2 = 2 \cdot 0.18 \cdot 0.82 = 0.2952$$

$$x_3 = 2 \cdot 0.2952 \cdot 0.7048 = 0.416$$

$$x_4 = 2 \cdot 0.416 \cdot 0.584 = 0.486$$

$$x_5 = 2 \cdot 0.486 \cdot 0.514 = 0.50$$

...

Der Wert erreicht rasch 0.5 und bleibt dort, das heißt die Hasenpopulation würde im Beispiel zunehmen bis zum Wert 0.5 und dort konstant bleiben.

Nun kann man aber noch die Vermehrungsrate ändern. Die Frage ist nun, was passiert mit der Hasenpopulation bei verschiedenen Werten von a ? Welches Gleichgewicht

wird erreicht? Wird überhaupt ein Gleichgewicht erreicht?

Dieses Modell ist sehr einfach gehalten, trotzdem zeigt es bereits eine überraschende Vielfalt an Zuständen.

1. Fall: $0 < a \leq 1$

In diesem Fall geht x gegen 0, was klar ist, denn a war ja die Vermehrungsrate, d.h. die Hasen sterben aus.

2. Fall: $1 < a \leq 2$

Hier geht die Population rasch in einen Gleichgewichtszustand über, die Folge der x wächst entweder monoton oder fällt monoton, je nach Anfangswert der Population

3. Fall: $2 < a \leq 3$

Auch hier gibt es wieder einen Gleichgewichtszustand, die Folge der x jedoch geht oszillierend gegen den Grenzwert, nicht monoton.

Nun sei $a > 3$ z.B. 3.1

$x_1 = 0.3$
 $x_2 = 0.651$
 $x_3 = 0.704$
 $x_4 = 0.646$
 $x_5 = \dots$

Rechnet man weiter, so erkennt man, daß x schließlich zwischen 2 Werten hin und her pendelt, nämlich zwischen den Werten 0.557 und 0.764

Also haben wir hier kein eindeutiges Gleichgewicht, die Hasenpopulation ändert sich von Jahr zu Jahr sprunghaft.

Vergrößert man a weiter bis zum Wert 3.449489, so pendelt die Population stets zwischen 2 Werten hin und her. Doch ab diesem Wert passiert es wieder: Eine Periodenverdoppelung,

d.h. ab hier pendelt der Wert von x schließlich zwischen 4 (!) Werten hin und her. Bei $a = 3.5441$ geht dann dieser 4-er Zyklus in einen 8er-Zyklus über. All diese Werte von a , bei denen die Zykluslänge verdoppelt wird, heißen Bifurkationspunkte.

Dieser 8er-Zyklus geht in einen 16er-Zyklus, dieser in einen 32er-Zyklus etc. über, bis

zu einem bestimmten Wert: $a = 3.569946$

Ab diesem Wert bis zum Wert $a = 4$ passiert wieder etwas sehr seltsames:

Erst wird das Ganze einmal chaotisch, d.h. x pendelt zufällig zwischen irgendwelchen

Werten hin und her, an diesen Stellen ist der Attraktor kein Zyklus irgendeiner Länge, sondern ein eindimensionales Fraktal.

In diesem Bereich bis 4 jedoch gibt es noch sogenannte "Fenster", z.B. bei $a = 3.83$, wo

wieder ein Zyklus der Länge 3 vorherrscht, der in einen Zyklus der Länge 6, dann 12, 24, 48, ... übergeht. Solche Fenster finden sich noch weit mehr.

Wenn man sich das ganze Modell betrachtet und sich vergegenwärtigt, daß wir es uns am Beispiel der Hasenpopulation deutlich gemacht haben, so überrascht es doch,

daß so seltsame Dinge auftreten können. Ein unwissender Leser hätte doch stets vermutet, es müsse irgendein Gleichgewicht vorherrschen.

1.30 2.3 Fraktale --- 2.3.3 Bifurkationsdiagramme

2.3.3.2 Parameterwindow 1

Formel

- Im theoretischen Kapitel wurde die Verhulst-Formel $a*x*(1-x)$ besprochen. Analog kann man jedoch die Gedankengänge auch auf andere Formeln anwenden. Im Programm wurden 5 der wichtigeren Formeln eingebaut. Von diesem kann man sich das

Bifurkationsdiagramm zeichnen lassen.

Iteration

- Um das Bifurkationsdiagramm korrekt zeichnen zu können, muß der Anfangswert erst einmal genügend oft in die Formel eingegeben (iteriert) werden, um ihm Gelegenheit zu geben, zu einem evtl. vorhandenen Attraktor hinzuwandern. Erst dann kann mit dem

Zeichnen begonnen werden.

In diesem Programm wird bei den Bifurkationsdiagrammen grundsätzlich so oft iteriert,

bis die Hälfte des Wertes, der hier angegeben ist, erreicht ist. Dann ist der Start-

Punkt hoffentlich bei seinem anziehenden Zyklus angelangt. Dann wird bis zum Erreichen

des hier angegebenen Wertes der Punkt weiter iteriert und nun dabei gezeichnet.

Will man das Bifurkationsdiagramm genauer haben (hat man z.B. eine Verästelung stärker vergrößert), so erkennt man, daß es sich eigentlich um keine plötzliche Verästelung handelt, sondern um ein "bandiges" auseinanderdriften der Werte. Dies ist

auf die Ungenauigkeit des Programms zurückzuführen! Es handelt sich wirklich um ganz plötzliche

Verästelungen. Erhöhen Sie den Iterationswert und das Programm erhöht die Genauigkeit, was

die Sache wieder wie eine Verästelung aussehen läßt.

Variable x/Variable y/beide

- Diese Option ist nur bei Formel Nr. 3 anwählbar. Dort handelt es sich nämlich um die Formel $x=a*x*(1-x-y)$ und $y=a*x*y$, es kommen also 2 Variablen vor, also z.B. Fuchs, Hase (und der Vermehrungsfaktor). Welche Variable gezeichnet werden soll, bestimmen Sie hier.

a: Minimum - Maximum

- Horizontal im Fraktalwindow wird der Parameter a aufgetragen. Hier bestimmen Sie ab welchem und bis zu welchem Wert.

x/y: Minimum - Maximum

- Vertikal im Fraktalwindow wird die Variable x - bzw. bei Formel 3 auch y - aufgetragen.

Wieder bestimmen Sie hier den Ausschnitt.

Theorie hierzu:

Kapitel 2.3.3.1

1.31 2.3 Fraktale --- 2.3.3 Bifurkationsdiagramme

2.3.3.3 Datenwindow

- In den Felder a bzw. x/y werden die der Bildschirmposition entsprechenden Werte für a bzw. x/y angezeigt.
- In den Felder x bzw. y und in den Felder Ende x bzw. Ende y werden die Startwerte, also die Initialisierungswerte, bzw. die Endwerte nach der darunter eingestellten Anzahl an Iterationen angezeigt.
- Im Feld Zyklus wird die Länge eines evtl. gefundenen Zyklus angezeigt.
- Durch das Feld 'Zeige Iteration' in Verbindung mit dem Slider wird eingestellt, nach welcher Iterationstiefe der Wert von x bzw. y in den beiden Feldern Ende x bzw. Ende y angezeigt wird. Hierdurch kann man untersuchen, welche Werte nach und nach angenommen werden, ohne den Taschenrechner zu bemühen.

Hinweis:

Durch Drücken der Taste 'I' bzw. Shift+'I' kann dieser Wert aus dem Fraktalwindow heraus geändert werden. Es ist also nicht zwingend nötig, das Datenwindow zu aktivieren.

Theorie hierzu:

Kapitel 2.3.3.1

1.32 2.3 Fraktale --- 2.3.4 Dynamische Systeme

2.3.4 Dynamische Systeme

2.3.4.1 Theorie

1961 machte der Meteorologe Edward Lorenz Berechnungen mit einem System von \leftrightarrow mehreren

Differentialgleichungen, d.h. einem System, von dem nicht die konkreten Punkte bekannt sind, aus denen dann weitere Punkte irgendwie berechnet werden können, sondern von dem zu jedem beliebigen Punkt die Steigung der Tangente bekannt ist, also die Ableitung. Aus dieser kann dann eine Näherung für den nächsten Punkt berechnet werden.

Nun gut, er machte so seine Experimente und stellte fest, daß sein Ergebnis sehr \leftrightarrow stark

von der Rechengenauigkeit abhängte. Ein sehr kleiner Fehler am Anfang hatte nach kurzer Zeit ein völlig anderes Ergebnis zur Folge.

Der Titel einer seiner Publikationen war deshalb: "Kann das Flattern eines \leftrightarrow Schmetterlings

in Brasilien einen Orkan in Texas verursachen?"

Dieser von Lorenz entdeckte Effekt wurde sinnvollerweise dann " \leftrightarrow Schmetterlingseffekt "

genannt.

Edward Lorenz vereinfachte dann sein Modell stark und experimentierte damit herum. Dieses Modell bestand dann nur noch aus 3 Differentialgleichungen:

$$\begin{aligned} dx/dt &= -ax + ay \\ dy/dt &= cx - y - xz \\ dz/dt &= -bz + xy \end{aligned}$$

Die 3 Gleichungen lesen sich so:

Die Ableitung in x-Richtung nach der Zeit t ist $-ax + ay$

Die Ableitung in y-Richtung nach der Zeit t ist $cx - y - xz$

Die Ableitung in z-Richtung nach der Zeit t ist $-bz + xy$

Er gab a den Wert 10, b den Wert $8/3$ und experimentierte mit verschiedenen Werten von c. Die Ergebniskurve ist dreidimensional und nach dem Existenz- und

Eindeutigkeitssatz

(das ist nur was für Studenten der Mathematik oder zumindest von solchen, die Mathe

als Nebenfach haben...) aus mathematischer Sicht eindeutig, wenn ein Anfangspunkt gegeben ist (ein sog. Anfangswertproblem, kurz AWP...). Aber leider eben nur aus mathematischer Sicht.

Lorenz nahm für c den Wert 28 und berechnete die Kurve für verschiedene Anfangswerte.

Doch er fand, daß nach einer gewissen Zeit stets dieselbe Figur auftauchte. Sie war zwar kompliziert, nämlich aus unendlich vielen Schleifen aufgebaut, aber seltsam

war das schon. Es schien so, als ob der Raum durch diese 3 Gleichungen eine unendlich

filigrane Struktur aufgeprägt bekam, einen fraktalen Attraktor, zu dem unabhängig vom Anfangswert der Punkt hingezogen wird. Da diese Figur so merkwürdig war, wurde sie als "strange attractor" bezeichnet.

Nun, einige Numeriker bezeichnen all dies als "absoluten Schwachsinn" (Zitat!), da der Schmetterlingseffekt nur auf Rundungsfehler aufbaut, also die ganze

Geschichte

nur in der Phantasie der Computerwelt existiert und keinerlei praktische Bedeutung hat. Dieser Meinung kann ich mich nicht ganz anschließen. Es kommt oft - wenn auch nicht

immer - vor, daß die Natur selbst nur mit Integerzahlen rechnet, da das betreffende Objekt

nicht unendlich oft teilbar ist. Das heißt doch aber, daß auch in der Natur zwangsweise

stets Rundungsfehler auftreten. Somit sind die Computer als perfekte Nachahmer der Natur geeignet,

zumindest qualitativ, nicht quantitativ. Die Größenordnung der Integer-Zahlen in der

Natur ist doch um einige Äonen größer als die im Computer.

1.33 2.3 Fraktale --- 2.3.4 Dynamische Systeme

2.3.4.2 Parameterwindow 1

Bereich

- Da der Lorenz/Roessler-Attraktor dreidimensional ist, fällt es auf den ersten Blick schwer, den zu zeichnenden Ausschnitt anzugeben. Ich habe mir dabei auf folgende Art befolgt:
Grundsätzlich sieht man von gerade vorn auf den Attraktor und gibt von diesem Standort ausgehend die Bereichswerte Links/Oben/Rechts/Unten an, also die x- und y-Koordinaten des Bereichs.

Ansichtswinkel

- Um nun den Attraktor nicht nur von vorne zu betrachten, sondern aus einer beliebigen Position, kann man die Ansichtswinkel ändern. Das System, das hier verwendet wird, gleicht dem bei der Erde verwendeten System: Längen- und Breitenwinkel. Mittels Alpha wird der Längenwinkel (entspricht dem Längengrad) eingestellt, mit Beta der Breitenwinkel (also der Breitengrad).

Parameter

- hier können die 3 bei den dynamischen Systemen verwendeten Parameter gewählt werden.
Es empfiehlt sich, die Parameter langsam zu ändern, da diese Systeme mitunter sehr stark auf eine Änderung reagieren.

Systemtyp

- Zur Zeit bietet das Programm 2 Typen aus der Klasse der kontinuierlichen dynamischen Systeme, den Lorenz und den Roessler-Attraktor. Beim Lorenzattraktor hat man als Ausgangssystem von Differentialgleichungen

$$\begin{aligned} dx/dt &= -ax + ay \\ dy/dt &= cx - y - xz \\ dz/dt &= -bz + xy \end{aligned}$$

beim Roessler-Attraktor dagegen

$$\begin{aligned} dx/dt &= -y - z \\ dy/dt &= x + ay \\ dz/dt &= b + xz - cz \end{aligned}$$

Theorie hierzu:

Kapitel 2.3.4.1

1.34 2.3 Fraktale --- 2.3.4 Dynamische Systeme

2.3.4.3 Parameterwindow 2

Startpunkt

- Im Theorieteil wurde erwähnt, daß unabhängig vom Startpunkt die Punktbahn stets von einem Attraktor, der wegen seines komplizierten Aufbaus als "strange attractor" gezeichnet wird, angezogen wird. Wer das nicht glaubt, hat hier die Möglichkeit,

unterschiedliche Startpunkte zu wählen. Er kann dann eine Animation berechnen ↵
lassen und
dort dann bildlich erkennen, welche Auswirkungen eine Änderung des Startpunktes
hat.

Zeiteinstellungen

- Mit 'Zeit' wird die Zeitdauer der Bahnverfolgung festgelegt. Das Differential\ ↵
gleichungs\system
beschreibt ja die Wegeänderung in Abhängigkeit der Zeit. Da aber der Computer mit ↵
der Ableitung nichts
anfangen kann, muß er dieses dx/dt (sprich "dx nach dt"), das gleichbedeutend ↵
mit dem Grenzwert von
 Δx geteilt durch Δt für t gegen 0 ist, durch Δx geteilt durch Δt ↵
mit hinreichend kleinem Δt ersetzen. Dieses Δt wird durch 'Delta' ↵
wählbar
gemacht.

Zeichengeschwindigkeit

- Der Lorenz- bzw. der Roessler-Attraktor werden recht flott gezeichnet. Das ist ↵
schön.
Aber zu Demonstrationszwecken bzw. wenn man die Punktebahn mit dem Auge ↵
mitverfolgen
will, ist es nötig, die Geschwindigkeit zu drosseln. Hiermit ist die ↵
Geschwindigkeit
einstellbar von 1 (große Verzögerung) bis 100 (keine Verzögerung)

Zeichenmodus

- Hier stehen 3 Möglichkeiten zur Auswahl:
Zeichne als Punkte/zeichne als Linien/Punktansammlung
Die ersten beiden Modi zeichnen den gesamten Lorenzattraktor vom Zeitpunkt 0 bis
zum Zeitpunkt 'Zeit' entweder jeden berechneten Punkt als Punkt oder aber mittels
einer Verbindungslinie zum jeweils letzten Punkt.
Der 3. Modus ist etwas spezielleres:
In Kapitel 2.3.4.1 über die Theorie wurde erwähnt, daß kleine Änderungen am Anfang
zu völlig anderen Ergebnissen führen (Schmetterlingseffekt). Dies kann mit diesem ↵
Modus
veranschaulicht werden. Dazu wird von einer mehr oder weniger großen Ansammlung ↵
von Punkten
ausgegangen, die alle recht nahe beieinanderliegen. Dann wird jeder Punkt auf ↵
seiner
Bahn verfolgt. Nach einiger Zeit sieht man, daß die anfangs als einzelner Punkt
erscheinende Punktwolke sich aufteilt und wiederum nach einiger Zeit deckt sie
den gesamten Attraktor ab. Dies verdeutlicht, daß eine mathematische Vorhersage
über die Position eines Punktes zu einer bestimmten Zeit bei diesem System sinnlos
ist, da in der Praxis die Anfangsposition nicht genau bestimmt bzw. festgelegt ↵
werden
kann. Jede noch so kleine Ungenauigkeit am Anfang hat auf Dauer eine ↵
unvorhersehbare
Wirkung. Das einzige, was man sagen kann, ist, daß sich der Punkt auf dem ↵
Attraktor befindet,
wo dort aber genau, ist nicht möglich.
Mittels des Gadgets 'Abstand' kann man den anfänglichen mittleren Punktabstand
der Punkte voneinander einstellen, also den Radius der Punktwolke.
Je näher die Punkte beieinanderliegen, desto länger dauert es, bis der ↵
Schmetterlingseffekt
zu sehen ist.

Im 'Punkte'-Gadget kann die Anzahl der in der Punktwolke vorhandenen Punkte ←
eingestellt
werden. Das richtet sich hauptsächlich nach der vorhandenen Rechenkapazität und
der Geschwindigkeit der Grafikausgabe, da sehr viele WritePixel-Systemaufrufe hier
nötig sind...
Spaß macht dieser Modus vor allem dann, wenn die Punktwolke aus vielen Punkten
besteht. Das erfordert allerdings viel Rechenkapazität, wenn man systemkonform
bleiben will. Damit man nun nicht einen 200-Mhz-68060 und eine ←
hardwareunterstützte
WritePixel-Routine braucht, habe ich ein Gadget hinzugefügt, mit dem man einen
Modus einschalten kann, in dem direkt in die Planes hineingezeichnet wird, was ←
enorme
Geschwindigkeitsvorteile bringt, Grafikkartenbesitzer aber alt aussehen läßt. In
diesem Modus wird radikal alles übermalt, was sich im Einzugsbereich des Windows
befindet, also z.B. das Menü, ein darüberliegendes Window etc. Also bitte darauf ←
achten,
daß das Window vollständig sichtbar ist...

Theorie hierzu:

Kapitel 2.3.4.1

1.35 2.3 Fraktale --- 2.3.5 Plasma

2.3.5 Plasma

2.3.5.1 Theorie

Plasma ist nichts anderes als eine 2-dimensionale Brown'sche Bewegung. Eine 1- ←
dimensionale
Brown'sche Bewegung wird folgendermaßen konstruiert: Man geht z.B. von einem Punkt
aus, besorgt sich eine Zufallszahl, wobei der erzeugende Zufalls\zahlen\generator
 $N(0;1)$ -normalverteilt sein muß. Man geht von dem Punkt aus horizontal einen ←
Schritt
nach rechts und um die Zufallszahl nach unten bzw. oben, je nach Vorzeichen ←
derselben.
Das wiederholt man. Das Ergebnis ist so etwas wie ein Schnitt durch ein Gebirge,
also eine Zickzack-Bewegung mal nach unten, mal nach oben.
Von dieser Methode zur Erzeugung einer Brown'schen Bewegung gibt es mehrere ←
Varianten.
Die eine ist, daß man z.B. eine Brown'sche Bewegung von Punkt A nach Punkt B ←
erzeugen
will. Dazu nimmt man die beiden Punkte, verbindet sie durch eine (imaginäre) ←
Strecke,
holt sich den Mittelpunkt dieser Strecke und versetzt den Mittelpunkt um eine ←
Zufallszahl,
die noch mit einer Zahl multipliziert wird, die von der gewünschten Dimension der
Bewegung und der Länge des Teilungsintervalls abhängt. Man erhält 2 Strecken, für
die man analog verfährt. Von diesem Algorithmus ist leicht eine 2-dimensionale
Variante denkbar: Man hat ein Viereck und versetzt den Mittelpunkt des Vierecks
um eine Zufallszahl, man macht entsprechendes mit den Randpunkten und erhält ←
wieder
4 Vierecke, mit denen man analog verfährt. Ungefähr dieser Algorithmus wird in ←
meinem

Programm angewendet.

1.36 2.3 Fraktale --- 2.3.5 Plasma

2.3.5.2 Parameterwindow 1

Sigma

- Bei jeder Mittelpunktsverschiebung wird eine Zufallszahl geholt, diese mit einer Zahl multipliziert, die von der Dimension und von der Länge des Intervalls abhängt ←

Diese Zahl ist die Basis dieses Multiplikators, d.h. bei der ersten Mittelpunkts\ ←
verschiebung

wird die Zufallszahl mit dieser Zahl multipliziert, die weiteren Mittelpunkte der kleineren Vierecke werden mit Teilen dieser Zahl multipliziert. Die genauen 'Teile ←

hängen von der gewünschten Dimension ab.

H

- Diese Zahl bestimmt direkt die Dimension des Objekts. Die resultierende ←
Dimension

ist $3-H$, d.h. ist H z.B. 0.9 , so ist die Dimension $3-0.9=2.1$, also ist das ←
Ergebnis

ein bisschen rauher als eine Fläche. Ist $H=0.1$, so ist die Dimension allerdings ←
 2.9 ,

was eine sehr rauhe Fläche, die lokal beinahe raumfüllend ist, zur Folge hat. Man kann sich das Gebilde als einen Berg mit sehr vielen sehr hohen Zacken vorstellen, eben beinahe raumfüllend...

ColorMult

- Der Ergebniswert wird mit dieser Zahl multipliziert, das Ergebnis als Farbnummer interpretiert. Dieser Parameter hat eine ähnliche Wirkung wie Sigma, allerdings ←
ändert

er nicht die Werte im Puffer, sondern veranlaßt eine andere Interpretierung. Das Plasmafraktal wird bei einer Änderung dieses Parameters nicht neu berechnet, es ←
wird

lediglich der Puffer neu dargestellt, was die Geschwindigkeit erhöht.

Reihe

- Da mit Zufallszahlen gearbeitet wird und eine Zufallszahlenreihe auf einem ←
Computer

deterministisch ist, ist es nötig, einen Startwert für die Reihe zu wählen, damit ←
nicht

immer dasselbe Gebilde herauskommt.

Theorie hierzu:

Kapitel 2.3.5.1

Anmerkung:

Seit ChaosProV2.0 darf Plasma nun auch in 3D berechnet werden, was logischerweise ←
zu mehr oder weniger natürlichen Bergen

führt. In Verbindung mit dem 3D-Puffer kann das Ganze dann auch in 24 Bit ←
gespeichert werden, was der Realitätsnähe des

Bildes dann durchaus zuträglich ist. 'H' bestimmt die Dimension und hat somit ←
großen Einfluß auf das 3D-Bild. $H=1.1$

ist recht gut, der Normalwert von 0.8 ist für 3D zu niedrig. 0.8 hat starke Verklüftungen zur Folge...

1.37 2.3 Fraktale --- 2.3.6 Lyapunov-Raum

2.3.6 Lyapunov-Raum

2.3.6.1 Theorie

Der Lyapunov-Raum baut auf den Bifurkationsdiagrammen auf. Dort wurde eine Formel benutzt, die z.B. die Populationsentwicklung beschreibt. In Abhängigkeit von der Wachstumsrate wurde graphisch dargestellt, ob , und wenn ja, welches Gleichgewicht in der Bevölkerung herrscht (kein Gleichgewicht, also chaotisches Verhalten, eindeutiges Gleichgewicht, oder ein Zyklus einer bestimmten Länge).

Beim Lyapunov-Raum führt man noch einen 2. Wachstumsfaktor ein, der sich mit dem ersten in einer definierten (aber einstellbaren) Art und Weise abwechselt.

Als Beispielsequenz sei z.B. AAABB gegeben, sei $A=3$, $B=2$

Veranschaulicht heißt das, daß meine Hasenpopulation im einen Jahr z.B. mit einem Faktor von 3 wächst, im nächsten dann wieder um den Faktor 3, dann wieder um den Faktor

3, dann plötzlich um den Faktor 2, dann wieder um den Faktor 2, dann beginnt das ganze wieder von vorn, also sie wächst wieder um den Faktor 3 usw.

Das ganze spielt man nun für alle möglichen Werte von A und B als Wachstumsraten durch, trägt dabei A in der Horizontalen und B in der Vertikalen graphisch auf. Nun muß man sich noch überlegen, was man an der Stelle macht. Es bietet sich an, an der durch A und B definierten Stelle einen Bildschirmpunkt in einer bestimmten Farbe zu zeichnen. Doch in welcher Farbe?

Bei den Bifurkationsdiagrammen war eine globale Unterscheidung in 2 Hauptklassen möglich:

1. Klasse: Werte des Wachstumsfaktors, die zu einem Gleichgewicht führt, zu einem Zyklus irgendeiner Länge
2. Klasse: Werte des Wachstumsfaktors, die zu keinem Gleichgewicht führen.

Diese Klasseneinteilung machen wir hier nun auch und färben einen Punkt entsprechend ein.

Die Frage bleibt, wie wir entscheiden wollen, ob ein Gleichgewicht oder Chaos vorherrscht.

Nun, betrachten wir die Formel $f(x)=a*x*(1-x)$

Wir legen fest, daß Ordnung vorherrscht, wenn der Durchschnitt aus dem Betrag der Ableitung der Funktion $f(x)$ nach hinreichend vielen Jahren kleiner 1 ist, sonst liegt

Chaos vor. Doch müssen wir, wie bei den Bifurkationsdiagrammen, dem Wert erst einmal

Gelegenheit geben, zu einem Attraktor hinzuwandern. In der Praxis bilden wir noch kurz den

Logarithmus, so daß wir algorithmisch erhalten:

$X=0.5$

(müssen X erst mal Gelegenheit geben, zum Attraktor zu wandern...)

```
FOR N=1 TO 4000
  (R ist A oder B, je nachdem, wie die Sequenz aussieht)
  X=R*X*(1-X)
NEXT N
```

```
Summe=0
FOR N=1 TO 6000
  (R ist A oder B, je nachdem, wie die Sequenz aussieht)
  X=R*X*(1-X)
  Summe=Summe+Ln|R-2*R*X|
NEXT N
Summe=Summe/6000
```

So, das Ergebnis, in der Variablen 'Summe' ist nun der Durchschnitt aus den Größen $\ln|R-2*R*X|$, also dem Logarithmus der Ableitung der Formel $f(x)$ und repräsentiert die Rate, mit der die Größe von aufeinanderfolgenden Werten der Funktion wächst. ↔

Das Ergebnis wird auch Lyapunov-Exponent genannt.

Ist der Durchschnitt negativ (d.h. also $|R-2*R*X|$ ist kleiner als 1 im Durchschnitt),

so herrscht Ordnung vor, ansonsten Chaos.

Chaos färben wir einfarbig ein, meistens schwarz. Ich habe testweise das Chaos eingefärbt und festgestellt, daß es nicht umsonst als Chaos bezeichnet wird... Falls ein Wert kleiner 0 herauskommt (d.h. Ordnung, in der Praxis erhält man

Werte meist

so bis -5) so multiplizieren wir die Zahl noch geeignet, wandeln es in eine ganze Zahl

um und färben den Punkt mit dieser Farbe ein. Das wars dann schon...

1.38 2.3 Fraktale --- 2.3.6 Lyapunov-Raum

2.3.6.2 Parameterwindow 1

Formel

- Diese Formeln sind identisch mit den Formeln bei den Bifurkationsdiagrammen, lediglich Formel Nr. 3 fehlt, da ich nicht wußte, was ich hier mit der Ableitung der Funktion anfangen sollte.

ExpMin

- Hier kann man den minimalen Exponenten einstellen. Die Farben werden automatisch bestmöglich auf die Lyapunov-Exponenten von ExpMin bis 0 verteilt. Alle Werte, die einen kleineren Lyapunov-Exponenten ergeben als ExpMin, werden mit Farbe 4 eingefärbt. ↔

Start x

- Wer den Lyapunov-Raum betrachtet, dem werden gewisse Zacken auffallen, die sich kreuzen. Sehr bemerkenswert ist, daß die Lage der Zacken, also ob der eine vor dem anderen oder hinter dem anderen liegt, vom Startwert für x abhängt, den man hiermit einstellen kann. ↔

Sequenz

- Hier kann man das Schema der Abwechslung zwischen den beiden Wachstumsraten angeben. ↔

Damit die Änderung wirksam wird, muß das Gadget mittels Tab-oder Return-Taste ↵ verlassen werden.

Wer also die 7 fetten und die 7 mageren Jahre nachbilden will, muß hier nur ↵
AAAAAABBBBBBBB
eingeben...

Passes

- Um einen schnelleren Eindruck vom Aussehen des Lyapunov-Raumes zu bekommen, kann man erst einmal die Auflösung künstlich verkleinern wie man es schon von den Julia-↵
Mandel-Mengen gewohnt ist. Wem sich nun die Nackenhaare aufstellen, weil er meint, daß ich diese Methode zur Verschnellerung einsetzte (wie bei Julia/Mandel), der darf sich nun ↵
gleich wieder kämmen. Keine Angst, hier wird nur vergrößert, durch Wahl von 3 ↵
Zeichendurchgängen wird das Fraktal langsamer gezeichnet als mit einem (wenn auch nur unbedeutend ↵
langsamer), aber man bekommt sehr viel schneller einen Eindruck vom Aussehen und kann gleich weiter hineinzoomen.

Chaosfarbe

- Name sagt alles, oder?

Stabilisation

- Dieser Wert gibt an, wie oft die Formel iteriert wird, ohne daß der Exponent ↵
berechnet wird. Dies gibt dem Startwert Gelegenheit, zu seinem Attraktor hinzuwandern.

Iteration

- Die hier angegebene Zahl gibt an, wie oft die Formel dannach noch iteriert wird, ↵
um den Lyapunov-Exponenten zu berechnen. Der Geschwindigkeit zuliebe rate ich euch, diese Werte erst recht klein zu wählen ↵
und sie dann testweise zu erhöhen, um zu sehen, ob sich was weltbewegendes ändert...

Bereich

- dürfte allen klar sein...
A wird horizontal aufgetragen, B vertikal

Theorie hierzu:

Kapitel 2.3.6.1

1.39 2.3 Fraktale --- 2.3.6 Lyapunov-Raum

2.3.6.3 Datenwindow

Im Datenwindow werden die dem Bildschirmpunkt entsprechenden Wachstumsraten, also ↵
A und B, angezeigt und außerdem der Lyapunov-Exponent nochmal kurz berechnet.

Theorie hierzu:

Kapitel 2.3.6.1

1.40 2.3 Fraktale --- 2.3.7 IFS

2.3.7.1 IFS

In letzter Zeit sorgte die fraktale Bildkompression für einigen Wirbel, versprach sie doch ungeheure Packraten. Angeblich sollte es möglich sein, mit ein paar wenigen Zahlen Bilder zu codieren. Der Grundstein für dieses effektive Packen von Bilddaten ist ein iteriertes Funktionensystem, kurz IFS. Letztendlich sind auch die Julia- /Mandelbrot\mengen iterierte Funktionensysteme, doch auf sie soll hier nicht eingegangen werden. Hier wird versucht, lineare iterierte Funktionensysteme zu betrachten. Wie schon bei den Julia- /Mandelbrot\mengen interessiert man sich hier auch für den 'Attraktor'. Daß dieser Attraktor sehr kompliziert sein kann, hat man schon durch die Julia- / Mandelbrot\mengen erfahren. Das besondere an den (linearen) IFS ist, daß man nicht nur aus einem IFS den Attraktor berechnen kann, sondern umgekehrt, daß es möglich ist, zu einem gegebenen Attraktor ein IFS zu bestimmen, das man berechnen kann, so daß der Attraktor, zu dem man es bestimmt hat, in etwa wieder herauskommt. Dadurch erreicht man natürlich Bildkompression. Man muß lediglich die Funktionen so wählen, daß der Attraktor eben das Bild ist, das man 'packen' will. Klingt recht einfach, oder? Wie sieht die Sache nun genauer aus?

Ein IFS besteht aus mehreren Funktionen, jede einzelne von ihnen sei eine affine Transformation, d.h. eine Funktion, die nichts anderes macht, als ein Objekt um einen bestimmten Winkel zu drehen, zu stauchen und zu verschieben. Zu jeder dieser affinen Transformationen existiert noch eine sog. Ausführungs\wahrschein\lichkeit. Solche affinen Transformationen können einfach durch eine Matrix (für Drehung und Stauchung zuständig) und einen Vektor (für Verschieben zuständig) beschrieben werden.

Nun, wie wird ein Bild, der Attraktor des IFS, berechnet?

1. Man startet mit einem Punkt, der sicher auf dem Attraktor liegt, hier bietet sich z.B. der Fixpunkt der ersten Transformation an.
2. Man sucht sich eine Funktion aus dem IFS aus, so daß auf lange Sicht eben die gewünschte Ausführungs\wahrschein\lichkeit erreicht wird.
3. Man wendet die affine Transformation auf den letzten Punkt an und geht wieder zu Schritt 2

Mehr oder weniger schnell erscheint dann der Attraktor.

Nun zu der Frage, warum und vor allem wie es möglich ist, ein IFS zu einem
gegebenen Attraktor zu bestimmen.

Es gibt einen mathematischen Satz dafür, daß dies möglich ist, und der hat sogar
einen Namen.

Er lautet Kollage Theorem, und besagt:

Gegeben sei ein IFS mit den Transformationen w_1 bis w_n , desweiteren eine Menge T .
Sei s die größte Lipschitzkonstante der Transformationen w_1 bis w_n , sie muß
kleiner 1 sein. Tja, s soll, informell gesagt, der
maximale Kontraktionsfaktor sein. Jede Transformation muß ja eine Mischung aus
Drehung, Stauchung und Verschiebung sein,
 s soll dann eben der maximale Stauchungsfaktor sein, das ist dann die
Lipschitzkonstante.

Der Hausdorffabstand von T und der Vereinigung der $w_1(T)$ bis $w_n(T)$ soll kleiner
einer vorgegebenen Schranke ϵ sein.

Dann ist der Hausdorffabstand zwischen dem Attraktor des IFS und T kleiner als
 $\epsilon/(1-s)$.

Der Hausdorffabstand zwischen 2 Mengen ist, etwas flapsig und mathematisch falsch,
aber eben klarer ausgedrückt, der
'normale' Abstand zwischen 2 Mengen.

Was besagt nun der Satz?

Sei eine Menge T gegeben, das kann z.B. ein Bild sein. Desweiteren hat man sich
die Transformationen w_1 bis w_n so
gewählt, daß jeweils die Vereinigung der Mengen $w_1(T)$ bis $w_n(T)$, also die
Bildmengen, die entstehen, wenn man w_1 bis w_n jeweils getrennt auf T
anwendet, ziemlich genau T überdecken (also sie alle zusammen eben eine Kollage
ergeben, das Original wird aus verkleinerten, gedrehten und verschobenen
Kopieen des Originals aufgebaut, daher auch der Name des Satzes). Dann ist die
die Bedingung des Satzes in etwa erfüllt, nämlich daß der Hausdorffabstand
zwischen T und der Vereinigung der $w_1(T)$ bis $w_n(T)$ kleiner einer gewissen
Schranke

ϵ ist. Die Größe der Schranke richtet sich natürlich nach der Qualität der
Kollage.

Dann sagt der Satz: Der Attraktor des IFS, also das Bild, das man mit den
Transformationen berechnen kann, stimmt mit dem

Original T (evtl. ein Bild) derart überein, daß der Hausdorffabstand zwischen
beiden kleiner als $\epsilon/(1-s)$ ist.

D.h., man braucht das Bild nicht mehr, man braucht nur noch die w_1 bis w_n und kann
daraus dann das Bild berechnen, der Fehler ist kleiner
als $\epsilon/(1-s)$.

Hier sieht man schon, daß es sinnvoll ist, s so klein wie möglich zu machen.
Andererseits dürfte klar sein,

daß man dann das Original aus sehr vielen sehr stark verkleinerten Kopieen seiner
selbst

zusammenbasteln muß, d.h. man hat sehr viele Transformationen, was die Packrate
senkt,

und was vor allem die Berechnungszeit für den Attraktor, also quasi das Entpacken,
sehr langsam macht. Sinnvoll ist es z.B., das Bild aus vielleicht Vierteln seiner
selbst
aufzubauen.

Tja, das ist die Idee der fraktalen Bildkompression. Man muß lediglich die
Transformationen w_1

bis w_n herausfinden. Das macht man am besten so: Gegeben ein Bild, das kopiert man
und stellt

es verkleinert dar. Der Benutzer soll nun durch Drehen, Stauchen und Verschieben
 die
 Kopie des Bildes so über dem Original plazieren, daß es einen Teil des Originals
 überdeckt.
 Anschließend wird es fixiert, die erste Transformation ist fertig (man hat sich
 halt
 die durchgeführten Drehungen, Stauchungen und Verschiebungen gemerkt). Dannach
 holt man sich
 eine weitere Kopie des Originals, dreht, staucht und verschiebt sie wieder, bis
 sie
 einen noch nicht überdeckten Teil des Originals überdeckt, fixiert sie dort und
 schon
 hat man die 2. Transformation. So fährt man fort, bis man das Original aus
 verkleinerten Kopieen
 seiner selbst zusammengesetzt hat, genau wie in einer Kollage. Anzumerken ist noch
 ,
 daß sich die Teile überlappen dürfen, das führt lediglich zu einer erhöhten
 Rechenzeit, der
 Attraktor ist eben nicht so schnell sichtbar.

Ich habe mich natürlich bereits an einem Programm, das dies interaktiv ermöglicht,
 versucht,
 leider kam nicht das Richtige heraus. Vielleicht liegt es daran, daß ich im
 Kunstunterricht
 beim Thema 'Kollagen' nicht aufgepaßt habe, vielleicht aber auch an meinen
 mangelhaften Programmierkenntnissen. Wer weiß, vielleicht schaffe
 ich es ja noch einmal.

1.41 2.3 Fraktale --- 2.3.7 IFS

2.3.7.2 IFS-Parameter 1

Die meisten Parameter sollten eigentlich selbsterklärend sein.

Formel

Im Listview wählt man die Formel aus, die man mit dem
 Formeleditor
 erstellt

hat.

Parameter

Rechts neben dem Listview kann man bis zu 4 Parameter wählen, die evtl. von der
 gewählten Formel
 benutzt werden.

Iteration

Diese Zahl gibt die Anzahl der Punkte an, die berechnet werden.

Farbmodus

Hiermit kann man den Einfärbungstyp festlegen.

Es stehen 3 Typen zur Auswahl:

1. Transformation: Die Farbe des Punktes wird durch die zuletzt ausgeführte
 Transformation

bestimmt.

2. Wahrscheinlichkeit: Selbes wie 1.: Zur Bestimmung der anzuwendenden
 Transformation

wird eine Zufallszahl berechnet. Hier bestimmt dann die Zufallszahl die Farbe des Punktes.

3. Maß: Dies ist der bevorzugte Modus. Je größer das Maß des IFS an der betreffenden Stelle ist, desto größer ist die Farbnummer.

Ausschnitt
Sollte klar sein...

Winkel

Ein IFS ist in ChaosPro stets ein 3 dimensionales IFS. Deshalb existieren diese beiden Gadgets, mit denen man den Blickwinkel einstellen kann.

1.42 2.3 Fraktale --- 2.3.8 L-System

2.3.8.1 L-System

Nun, die korrekte Bezeichnung sollte DOL-Systeme, oder auf deutsch, deterministisch kontextfreie Lindenmayer-Systeme, lauten. Deterministisch, da eben keinerlei Zufall eine Rolle spielt im Gegensatz zu den IFS, kontextfrei, da es sich um kontextfreie Grammatiken handelt (Informatikerjargon, hochtrabend ausgedrückt heißt es, daß es Regeln zur Textersetzung gibt, die unabhängig vom Textzusammenhang anzuwenden sind, d.h. Regeln der Form "Buchstabe wird ersetzt durch Zeichenkette" und nicht eben kontextsensitiv, "Zeichenkette wird durch Zeichenkette ersetzt".)

Soviel zum Wort, um was handelt es sich? Nun, ausgehend vom einer Anfangszeichenkette, dem Axiom, werden fest vorgegebene Regeln zur Textersetzung angewandt. Sie alle die Form "Buchstabe wird durch Zeichenkette ersetzt". Man fängt beim ersten Zeichen an, prüft, ob es eine Regel gibt, die dieses Zeichen ersetzt, wenn ja, ersetze, wenn nein, mache nichts. Man macht beim 2. Zeichen weiter, sucht wieder nach einer Regel, ersetzt, falls es eine Regel gibt, und geht zum dritten Zeichen. So fährt man fort bis zum Ende der Zeichenkette. Dann ist die erste Iteration fertig. Nun fängt Iteration 2 an, man beginnt wieder mit dem ersten Zeichen, prüft, ersetzt, etc. bis zum letzten Zeichen, dritte Iteration, 4. Iteration, usw., bis die maximale Iteration erreicht ist. Werte kleiner 10 sind da durchaus sehr sinnvoll, die Zeichenketten wachsen nämlich ganz schön schnell.

Am Ende kommt es nun: Gewisse Sonderzeichen werden als Zeichenkommandos interpretiert, und so wird der Endstring 'gezeichnet'. Das Ergebnis sind dann Fraktale, zumindest wenn man unendlich oft iterieren würde. Die Lindenmayer-Systeme haben eine gewisse Bedeutung in der computergestützten Landschaftserzeugung erlangt, da die Ergebnisse oft Elementen aus der Natur, vornehmlich Pflanzen, ähneln. Leider ist es so, daß man mit

deterministisch kontextfreien Systemen nur sehr schwer in der Lage ist, wirklich realitätsnahe Elemente darzustellen. Deshalb gibt es Erweiterungen der DOL-Systeme \leftrightarrow , die dann eben kontextsensitiv und nicht deterministisch sind.

Um Klarheit zu schaffen, denke ich, ist es sinnvoll, ein einfaches Beispiel \leftrightarrow anzuführen:

Die Kochkurve: Jeder sollte sie kennen, sie ist ideal für DOL-Systeme:

Das Axiom, also der Anfangsstring: F

Die Regel: F wird ersetzt durch F-F++F-F

Das Ergebnis: Die Kochkurve, zumindest wenn man F beim Zeichnen als eine gerade \leftrightarrow Linie auffaßt, '-' eine Drehung um -60 Grad, '+' eine Drehung um +60 Grad ist. Allgemein ist es so, daß die \leftrightarrow Zeichenkommandos und vor allem die Art des Zeichnens sehr den Turtle-Grafiken ähnelt. Aber, wer weiß heutzutage noch was von \leftrightarrow Turtle-Grafiken? Früher war diese Art des Zeichens sehr beliebt. Man stellte sich eine virtuelle Schildkröte (turtle) \leftrightarrow vor, der man eben Kommandos geben konnte wie 'drehe dich um 30 Grad', 'bewege dich \leftrightarrow gerade vorwärts und hinterlasse dabei eine Schleimspur' (Zeiche), oder 'bewege dich \leftrightarrow gerade vorwärts und zeichne dabei nichts' (wie macht sie das bloß?) und Ähnliches.

Zu den Sonderkommandos, die als Zeichenkommandos interpretiert werden, siehe \leftrightarrow Kapitel über den L-System-Formeleditor.

1.43 2.3 Fraktale --- 2.3.8 L-System

2.3.8.2 LSystem-Parameter 1

Im Listview kann die Formel gewählt werden, von der man eine Darstellung berechnen \leftrightarrow lassen will.

Die Formeln können im Formelwindow für L-Systeme geändert werden.

Rechts neben dem Listview finden sich 4 Parameter, die derzeit noch unbenutzt sind \leftrightarrow

Darunter kann Anzahl an Iterationen eingestellt werden, also wie tief die \leftrightarrow Stringersetzung durchgeführt werden soll.

Normal sind hier Werte von 3 bis 15. Es ist sehr sinnvoll, mit 3 anzufangen und \leftrightarrow diese Zahl schrittweise um 1 zu erhöhen und

die Auswirkungen zu beobachten. Die Zeit, die für die Berechnung benötigt wird, \leftrightarrow nimmt nämlich exponentiell zu, d.h. eine

Erhöhung der Iterationszahl um 1 hat evtl. eine Vervierfachung der benötigten \leftrightarrow Rechenzeit zur

Folge.

Unter dem Listview kann man noch den Ausschnitt wählen. Andere Programme gehen so \leftrightarrow vor, daß sie zuerst das

Bild komplett berechnen, die Größe des Bildes berechnen und dann die \leftrightarrow Ausschnittswerte so wählen, daß

das Bild optimal in das Window paßt. Das jedoch hat den gravierenden Nachteil, daß man unter Umständen erst ein paar Minuten warten muß und dann erst das Bild sieht, was ich für unangenehm halte.

1.44 2.3 Fraktale --- 2.3.9 Diffusion

2.3.9.1 Diffusion - Theorie

Tja, was sauge ich mir jetzt wieder aus den Finger?

Manche Phänomene in der Natur erinnern recht stark an Fraktale, so z.B. die elektrostatische Entladung einer Wolke, auch Blitz genannt. Oder aber Diffusionsphänomene, z.B. wenn man

Öl und Wasser mischt. Diese beiden Flüssigkeiten bilden gewisse Grenzlinien, die sehr stark Blitzen ähneln.

Nun, solch eine Diffusion kann man auch mit dem Computer simulieren. Dazu geht man folgendermaßen vor:

Man startet mit einem Objekt auf dem Bildschirm, z.B. einem Punkt. Dann läßt man irgendwo einen weiteren Punkt entstehen. Diesen weiteren Punkt bewegt man nun zufällig herum, so daß er einen sogenannten 'Random Walk', eine Brownsche Bewegung, ausführt. Genauer gesagt macht man es so: Man holt sich eine Zufallszahl zwischen 0 und 3, interpretiert diese Zahl als Richtung, also 0 heißt nach rechts, 1 nach unten, 2 nach links, 3 nach oben, bewegt den Punkt ein Pixel weit in die entsprechende Richtung und fährt so fort...

Klar, daß das ewig weitergeht, also braucht es noch ein Abbruchkriterium. Es soll ja ein Diffusionsfraktal entstehen.

Also, sobald der sich bewegende Punkt in die Nachbarschaft eines festen Punktes kommt, bleibt

er an diesem Punkt kleben und bewegt sich nicht mehr. Das ist das Ende dieses Punktes, ein neuer Punkt wird erzeugt, bewegt sich wieder ziellos hin und her, bis er

an einem festen Punkt kleben bleibt. Das resultierende Bild ist dann eine Simulation eines

Diffusionsprozesses.

1.45 2.3 Fraktale --- 2.3.9 Diffusion

2.3.9.2 Diffusionsparameter 1

Tja, welche Parameter gibt es hier?

Typ

Hier gibt es 2 Möglichkeiten:

1. Diffusion: Berechnet ein Diffusionsfraktal

2. Random Walk: Zeigt lediglich einen 'Random Walk', d.h. wie sich ein Punkt hin- und

herbewegt.

Form

4 Möglichkeiten stehen zur Auswahl, die die Anfangsform der festen Punkte bestimmt ↔

Ist die Anfangsform eine Linie, so ist klar, daß die herumfahrenden Punkte entlang ↔
dieser Linie

irgendwann kleben bleiben, und so etwas in der Art von 'Moos' entsteht, eben etwas ↔
ähnliches wie

'viele Blitze aus einer Linie'.

- a) Punkt: Bloß ein Punkt in der Mitte
- b) Linie: Zeichnet eine Linie
- c) Quadrat: Zeichnet ein Quadrat
- d) Zufall: setzt zufällig ein paar Punkte

Zeige Weg

Falls angewählt, wird der Weg des Punktes gezeigt, ansonsten wird bloß das ↔

Resultat gezeigt, also
der Punkt, sobald er festgeklebt ist...

Reihe

Bestimmt den Startwert für den Zufallszahlengenerator

Farbmodus und Farbadd

a) Nachbar: Die Farbe des Punktes berechnet sich aus der Farbe des Nachbarpunktes
+Farbadd

b) Zeit: Die Farbe des Punktes berechnet sich direkt aus der Zeit, die der Punkt ↔
herumgewandert
ist.

Haftungs\wahrscheinlichkeit

Werte zwischen 0 und 1 sind erlaubt. Dieser Wert bestimmt, mit welcher ↔

Wahrscheinlichkeit ein Punkt, der
auf einen festen Punkt trifft, dort hängenbleibt. Bisher wurde angenommen, daß der ↔
Punkt stets haften bleibt, sobald

er auf einen festen Punkt trifft. Doch das muß natürlich nicht sein. Mit diesem ↔
Wert stellt man diese Wahrscheinlichkeit ein.

1 heißt, der Punkt bleibt haften, sobald er auf einen festen Punkt trifft, 0.5 ↔
heißt, der Punkt bleibt in der Hälfte

aller Fälle am erstbesten festen Punkt haften, im Durchschnitt allerdings erst am ↔
2. festen

Punkt, auf den er trifft.

Boxgröße

Die Berechnung eines Diffusionsfraktals dauert eine wahre Ewigkeit. Um das ↔
wenigstens etwas zu beschleunigen, wird

um die Menge aller festen Punkte ein Rechteck definiert, das um 'Boxgröße' Pixel ↔
größer ist als das kleinste Rechteck, das

alle festen Punkte umfaßt. In diesem größeren Rechteck erscheinen dann alle neuen ↔
Punkte. Sobald ein herumfahrender Punkt dieses

Rechteck verläßt, wird er verworfen (es dauert zu lange, bis er wiederkommt) und ↔
ein neuer Punkt in dem Rechteck geschaffen, der dann

eben hoffentlich schneller hängenbleibt.

1.46 2.3 Fraktale --- 2.3.10 Brown

2.3.10.1 Brown - Theorie

Und weiter geht's mit dem nächsten Fraktaltyp, der Brownschen Bewegung. Keine Angst, ich fange jetzt nicht mit einer wahr\schein\lichkeits\theore\tischen Betrachtung der Brownschen Bewegung an, also nichts über unendliche Pfade, differenzierbar an keiner Stelle und solchen Sachen. Dazu verweise ich auf Bandelow ;-) (Priv. Doz. Dr. Stefan Schäfflers Lieblichsbuch...)

Eigentlich brauche ich sowieso kaum etwas erklären. Der Fraktaltyp 'Plasma' ist ja eine 2 dimensionale Brownsche Bewegung. Eine Brownsche Bewegung hat einige sehr angenehme Eigenschaften. So kann man eine 1-dimensionale Brownsche Bewegung aufzeichnen, und sie sieht eher nach einem Musiksample aus. Mit Hilfe eines Parameters kann man die Dimension der Brownschen Bewegung bestimmen. Es ist klar, daß, wenn etwas nach einem Sample aussieht, man dieses Sample auch spielen sollte. Das kann ChaosPro auch, keine Angst. Letztendlich hat sich herausgestellt, daß Musik sehr einer Brownschen Bewegung ähnelt, d.h. man kann eine Brownsche Bewegung abspielen, und das macht auch noch Sinn... Es sollte dies das Vorstadium für fraktale Musik sein. Man darf allerdings nicht erwarten, daß da dann plötzlich Spärenklänge ertönen. Laut Literatur hört sich die 'Musik' eben wie Musik an, allerdings etwas arg fremdartig für uns... Einmal davon abgesehen, daß typische Elemente, die in jedem Musikstück auftreten, wie Refrain, Themenabwandlung, Tonhöhenänderung bei gleichbleibendem Thema etc., natürlich hier nicht auftreten. Mozart und Beethoven haben also nach wie vor ihre Berechtigung...

1.47 2.3 Fraktale --- 2.3.10 Brown

2.3.10.2 Brown-Parameter 1

Skalierung
Bestimmt einen Verkleinerungsfaktor, der für die Berechnung benötigt wird. Er muß zwischen 0 und 1 liegen, je näher an 1, desto länger dauert die Berechnung. Dieser Parameter hat nichts mit der graphischen Darstellung zu tun, vielmehr bestimmt er die Anzahl an Durchgängen, die benötigt werden, bis die Brownsche Bewegung berechnet wird.

H
Bestimmt die Dimension, die ist nämlich 2-H...

Samples
Bestimmt die Anzahl an Punkten, die für die Brownsche Bewegung berechnet werden

.

Spiele & Stop

Startet/stoppt das Spielen der Brownschen Bewegung. Die Höhe wird durch 'Rate' ←
bestimmt, ob
als Musik oder als Sample wird durch 'Modus' bestimmt.

Modus

Bestimmt, ob die Brownsche Bewegung als eine Folge von Tönen gespielt werden soll ←
oder
ob sie als ein Sample abgespielt werden soll.

Rate

Bestimmt die Tonhöhe des Abspielens.

1.48 2.3 Fraktale --- 2.3.11 3D-Ansichten

2.3.11 3D-Ansichten

2.3.11.1 3D-Einführung

Das 3D-Modul ist als recht eigenständiges Modul ausgelegt. Es nimmt keine ←
Rücksicht darauf,
zu welchem Fraktal (wenn überhaupt) die Daten gehören, die es in einem Array ←
übergeben
bekommt. Das hat natürlich Nachteile, da so keine Rückkoppelung zum Fraktal ←
möglich
ist, falls mehr Werte zur korrekten Berechnung benötigt würden. Der Vorteil ist ←
jedoch,
daß in einer späteren Version des Programms ein Außenstehender eine Tabelle mit ←
Höhenwerten
aufbauen und von meinem Programm dann umsetzen lassen kann. So spart man sich eine ←
Menge Arbeit.
Außerdem kann ich irgendwelche Höhenwerttabellen von z.B. Scenery Animator oder
einem ähnlichem Programm kurz in mein Format umrechnen, so daß sie von diesem ←
benutzt
werden können. Wie gesagt, vieles ist möglich, was letztendlich eingebaut wird, ←
entscheidet
sich in der Hauptsache durch eure Reaktionen...

1.49 2.3 Fraktale --- 2.3.11 3D-Ansichten

2.3.11.2 3D-Parameterwindow 1

Projektionsart

Es stehen 2 Projektionsarten zur Verfügung:

1. Orthogonal: Die bevorzugte, gute Methode. Hier wird jeder Punkt einfach per ←
orthogonaler

Projektion auf eine 2D-Fläche abgebildet. Eine Entfernung spielt keine Rolle, die
Implementation eines horizontalen Ansichtswinkels ist nicht möglich, ich habe es
versucht. Man benutze statt dessen eine Rotation des 2D-Bildes. In diesem Modus ←
wird

stets das gesamte Bild in bestmöglicher Qualität gezeichnet.

2. Perspektive: Die alte, schlechte, Methode, nur auf Wunsch eines meiner ←
Lieblingsbetatesters

wieder implementiert. Hier wird das 3D-Objekt auf eine 2D-Fläche projiziert, so wie es das menschliche Auge macht. Hier können auch noch andere Dinge gewählt werden.

Der große Nachteil bei diesem Algorithmus ist, daß das gesamte Zeichnen in keinsten

Weise für den Computer vereinfacht werden kann, was diesen Modus sehr langsam macht.

Der gesamte Programmcode ist auch sehr unübersichtlich, was Erweiterungen zu einer wahren Qual werden läßt.

Zeichenmodus

Nur bei Projektionsart = Projektion...

Hier kann man wählen, wie die einzelnen Werte dargestellt werden sollen:

1. Punkte: zeichnet nur die Punkte
2. Gridlinien: zeichnet die Verbindungslinien zwischen den Punkten
3. Vierecke: zeichnet aus jeweils 4 Punkten ein Viereck
4. Spitzen: zeichnet Spitzen vom Grund aus.
5. Mosaik: zeichnet einfach Plättchen der Größe 2x2 an den Kreuzungspunkten des Rasters
6. Kreuz: zeichnet an den Kreuzungspunkten Kreuze.

Am besten, man probiert es aus...

Entfernung

Nur bei Projektionsart = Projektion...

Name sagt alles...

H-Winkel

Nur bei Projektionsart = Projektion...

- Der Horizontalwinkel, von dem aus der Beobachter auf das Objekt blickt. Er entspricht dem Breitengrad der Weltkugel.

V-Winkel

- Der Beobachter steht stets direkt vor dem Fraktal und blickt von einer gewissen Höhe aus auf das Fraktal. Die Höhe wird durch den Vertikalwinkel festgelegt, der dem Breitengrad der Weltkugel entspricht.

Licht

- Falls angewählt, ist eine Lichtquelle vorhanden, die unendlich weit entfernt ist. Die Position der Lichtquelle wird durch den Horizontal- und den

Vertikalwinkel

definiert, die dem Längen- bzw. Breitengrad auf der Erdkugel entsprechen.

Angenommen,

Amerika liegt bei 0, dann wäre Europa ungefähr bei H-Winkel 90, Japan ungefähr bei -90. Falls nicht angewählt, werden

die Originalfarben zur Einfärbung herangezogen.

Intensität

Diffuse

Umgebung

Reflektion

- Die Helligkeit einer Fläche wird durch diese 4 Werte festgelegt:

'Intensität' bestimmt die Intensität der Lichtquelle, so daß man nun das langsame Einschalten der Lichtquelle durch eine Animation

berechnen kann. Der Wert muß zwischen 0 und 1 liegen. 'Umgebung' ist eine Zahl zwischen 0 und 1 und bestimmt, wieviel Licht auf jede Fläche fällt, Betonung

auf -jede-

unabhängig davon, ob nun das Licht darauf scheint oder nicht. Hier einen Wert von 1 einzutragen, ist sinnlos, da dann jede Fläche mit einer Helligkeit von 1 (volle Lichtstärke) gezeichnet würde.

'Diffuse' bestimmt, in welchem Verhältnis das Licht von der Lichtquelle zu dem reflektierten Licht steht.

Ein Wert von 0.8 heißt, daß 80% der Intensität einer Fläche durch den Winkel, in dem die Lichtquelle darauf scheint, bestimmt wird und 20% der Intensität einer Fläche durch den Winkel, den der Vektor des reflektierten Lichtes mit dem Sichtvektor des Beobachters einschließt. Allgemein definiert er, ob das 3D-Bild durch reflektiertes Licht oder durch Lichtquellenlicht sichtbar wird.

'Reflektion' bestimmt, wie stark die Flächen das Licht reflektieren. 1 heißt stark, 2 weniger stark, 0.5 sehr stark, etc.

GridX und GridY

Nur bei Projektionsart = Projektion...

- Hier kann man die Auflösung in X- und in Y-Richtung festlegen. Eine geringere Auflösung hat eine Verschnellerung des Zeichenvorganges zur Folge...

1.50 2.3 Fraktale --- 2.3.11 3D-Ansichten

2.3.11.3 3D-Parameterwindow 2

DeltaX/Y

- Das Objekt selber wird um den Nullpunkt herum gezeichnet. Um es zu verschieben, dienen diese beiden Schieberegler.

DeltaZ

- Hiermit wird das Objekt in der Höhe verschoben.

Invers

- Invertiert alle Höhen. Aus dem ehemaligen Mandelbrotberg wird ein Mandelbrottal.

Autoadjust

- Versucht, das Fraktal so zu verschieben und zu strecken, daß es gut ins Window paßt.

FrontMult

BackMult

- Bestimmt, mit welchen Zahlen die Höhenwerte vorne bzw. hinten noch multipliziert werden. Normalerweise wählt man beide Werte gleich. Will man jedoch dem Objekt mehr

Plastizität verleihen, empfiehlt es sich, BackMult etwas größer zu wählen.

Die Werte in der Mitte werden mit $(\text{FrontMult} + \text{BackMult}) / 2$ multipliziert, es geht also linear von vorne nach hinten.

Steigung

- Dieser Wert bestimmt, wie steil die Hänge bzw. Täler des Bildes sein sollen.

Kleinere Werte ==> weniger steil

Größere Werte ==> steiler

Die Funktion lautet exakt:

$x^{(1/\text{Steigung})}$
 ,wobei x die Höhe bezeichnet, d.h. $\text{Steigung}=2 \implies x^{0.5}$, also Wurzel aus x .

YStretch

Dieser Wert bestimmt den Multiplikationsfaktor für die y -Richtung des 3D-Objektes. Diese Richtung geht sozusagen nach 'hinten'. Falls das 3D-Bild zu 'kurz' erscheint \leftrightarrow
 ist dieser Wert eben größer als 1 zu wählen.

Wasser

Plateau

Mit diesen Werten kann man Höhen abschneiden. Ist eine Höhe kleiner als der durch \leftrightarrow
 'Wasser'
 eingestellte Wert, dann wird diese Höhe auf 'Wasser' gesetzt. Ist eine Höhe größer \leftrightarrow
 als
 'Plateau', dann wird die Höhe auf 'Plateau' gesetzt. Die Farbe dieser Regionen wird ganz natürlich durch den Winkel bestimmt, in dem das Licht einfällt.

1.51 2.3 Fraktale --- 2.3.11 3D-Ansichten

2.3.11.4 3D-Parameterwindow 3

zu benutzende Farben

Hier stellt man die Farben ein, die zur Einfärbung benutzt werden sollen. Die \leftrightarrow
 erste 3D-Farbe wird
 als 'schwarz' angesehen, die letzte 3D-Farbe als 'weiß' und entsprechend \leftrightarrow
 eingefärbt, d.h. je nachdem,
 wie stark das Licht auf die Fläche scheint, wird eine Farbe aus dem Bereich von ' \leftrightarrow
 erste 3D-Farbe'
 bis 'letzte 3D-Farbe' genommen.

Hintergrund

Zu Beginn des Zeichenvorganges wird mit dieser Farbe das Window gelöscht. Falls \leftrightarrow
 Flächen freibleiben, so
 erscheinen sie eben in dieser Farbe.

Dithering

Es sind 3 Modi wählbar. Der erste, gar kein dithern, der zweite, in dem versucht \leftrightarrow
 wird, mittels
 Dithering die doppelte Anzahl an Farben zu simulieren und der dritte, in dem \leftrightarrow
 versucht wird, durch Dithering die
 4-fache Anzahl an Farben zu simulieren.

ExtBuffer

Falls ein Fraktal von 2D nach 3D transformiert wird, werden natürlich bloß alle
 im Puffer verfügbaren Werte transformiert. Aber das resultiert \leftrightarrow
 höchstwahrscheinlich
 in einem Bild, das "vorne" bzw. "hinten" abgeschnitten wirkt. 'ExtBuffer' erlaubt \leftrightarrow
 es
 nun, die vertikale Puffergröße zu erhöhen. Die Angabe erfolgt in Prozent der \leftrightarrow
 Originalpuffergröße,
 d.h. $\text{ExtBuffer}=30$ heißt, daß ein Puffer derselben Breite, aber der Höhe ' \leftrightarrow
 Normalhöhe + 30% des Normalen'
 angelegt und auch berechnet wird. Für ein 2D-Fraktal ist all das absolut sinnlos, \leftrightarrow
 hier sollte man

dem Speicher zuliebe ExtBuffer auf 0 setzen. Für 3D-Transformationen ist ein Wert so zwischen 30 und 50 sinnvoll.

Sättigung

Helligkeit

Diese Werte haben nur beim Abspeichern des 3D-Bildes in 24 Bit eine Auswirkung. Im 3D-Puffer stehen die Originalfarben und die Stärke des Lichtes, das an die Stelle fällt. Wie nun diese beiden Informationen gekoppelt werden, um zur 24-Bit-Farbe zu gelangen, wird durch diese beiden Gadgets bestimmt.

Sättigung: Bestimmt, wie stark der Sättigungsgrad der Originalfarbe durch das Licht beeinflusst wird. Wertebereich reicht von 0 (Originalsättigung, Licht spielt gar keine Rolle) bis 100 (Originalsättigung wird komplett durch die Lichtintensität ersetzt)

Helligkeit: Siehe bei Sättigung, bloß wird nun die Helligkeit der Farbe evtl. durch das Licht beeinflusst.

Normalerweise setzt man Helligkeit auf 100, Sättigung auf 0, d.h. die Originalfarbe wird benutzt, ins FSH-Farbmodell umgerechnet, die Helligkeit durch das Licht, das an exakt diese Stelle scheint, ersetzt, das ganze nach RGB-Format und dann abgespeichert. Wer sich für die Wirkung des ganzen interessiert, kann ja eine 24-Bit-3D-Animation darüber berechnen lassen ;-)

Riemann

(nicht implementiert!)

- Fraktale sind fast immer Repräsentationen dafür, was für verschiedene komplexe Zahlenwerte

passiert. Jeder Bildschirmpunkt entspricht einfach einem komplexen Zahlenwert.

Riemann nun hat sich eine andere Darstellung für die komplexen Zahlen einfallen lassen.

Normalerweise spricht man ja von der komplexen Zahlenebene, die eine Achse ist die reelle Achse, die

andere die Imaginäre Achse. Riemann hat diese Ebene nun auf eine Kugel transformiert.

Diese Kugel hat den Radius 1 und berührt mit ihrem Südpol (also grob gesprochen halt unten...)

die komplexe Zahlenebene im Ursprung. Wenn man nun einen Punkt auf der komplexen Zahlenebene

hat, dann zeichnet man einfach die Verbindungsgerade zwischen diesem Punkt und dem Nordpol der

Kugel. Da, wo diese Gerade die Kugel durchstößt, ist der Punkt auf der Kugeloberfläche zu

finden, der dem komplexen Zahlenwert entspricht.

Alles schön und gut, jedem festen komplexen Zahlenwert wird so genau ein Punkt auf der Kugeloberfläche zugeordnet, bis auf eine Ausnahme: Dem unendlich fernen Punkt. Genauer gesagt, gibt es keinen unendlich fernen Punkt, es gibt unendlich

viele

von dieser Sorte. Bloß allen ist eines gemeinsam: Sie werden allesamt auf den Nordpol der Kugel abgebildet. Dies ist vermutlich auch der Grund dafür, daß in der Fraktaltheorie der unendlich ferne Punkt fast immer als möglicher Attraktor in Erwägung gezogen wird. Man beachte den Singular (Punkt). Es sind unendlich viele

solche da, bloß auf der Riemannkugel sind sie halt alle beisammen am Nordpol.

Nun, mein Programm kann bei der 3D-Darstellung auch die Riemann-Kugel zeichnen. Allerdings ist es nicht die Standardriemannkugel, denn die berührt die komplexe Zahlenebene ja im Ursprung. Dieses Programm allerdings berechnet einen graphisch günstigen Berührungspunkt selbst. Der Radius ist auch einstellbar, bei der Originalkugel wäre er ja fest 1. Man kann sich wohl denken, wie es aussehen würde, wenn man in ein Fraktal hineinzoomt und dann sich die Riemann-Darstellung betrachtet. Für Leute mit weniger Fantasie: Viel zu wenig Werte sind da, hat man also die ganze Kugel vor Augen, so ist darauf evtl. nur ein Punkt sichtbar, eben der, der für den Ausschnitt des gerade berechneten Fraktals steht. Kommt man auf die Idee, sich die Kugel doch näher heranzuholen, sprich hinzuzoomen, dann ist klar, daß die lokale Krümmung der Kugel immer weiter abnimmt, und wenn man das Fraktal dnn vor Augen hat, wird es sich nicht merklich von der 2D-Darstellung unterscheiden. Den Radius einzustellen, kann man auch dem Programm überlassen. Einfach auf 'Radius setzen' klicken und das Programm berechnet ihn so, daß das Fraktal schön auf einer Kugel Platz hat.

1.52 2.3 Fraktale --- 2.3.12 Das Wizardwindow

2.3.12 Das Wizardwindow

Dieses Window gibt es auch für das Animationssystem. Allerdings sind dann nur Informationen verfügbar, keine Analyse oder Höhenanzeige.

Abhängig vom Fraktaltyp enthält dieses Window unterschiedliche Daten.

Information

Hier finden sich die Dauer der Berechnung, evtl. wie oft hineingezoomt wurde, die Anzahl an Bytes, die für den Puffer in Anspruch genommen werden, einen Fortschrittindikator, die benutzte Arithmetik und vielleicht noch andere derartigen Werte.

Analyse

Hier wird das Profil der Iterationswerte angezeigt, d.h. wie viele Punkte von welcher Iterationstiefe gibt es. Dies ist sehr nützlich, da man z.B. sieht, ob evtl. die maximale Anzahl in Iterationen zu hoch ist. Auch gut, um optimale Werte für die Farbverteilung zu erreichen, da man so leichter entscheiden kann, wie die Farbverteilung auszusehen hat. Es sollte nach Möglichkeit immer so sein, daß die Steigung des Graphen der Farbverteilung an den Stellen, an denen es viele Punkte mit ähnlichen Iterationswerten gibt, hoch ist, sonst niedrig. Nochmal zum Mitdenken: Da, wo im Analyse-Graph die 'Hügel' sind, sollte sich die Farbverteilungsfunktion stark ändern, d.h. stark wachsen oder

stark fallen (ja, meist wohl wachsen...).

Höhen

Tja, eines der größten Probleme mit ChaosPro ist es, geeignete Werte für die 3D-Transformation zu finden. Wählt man diesen Modus an, so zeichnet das Programm einen Graphen, um die gerade aktuelle Transformationsfunktion zu zeichnen. So sieht man gleich, welche Auswirkungen die Parameter haben (und vor allem in welchen Größenordnungen).

1.53 2.3 Fraktale --- 2.3.13 Das Kommentarwindow

2.3.13 Das Kommentarwindow

Na ja, manchmal ist es nützlich, Tips zu Fraktalen zu geben. Manche Leute sehen es auch gern, wenn ihr Name in den von ihnen gefundenen Fraktalen auftaucht. Gut, man sieht ja, wofür dieses Window und die Gadgets darin gut sind, also spare ich mir lange Erklärungen.

1.54 2.4 Menüs

2.4 Die Menüs

2.4.1 Systemmenü

Weitere Menüpunkte:

Fraktalmenü

Fraktalwindows

Windows

Extras

Daten laden/speichern

- Damit werden die Fraktaldateien aus einer Datei geladen/in einer Datei gespeichert. Wird die Datei im Verzeichnis ChaosPro/FractPic abgespeichert, so wird sie beim nächsten

Start des Programms automatisch geladen und im Gadget für die Fraktalbilder angezeigt.

Wird eine Datei nachträglich geladen, so wird sie nach dem Ladevorgang in das Listview eingehängt.

Das Programm merkt beim Abspeichern, ob das 3D-Window offen ist und speichert in diesem

Fall die Fraktaldatei und die Information, daß es sich um ein 3D-Fraktal handelt. Wenn

zukünftig dann auf das Gadget 'Bild berechnen' geklickt wird, werden gleich beide Windows,

das 2D- und das 3D-Window geöffnet.

Seit V2.0 ist ChaosPro in der Lage, auch Mand2000, Mandelmania und kompatible Datenfiles zu

laden.

ChaosPro speichert nun auch die Datenfiles als Chunk in jedem IFF-ILBM Bild mit, ←
so
daß man diese Bilder laden und weiter mit dem Fraktal arbeiten kann.

Bild laden

Sofern man über die datatypes.library verfügt (ab OS3.0 im Lieferumfang dabei), ←
kann man mit ChaosPro
auch Bilder laden. Wählt man diesen Menüpunkt an, so erscheint ein Requester, die ←
datatypes.library wird geöffnet und
das gewählte Bild ins Fraktalwindow des gerade aktuellen Fraktals geladen. Dieses ←
Window
wird auch noch in der Größe verändert, so daß das zu ladende Bild hineinpaßt. ←
Durch die Verwendung der datatypes.library habe
ich mir sehr viel Zeit gespart, als Nebeneffekt kann man damit natürlich jeden ←
Bildtyp, für den ein DataType existiert,
einladen. Ein weiterer Menüpunkt existiert noch, der mit diesem eng verbunden ist. ←
Es ist dies
der Punkt in 'Einstellungen/ Verschiedenes/ Bild Remap'. Falls dieser Punkt mit ←
einem Haken versehen ist,
so wird die Farbpalette des Bildes beim Einladen der aktuellen Bildschirmpalette ←
so
gut wie möglich angepaßt. Der Ladevorgang dauert dann natürlich etwas länger. Ich ←
selber wähle diesen Punkt nicht an, da ich sowieso nur Fraktale einlade und da ←
gewisse
'Falschfarben' gar nicht auffallen.

Bild speichern/ins Clipboard

- Damit kann das Bild als IFF-Bild abgespeichert werden. Bei den Fraktaltypen ←
Juliamenge/ Mandelbrot/ Plasma/ Lyapunov-Raum ist es möglich, das 2D-Bild
in einer beliebigen Farbtiefe bis 256 Farben unabhängig von der Hardware bzw. vom ←
Betriebs (bildschirm-) modus
abzuspeichern. In welcher Tiefe Sie es haben wollen, können Sie nach der Wahl des
Dateinamens bestimmen. Damit können dann Besitzer alter Amigas mit ECS-Chipsatz ←
die Bilder in 256 Farben
speichern und sie von einem anderen Programm in den HAM6-Modus umrechnen lassen.
Außerdem wurde der Wunsch nach 24 Bit geäußert. Auch das ist hier möglich. Dabei
stehen 2 Möglichkeiten zur Auswahl: 24 (Screen) und 24 (256)
Ersteres speichert das Fraktal so, wie es auf dem Screen angezeigt wird, aber eben ←
mit
24 Bit. Das Programm berechnet eigentlich mehr Farben als es anzeigen kann und ←
rechnet dann auf die Anzahl der
zur Verfügung stehenden Farben herunter. Falls nun auf dem Screen Flächen ←
derselben Farbe vorkommen, ist es möglich, daß
das Programm gezwungen war, hier verschiedenen eng beieinander liegenden Werten ←
dieselbe
Farbe zuzuweisen. Wenn das 24 Bit-Bild abgespeichert wird, wird die benutzte ←
Farbpalette 'aufgeblasen'
und so die korrekte Farbe abgespeichert, was die Fläche zu einem (mehr oder ←
weniger) sanften
Farbübergang werden läßt. Beachten Sie bitte, daß bei Verwendung der ←
Iterationseinfärbung bei Julia/Mandel das
24Bit-Bild optisch identisch ist mit dem Bild, das Sie erhalten, wenn Sie es in ←
256 Farben speichern.
Wollen Sie hier wirklich das, was Sie wahrscheinlich wollen werden, nämlich viele, ←
viele Farben, sanfte
Farbübergänge etc., dann müssen Sie die CPM-Einfärbung einstellen.

Die andere Wahlmöglichkeit, 24 (256), nimmt die komplette Farbpalette von 256 Farben, bläst sie auf und speichert so das Bild ab. Das entspricht der Anwahl von 8 Planes, nur daß hier eben Flächen durch Farbübergänge ersetzt werden.

Damit keine Mißverständnisse aufkommen: Wenn ein Fraktal keine Flächen enthält, wird das 24 Bit-Fraktal nicht (merkbar) anders aussehen, als eines mit z.B. 64 Farben. Also ein Plasma-Fraktal mit hoher Granulation als 24 Bit abzuspeichern ist sinnlos. Es muß schon Flächen aufweisen.

Bei den Bifurkationsdiagrammen und den dynamischen Systemen und auch bei den 3D-Ansichten der Fraktale besteht all diese Möglichkeit nicht. Das Fraktal wird einfach in der berechneten Tiefe, also in der Screentiefe, abgespeichert. Ausnahme: Wenn das Fraktal einen 3D-Puffer bekommen hat, dann kann es in 24 Bit Farbtiefe abgespeichert werden.

Drucken

- Ich habe mir längere Zeit überlegt, ob ich eine Druckroutine einbauen soll. Letztendlich bin ich zu dem Entschluß gekommen, keine Druckoption einzubauen, da ich denke, daß meine Druckroutine viel zu schlecht wäre. Fraktale will man stets in hervorragender Qualität ausdrucken, allein dazu ist schon ein Colormanagement System von großem Vorteil. Doch das bietet das AmigaOS nicht. Aus diesem Grund habe ich einfach die Server-Funktion des Programmes 'Studio' von Wolf Faust benutzt. Dadurch steht mir eine exzellente Druckoption zur Verfügung, ein Colormanagement System, so daß die Fraktale auf dem Papier genauso gut aussehen wie auf dem Bildschirm, viele Parameter, eine benutzerfreundliche Bedienung, etc. Natürlich hat das den Nachteil, daß man Studio erst mal besitzen muß, um drucken zu können.

Anmerkung:

Bei Anwahl dieses Menüpunktes speichert ChaosPro zuerst das Fraktal in das Temporärverzeichnis ab, daher wird der Benutzer zuerst einmal mit Fragen zur Screentiefe etc. konfrontiert. Das Temporärverzeichnis ist 'ChaosPro:', kann aber durch das ToolType 'Virtual=<Dir>' überschrieben werden. Nach dem Abspeichervorgang wird dann Studio asynchron aufgerufen, das die erzeugte Datei nach dem Druck wieder löscht.

SystemInfo

- Dieser Punkt gibt Informationen über den Prozessor/ Koprozessor/ Grafikkchips/ Speicher/ Priorität aus

Über ChaosPro

- Informationen über den Autor, die Programmversion

Ende

- Erklärung nötig?

1.55 2.4 Menüs

2.4.2 Fraktalmenü

Weitere Menüpunkte:

Systemmenü

Fraktalwindows

Windows

Extras

Juliamenge

Mandelbrot

Bifurkation

Dynamisches System

Plasma

Lyapunov-Raum

Diffusion

IFS

L-System

Brownsche Bewegung

- Damit werden neue Fraktale dieses Typs hinzugefügt. Sie werden mit den Standardwerten für den jeweiligen Typ initialisiert.

Defaultwerte

- Falls man sich mit den Parameterwerten vertan hat und gar kein sinnvolles Fraktal mehr erscheinen will, so kann man diesen Punkt anwählen. Er setzt die Daten des Fraktals auf die im Programm gespeicherten Defaultwerte.

Windowgröße ändern

- Dieser Menüpunkt zeigt die aktuelle Windowgröße und erlaubt es auch, sie zu ändern.

Die maximale Größe des Windows wird nicht nur durch den Screen bestimmt, sondern ist natürlich

die Screengröße minus dem Rahmen des Windows. Diese Funktion kann derzeit ein normales Window

nicht in ein Backdropwindow umwandeln bzw. umgekehrt. Derzeit finde ich, daß dieses Verhalten

eher ein Feature als ein Fehler ist, da es nun möglich ist, ein Backdropwindow zu machen (Menüpunkt

'Windowtyp ändern') und anschließend dieses Backdropwindow in der Größe zu ändern. (Natürlich mag es

etwas seltsam erscheinen, wenn ein Window ohne Rahmen irgendwo auf dem Screen existiert...)

Zoomen

- Skalieren hinein - Skalieren heraus

- Skalieren hinein entspricht einem Doppelklick in die Mitte des 2D-Fraktalwindows. 'Skalieren heraus' macht das

Gegenteil inklusive der Skalierung, etc.

- Rahmen hinein - Rahmen heraus

Nun kann man irgendwo in das 2D-Fraktalwindow klicken (Maustaste darf wieder ↔ losgelassen werden), kann dann einen Rahmen ziehen und dann nochmal klicken. Wenn hineingezoomt werden ↔ soll, wird der durch den Rahmen bestimmten Ausschnitt zur vollen Windowgröße vergrößert. Beim ↔ Hinauszoomen wird das volle Window in den Rahmen hineinprojiziert.

Undo/Redo

- Unbeschränktes Undo/Redo für jedes Fraktal. Grundsätzlich wird ein Puffer von 10 KB pro Fraktal angelegt, in dem die alten Werte aufbewahrt werden. Mir hat das bisher in jedem Fall gereicht...

Verschieben nach...

- verschiebt das Fraktal, selbes kann mit den Cursortasten einfacher erreicht werden

Proportion

- falls das Fraktal stark verzerrt ist, stimmt das Verhältnis der Ausschnittwerte überhaupt nicht mehr mit dem Verhältnis der Breite zur Höhe des Windows überein. Mit diesem Punkt kann das Verhältnis wiederhergestellt werden.

Berechnung

- Stopp/Fortsetzen: Da das Programm im Multitasking läuft, kommt es des öfteren ↔ vor, daß man mehrere Fraktale berechnen läßt aber nun eben nur eines so schnell wie möglich fertig ↔ haben will. Mit diesem Punkt stoppt man die Berechnung des aktiven Fraktals. Der Task wird ↔ dabei schlafen gelegt und kann durch Anwahl des Punktes 'Fortsetzen' wieder aufgeweckt ↔ werden. - Neustart: Erzwingt eine Neuberechnung des Fraktals.

Picasso

Picasso schließen

- Sollten diese Punkt funktionieren, so würde er das 24 Bit-Bild direkt auf der ↔ Picasso Grafikkarte von Village Tronic anzeigen. (nicht implementiert)

EGS ansteuern

- Dieser Punkt öffnet ein Window auf dem EGS-Default-Screen und zeichnet darin das ↔ Fraktal in 24 Bit, sofern dies möglich ist. (nicht implementiert)

Starte virtuell

Dieser Menüpunkt läßt sich nur bei Julia- /Mandelbrot\mengen anwählen. Im ↔ Parameterwindow Nr.3 existieren 4 Gadgets, in denen man die entsprechenden Werte wie Größe/Tiefe/3D eintragen kann. Durch ↔ Anwählen dieses Menüpunktes schaltet der Subtask in den virtuellen Modus und startet somit die virtuelle Berechnung des ↔ Fraktals direkt auf die Festplatte. Näheres dazu findet sich im Parameterwindow ↔ Nr.3, Punkt

virtuelles Berechnen
Export/Reflections

Tja, dies ist der 1. Versuch eines Exports in ein Raytracing-Format. Am besten, ↔
 man testet
 es einfach aus. Es erscheinen Requester, die nach der Auflösung des Objekts fragen ↔
 , da man ja die
 volle Auflösung nicht verwenden kann, da zumindest ReflectionsV2.0 noch eine ↔
 Beschränkung der Anzahl an
 Dreiecken auf 32767 hat. Es werden 2 Dateien abgespeichert, eine Datei mit Namen ↔
 *.obj, und eine mit Namen
 *.mat.

1.56 2.4 Menüs

2.4.3 Fraktalwindows

Weitere Menüpunkte:

Systemmenü

Fraktalmenü

Windows

Extras

Parameter 1...

Parameter 2...

Parameter 3...

- Hiermit wird das Parameterwindow 1/2/3 geöffnet/geschlossen. Welche Parameter einstellbar sind und was sie bedeuten, kann in den entsprechenden Kapiteln nachgelesen werden

3D-Parameter 1

3D-Parameter 2

3D-Parameter 3

- Die 3D-Parameterwindows werden hiermit geöffnet/geschlossen. Näheres ist in ↔
 Kapitel 2.3.11
 nachzulesen.

Datenwindow

- Einige Fraktaltypen haben Datenwindows zur Verfügung. Wenn man dieses Window öffnet und mit der Maus über das Fraktal hinwegfährt, so werden hier die Daten entsprechend der Mauszeigerposition angezeigt.

Wizardwindow

- Öffnet/Schließt das

Wizardwindow

Kommentarwindow

- Öffnet/Schließt das

Kommentarwindow

Formelwindow

- Öffnet/Schließt das Formelwindow.

Zeige Position

- Juliamenge, Mandelbrotmenge, Bifurkation, Lyapunov-Raum haben einen 2D- ↔
 Ausschnitt. Es kann

hineingezoomt werden. Und plötzlich weiß man nicht mehr ganz genau, wo man denn ↵
 nun hineingezoomt
 ist. Die Parameter in den Parameterwindows geben zwar Auskunft, aber ganz klar ist
 es trotzdem nicht. Dieser Menüpunkt öffnet für dieses Fraktal ein Window. In ↵
 diesem
 Window werden alle Fraktale desselben Typs angezeigt. Wählt man einen Eintrag an, ↵
 so
 wird im Fraktalwindow der Bereich, in dem das angewählte Fraktal gezeichnet wurde, ↵
 als Rahmen
 gezeichnet.

Juliaparameter setzen

- Dieser Menüpunkt ist nur bei Mandelbrotmengen anwählbar. Er öffnet ein Window, ↵
 in dem
 sämtliche Juliamengen angezeigt werden. Wählt man eine Juliamenge an, so wird in ↵
 der Mandelbrotmenge
 der Parameterwert der Juliamenge als Kreuz angezeigt. Dieses Kreuz kann man ↵
 natürlich verschieben
 und so den Parameterwert der Juliamenge ändern. Es wurde bereits erwähnt, daß ↵
 interessante Juliamengen
 am Rande der Mandelbrotmenge zu finden sind. Doch wo ist der Rand der ↵
 Mandelbrotmenge, wenn man nur
 eine komplexe Zahl vor sich hat? Mit Hilfe des Kreuzes weiß man es ganz genau.
 Aber Achtung: Julia- und Mandelbrotmenge sollten schon zusammenpassen, also sollte ↵
 die
 Juliamenge z.B. vom Subtyp z^2+c sein, dann sollte auch die Mandelbrotmenge vom ↵
 Subtyp z^2+c sein.
 Ansonsten ist dieses Gerede von interessanten Juliamengen am Rande der ↵
 Mandelbrotmenge ziemlicher
 Schwachfug...

Colormapping Window

- Öffnet/Schließt das
 Colormapping Window
 Windowtyp als Backdrop/normales Window
 - Evtl. will man den gesamten Platz auf dem Screen für ein Fraktal ausnutzen. Wenn
 ein Windowrahmen vorhanden ist, so ist das aber nicht möglich. Deshalb kann man ↵
 ein
 Window zum Backdrop-Window erklären. Dabei wird automatisch das Window geschlossen ↵
 ,
 der Rahmen, die Titelleiste, die Systemgadgets entfernt, das Window auf volle ↵
 Größe gebracht
 und erneut als Backdrop geöffnet. Das funktioniert mit dem 2D- bzw. 3D-Window.
 Aber Achtung: Da man nun plötzlich ohne Depth-gadget da steht, kann man die ↵
 Reihenfolge
 der Windows nicht mehr ändern. Also über Sinn und Unsinn von mehreren Backdrop- ↵
 Windows
 läßt sich streiten...

1.57 2.4 Menüs

2.4.4 Windows

Weitere Menüpunkte:

- Systemmenü
- Fraktalmenü
- Fraktalwindows
- Extras
- Formeleditor für Julia/Mandel
- Formeleditor für LSystem
- Formeleditor für IFS
- Farbpalettenwindow
- Palettenbearbeitung
- Animation 1&2
- CycleControl
- AnimWizard
- Output
- Benutzerwindow

Hier kann ich wieder auf das Kapitel 2.2 verweisen. Dort ist alles Wissenswerte

nachzulesen.

Vielleicht noch eines: Über das Menü können nur die ersten 4 benutzerdefinierten Windows geöffnet werden. Wer mehr hat, muß diese dann über den Arexx-Port öffnen.

1.58 2.4 Menüs

2.4.5 Extras

Weitere Menüpunkte:

- Systemmenü
- Fraktalmenü
- Fraktalwindows
- Windows
- Hilfe

- zeigt das Inhaltsverzeichnis der Online-Hilfe an. Wer schneller zum Ziel gelangen will, benutzt:

1. Menuhelp

Ihr wählt einen Menüpunkt an, lasst aber die rechte Maustaste nicht los, also ←
eigentlich nicht anwählen, sondern den Menüpunkt
'unterlegen'. Dann drückt ihr die Help-Taste. Das Betriebssystem meldet nun meinem
Programm, daß der Benutzer Hilfe zu einem Menüpunkt will und sagt mir auch, zu ←
welchem.
Das Programm kann nun geeignet reagieren.

2. Selbstgestricktes Gadgethelp

Mein Programm hält umfangreiche Datenlisten aufrecht, in denen die Positionen und ←
Ausmaße
der Gadgets festgehalten werden. Wird die Help-Taste gedrückt, so sucht mein ←
Programm
nach dem Gadget, über dem der Mauszeiger steht und handelt entsprechend. Ist der
Mauszeiger über keinem Gadget, so zeigt er den Hilfe-Text an, der als Default für
dieses Window vorgesehen ist.

globaler Stop

globales Fortsetzen

- Stoppt die Berechnung aller Fraktale auf einen Schlag, falls ein anderes ←
Programm
sämtliche Rechenzeit braucht...
'Fortsetzen' weckt alle Fraktalberechnungstasks auf, falls und nur falls sie ←
schlafen.

Colorcycling

- An

Damit kann man Colorcycling an-/ausschalten

- Aufwärts

Hiermit kann eingestellt werden, ob aufwärts (zu höheren Farbnummern hin) oder ←
abwärts

gecycled werden soll.

- Schneller/Langsam

Geschwindigkeit des Colorcyclings. Zum Colorcyclen wird ein separater Task ←
geschaffen.

Im Interrupt darf ja laut RKM: Libraries nichts mit den Farbtabelle angestellt
werden.

Da nun das Taskswitching nicht beliebig oft auftritt, sondern nur ca 50 mal in
der Sekunde, so ist dadurch natürlich die Geschwindigkeit des Colorcyclings ←
begrenzt.

Wer mit 256 Farben arbeitet, muß natürlich auch bedenken, daß es fürs ←
Betriessystem

ein ganz schöner Happen ist, 256 Farbwerte neu einzustellen, die Copperlisten ←
hierfür neu zu berechnen, zusammenzulinken mit
den anderen und alles anzuzeigen...

Taskpriorität

- Damit wird die Taskpriorität des Haupttasks und des Colorcycling-Tasks ←
eingestellt.

Sämtliche Berechnungstasks laufen mit einer um 1 niedrigeren Priorität. Default- ←
Wert

ist 0, Fraktalberechnung also bei -1, was einen ungestört auf der Workbench oder
mit irgendeinem anderen Programm weiterarbeiten läßt, während ein Fraktal ←
berechnet

wird.

Window auslagern

- auf Fraktalscreen / auf Parameterscreen / auf Workbench / auf Publicscreen
Damit kann jedes Window bis auf das 2D- und 3D-Fraktalwindow auf den betreffenden anderen Screen ausgelagert werden. Sehr sinnvoll, da der Platz auf einem noch so großen

Screen recht schnell knapp wird. Auch für den Speicher gut, denn ein Parameter-Window auf einem 256 Farb-Fraktalscreen schluckt auch entsprechend mehr Speicher als wenn es auf einer 4-Farb-Workbench liegt. Durch das Preferences-Programm kann eingestellt werden, auf welchem Screen die Windows defaultmäßig aufmachen.

Windowpositionen

Hier können für die Windows, die mehrmals zu öffnen sind, die Screens angegeben werden, auf denen die Windows sich dann defaultmäßig öffnen.

Verschiedenes/Bild remap

Ist für 'Bild laden' zuständig und bestimmt, ob das Bild, das geladen wird, an die aktuelle Farbpalette angepaßt wird oder nicht.

Verschiedenes/Koordinaten anzeigen

Falls angewählt, wird die Screenleiste dazu benutzt, die Koordinate in der komplexen Zahlenebene anzuzeigen, die der Mauszeigerposition bzgl. des aktuellen Fraktals entspricht.

Da dies recht viel Rechenzeit schluckt, kann man es abschalten, da sowieso dieser Wert auch

im Datenwindow zur Verfügung steht. Das ToolType 'CoordShow' bestimmt, ob diese Anzeige defaultmäßig an- oder ausgeschaltet ist.

Verschiedenes/GuiFaktorX

Verschiedenes/GuiFaktorY

In ChaosPro sind alle Gadgets und Windows nicht in Pixeleinheiten, sondern in Fonteinheiten angegeben.

Es existiert eine globale Routine zum Windowöffnen mit Gadgets, Borders, etc. Diese Routine

rechnet dann die Fontgrößen einfach in Pixelgrößen um. Dadurch wird das gesamte GUI von

ChaosPro fontsensitiv. Da es aber viele verschiedene Fonts gibt und ich selber das GUI so gestaltet habe, daß es bei mir am besten aussieht, also mit dem Font XHelvetica 11 von Martin

Huttenloher (Bestandteil von MagicWB), kommt es des öfteren vor, daß bei anderen Benutzern das GUI etwas arg vermüllt aussieht,

z.B. wenn man topaz 8 benutzt.

Diese beiden Faktoren nun erlauben es, das gesamte GUI in x- bzw. in y-Richtung zu strecken. Der Effekt ist

derselbe, als wenn die Windows ein Sizingadget hätten, das man benutzen könnte. Bloß haben die Windows

eben kein Sizingadget, da es relativ viel Arbeit wäre, dies einzubauen. Diese beiden Faktoren wählt man eben passend

zum Font oder passend dem persönlichen Geschmack und speichert dannach das Config-File, damit die Änderungen auch beim

nächsten Start von ChaosPro erhalten bleiben. Diese beiden Faktoren werden nämlich im Config-File eingetragen...

Verschiedenes/Publicscreen

Hiermit kann man den Publicscreen per Namen auswählen, auf den dann Windows ausgelagert werden können. Es empfiehlt sich, dannach die Konfiguration zu

speichern, um nicht immer wieder den Namen anzugeben.

Verschiedenes/Genauigkeit

Hier kann gewählt werden, ab welcher Zoomtiefe der Wechsel zur nächsthöheren Genauigkeit gewechselt wird. Es stehen derzeit 4 unterschiedliche Arithmetiken zur Verfügung: FPU (ungefähr 64 Bit), 96 Bit, 128 Bit und 160 Bit. Folglich existieren 3 Stellen, an denen zur nächstgenaueren Arithmetik gewechselt wird. Defaultmäßig ist es so, daß ab einer Zoomtiefe von 53 die 96 Bit-Arithmetik benutzt wird, ab einer Zoomtiefe von 86 die 128 Bit-Arithmetik und ab einer Zoomtiefe von 116 die 160 Bit-Arithmetik. Dies kann mit diesem Menüpunkt geändert werden. Falls im Fraktal seltsame offensichtlich falsche Pixel auftreten, so ist dies ein Anzeichen dafür, daß die Genauigkeit nicht mehr ausreicht. In diesem Fall sollte dieser Menüpunkt aufgerufen werden und der Zeitpunkt, ab dem die nächsthöhere Arithmetik benutzt wird, vorverlegt werden.

Screenmodes einstellen

Font einstellen

- Ist hier eine umfangreiche Erklärung nötig?

Falls sie angewählt werden, werden alle möglichen Windows geschlossen, die Werte geändert und dann alles erneut wieder geöffnet.

Die minimalen Screentiefen sind:

1. Für den Parameterscreen: 1
2. Für den Fraktalscreen: 3
3. Für den Colorscreen: 4

- Für den Font gibt es keine Einschränkung. Aber: Falls ein Window nicht mehr auf dem Screen Platz hat, ist halt ein kleinerer Font zu wählen. Das betrifft vor allem Leute, die das Programm in einer Auflösung von 640x200 bzw. 640x256 betreiben. Topaz 8 ist da fast schon zu groß.

Bildschirmfarben

Hier können die ersten 4 Farben des Bildschirms geändert werden. Dies sind die Farben,

die von ChaosPro nicht verändert werden, damit alle Bildelemente auch sichtbar

bleiben. Leider erscheinen in dem Palettenrequester alle Farben, die der Bildschirm

anbietet. Es können jedoch nur die ersten 4 Farben geändert werden, die anderen wird

ChaosPro bei nächster Gelegenheit gleich wieder ändern. Nach dem Einstellen sollte das Config-File gespeichert werden, wenn die Änderungen dauerhaft bleiben sollen.

Konfiguration laden/speichern

- Hiermit können verschiedene Konfigurationen geladen bzw. gespeichert werden.

In diesen Dateien sind die änderbaren Daten wie Screenmodus, Font, geöffnete Windows,

Positionen dieser Windows, etc. gespeichert.

1.59 2.4 Menüs

2.4.6 Benutzerdefiniertes Menü

Hierfür ist wieder eine ASCII-Datei im Verzeichnis ChaosPro/Prefs/ mit Namen Menu.asc erforderlich. Diese muß dann mit dem Preferences-Programm übersetzt werden. ↵

Dabei wird die Datei Menu.prefs generiert.

Der Aufbau der Datei ist:

```
MENU <Menütext> <Tastaturshortcut> <Arexx-Script>
ITEM <Itemtext> <Tastaturshortcut> <Arexx-Script>
...
ITEM <Itemtext> <Tastaturshortcut> <Arexx-Script>
MENU <Menütext> <Tastaturshortcut> <Arexx-Script>
...
END <Menütext> <Tastaturshortcut> <Arexx-Script>
```

Dabei muß bei MENU und END natürlich nichts Sinnvolles bei Tastaturshortcut bzw. Arexx-Script angegeben werden. ↵

für Menütext kann auch die Konstante BARLABEL eingetragen werden. Sie erzeugt einen Trennstrich. ↵

Für Tastaturshortcut kann auch NONE angegeben werden, falls eben keiner gewünscht wird.

Beispielsweise sieht es so aus:

```
MENU MenüNr1 NONE dummy.rexx
ITEM Data B Daten.rexx
ITEM BARLABEL NONE dummy.rexx
ITEM Nochwas C Nochwas.rexx
MENU MenüNr2 NONE dummy.rexx
ITEM InOut D InOut.rexx
END BARLABEL NONE dummy.rexx
```

1.60 2.5 Programmverzeichnisse

2.5 Programmverzeichnisse

Basisverzeichnis, vom dem aus das Programm die verschiedenen Unter\verzeichnisse anspricht, ist stets das logische Verzeichnis 'ChaosPro:'. Ist dieses beim Programmstart vorhanden, ist alles ok. Ist es nicht vorhanden, so versucht das Programm herauszufinden, von welchem Verzeichnis aus es gestartet wurde. Anschließend erzeugt es den Assign mit Namen ChaosPro: auf das so gefundene Verzeichnis. ↵

ChaosPro:libs/

hier befinden sich alle Bibliotheken, die das Programm benötigt. Es ist nicht

nötig, sie nach libs: zu kopieren, ganz davon abgesehen, daß sie m.E. dort absolut
nichts
zu suchen haben. Es sind Bibliotheken für ChaosPro, also sollen sie auch da sein,
wo sich
das Hauptprogramm befindet. Daß sich das ganze Programm dadurch so ganz nebenbei
sehr
einfach wieder deinstallieren läßt, ist volle Absicht...

ChaosPro:Guides/
Hier befindet sich die Dokumentation

ChaosPro:Prefs/
Alle Einstellungen des Programms sind dort zu finden

ChaosPro:Palette/
ChaosPro braucht mindestens eine Farbpalette in diesem Verzeichnis, sonst bricht
es mit
einer Fehlermeldung ab. Zum Programmstart liest es selbstständig alle Dateien aus
diesem Verzeichnis ein, filtert eventuell vorhandene Farbinformationen heraus und
verwirft den Rest. Es ist daher auch möglich, hier ganze Bilder zu lagern. In
diesem Fall werden eben die Farbtabelle der Bilder eingeladen.

ChaosPro:Catalogs/
Hier finden sich alle Übersetzungskataloge

ChaosPro:FractPic/
Beim Start des Programms wird dieses Verzeichnis auf mit dem Programm
abgespeicherte Daten von Fraktalen durchsucht.
Alle gefundenen Fraktale werden geladen. Benötigt ein Fraktal eine nicht im
Programm eingebaute Formel, so lädt es sie
ebenso ein (sie wurde in der gleichen Datei in einem anderen Chunk gespeichert)
und fügt sie der
internen Formelliste hinzu.

ChaosPro:Anims/
ChaosPro:AnimData/
In diesen Verzeichnissen versucht ChaosPro zuallererst, Animationen bzw.
Animationsdaten
abzuspeichern.

ChaosPro:Formula/
Dieses Direktory wird beim Programmstart gescannt. Alle dort gefundenen Formeln,
die mit dem Programm
abgespeichert wurden, werden geladen und können verwendet werden.

1.61 2.6 Preferencesprogramm

2.6 Preferencesprogramm

Zur Übersetzung von speziellen Files existiert ein externes Preferences\programm.
Folgende Optionen bieten es an:

Compile Userwindows

Benutzerdefinierte Windows

werden vorerst in einer ASCII-Datei definiert.

Dannach muß diese noch in ein Format konvertiert werden, das möglichst leicht vom Hauptprogramm verstanden wird. Dazu dient dieses Gadget. Einfach drauf klicken und mit Hilfe der ASCII-Datei ChaosPro: Prefs/ Windows.asc wird die Datei ChaosPro/ Prefs/ Windows.prefs geschaffen.

Compile Usermenus

Benutzerdefinierte Menüs

werden ebenfalls vorerst in einer ASCII-Datei festgelegt, die anschließend noch für das Hauptprogramm konvertiert werden muß, was dieses Gadget erledigt. Es erzeugt aus der Datei ChaosPro/ Prefs/ Menu.asc die Datei ChaosPro/ Prefs/ Menu.prefs.

Die Online-Hilfe

Da das Programm auf beliebigen Screens laufen kann, war es nötig, auch die Online-Hilfe

entsprechend anzupassen. Denn was nützt es, wenn die Online-Hilfe, die typischerweise für einen Screen mit einer Breite von 640 Pixeln formatiert ist, auf einem Screen mit 1024 Pixeln

Breite nur knapp über die Hälfte des Platzes ausnutzt? Was passiert, wenn der Benutzer sich entschließt, die

Online-Hilfe in einer anderen Schriftart anzuzeigen und deshalb die @FONT-Direktive in der Guide-Datei ändert?

Ganz klar, es sieht schrecklich aus, da plötzlich jede Zeile eine andere Breite hat, zumindest wenn man einen Proportional-Font verwendet.

Aus diesem Grund wird auch die Guide-Datei mit Hilfe des Preferences-Programms kompiliert,

um eine optimale Ausnutzung des Platzes zu erreichen und das Aussehen durch Verwendung von Blocksatz

(jede Zeile hat dieselbe Breite) ansprechender zu gestalten.

- GuideWidth

Hiermit legt man fest, in welcher Breite der Guide formatiert werden soll. Da der Windowrahmen

auch noch Platz beansprucht, sind für gewöhnlich von der Screen-Breite 40 Pixel abzuziehen.

- Language

Damit legt man fest, in welcher Sprache die Datei geschaffen werden soll. Dies ist unabhängig davon, ob

das Locale-System installiert ist, denn es existieren (hoffentlich) 2 Guide-Dateien in ChaosPro/Guides, nämlich deutsch.guide

und englisch.guide, die eben entsprechend herangezogen werden.

- Build Guide

Damit wird im Verzeichnis ChaosPro/Guides/ die Guide-Datei ChaosPro.guide aus der Datei deutsch.guide bzw. englisch.guide geschaffen. Die Originaldateien bleiben natürlich erhalten. ChaosPro.guide ist eine absolut normale Guide-Datei, die man sich mit Multiview, Hyper oder AmigaGuide anzeigen lassen kann, die aber eben auf das

richtige Format und Blocksatz gebracht worden ist. Diese Operation kann recht lange

dauern (auf meinem A4000/040 ca 60 Sekunden!), da 1. die Guide-Datei recht groß ist

und 2. ich mir keine Mühe gegeben habe, die Routine schnell zu machen, da ich 3. denke daß man diese Operation nicht besonders oft ausführen läßt.

Abhängig vom Font kann es vorkommen, daß plötzlich ein ominöser Requester aufgeht ←
mit der Fehlermeldung
"Failed to create a line". Hier kann man nur 'Ok' klicken, und genau das ist auch ←
zu tun. Mit dieser
Meldung hat es folgendes auf sich: Bei Erstellen des Blocksatzes kann es vorkommen ←
, daß ein Wort länger als
eine gesamte Zeile ist und auch nicht getrennt werden kann. Zumindest weiß der ←
Computer nicht, wo er es
trennen soll. In diesem Fall wird diese Meldung gebracht. Doch denke ich, daß ←
diese Meldung in der
Regel mit einem Programmfehler zusammenhängt. Bei einer daraufhin gestarteten ←
Bugsuche habe ich allerdings
keinen Fehler festgestellt, dort waren stets Wörter (es waren irgendwelche ←
komplexe mathematischen Ausdrücke)
wirklich zu lang. Das heißt für die Benutzer: Warten, bis ich da voll durchblicke ←
und evtl. eine genauere Fehlerbeschreibung
(vielleicht sogar mit der Nummer der Zeile, in der der Fehler aufgetreten ist...) ←
einbaue.

Um den Font der Guide-Datei oder sogar ganze Textpassagen zu ändern, kann man ←
natürlich einfach
die Datei ChaosPro.guide ändern. Doch das ist so ziemlich das Dümme, was man ←
machen kann.
Denn dann steht man ohne Blocksatz da und außerdem sind die Änderungen beim ←
nächsten Anklicken von
'Build Guide' wieder verloren...
Will man also etwas ändern, so ist dazu deutsch.guide bzw. englisch.guide zu ←
ändern.
Dies sind ebenfalls normale Guide-Dateien für AmigaGuide bzw. MultiView oder auch ←
Hyper, aber eben noch nicht im richtigen
Format. Dort kann man in Zeile 6 die @FONT-Direktive ändern und einen anderen ←
Fontnamen
sowie eine andere Fontgröße einstellen.
Will man Testpassagen ändern, so ist darauf zu achten, daß ein Absatz durch eine
Zeile mit weniger als 76 Zeichen beendet wird. Zeilen mit 76 Zeichen oder mehr
werden mit der nächsten Zeile, diese evtl. mit der übernächsten etc. zu einem ←
Absatz
verbunden. Falls also nach der Konvertierung plötzlich ein neuer 'Absatz' mitten ←
im Absatz
begonnen wird, hat die letzte Zeile weniger als 76 Zeichen...
Falls ein Wort zu lang ist, sieht es evtl. im konvertierten Guide nicht so toll ←
aus,
da um des Blocksatzes willen zu viele Spaces eingefügt werden mußten. Deshalb ←
besteht
die Möglichkeit, der Konvertierungsroutine mitzuteilen, wo ein zu langes Wort ←
getrennt
werden kann. Dies geschieht mit dem Backslash-Zeichen links neben der Backspace- ←
Taste
rechts oben auf der Tastatur. Also einfach an den Trennstellen dieses Zeichen ←
einfügen.
Wird das Wort getrennt, so wird ein '-'-Zeichen eingefügt und getrennt, ansonsten ←
wird
der Backslash einfach entfernt.

1.62 2.7 Problemecke

2.7 Problemecke

1. Problem

Gelegentlich hängt das gesamte System, wenn ich die Online-Hilfe aufrufe.

Lösung:

Keine. Ich habe keine Ahnung, woran das liegen kann. Doch da ich weiß, daß der AmigaGuide wirklich noch nicht ausgereift ist (man denke daran, daß Commodore beinahe zeitgleich die lokalisierte Workbench herausbrachte und deshalb das SPEAK ←
:-Device

nicht mehr mitlieferte, aber nicht daran dachte, daß dieselbe Online-Hilfe in ←
einer anderen

Sprache andere Zeilennummerierungen hat, was die Lokalisierung einer Online-Hilfe ←
stark erschwert!), denke ich, daß der Fehler
mehr im AmigaGuide steckt.

2. Problem

Wie kann ich die Größe des AmigaGuide-Fensters dauerhaft einstellen?

Lösung:

Durch Auswahl des Menüpunktes 'Vorgaben speichern'...

3. Problem

Bei Druck auf die Help-Taste kommt kein AmigaGuide-Fenster, die Online-Hilfe ←
funktioniert also

nicht.

Lösung:

1. Evtl. ist das AmigaGuide-System nicht richtig installiert, dann sollten Sie ←
sich die offizielle Distribution
des AmigaGuide-Paketes (verfügbar bei PD-Händlern oder auf dem Aminet) besorgen und ←
es mittels des
Installerscripts installieren.

2. Evtl. ist ChaosPro.guide oder ChaosPro.Topics nicht vorhanden. In diesem Fall ←
sollten

Sie das Preferences-Programm starten und auf 'Build Guide' klicken. Mehr ←
Informationen hierfür finden sich im Kapitel über das

Preferences-Programm

3. Es existiert ein ToolType, mit dem man die Online-Hilfe ←
abschalten kann, was Speicherplatz
spart. Es heißt sinnigerweise NO_AGUIDE und, falls angegeben, wird die Online- ←
Hilfe nicht
initialisiert.

4. Problem

ChaosPro stürzt beim Start ab.

Lösung 1:

Hier gibt es viele Möglichkeiten. Eine Möglichkeit, die in 60% aller Fälle auftrat ←

war, daß die Benutzer FastMathV40.5 installiert hatten, und in dieser ist ein ←
schlimmer

Fehler. Erst die neuere Version V40.6 besitzt diesen Fehler nicht mehr. Also:
Upgraden...

Lösung 2:

Benutzen Sie PointerEyes in einer Version kleiner als V4.1? In diesem Fall gibt es eine Betriebssystemversion, unter der ChaosPro beim Aufruf von OpenWorkBench hängenbleibt. Hier also scheint ChaosPro einfach zu schlafen und nichts mehr zu tun, man kann ganz normal mit dem Amiga weiterarbeiten. Seit PointerEyesV4.1 ist dieser Fehler behoben.

Lösung 2:

Haben Sie das Icon von ChaosPro geändert? In diesem Fall empfehle ich nachzusehen, ob der Stack groß genug ist. 20000 Bytes sollten genügen...

1.63 2.8 Sonstiges Erwähnenswertes

2.8 Sonstiges Erwähnenswertes

Wie gibt man Zahlen ein? Ganz klar, mit Integergadgets. Aber was ist mit Fließkommazahlen? So etwas gibt es bisher noch nicht im Betriebssystem. Deshalb mußte ich mir meine eigene Hook-Routine schreiben, um aus einem String-Gadget ein Float-Gadget zu machen. In diesem Float-Gadgets werden alle sinnlosen Tastendrucke herausgefiltert. Gewisse andere Tastaturkombinationen wie z.B. RAmiga+X, um das Eingabefeld zu löschen, bewirken entsprechend Sinnvolles für ein Floatgadget, nämlich den Eintrag '+0.0'. Um das Vorzeichen einer Zahl zu ändern, ist an irgendeiner Stelle das '+'- oder '-'-Zeichen zu drücken. Das Vorzeichen an allererster Stelle ändert sich sogleich. Um den Dezimalpunkt an eine andere Stelle zu setzen, drückt man an der betreffenden Position einfach die '.'-Taste. Dadurch wird ein evtl. vorhandener Dezimalpunkt gelöscht und an die neue Stelle gesetzt. Zahlen in Exponential Schreibweise sind in der vorliegenden Version nicht möglich.

Wer das Programm bereits benutzt hat, dem wird aufgefallen sein, daß die im PicTask-Window angewählten Einträge sich ständig ändern, wenn ein anderes Window aktiviert wird. Das ist Absicht. Das aktive Window bestimmt den aktiven Task. Immer wenn ein neues Window aktiv wird, sucht das Programm nach dem Fraktaltask, zu dem das Window gehört und erklärt ihn zum aktiven Task. Außerdem geht es noch das gesamte Menüsystem durch und verbietet die Auswahl von unter Umständen für den jeweiligen Fraktaltyp nicht ausführbaren Menüpunkten, wie z.B. das Zeigen des Juliaparameters für Bifurkationsdiagramme.

Wenn man nicht mehr weiß, welcher Fraktaltyp nun überhaupt vorliegt, bzw. zu welchem Task ein Window gehört, dann sollte man das Window aktivieren und sich den Screentitel

ansehen. Dort steht noch einmal kurz, welchen Namen das Fraktal hat und welcher Typ es ist. ↵
 Gut z.B. für Julia- und Mandelbrotmengen, wenn man hineingezoomt hat. Dann fällt es recht ↵
 schwer, die beiden Typen auseinanderzuhalten.

Ich habe mich bemüht, das Programm style-guide konform zu programmieren, da ich ein ↵
 Herz für Grafik\karten\besitzer habe. Aus diesem Grund ist mein Programm auch nicht ↵
 unbedingt das schnellste. Vor allem die Benutzer von Mand2000 werden die Geschwindigkeit ↵
 des Hineinzoomens als nicht besonders hoch ansehen.

Manch einer will, daß beim Programmstart sofort das Palettenwindow aufgeht oder ↵
 irgendetwas anderes
 gleich mal passiert. Diese Möglichkeit bietet das Programm über den AREXX-Port.
 Beim Start wird das Arexx-Programm ChaosPro/ Rexx/ ChaosProInit ausgeführt. Dort
 könnt ihr gleich mal ein Bild berechnen lassen.

Tja, so gut wie alle Programme wollen einen logischen Assign zum Arbeiten. Mein ↵
 Programm macht
 hier keine Ausnahme. Doch habe ich einen anderen Weg als (fast) alle anderen ↵
 Programme beschritten:
 Das Programm sucht beim Programmstart nach dem logischen Assign 'ChaosPro:'. Ist ↵
 er vorhanden,
 dann sucht er ausgehend von dort in den verschiedenen Unterverzeichnissen z.B. ↵
 ChaosPro:Prefs, ChaosPro:Palette, ChaosPro/Formula etc.
 nach den jeweiligen Dateien. Ist dieser Assign aber nicht vorhanden, dann wird er ↵
 vom Programm kreierte und bei Programmende wieder entfernt.
 Das heißt nun, daß z.B. für die Rexx-Scripts als Pfadangabe stets z.B. ChaosPro: ↵
 Rexx/ Default.rexx verwendet werden kann, da zur Laufzeit
 immer ein Assign mit Namen 'ChaosPro:' vorhanden ist.

1.64 2.9 Tooltypes

2.9 Tooltypes

ChaosPro kann ganz normal vom CLI aus gestartet werden. Die Tooltypes werden ohne ↵
 Änderung
 dort auch akzeptiert, z.B. ist folgendes möglich:

```
ChaosPro NOJOYSTICK CHUNKYMODE NO_AGUIDE VIRTUAL=dh2: PALETTES=dh3:Palettes
```

ChaosPro ist es vollkommen egal, ob die Parameter über das CLI oder über das ↵
 Piktogramm als Tooltypes kommen. Es erledigt sowieso alles der Startupcode
 des SAS C-Compilers...

Das Programm unterstützt derzeit die folgenden ToolTypes:

NOJOYSTICK

Dieses Tooltype verbietet die Benutzung eines Joysticks in Port zum Zoomen und ↵
 Herumfahren
 im Fraktal. Dies war nötig, da evtl. ein Dongle (z.B. der von Real3D) in diesem ↵
 Port stecken kann,

was einige Verwirrung stiften kann. Falls das Fraktal, sobald das Window aktiviert wird, unkontrolliert herumfährt, so ist evtl. so ein Dongle schuld und man sollte dieses ToolType angeben.

CHUNKYMODE

Dieses ToolType bestimmt die Routine, die zum Skalieren des Fraktals bei einem Doppelklick benutzt wird. Normalerweise wird dies auf die folgende Art erledigt: Der Windowinhalt wird mittels ReadPixelFormat8 in einen Puffer ausgelesen, dieser dann skaliert, das Ergebnis mit einer eigenen Routine ins Planeformat umgewandelt und anschließend mittels ClipBlit ins Window kopiert. Wenn nun jemand eine Grafikkarte besitzt und der Screen, der für das Fraktalprogramm benutzt wird, eigentlich ein ChunkyScreen ist, so wird natürlich in dem Augenblick, in dem mein Programm ClipBlit aufruft, alles wieder ins Chunkyformat umgewandelt... Wenn man dieses ToolType angibt, dann benutzt mein Programm nach dem Skalieren einfach WritePixelFormat8 und überläßt es damit dem Grafikkartentreiber, die Werte, die evtl. schon im richtigen Format, eben dem Chunkyformat, vorliegen, in den Screen zu kopieren. Bemerkung: Das Programm schreibt nirgendwo direkt in die Planes von Windows oder gar dem Screen. Wenn alle Programme so geschrieben wären, dann gäbe es heute Grafikkartentreiber, die um einiges schneller wären.

COLORWHEEL

Dieses ToolType bestimmt, ob beim Paletteneditieren das Farbrad angezeigt werden soll. Um die korrekte Darstellung des Farbrads zu ermöglichen, sind natürlich viele Farben dafür nötig, genauer gesagt, werden für die Darstellung die Hälfte der Farben auf dem Screen benutzt. Aus diesem Grund kann man mit diesem ToolType explizit angeben, ob man das Farbrad auch wirklich will. Wenn Sie alle Farben auf dem Screen für das Anzeigen der Farben der Palette benutzen wollen, dann geben Sie dieses ToolType nicht an.

BUILTIN

Falls angegeben, wird die eingebaute Sprache, also englisch, benutzt. Ansonsten natürlich die Sprache, die im Locale-System voreingestellt ist. Dieses ToolType wird eigentlich nur von mir zum Testen benötigt.

BACKFILL

Falls gesetzt, dann wird das Window vor dem Hinzufügen durch ein Raster ausgefüllt. Dadurch hebt es sich besser vom Screen ab und sieht evtl. schöner aus.

PGA_NEWLOOK

Falls gesetzt, werden die Proportional-Gadgets im Aussehen verschönert. GadTools unterstützt dies jedoch an sich nicht. Das Bit wird also von Hand gesetzt. Es funktioniert, ist jedoch ein undokumentiertes Feature, das nicht zwingend funktionieren muß, bzw. sogar zu Fehlverhalten führen kann. Bisher wurden jedoch keinerlei Nebenwirkungen bekannt.

NO_EGS

Falls nicht angegeben, kann das EGS-System, sofern korrekt installiert, durch einen Menüpunkt angesprochen werden. Dieser Menüpunkt zeichnet dann das aktuelle Fraktal auf dem EGS-Default-Screen.

NO_AGUIDE

Falls nicht angegeben, wird die amigaguide.library geöffnet. Somit ist eine kontextsensitive Online-Hilfe verfügbar. Evtl. empfiehlt es sich, aus Speichergründen darauf zu verzichten.

NO_REXX

Falls nicht angegeben, wird der Arexx-Port des Programms initialisiert und steht zur Benutzung bereit. Wer diesen nicht will und den Speicher für etwas anderes besser nutzen kann, der gebe dieses ToolType an.

PICTURES=<Dir>

Dieses Tooltype legt das Verzeichnis für Bilder fest. Default ist 'ChaosPro: Pictures'. Da jeder ein anderes Verzeichnis für seine Bilder hat, kann man dies hier einstellen.

EXPORT=<Dir>

Dieses Tooltype legt das Verzeichnis für die exportierten Objekte im Reflections-Format fest. Default ist 'Szenen:'.

PALETTES=<Dir>

Dieses Tooltype legt das Verzeichnis für die Farbpaletten fest. Default ist 'ChaosPro:Palette'. Da jeder ein anderes Verzeichnis für seine Paletten hat, kann man dies hier einstellen. Trotzdem wird ChaosPro beim Start stets immer nur die Paletten aus dem Verzeichnis ChaosPro:Palette laden. Dieses Tooltype legt lediglich das Default-Verzeichnis für den Palettenfilerequester fest.

VIRTUAL=<Dir>

Dieses Tooltype legt das Verzeichnis fest, in dem das Temporärfile für den virtuellen Puffer angelegt wird. Es sollte dies nach Möglichkeit auf einer Partition mit viel Platz sein. Default ist das Verzeichnis 'ChaosPro:'.

ANIMS=<Dir>

Dieses Tooltype legt das Verzeichnis für die Animationen fest. Default ist 'ChaosPro:Anims'. Da jeder ein anderes Verzeichnis für seine Animationen hat, kann man dies hier

einstellen.

STARTPRI=<Priority>

Dieses Tooltype legt die Priorität von ChaosPro beim Programmstart fest. Das Hauptprogramm wird mit dieser Priorität laufen, alle Berechnungstasks mit StartPri-1. Default ist StartPri=0.

AUTOSAVE

Falls angegeben, wird beim Beenden des Programms automatisch der Menüpunkt 'Konfiguration speichern' ausgeführt und die Datei 'ChaosPro.config' ins Verzeichnis 'ChaosPro:Prefs' geschrieben, so daß beim nächsten Programmstart die Windows automatisch an der alten (gewohnten) Stelle erscheinen.

DEBUG

Falls dieses Tooltype angegeben wird, informiert ChaosPro in der Initialisierungsprozedur den Benutzer über den Verlauf der Initialisierung. Dies ist besonders dann nützlich, wenn das Programm sich nicht starten läßt. In diesem Fall sieht man nämlich in etwa, wo das Programm Probleme hat. Einerseits nützlich für den Benutzer, andererseits bitte ich, bei Startproblemen mir die Ausgabe des Programmes zukommen zu lassen, da die Information, daß sich das Programm nicht starten läßt, einfach nicht ausreichend ist.

EHB_HAM

Falls dieses Tooltype angegeben ist, werden im Screenmoderequester auch diese speziellen Modi EHB und HAM angezeigt. ChaosPro unterstützt diese Modi nicht, aber funktioniert natürlich. Den Sinn des Tooltypes habe ich noch nicht ganz herausgefunden, ich habe es lediglich auf Benutzerwunsch hin implementiert...

SAVEGFX

Falls angegeben, wird ein Forbid/Permit-Paar um jeden Aufruf von WritePixelLine8 plazierte. Falls Sie einen OCS/ECS/AGA-Amiga haben, empfehle ich, dieses Tooltype anzugeben, da es scheint, als ob im OS ein Fehler ist: Man berechne mehrere Fraktale zur selben Zeit, auf meinem A4000/040 kann ich ungefähr 3 oder 4 Stück berechnen. Dann plötzlich hängt das gesamte System, der Mauszeiger bewegt sich nicht mehr, es geht gar nichts mehr. Es kommt kein Enforcerhit, kein Mungwall report, gar nichts. Falls Sie eine Grafikkarte mit Intuitionemulator haben, können Sie versuchen, dieses Tooltype nicht anzugeben. Vielleicht wurde dieser Fehler bei der Treibersoftware erkannt und behoben.

WIZARDOPEN

Falls angegeben öffnet sich das Wizard window automatisch, wenn ein Fraktal berechnet wird. Dies gilt auch für das AnimWizardwindow.

OUTPUTLINES=xxx

OUTPUTLINES=xxx gibt die Anzahl zu allozierender Zeilen für das Outputwindow an

.

REQHEIGHT=xxx

Dieses Tooltype läßt Sie die Höhe aller File-, Font- und Screenmoderequester angeben

.

COORDSHOW

Bestimmt, ob die (komplexen) Koordinaten des aktuellen Fraktals in der Screenleiste angezeigt werden sollen.

DEFFRACTSCREEN=<PubScreenName>

DEFPARMSCREEN=<PubScreenName>

Durch diese Tooltypes lassen sich 2 Publicscreens angeben, die ChaosPro benutzen soll, statt selber

den Fraktalscreen und (vielleicht) auch noch den Parameterscreen zu öffnen.

Hiermit können Sie sich Ihren

Liblingspublicscreen definieren, zum Beispiel mit einem Hintergrundbild mittels des Programms 'ScreenWizard'

und ChaosPro wird dann diesen Publicscreen benutzen.

ACHTUNG: SIE HABEN SICHERZUSTELLEN, DAß DER FRAKTALSCREEN IMMER EINE TIEFE VON MINDESTENS 3 PLANES, ALSO 8 FARBEN

HAT, WENN SIE DIESES FEATURE BENUTZEN.

1.65 2.10 Rechtliches

2.10 Rechtliches

Im Laufe der Nutzung und Programmierung dieses Programms passierte es 2 mal, daß meine Festplatte durch einen Programmfehler einen Defekt aufwies. Ich bin nicht perfekt.

Deshalb sind mit einer Wahrscheinlichkeit von 100% Fehler in diesem Programm, die unter Umständen schlimme Folgen haben können.

Deshalb:

Keine Garantie

Es besteht keine Garantie für dieses Program zusätzlich zur augenblicklich geltenden

Rechtssprechung. Der Copyright-Inhaber und/oder Drittanbieter stellen dieses Programm "so

wie es ist" zur Verfügung, ohne jegliche Garantie, sei es eine Garantie ausdrücklicher

oder impliziter Art, einschließlich, aber nicht darauf beschränkt, den impliziten Gepflogenheiten

des Handels und der Tauglichkeit für einen bestimmten Zweck. Das gesamte Risiko, das

durch die Benutzung des Programms entsteht, tragen allein Sie. Sollte das Programm fehlerhaft sein, so tragen Sie alle Kosten, die Ihnen durch den/die Fehler

unmittelbar oder

mittelbar entstehen.

Außer die augenblickliche Rechtsprechung fordert es, kann in keinem Fall ein
 Copyright-Inhaber
 oder irgendein Drittanbieter, der dieses Programm wie oben erlaubt weiterverbreitet
 , haftbar
 gemacht werden für irgendwelche Schäden, seien es allgemeine, spezielle, zufällige
 ,
 direkte oder indirekte Schäden, die durch die Benutzung oder die Fehlbenutzung des
 Programms entstehen (eingeschlossen, aber nicht darauf beschränkt, der Verlust von
 Daten,
 die fehlerhafte Berechnung von Daten, Verluste ihrerseits oder von Drittanbietern,
 oder
 die Unfähigkeit dieses Programms, mit einem anderen zusammenzuarbeiten), sogar,
 wenn
 der Copyright-Inhaber oder irgendein Drittanbieter von der Möglichkeit eines
 solchen Schadens
 unterrichtet wurde.

ChaosPro 1995 Martin Pfingstl

ChaosProV2.0 ist Public Domain.

ChaosProV2.0 ist frei vertreibbar

1.66 2.11 Gesucht...

2.11 Gesucht...

Gesucht werden:

1. Übersetzer der Katalogdateien in verschiedene Sprachen, da ich selber nur des Deutschen und Englischen mächtig bin. Es ist ratsam, mich vorher zu fragen, ob
 evtl.
 nicht bereits ein hilfreicher Zeitgenosse bereits beschlossen hat, mir zu helfen.
 Außerdem
 braucht man ja auch die .cd und .ct-Dateien.
2. Falls jemand Vorschläge zur Verbesserung der Dokumentation hat, so kann er es
 gerne
 tun. Falls jemand meint, ein Kapitel im Programm wäre etwas zu kurz und knapp
 geraten,
 darf er mir ruhig eine verbesserte Version zuschicken. Dies betrifft natürlich
 auch
 die theoretischen Teile, also die Kapitel, in denen ich versucht habe, in Kürze
 das Wichtigste an theoretischem Wissen zu
 vermitteln, aufarbeiten.
3. Was ich mir von Anfang an gewünscht habe, war ein Programm, das nicht einfach
 nur Fraktale berechnet, sondern mehr einem Buch über Fraktale gleicht, bloß eben
 mit den interaktiven Möglichkeiten eines Computerprogrammes. Ich stelle mir nun
 folgendes vor: Ein AmigaGuide-Kapitel über Fraktale, in dem die Knöpfe, die der
 Benutzer anklicken kann, keine Links zu Kapiteln darstellen, sondern Arexx-Scripts
 ausführen. Dies geht mittels des Schlüsselwortes 'RX' <Arexx-Script-File> statt '
 LINK'
 <Nodename>. Da nun ChaosPro ein Arexx-Interface hat und außerdem der AmigaGuide
 bei

- Druck auf die Help-Taste asynchron innerhalb von ChaosPro läuft, sind alle Voraussetzungen geschaffen, über ein AmigaGuide-Dokument durch Knöpfe ChaosPro fernzusteuern. Wer möchte sich als Autor versuchen? Mir liegt programmieren mehr als Texte schreiben...
4. Wer hübsche Bilder berechnet hat, der kann mir natürlich die Datenfiles davon schicken. Auch an Farbpaletten bin ich interessiert. Aber denkt nicht daran, die Paletten von FractInt oder aus einem Archiv mit Namen FractInt_Extra (oder ähnlich, in der Dosenwelt wohl eher was in der Form FRAXTRA4.ARJ oder FRAXTRA5.ARJ zu konvertieren - das habe ich bereits getan, und werde es auch tun, falls da noch mehr kommt...)
5. An Verbesserungsvorschlägen leglicher Art bin ich natürlich sehr interessiert. Sei es ein Rechtschreibfehler in der Hilfe-Datei in Zeile xxx oder aber der Vorschlag, das Programm netzwerkfähig (Stichwort: verteiltes Rechnen) zu machen...
6. Ich weiß nicht, wo die Freizeit des letzten Jahres geblieben ist, wer kann helfen ;-)?
7. Wer kann mir bei meinen Mathematik-Aufgaben helfen? Erwünscht sind vor allem Kenntnisse in der Martingaltheorie, partiellen Differentialgleichungen, gute Kenntnisse in Funktionentheorie und Funktionalanalysis.

1.67 2.12 Zur Geschwindigkeit

2.12 Zur Geschwindigkeit

Tja, und nun das leidige Thema der Geschwindigkeit: Viele Leute lobten ChaosPro über alles, nur an der Geschwindigkeit hatten die Leute meistens etwas auszusetzen

Dazu möchte ich nun folgendes sagen:

ChaosPro wurde nicht geschrieben, um alle anderen Fraktalgeneratoren an Geschwindigkeit zu überbieten. Andere Programme sind schneller. Doch andererseits habe ich nicht absichtlich langsame Routinen geschrieben. ChaosPro ist durchaus nicht so langsam, wie manch einer beim ersten Geschwindigkeitsvergleich denken mag.

Was anscheinend einige Leute nicht ganz verstanden haben, ist, daß die Parametereinstellungen die Geschwindigkeit beeinflussen. Beispiel: Außenfärbung=CPM. Was passiert bei einem Vergleich? Die anderen Programme berechnen einfach nach dem Escape Time Algorithmus (der ganz normale, den jeder kennt) zu jedem Bildschirmpixel eine Integerzahl. Dannach schauen sie, ob in einem

kleinen Quadrat nur eine bestimmte Anzahl an Iterationen auftrat. Wenn ja, wird das gesamte Quadrat mit dieser Integerzahl aufgefüllt, was natürlich einen enormen Geschwindigkeitsvorteil bringt, vor allem im Innengebiet. Diese Leute nun denken, daß es keinen großen Unterschied macht, wenn sie mal kurz in ChaosPro auf CPM gehen. Falsch gedacht! In diesem Modus wird zu jedem Pixel auf dem Bildschirm eine reelle Zahl berechnet, und es ist äußerst unwahrscheinlich, daß in einem noch so kleinen Gebiet stets dieselbe reelle Zahl auftritt.

Nichtsdestotrotz war ChaosPro langsamer als andere Programme, auch wenn nur die absoluten Standardparameter eingestellt/benutzt wurden. Da es mich ärgerte, daß sich Leute darüber beschwerten, habe ich mir kurz einen kleinen Trick zur Beschleunigung ausgedacht. Diese Beschleunigungen wirken sich bei den Iterationsberechnungen aus, treten also vor allem dann merklich auf, wenn das Bild lange zur Berechnung braucht. Die Standardmandelbrotmenge wird trotzdem noch langsamer berechnet als mit fast allen anderen Programmen, was mit großer Wahrscheinlichkeit daran liegt, daß ChaosPro systemkonform mittels WritePixelLine8 bzw. ClipBlit zeichnet. Mit diesen Routinen ist es unmöglich, anderen Programmen beim Zeichnen nachzukommen. Zommt man jedoch hinein, so sieht das Ergebnis nicht mehr so schlecht für ChaosPro aus, da dann die Berechnung wesentlich länger dauert als das Zeichnen. Sollte jemand Ideen für schnelle Zeichenroutinen haben, so kann er/sie sie mir natürlich schreiben.

1.68 2.13 Geschichtliches & Änderungen seit V1.0

2.13 Geschichtliches & Änderungen seit V1.0

Geschichtliches

Im März 1993 fing alles an. Ich begann, an einem Fraktalprogramm herumzuschreiben, der Name stand nicht fest, der Funktionsumfang auch nicht. Im wesentlichen sollte es so ähnlich werden wie MandelMania, allerdings leistungsfähiger, mit 24 Bit Fähigkeiten, 3D-Transformationen und vielen Parametern. Tja, im September 1993 war ich dann in etwa fertig, dachte ich zumindest. Zur Sicherheit fragte ich im Usenet noch einmal herum, ob die Anwender noch weitere Vorschläge hätten. Das hätte ich lieber nicht tun sollen. Ein Artikel - der Autor war Holger Lubitz, ich werd's nicht vergessen;-) - schlug vor, ich solle mir Mand2000 ansehen, das gerade als Demo-Version auf das Aminet gelegt wurde. Und das

war es dann.

Nachdem ich dieses Programm gesehen hatte, stand für mich fest, daß ich noch ↵
einmal ganz von vorne
anzufangen hatte. Die ganze Sache zog sich dann bis Anfang Mai 1994 hin. Im ↵
wesentlichen wurde Oktober, Weihnachts\ferien und
März/April herumprogrammiert. Dann dachte ich wieder, daß das Programm fertig sei. ↵
Mittlerweile hieß es
ChaosPro, da der Name FractalPro ja schon vergeben war, etwas besseres fiel mir ↵
leider nicht
ein. Ich beschloß allerdings, erst noch ein paar Anwender Betatester spielen zu ↵
lassen. Und das war gut so.
Es waren, wie sich herausstellte, noch viel zu viele Fehler enthalten. Die ganze ↵
Sache zog sich
dann bis Dezember hin, vor allem da von Mai bis Juli das Sommersemester war, und ↵
dannach das
Vordiplom anstand, auf das ich dann doch lernen mußte. Die letzte Prüfung war Ende ↵
Oktober, anschließend fing das
Wintersemester an.
Im Dezember wurde es mir schließlich zu bunt, man kann ein Programm ewig
zwischen Betatestern hin- und herschicken, Fehler finden sich immer. Da ich nicht ↵
damit rechnete, auf
dem Sharewaremarkt mit ChaosPro Erfolg zu haben, schickte ich es ans Amiga Magazin
als Programm des Monats. Nun gut, das Amiga Magazin erkor schließlich ChaosPro zum ↵
Programm des Monats
in Ausgabe Februar 1995. Damit erlangte ChaosProV1.0, Revision 180, schließlich ↵
zum ersten Mal
Verbreitung. ChaosProV1.0 war von da an Public Domain, lediglich die Verbreitung ↵
beschränkte sich auf die Amiga Magazin
Public Domain Disketten.
Ich programmierte dann noch weiter, beseitigte Fehler, fügte neue Features hinzu, ↵
und legte
schließlich ChaosProV2.0, Revision 73, eine Betaversion, aufs Aminet. Und da bin ↵
ich nun und
beseitige Fehler...

Änderungen seit V1.0

Änderungen seit V2.0

Ich möchte die Änderungen in 2 Teile gliedern:

Erster Teil: Änderungen, die man merkt, neue Features, das, was die Leute ↵
letztendlich
interessiert.

Zweiter Teil: Änderungen, die man eigentlich nicht merkt, z.B. Bugfixes, leicht
geändertes Verhalten, Schönheitskorrekturen oder ähnliches...

a) Änderungen, die man (ich?) merkt

- ChaosPro beschleunigt (Black_Hole, Auflösung 640x495, 1 Pass, etc. davor: ca 8 ↵
min,
jetzt ca. 6 min)

- ChaosPro liest nun auch Mand2000, Mandelmania und kompatible Fraktalfiles. ↵
Mandelmania
konnte ich testen, aber nicht Mand2000, da die Demo-Version kein Speichern
erlaubt.

- Virtuelle Julia- /Mandelbrot\fraktale: Im Parameterwindow Nr. 3 kann man Werte angeben, dannach wählt man den Menüpunkt 'Starte virtuell'.
 - Farbverteilungswindow hinzugefügt
 - graphische Ausgabe der Farbverteilung hinzugefügt.
 - Wizardwindow hinzugefügt
 - Commentwindow für jedes FractPic hinzugefügt
 - FractInt-Paletten (diese *.map-files) werden nun auch von ChaosPro akzeptiert.
 - Nun gibt es 3 verschiedene Dockwindows mit Gadgets.
 - Neues Feature, neues Tooltype: AutoSave, falls angegeben, speichert ChaosPro beim Verlassen die Positionen folgender Windows: Palettenwindow, Anim1,Anim2,CycleControl, PicTask, Dock1, Dock2, Dock3. Zusätzlich wird noch gespeichert, ob dieser Windows geöffnet waren. Beim nächsten Start von ChaosPro werden dann alle geöffneten Windows wieder an der alten Stelle geöffnet.
 - Altes Formeleditorwindow hat ausgedient. In diesem Window kann man nun nur noch die Formel einstellen, zum Ändern der Formeln gibt es ein anderes Window.
 - Der Formeleditor versteht nun auch logische Operatoren: '==' '!=' '>' '<' '>=' '<=' 'T' 'F' '!'.
 - Neuer Modus für Wizardwindow: Falls angewählt, wird im Wizardwindow der Graph der Höhen\trans\formations\funktion gezeichnet.
 - Formeleditor verbessert.
 - ChaosPro kann nun ab OS3.0 Bilder laden. Dazu wird die datatypes.library verwendet.
 - DEM (Distance Estimator Method) eingebaut. Damit wird der Abstand eines Punktes zur Julia/Mandelbrotmenge geschätzt. Ist langsam, da hierbei nicht nur immer $z=z^2+c$, sondern gleichzeitig auch $f=2*z*f+1$ gerechnet werden muß, zusammen mit der Abbruchbedingung verlangsamt das die Sache doch enorm.
 - Neue Fraktaltypen: IFS/ LSystem/ Brown/ Diffusion, wobei IFS und LSystem zusätzlich noch ein Formelwindow haben.
 - Plasma darf nun auch in 3D berechnet werden, was logischerweise zu mehr oder weniger natürlichen Bergen führt.
-

- Lightintensity-Gadget eingebaut, bestimmt die Lichtstärke, so daß man z.B. das Einschalten einer Lichtquelle simulieren kann. ↔

- Reflections-Export hinzugefügt

b) Änderungen, die man nicht so schnell merkt

- Das Auslagern eines Windows funktionierte nicht, wenn das Fraktal nicht berechnet war. ↔

- Wenn der Screenmodus von ChaosPro aus geändert wurde, so bekamen alle Fraktalwindows beim Wiederöffnen die Defaultgröße, das Programm arbeitete intern jedoch mit der alten Größe weiter. ↔

- Wenn ein AnimData-File mit einer eingestellten Tiefe von 24 Bit gespeichert wurde, so konnte es nicht mehr geladen werden. ↔

- CPPrefs-Window konnte leicht zu groß werden, wenn der Font zu groß war, da WA_AutoAdjust nicht angegeben war. ↔

- Stacksizecheck war falsch, falls das Programm vom CLI aus aufgerufen wurde. Das Programm beschwerte sich dann immer über einen zu kleinen Stack. ↔

- Falls ChaosPro.Topics irgendwie verstümmelt ist, d.h. eine größe von 0 Bytes hat, beschwert sich ChaosPro über zu wenig Speicher, weil es bei AllocMem(0, MEMF_CLEAR) einfach immer 0 zurückbekommt. Dieser Fehler könnte auch in Verbindung mit einem Netzwerk auftreten. ↔

- Falls Window zu groß für den Screen, dann hätte das Programm automatisch den in CPPrefs eingestellten 'SmallFont' nehmen müssen. Es berechnete allerdings alle Größen, als ob es 'SmallFont' nehmen würde, und nahm dann trotzdem den normalen Font... ↔

- Tooltypes hinzugefügt: Anims, Pictures, Palettes, Virtual, StartPri, um die entsprechenden Directories etc. anzugeben. ↔

- Maximale Anzahl an Iterationen im Innen-/Außengebiet aus dem Datenwindow entfernt

.

- Das Formeltextgadget im Formelwindow war nicht ans Listview angehängt.

- Neuen Farbverteilungstyp hinzugefügt: Sqrt

- Die Taskpriorität von ChaosPro ist nun im SystemInfo-Requester zu finden.
 - Gadgetleiste mit ein paar Buttongadgets für die Parameterwindows und ein paar andere ausgewählte Operationen. Wer will, kann mir ja für mehr Operationen kleine Bildchen malen (Größe: 32x24, in 2 Planes), die ich dann auch noch einbaue...ich bin kein Künstler... ←
 - Programm gab beim Screenöffnen sowohl SA_DClip als auch SA_Overscan an, obwohl das nicht erlaubt ist. ←
Nun werden die Standardoverscanmodi korrekt unterstützt. (Vielleicht ist das das Problem mit der Piccolo?) ←
 - Beim Fehlschlagen des Screenöffnens wird nun eine möglichst exakte Fehlermeldung zurückgegeben.
 - Falls Fraktal 'verdreht' war, gab es falsche Werte bei 'zoomed in x times' an.
 - Programm löschte bei einer Iterationsänderung nur die ersten 65535 Elemente des Puffers.
 - Bei Änderung der Iterationstiefe wurde das Fraktal nur neu anhand der Pufferdaten gezeichnet, und nicht der Puffer selbst neu überdacht. ←
 - Formelparser mochte '-' nicht. Für ihn war es fast immer ein unaeres '-'.
 - Abbruchbedingung Nr 3 ausgebaut (benutzerdefinierter Punkt als Abbruchattraktor) . Wird nun für Formeln verwendet... ←
 - Parameterwindow Nr. 1 hatte die Gadgets für den 1. Parameter im Falle der Mandelbrotmenge disabled. Bei Formeln ist nun jedoch keine Unterscheidung zwischen Julia- /Mandelbrot\mengen möglich ←
...
 - Neues Tooltype: DEBUG, falls angegeben, gibt das Programm beim Initialisieren Meldungen über den Ablauf aus. ←
 - DEM Gadget ins JulMand Datenwindow eingebaut.
 - Eliminieren hatte einen Fehler.
 - PicTask-Window wird nicht mehr geupdated, wenn sich die Werte nicht geändert haben. Früher wurde es bei jeder Windowaktivierung geupdated, was evtl. zu leichtem 'Flimmern' beim Gadgetaufbau führte, da das alte gelöscht und das neue (dasselbe) dann gezeichnet wurde. ←
 - Palettenänderung funktionierte nicht immer wie erwartet: Grautonpaletten waren aufgrund von Rundungsfehlern leicht falsch, Funktion 'Spread' funktionierte in Verbindung mit Grautönen auch nicht wie gewünscht. ←
-

- Bei Bildern wird nun auch die Colorcycling-Information im CRNG-Chunk ↔ abgespeichert. Jetzt brauche ich nur noch ein Programm, das diesen Chunk auch auswertet...
- Fraktal nach links rausbewegen ging nicht. Sobald es über den linken Windowrand ↔ gezogen wurde, wurde der 'Move'-Vorgang abgebrochen.
- Funktion imag(...) war falsch implementiert.
- Spaces in den Formeln wurden nicht akzeptiert.

Nun die Änderungen, die seit V2.0 durchgeführt wurden:

- DeepZoom hinzugefügt.
 - Fehler in der Routine SetData2Default entfernt, der den Computer zum Absturz ↔ bringen konnte, falls zu viele Daten verändert waren
 - Veränderlicher Rotationspunkt für Julia/Mandel hinzugefügt
.
 - Palette morphing während einer Animation ist nun möglich
.
 - Wenn die Anzahl an Iterationen bei Julia/Mandel ungefähr 40000 oder mehr war, so zeigte der Slider ein etwas seltsames Verhalten, da so ein Slider vom OS aus nur einen Wertebereich von -32768 bis 32767 hat.
 - Wahl der Interpolations\knoten\punkte verbessert. Wenn man z.B. eine Animation ↔ kreiert, in der die Iteration von 10 auf 150 in 15 Schritten verändert werden soll, dann ↔ erhalten die einzelnen Frames die Iterationstiefen 10, 12, 14, 17, 21, 27, 33, 40, 50, 62, ↔ 77, 90, 107, 127, 150. Dies wird nun auch auf den Steigungswert bei 3D-Transformationen angewandt.
 - 5 Listen für Fraktale sind nun im PicTaskwindow möglich.
 - Programm teilte durch 0, falls Wizardwindow offen und das Fraktal nicht ↔ berechnet war.
 - Pseudo-Mandelbrot hinzugefügt: Der erste Parameter, der für Juliamengen wählbar, ↔ für Mandelbrotmengen jedoch nicht wählbar war, darf nun einen Initialisierungswert ↔ enthalten, der dazu führt, daß das Ergebnis eine Pseudo-Mandelbrotmenge wird.
 - ToolType SaveGfx hinzugefügt.
-

- Wenn ChaosPro.topics nicht vorhanden war, so produzierte ChaosPro einen Enforcer Hit. ←
 - VMM sollte jetzt mit ChaosPro funktionieren.
 - Netzwerkberechnung sollte nun möglich sein.
 - Zeichnen von LSystem war falsch. Negative Zeichenkoordinaten wurden nicht richtig gehandhabt. ←
 - 'Zeige Fortschritt' ist nun nicht mehr im Windowtitel des Fraktals, sondern im Wizardwindow des Fraktals. ←
 - Defaultparameter können nun geändert werden.
 - Virtuelles Zeichnen funktionierte nicht, wenn der Puffer 16-Bit-Int war .
 - Neue Arexx-Kommandos:
 - StartAnim
 - AbortAnim
 - LoadAnimData
 - SaveAnimData
 - MoveAnimKey
 - DelAnimKey
 - AddAnimKey
 - GetAnimKey
 - SetAnimData Width/Height/Depth/3DAnim/Buffer/Interpolation/Savemode/Startframe /EndFrame/3DBuffer ←
 - GetAnimData Width/Height/Depth/3DAnim/Buffer/Interpolation/Savemode/Startframe /EndFrame/3DBuffer ←
 - AnimKeyPalette Set/Clear/Get <Name>
 - AnimKeyFrames Set/Get <Name>
 - TimeUnit
 - CalculateFrames
 - FrameToPic
 - Verzeichnis ChaosPro:Docking kann nun alternative Bilder für die Dockwindows enthalten.
 - Neuer Menüpunkt Screencolors erlaubt es nun, die ersten 4 Farben des Screens zu ändern.
 - ChaosPro lädt nun Verzeichnisse erneut ein, wenn der Filerequester erscheint. ←
Vorher wurde der Puffer von ReqTools benutzt, was Verwirrung stiftete.
 - Accuracy setting (Genauigkeits\einstellungen) hinzugefügt .
 - Neues Window:
 - AnimWizard
 - , Wizardwindow für das Animationssystem.
.
-

- Neues ToolType: WizardOpen.
 - Wenn ein AnimKey auf/ab bewegt wurde, so konnte er den sichtbaren Bereich verlassen. Ab OS3.0 ist das nun nicht mehr der Fall.
 - 'Endframe' im Windowtitel einer Animation war leicht verwirrend. Wenn man 200 Frames in der Animation hatte, aber nur bis Frame 100 berechnen ließ, so zeigte es 'xx/200' statt 'xx/100' an.
 - - Outputwindow hinzugefügt.
 - Neues ToolType: OUTPUTLINES=xxx gibt die Anzahl an Ausgabezeilen an, die allokiert werden sollen.
 - Wenn der Screenmodus verändert wurde, während der Änderung ein Fehler auftrat und der Benutzer dann ChaosPro verlassen wollte, so stürzte ChaosPro ab.
 - Requesterfont kann nun geändert werden. Menüpunkt 'Einstellungen/Wähle Font/Requester' erlaubt dies.
 - Neues ToolType: ReqHeight
 - Neues ToolType: CoordShow
 - Neuer Menüpunkt: CoordShow
 - Fehler entfernt: Falls zu wenig Speicher vorhanden war, um ein Fraktal berechnet zu lassen, dann entfernte ChaosPro das Fraktal aus der Liste, in der es noch gar nicht vorhanden war, was zu einem Absturz führen konnte (bzw. führte).
 - Arexx-script Show_all_fractals hinzugefügt. Benutzerwindow 1 enthält einen Eintrag hierfür.
 - Wizardwindow zeigt nun die zur Fraktalberechnung benötigte Zeit an, wobei sie nun auch regelmäßig auf den neuesten Stand gebracht wird.
 - FPU wurde zu spät initialisiert, was zu fehlerhaften Daten bei zu Beginn eingeladenen Fraktalen führen konnte.
 - 2 neue ToolTypes:
DEFFRACTSCREEN=<PubScreenName>
DEFPARMSCREEN=<PubScreenName>
 - Speichern eines Bildes durch Anwahl des entsprechenden Gadgets im Dockwindow nicht...
-

- 3D-Window wurde zu groß geöffnet. Falls das 3D-Bild gespeichert wurde, war es ←
daher stets um ein Pixel zu breit und hatte deshalb eine schwarze
Linie am rechten Rand.
 - Wenn eine Animation im Einzelbildmodus erzeugt wurde, überprüfte ChaosPro nicht ←
korrekt darauf, ob dadurch nicht ein bereits existierendes File überschrieben
wird.
 - AnimWizard-Window wurde nicht wie das normale Wizzardwindow geupdated.
 - Zusätzliches ASCII-Format für Formeln vom Typ Julia/Mandel. Beschreibung findet ←
sich im Kapitel über den

 Formelparser
 -
 Network-Window
 hinzugefügt.
 - Bei Benutzung des Undo/Redo-Mechanismus konnte der Rechner aufgrund einer ←
fehlerhaften Listenverkettung
zum Absturz gebracht werden.
 - Falls sich ein Fraktal mit dem Namen 'StartUp' nach der Initialisierung von ←
ChaosPro in der Fraktalliste
befindet, so wird es sofort automatisch geöffnet und berechnet. Also könnte man z. ←
B. ein Fraktal mit dem
Namen 'StartUp' erzeugen (evtl. ein vorhandenes Fraktal einfach umbenennen), ins ←
Verzeichnis 'ChaosPro:FractPic' legen und dieses Fraktal würde bei jedem Start
von ChaosPro automatisch berechnet.
 - Wenn man den Namen eines AnimKeys änderte, so prüfte ChaosPro nicht, ob der Name ←
evtl.
schon vergeben war.
 - ChaosPro erinnert sich nun an die Größe und Position der Fraktalwindows. Dazu ←
muß das Fraktal erst berechnet, dannach
das Window geschlossen werden. Die Daten stehen nun fest und werden beim ←
Abspeichern des
Fraktalfiles mit abgespeichert.
 - Virtuelle Berechnung: Drehung des Fraktals um 90 Grad funktionierte nicht ←
richtig, evtl. Fehler in
der Seek-Routine bei großen Seekabständen.
 - ChaosPro kam evtl. in den Deadlockzustand, falls das Wizardwindow geöffnet wurde
.
 - Virtuelle Berechnung: Im Fraktalwindow wird nun angezeigt, was gerade passiert
.
 - Virtuelle Berechnung: Anzeige des Fortschritts bei Drehung um 90 Grad wurde ←
falsch
berechnet.
 - Neues Materialfile für Reflections von Rolf Schulz.
-

- 5 unterschiedliche Verzeichnisse werden beim Programmstart gescannt und die ↵
darin enthaltenen Fraktaldateien werden in die 5 verfügbaren
Fraktallisten aufgenommen.
- 5 neue Tooltypes für diese 5 Verzeichnisse:
- DEFFRACTDIR1=<Directory 1> ((Default ist ChaosPro:FractPic))
- DEFFRACTDIR2=<Directory 2> ((Default ist ChaosPro:FractPic_2))
- DEFFRACTDIR3=<Directory 3> ((Default ist ChaosPro:FractPic_3))
- DEFFRACTDIR4=<Directory 4> ((Default ist ChaosPro:FractPic_4))
- DEFFRACTDIR5=<Directory 5> ((Default ist ChaosPro:FractPic_5))

1.69 Humor, die zweite

Dieses Kapitel ist für all diejenigen gedacht, die sich die Mühe machen, die ↵
Dokumentation schön brav
durchzulesen. Dieses Kapitel ist mit voller Absicht nicht im Inhaltsverzeichnis ↵
aufgeführt. Keine Angst, dieses
Kapitel ist auch das einzige versteckte Kapitel...
Noch eines: Komischerweise lesen die Leute, die ChaosPro benutzen, die ↵
Dokumentation nicht durch, doch dieses Kapitel finden
sie. Seltsam...

(Gesammelt aus verschiedenen Newsgroups aus dem Usenet):

69 Woerter = Die Zehn Gebote
270 Woerter = Die amerikanische Unabhaengig\keits\erklaerung
24793 Woerter = die EU-Verordnung ueber den Import von Bueroklammern.

Computer sind nicht intelligent -
Sie denken bloß, daß sie es seien!

Coffee not found, Operator halted!

computo ergo sum

Wen stoert's, daß der Pentium Fehler macht - Hauptsache, er macht sie schnell
!

Ich habe nichts gegen spontane Einfälle, sofern sie wohlüberlegt sind.

Stellenanzeige in der Zeitung: "Gesucht wird: Gleichstellungs\beauftragter"

* Aus der Frankfurter Rundschau:
POLITIKVERDROSSENHEIT
Das für Samstag, 29. April, vorgesehene Jugendforum der
evangelischen Kirche zur ``Politikverdrossenheit Jugendlicher``
muß mangels Teilnehmer ausfallen.

- WindowsError 002: No error . . . yet
- WindowsError 003: Dynamic linking error...Your mistake is now in every file
.
- WindowsError 010: Reserved for future mistakes
- WindowsError 011: Virtual Error, just for fun...
- WindowsError 014: Nonexistent error. This cannot really be happening.
- WindowsError 015: Impossible error. Tell us how it happened

- WindowsError 056: Operator fell asleep while waiting.
- WindowsError 058: Intel inside
- WindowsError 076: Harter Fehler: Sitzen Sie?
- WindowsError 078: BEAMTEN-ERROR:
Ihr Vorgang ist in Bearbeitung, bitte verwenden Sie beim nächsten Systemaufruf ←
Ihre Personalnummer, die
wir Ihnen binnen drei Monaten zuteilen.
- WindowsError 31: Program too small, lost in memory
- WindowsError 33: Too much CPU-power detected! Run Windows4.0 to slow down
- WindowsError 35: CPU just exploding
- WindowsError 34: Unspeakable error in module cthulhids at address R'LYEH.

Student im Examen.

Der Prof fragt: "Sie sehen dieses Skelett hier. Ist das nun maennlich oder ←
weiblich

?"

Student: "--- aeh ? ---"

Prof: "Schauen Sie sich doch mal dieses Becken an. Hat darin ein maennliches oder ←
ein

weibliches Geschlechtsteil geruht?"

Student: "Aeh - ein maennliches."

Prof: "Nunja, gelegentlich!"

Wo liegt der Unterschied zwischen einem Medizin- und einem Physik-Studenten? Wenn ←
beide die Aufgabe bekommen,

ein Telefonbuch auswendig zu lernen, so fragt der Physik-Student: "Warum?", der ←
Medizin-Student: "Bis

wann?"

Stilblueten aus Briefen an die Versicherung

'Ein Stopp-Zeichen erschien ploetzlich, wo noch nie vorher eines war.
Deshalb konnte ich nicht rechtzeitig bremsen.'

'Mein Auto war vollgepackt mit Straeuchern fuer den Garten. Ich hatte
immer Gruen vor Augen...'

'Ein unsichtbares Auto tauchte aus dem Nichts auf, streifte meinen Wagen
und verschwand wieder.'

'Der andere war ueberall auf der Strasse. Ich musste einige Male
schleudern, bevor ich ihn traf.'

'Das andere Auto rammte mich, ohne vorher seine Absicht kundzutun.'

'Als ich am Steuer Einschlief, war ich schon 40 Jahre gefahren.'

'Der alte Knabe konnte sich nicht entscheiden, in welche Richtung er gehen
wollte. Also ueberrollte ich ihn.'

'Die Telefonzelle kam rasend schnell naeher. Ich wollte ihr ausweichen,
aber da hatte sie schon meinen Kotfluegel getroffen.'

'Der Nebel war so stark, dass ich das Schild mit der Geschwindigkeits-
begrenzung nicht sehen konnte.'

'Ein Fussgaenger rammte mich und verschwand wortlos unter dem Wagen.'

Ich fuhr mit meinem Wagen gegen eine Leitplanke, ueberschlug mich und prallte gegen eine Baum. Dann verlor ich die Herrschaft ueber mein Auto.

Mein Dachschaden wurde wie vorgesehen am Freitagmorgen behoben.

An der Kreuzung hatte ich einen unvorhergesehenen Anfall von Farbenblindheit.

Ich habe schon so viele Formulare ausfuellen muessen, dass es mir bald lieber waere, mein Mann waere ueberhaupt nicht gestorben.

Ich habe noch nie Fahrerflucht begannen: Im Gegenteil, ich musste immer Weggetragen werden.

'Der Fussgaenger kollidierte in der Marienstr. mit meinem fahrenden PKW und richtete erheblichen Schaden an.'

'Dann sprang der Verfolgte ins Wasser und tauchte trotz mehrmaliger Aufforderung nicht mehr auf.'

'Wegen meinen Zaehnen konnte ich keine Nacht die Augen zu machen.'

'Mein Sohn hat bei Frau D. ein Zwei Tage altes Schwein gebadet, das dann an den Folgen gestorben ist.'

'JESUS ist in unser Leben getreten. Anbei geben wir die zu unrecht erhaltenen 1000 DM befreit zurueck. IHM sei Ruhm, Preis und Ehre in Ewigkeit! Mit freudigen Gruessen, eine "erloeste" Familie.'

'Ihr Computer hat mir ein Kind zugelegt. Ich habe kein Kind, schon gar nicht von Ihrem Computer.'

'Da mir nicht jeden Tag ein Fussgaenger ins Auto laeuft, habe ich verstaendlicherweise keine Erfahrung in solchen Dingen.'

'Ich fahre mit meinem Hausmaedchen in Urlaub. Wie kann ich mich versicherungsmaessig schuetzen? Die verhaeltnisse sind wie gehabt.'

'Frl. S. naeherte sich der Ampel und sprang von Gelb auf Gruen um.'

'Ich habe seinerzeit meinen Wagen nicht umgemeldet, da ich meinen Wohnsitz verlegte und diesen nicht verschrotten wollte.'

'Meine Tochter ging mit mir deshalb auf die Strasse, weil sie nicht mit dem Fahrrad fahren durfte, denn sie hatte es im Halse.'

1.70 2.14 Danksagungen & liebe GrüBe...

2.14 Danksagungen & liebe GrüBe...

Tja, an dieser Stelle soll nun allen Leuten gedankt werden, die mich bei meiner Arbeit an ChaosPro tatkräftig und moralisch unterstützt haben:

Rolf Schulz: Derjenige, der so viele Vorschläge bringt, daß ich Wochen allein mit seinen Anregungen zu tun habe. Die durchschnittliche Mailgröße zwischen uns beträgt wohl ungefähr 25 KB (oder Rolf?) , wohlgermerkt, ASCII-Text, keine Binärdaten. Er ist der Grund dafür, daß ich für das .todo file jetzt eine größere Festplatte anschaffen mußte... ;-)

Olaf Krolzig: Ein Mann der ersten Stunde. Einer, der mich durch viele Anregungen und Bugreports erfreut hat. Ihm verdankt ihr es, daß ich den 3D-Teil mehrmals überarbeitet habe.

Lutz Uhlmann: Ein Mann der allerersten Stunde. Er war der erste Betatester, der ChaosPro zum Laufen brachte. Ihm verdankt ihr den Screenmode Requester, das Colorcycling und solche Sachen ... Unglücklicherweise rieß der Kontakt nach ein paar Monaten ab, vielleicht habe ich ihn gar zu sehr genervt?

Kay Gehrke: Ein fleißiger Betatester, mit dessen Hilfe es mir gelungen ist, ChaosPro an Grafikkarten anzupassen. Es hat eine ganze Weile gedauert, bis ChaosPro auf seiner Picasso lief ...

Manfred Ambros: Betatester quer durch den Gemüsegarten. Er hat sich auch die Mühe gemacht, sich die Doku anzusehen und mich auf Rechtschreibfehler und falsche Links aufmerksam zu machen. Respekt, so was macht sonst kaum einer...

Jake und Mac Melon: Diese beiden Betatester haben mich dazu veranlaßt, das Animationssystem, die Splines, 24 Bit, 3D und solche Sachen zu schreiben bzw. zu verbessern. Desweiteren fanden sie Fehler in der Arithmetik, die die Genauigkeit beeinflussten, was ihnen außerordentlich mißfiel...

Roberto Patriarca: Er ist für die Übersetzung der Katalog-Datei ins Italienische verantwortlich. Er hat sich sogar angeboten, die Dokumentation ins Italienische zu übersetzen, was ich jedoch ablehnte, da ich denke, daß er nach der Übersetzung der Dokumentation wohl am liebsten für den Rest seines Lebens nichts mehr von ChaosPro hören konnte. So habe ich mir einen begeisterten Freund erhalten...

Bruce Dawson: Er hat Mand2000 (Cygnussoft) mit diesem herrlichen Benutzerinterface geschrieben. Wenn man mir früher einmal vorgeschlagen hätte, ich solle doch ein Fraktalprogramm schreiben, das die Fraktale

in Windows hinein berechnet und in der Lage ist, mehrere Fraktale gleichzeitig zu berechnen, so hätte ich ihn gefragt, wozu das denn gut sein soll und hätte es gelassen. Es mußte erst Bruce Dawson und sein Mand2000 kommen und zeigen, daß dies sehr viel Sinn macht. Mand2000 diene also als Vorbild für ChaosPro. Herzlichen Dank, Bruce Dawson!

1.71 2.15 Fähigkeiten von ChaosPro

2.15 Fähigkeiten von ChaosPro

Dieses Kapitel beschreibt folgende Features von ChaosPro:

1. Berechnen von 24 Bit Bildern
2. 3D-Transformationen
3. Animationen
4. Netzwerkkunterstützung
5. Virtueller Berechnungsmodus
6. Konfigurationsmöglichkeiten

1. Berechnen von 24 Bit Bildern

Eine der Besonderheiten von ChaosPro ist, daß ein Puffer verwendet werden kann, der die berechneten Iterationswerte aufnimmt. Dies ermöglicht es dann natürlich, Bilder in verschiedenen Planetiefen abzuspeichern. Das Programm muß lediglich so tun, als ob ein Screen mit entsprechender Farbzahl vorhanden ist und die Werte aus dem Puffer in Farben umrechnen. Dadurch ist es möglich, Bilder unabhängig von der Anzahl an Farben auf dem Bildschirm abzuspeichern. Will man nun Bilder in 24 Bit abspeichern, so müssen natürlich auch entsprechend viele verschiedene Werte im Puffer vorhanden sein. Nimmt man die 'normale' Einfärbung, d.h. man berechnet einfach zu jedem Pixel im Window den Iterationswert, so ist klar, daß hierbei nicht besonders attraktive 24 Bit Bilder zustandekommen. Es sind hierbei ja nur Werte von 0 bis 'MaxIt' möglich und dabei ist nicht einmal gesagt, daß auch nur annähernd alle Iterationswerte vorkommen. Das ist natürlich absolut unbefriedigend. Es muß also ein Algorithmus verwendet werden, der den verschiedenen Pixel im Window keine ganze Zahl

von 0 bis MaxIt, sondern eine reelle Zahl, im wesentlichen also eine Zahl mit Nachkommastellen, zuordnet. Dann bekommt man z.B. für die Standardmandelbrotmenge nicht nur die Zahlen 0, 1, 2, ..., 148, 149, 150, was in maximal 150 verschiedenen Farben resultiert, sondern auch alle möglichen Zwischenwerte wie 3.45, 6.79, 76.123. Nur dadurch ist es dann möglich, den immens großen 24 Bit Farbraum annähernd auszunutzen. ChaosPro bietet 2 Algorithmen, die dies leisten: CPM und DEM. Näheres bitte bei Außenfärbung nachlesen.

Will man also 24 Bit Bilder abspeichern, so muß man einen dieser beiden Algorithmen verwenden. Außerdem muß darauf geachtet werden, daß der Puffer auch entsprechend viele unterschiedliche Werte aufnehmen kann. Ihn sollte man dann am besten auf IEEEESP setzen. Dannach steht hübschen 24 Bit Bildern nichts mehr im Wege.

2. 3D-Transformationen

Wie schon beim Abspeichern von 24 Bit Bildern ist auch hier darauf zu achten, daß viele verschiedene Werte vorkommen. Benutzt man den 'normalen' Modus, so kommen nur ganze Zahlen vor. Daß es hierbei zu äußerst unattraktiven Treppenstufen kommt, dürfte allen klar sein. Also auch bei der 3D-Transformation sollte tunlichst CPM oder DEM gewählt werden. Desweiteren sollte der Puffer auf IEEEESP gesetzt werden, um unschöne Effekte zu vermeiden. Dannach ist es nur noch eine Sache der Parameter, die richtig gewählt werden müssen, um schöne 3D-Transformationen berechnen zu können. Eines sollte noch erwähnt werden. Aus gewissen mir bekannten aber schlecht erklärbaren Gründen ist DEM für 3D-Transformationen wesentlich besser geeignet als CPM. Allerdings ist DEM auch wesentlich langsamer, da hierfür gleichzeitig 2 Iterationsfunktionen berechnet werden müssen. Es war in ChaosProV1.0 ein großes Problem, gewisse Mandelbrot/Juliamengen in 3D vernünftig darzustellen. Stellte man ein, so traten sehr starke Sprünge auf. Dies passiert mit DEM nicht mehr in dieser Weise. 2 Punkte, die dicht beieinanderliegen, haben bei Färbung=CPM einen beliebig stark unterschiedlichen (Potential-)Wert, bei Färbung=DEM jedoch nicht. Falls es also einfach nicht gelungen ist, auftretende Zacken und Spitzen zu eliminieren, so sollte man es nun mit DEM versuchen...

Das Hauptwindow für die 3D-Transformationen ist das 3D Parameterwindow Nr. 2. Hier wird festgelegt, wie die Werte aus dem Puffer in Höhenwerte transformiert werden. Man sollte zuerst einmal auf 'Vorschlag' klicken, um überhaupt etwas zu sehen. Dann sehe ich mir die 'Wasser' und 'Plateau'-Werte an und wähle 'Plateau' so groß wie die Iterationstiefe, nicht damit

es passiert, daß alles oberhalb von 100 eingeebnet wird und somit entweder gar ←
nichts oder nur eine Ebene
zu sehen ist. Dann sehe ich etwas und wähle eine vernünftige Farbpalette wie z.B. ←
Volcano oder Grey.
Dannach entscheide ich mich, ob ich evtl. die Zacken verstärkt oder geschwächt ←
haben möchte und setze
'Steigung' höher (für Verstärken) oder niedriger (für Abschwächen). Dannach ←
versuche ich
immer, 'FrontMult' und 'BackMult' so zu wählen, daß das Bild wieder einigermaßen ' ←
paßt'. Es kann sehr leicht
passieren, daß sich riesige Zacken bilden, daß 'Steigung' ein sehr diffiziler ←
Parameter ist. Evtl. muß dannach
'FrontMult' und 'BackMult' um den Faktor 1000 verkleinert werden. Nach ein bisschen ←
herumexperimentieren mit
'Max' und 'Min' wählt man noch 'Plateau', um Zacken abzuschneiden, versucht alles ←
nocheinmal mit 'Invers' und
erhält früher oder später einmal ein brauchbares 3D-Bild. Gelingt es einfach nicht ←
, so sollte man
den Algorithmus DEM verwenden. Zum Schluß kann man noch mit den anderen 3D-Windows ←
herumexperimentieren und
den Ansichtswinkel etc. verändern.

3. Animationen

ChaosPro kann Animationen in beliebiger Farbtiefe erzeugen. Dabei kann jeder ←
sinnvolle Parameter
geändert werden, um so möglichst keine Wünsche offenzulassen. Die normale ←
Vorgehensweise ist
folgende:
Man berechnet ein beliebiges Fraktal, das den Anfangspunkt der Animation ←
definieren soll. Dieses Fraktal
darf man noch beliebig verändern. Entspricht es den Wünschen den Benutzers, so ←
fügt man
es als AnimKey in die AnimKeyliste an Position 0 ein. Dadurch werden sämtliche ←
Daten in den AnimKey
hineinkopiert und lassen sich nicht mehr direkt verändern. Nun ändert man den ein ←
oder anderen Wert des
berechneten Fraktals. So könnte man ein paar mal hineinzoomen und ein paar andere ←
Werte ändern. Dannach kann
man das Fraktal erneut als AnimKey nach dem bereits eingefügten AnimKey einfügen.
ChaosPro ist in der Lage, die Parameter, die sich bei diesen AnimKeys ←
unterscheiden, zu erkennen und kann 'Zwischenbilder'
erzeugen. Es sollte klar sein, daß man nicht allzuviel von einem AnimKey auf den ←
anderen ändern sollte, da es doch
sehr oft zu seltsamen Nebeneffekten kommen kann. Ich selber zoome meist von einem ←
AnimKey auf den anderen 1 mal hinein,
fahre noch vielleicht 50% der Windowbreite herum und füge es dann erneut als ←
AnimKey ein. Will man öfter hineinzoomen, so
empfehle ich, eben mehrere AnimKeys einzufügen.
So, das ist also die Erzeugung einer Animation. Nach Festlegung der Breite, der ←
Höhe, der Tiefe in Planes, des Puffers etc.
kann die Berechnung dann gestartet werden.
Seit ChaosProV3.0 nun kann man auch die Farbpalette während der ←
Animationsberechnung ändern.
Dazu kann man den verschiedenen AnimKeys auf Wunsch eine Farbpalette zuordnen. So ←
könnte man

z.B. während einer Zoom-In-Animation von der Farbpalette 'Volcano' auf 'Ice' überblenden. ←
Es besteht kein Zwang, jedem AnimKey eine Palette zuzuordnen. Hat man z.B. 5 AnimKeys und möchte einfach ←
von einer Farbpalette zu einer anderen wechseln, so ordnet man dem ersten AnimKey die eine Farbpalette ←
zu und dem letzten AnimKey eben die andere. ChaosPro kümmert sich dann um den Rest. Andernfalls müßte man ←
ja per Hand Zwischenpaletten berechnen und sie den einzelnen AnimKeys zuordnen, um den gewünschten ←
Effekt zu bekommen, eine Arbeit, bei deren Gedanken allein mir schon schlecht wird.
Das Zuweisen der Farbpalette geschieht denkbar einfach: AnimKey anwählen, im Palettenrequester eine Palette auswählen, ←
den gewünschten Offset- und Skip-Wert und im Anim1-Window auf 'Set' gehen. Will man sie wieder löschen, so eben auf ←
'Clear' klicken. Falls der erste AnimKey keine Palette besitzt, so benutzt ChaosPro automatisch die ←
aktuelle globale Palette. Der Grund ist klar: Für den ersten Frame braucht ChaosPro einfach unbedingt eine ←
Farbpalette. Es sollte auch nicht verwirren, bleibt ChaosPro doch so abwärtskompatibel zu V2.0 und V1.0. Dort ←
wies man keinerlei Farbpaletten zu, es wurde einfach die globale Farbpalette für die Animation benutzt. ←
Erwähnenswert ist noch, daß im Falle, daß ab einem bestimmten AnimKey keine Palette mehr zugewiesen wurde, ←
die Farbpalette fest bleibt. Hat z.B. AnimKey 1 eine Farbpalette, AnimKey 2 eine andere, AnimKey 3, 4, etc. keine mehr, ←
so wird die Palette beim Übergang von AnimKey 1 zu AnimKey 2 verändert und bleibt dann fest. ←

4. Netzwerkunterstützung

Seit ChaosProV3.0 besitzt ChaosPro die Fähigkeit, ein vorhandenes Netzwerk mit anderen Amigas auszunutzen. Dabei muß auf allen Amiga-Rechnern im Netzwerk CPChild gestartet werden, auf einem Rechner ChaosPro. Beim Berechnen der Fraktale wird dann die Aufgabe auf alle Rechner, auf denen CPChild gestartet wurde, verteilt. CPChild ist im Prinzip so ähnlich wie ChaosPro, bloß daß das GUI fehlt. CPChild erzeugt bei Bedarf Subtasks, die die eigentliche Berechnung übernehmen, genauso wie ChaosPro dies macht. ←
Nun, einige Voraussetzungen sind allerdings doch noch anzumerken: Zuerst muß als Netzwerksoftware Envoy, d.h. genauer gesagt die nipc.library installiert sein. Desweiteren müssen alle Amigas natürlich über eine FPU verfügen. Der Speicherbedarf ist auf den Rechnern, auf denen CPChild läuft, allerdings nicht sehr groß. Es wird ja nur berechnet. ChaosPro selber sammelt die Daten und speichert sie. ←
Um nun das Netzwerk auszunutzen, sollte man ChaosPro überall komplett installieren. Dannach muß man sich überlegen, auf welchem Rechner man ChaosPro laufen lassen will, und merke sich dabei den Hostnamen eben dieses Rechners. ←

Dannach muß man das ToolType CHAOSPROHOST von CPChild auf den Hostnamen setzen, ←
auf dem man ChaosPro zu starten gedenkt. ←
Der Sinn ist folgender: Nach dem Start von CPChild versucht dieses Programm, ←
Verbindung ←
zu ChaosPro aufzunehmen und ChaosPro zu sagen, daß ein neuer Host vorhanden ist, ←
der ←
zum Berechnen der Fraktale zur Verfügung steht. Dazu ist es allerdings nötig, daß ←
CPChild weiß, auf welchem Rechner ←
denn ChaosPro zu finden ist. ←
Ist dies mit allen CPChild-Programmen durchgeführt, so kann es losgehen: Man ←
starte ChaosPro, dannach starte man auf den anderen ←
Rechnern CPChild. Das wars dann schon. Beim Beenden von ChaosPro werden die ←
CPChild-Programme automatisch beendet, zuvor ist es nicht ←
möglich, diese zu beenden. ←
Sonstiges Erwähnenswertes: Es ist durchaus legitim, CPChild auf einem Rechner ←
mehrmals zu starten. Insbesondere macht es viel Sinn, CPChild ←
auch auf dem Rechner zu starten, auf dem man ChaosPro gestartet hat, da ChaosPro ←
im Netzwerkmodus gar nichts mehr berechnet und alles von ←
den verschiedenen CPChild-Programmen berechnen läßt. Würde ChaosPro nämlich auch ←
berechnen, ←
so müßte ChaosPro ständig während der Berechnung darauf achten, ob Daten über das ←
Netzwerk angekommen sind. ←
Ich halte das für nicht gut, besser finde ich die Option, doch einfach CPChild ←
auch auf dem selben Host wie ←
ChaosPro zu starten.

5. Virtueller Berechnungsmodus

Seit V2.0 besitzt ChaosPro die Fähigkeit, Fraktale in einem virtuellen Modus zu ←
berechnen. Das heißt, es ist möglich, riesige Fraktale ←
zu berechnen, ohne viel Speicher zu benötigen. Bisher war es mit meinem Amiga z.B. ←
nicht möglich, ein 1024x768 Punkte großes 3D-Fraktal ←
zu berechnen, da ich nicht genügend Speicher für den nötigen Puffer hatte. Um nun ←
ein Fraktal im virtuellen Modus zu berechnen, muß ←
man im Parameterwindow Nr 3 von Julia- und Mandelbrotmengen den 4 Gadgets die ←
entsprechenden Daten angeben, also die Breite, die Höhe, die Tiefe in Planes ←
und ob nach dem Berechnen des Fraktals ←
eine 3D-Version davon berechnet werden soll. ←
Sind die Daten eingestellt, so muß man nur noch den Menüpunkt 'Fraktal/Starte ←
virtuell' anwählen, schon beginnt ChaosPro mit der Berechnung. ←
Als Puffer wird nun nicht RAM-Speicher, sondern Festplattenplatz benutzt. Es wird ←
dabei eine Datei erzeugt. Normalerweise wird diese ←
Datei im Verzeichnis 'ChaosPro:' erzeugt, was man jedoch mit dem Tooltype 'VIRTUAL ←
=<Dir>' ändern kann. ←
Nachdem das Fraktal berechnet wurde, erscheint ein Filerequester, der nach dem ←
Namen der zu speichernden IFF-ILBM-Datei ←
frägt. ←
Dieses virtuelle Berechnen ist insbesondere nützlich in Verbindung mit dem ←
Druckertool Studio, das den Ausdruck ←
von IFF-ILBM-Bildern in beliebiger Größe erlaubt, was es ermöglicht, große ←
Fraktale auszudrucken, die ←
wirklich exzellent aussehen. Normalerweise kommt beim Ausdruck eines Fraktals in ' ←
normaler' Größe ein ziemliches Klötzchenbild heraus, ←
was nicht unbedingt professionell aussieht.

6. Konfigurationsmöglichkeiten

Es gibt einige Möglichkeiten, wie man ChaosPro den eigenen Bedürfnissen anpassen kann. All diese Möglichkeiten waren bisher über das ganze Guide-File verstreut, so daß kaum jemand alle Möglichkeiten kannte. Meist lesen die Anwender sowieso nicht das Guide-File, vielleicht wird aber dieses Kapitel gelesen.

- Was natürlich als erstes auffällt, ist der verwendete Screenmodus. Es existieren dafür Menüpunkte, die es erlauben, den Screenmodus von Fraktalscreen, Parameterscreen und Colorscreen zu verändern.

- Desweiteren ist es möglich, ChaosPro auf einem Publicscreen zu öffnen. Allerdings ist ChaosPro recht radikal, was die Farben anbelangt. Es nimmt einfach alle Farben ab Nummer 4 in Beschlag und ändert sie je nach verwendeter Farbpalette. Den Publicscreen, auf dem ChaosPro dann die Berechnungen startet, wählt man über Tooltypes.

- Tja, die Tooltypes. Immer wenn ich mir zu faul war, eine komfortable Methode der Programmeinstellungen anzubieten, habe ich dies über Tooltypes erledigt. Es gibt einige davon. Es wird sehr (!) empfohlen, alles durchzulesen. Die Ansteuerung von EGS wird z.B. über ToolTypes an- und ausgeschaltet. Die Online-Hilfe ebenfalls, falls man z.B. wenig Speicher hat. Sie benötigt doch recht viel Speicher und außerdem einige Zeit zur Initialisierung. Zu den ToolTypes geht's

[hier](#)

.

- Fonts: ChaosPro bietet die Möglichkeit, die Fonts für das GUI, das Menü und für die Requester getrennt einzustellen. Dies verändert natürlich das Aussehen z.T. stark. Verändert werden die Fonts schließlich über Menüpunkte.

- Eine nach meinen Erfahrungen bisher kaum benutzte Möglichkeit der Einstellungen ist die Veränderung des Fonts, der Sprache und der Breite des Guide-Files. Das Guide-File sollte natürlich ChaosPro.guide heißen und in deutsch vorliegen. Man kann den Font des Guide-Files verändern, die Breite einer Zeile in Pixel angeben und die Sprache auswählen. Ich bin wirklich sehr erschrocken, als ich jemanden sah, der ChaosPro benutzte, die Online-Hilfe aufrief und dann in grober Schrift (topaz 8) und falscher Breite, zusätzlich noch in englisch, die Dokumentation angezeigt bekam. Ich hatte sofort Erbarmen mit ihm...Wie man das ganze ändert, findet sich

[hier](#)

- Windows, deren Positionen und das automatische Öffnen: Alle globalen Windows, d. h. alle Windows, die nur einmal geöffnet werden können, haben einstellbare Positionen und ein Flag, das es ermöglicht, daß sie beim Programmstart automatisch geöffnet werden. Dazu ist eigentlich nicht viel zu tun. Man plaziere die Windows so, wie man sie gerne hätte, dann öffne/schließe man sie, bis alles so aussieht, wie es aussehen soll, wenn ChaosPro startet. Dann speichere man das Konfig-File (wieder ein Menüpunkt). Die Windowpositionen und der Status, also ob geöffnet oder geschlossen, werden dann in das Konfig-File eingetragen und abgespeichert. Beim nächsten Start von ChaosPro sieht ChaosPro wieder so aus wie nach dem Abspeichern des Konfig-Files.

- Es besteht die Möglichkeit, benutzerdefinierte Windows oder Menüpunkte zu erzeugen. Falls diese aktiviert werden, wird das ihnen zugeordnete Arexxscript ausgeführt.

3 Links hierfür:

Benutzerdefiniertes Menü

Benutzerdefinierte Windows

Preferencesprogramm zum Übersetzen

- Wenig bekannt ist, daß das Arexxscript 'ChaosProInit' im Verzeichnis ChaosPro/Rexx stets beim Programmstart ausgeführt wird. Hierdurch können legliche anderen Konfigurationsmöglichkeiten, die bisher nicht direkt unterstützt werden, evtl. über Arexx verwirklicht werden.

- Defaultbildchen: Die Dockwindows wurden begeistert akzeptiert, lediglich an den Bildern an sich hatten die Leute etwas auszusetzen. Sie gefielen ihnen zumeist nicht. Deshalb existiert nun die Möglichkeit, diese Bilder zu ersetzen. Das Directory 'ChaosPro/Docking' wird nach Dateien mit bestimmten Namen untersucht. Falls Dateien mit den entsprechenden Namen vorhanden sind, werden die darin enthaltenen Daten zur Darstellung benutzt. Diese Dateien müssen allerdings im RAW-Format vorliegen, da ich mir zu faul war, eine universelle Laderoutine für IFF-Bilder zu schreiben oder aber Datatypes zu benutzen, worüber sich dann die OS2.0 Benutzer aufgeregt hätten. Zur Konvertierung von IFF-Bildern ins RAW-Format kann ich z.B. GfxMaster aus dem Aminet empfehlen.

Zu den Daten der Bilder: Die Bilder müssen eine Tiefe von 4 Farben, also 2 Planes haben, die Größe ist fest bei 32x24 Pixel. Am besten erzeugt man diese Bilder mit einem Malprogramm und konvertiert sie dann mit z.B. GfxMaster ins RAW-Format. Dann muß das File nur noch mit dem korrekten Namen im Directory Dockimg abgespeichert werden. Folgende Namen sind möglich, das evtl. vorhandene 'Sel' am Ende deutet auf das

Bild hin, das erscheinen soll, wenn das Gadget aktiviert worden ist:

ImAnim1 ImAnim1Sel
 ImAnim2 ImAnim2Sel
 ImCycle ImCycleSel
 ImPalette ImPaletteSel
 Im2D1 Im2D1Sel
 Im2D2 Im2D2Sel
 Im2D3 Im2D3Sel
 Im3D1 Im3D1Sel
 Im3D2 Im3D2Sel
 Im3D3 Im3D3Sel
 ImWizard ImWizardSel
 ImComment ImCommentSel
 ImColMap ImColMapSel
 ImBoxZoomIn ImBoxZoomInSel
 ImBoxZoomOut ImBoxZoomOutSel
 ImUndo ImUndoSel
 ImRedo ImRedoSel
 ImShowLoc ImShowLocSel
 ImShowJul ImShowJulSel
 ImSave ImSaveSel
 ImIFS ImIFSSel
 ImDOL ImDOLSel
 ImJM ImJMSel
 ImDock1 ImDock1Sel
 ImDock2 ImDock2Sel
 ImDock3 ImDock3Sel

- Default-Fraktal-Typen: Ein weiteres Directory wurde geschaffen: 'ChaosPro/ Defaults' ←
 Dieses Directory kann Fraktaldateien enthalten, die dann geladen und benutzt ←
 werden, ←
 wenn der Benutzer ein neues Defaultfraktal erzeugen will, oder aber die Daten ←
 eines Fraktals ←
 auf die Defaultwerte setzen will. Ist man mit den Defaultwerten, die ChaosPro ←
 standardmäßig ←
 liefert, nicht einverstanden, z.B. mit dem Puffer, den ChaosPro wählt, oder aber ←
 mit dem ←
 Parameter 'c' der Standardjuliamege oder mit irgendetwas anderem, so kann dies ←
 korrigiert werden: ←
 Man wähle die Parameter des entsprechenden Fraktaltyps so, wie man meint, daß das ←
 Fraktal defaultmäßig ←
 aussehen soll. Dann speichert man das Datenfile durch 'Projekt/Daten speichern' im ←
 Directory ←
 ChaosPro/Defaults mit dem korrekten Namen ab, der vom Fraktaltyp abhängt. Folgende ←
 Files werden ←
 geladen und erkannt:

"def_Julia"
 "def_Mandel"
 "def_Bifurcation"
 "def_DynamicSystem"
 "def_Plasma"
 "def_Lyapunov"
 "def_Diffusion"
 "def_LSystem"
 "def_IFS"

"def_Brown"

Ich bitte darum, hier nicht sinnlos herumzuspielen, also z.B. ein Juliafraktal unter dem Namen "def_LSystem" abzuspeichern. Ich glaube zwar nicht, daß ChaosPro dann viel Sinnvolles zustandebringen... Ich empfehle, den Bildnamen auch entsprechend zu setzen. Desweiteren empfehle ich, die Einträge im Kommentarwindow auf den eigenen Namen zu setzen. Dadurch erhält jedes Fraktal automatisch den Namen etc. des Benutzers. Es wird ja beim Erzeugen das gesamte File eingeladen, und in diesem File sind auch alle Einträge des Kommentarwindows enthalten.

1.72 Index

III. Index

24 Bit

3D:

-

3D Parameterwindow 1

-

3D Parameterwindow 2

-

3D Parameterwindow 3

-

Einführung

About

Animationen:

- 3D

- Ausmaße

- Ein-/Ausgabe

- Fraktaldatei

- Frameverteilungsmodus

- Key hinzufügen

- Key löschen

- Key verschieben

- Keys

- Planetiefe

- Start/Abbruch

-

Window

Backdropwindow

Benutzerdefinierte Windows

Berechnung stoppen

Berechnung stoppen

Berechnung fortsetzen

Berechnung fortsetzen

Bild berechnen

Bild duplizieren

Bild löschen

Bilder speichern

Bildname

Bifurkation:

- Iterationen
- Zykluslänge
- Variablenwerte
- Datenwindow
- Variablen
- A
- zu benutzende Variable
- Iterationen
- Formel
- Parameterwindow 1
-

 Theorie

 Boxzoom

Colorcycling

Daten speichern

Datenwindow

Defaultwerte einstellen

Dynamisches System:

- Ansichtswinkel
- Ausschnitt
- Geschwindigkeit
- Parameterwindow 1
- Parameterwindow 2
- Start
- Systemtyp
-

 Theorie

 - Zeichenmodus

- Zeiteinstellungen

Formeleditor:

- Formelein-/ausgabe
- Formel hinzufügen
- Formel verbessern

Fraktal verschieben

Fraktalbilder

Fraktaltasks

Fraktaltypen

Fraktalwindow

Hilfe anzeigen

Juliamenge:

- Abbruchbedingungen
- Ausschnitt
- Außenfärbung
- Bailin
- Bailout
- Biomorphie
- Datenwindow
- Dekomposition
- Formel

- Innenfärbung
- Iterationen
- Kreisinversion
- Parameter
- Parameterwindow 1
- Parameterwindow 2
- Parameterwindow 3
-

Theorie

- Zeichendurchgänge

Juliaparameter zeigen

Lyapunov-Raum:

- Ausschnitt
- Chaosfarbe
- Daten
- Formel
- Iterationen
- minimaler Exponent
- Parameter
- Sequenz
- Stabilisation
- Start
-

Theorie

- Zeichendurchgänge

Mandelbrotmenge:

- Abbruchbedingungen
- Ausschnitt
- Außenfärbung
- Bailin
- Bailout
- Biomorphie
- Datenwindow
- Dekomposition
- Formel
- Innenfärbung
- Iterationen
- Kreisinversion
- Parameter
- Parameterwindow 1
- Parameterwindow 2
- Parameterwindow 3
-

Theorie

- Zeichendurchgänge

Neuberechnung

Paletten/Palette bearbeiten:

- Aktionen
 - Bearbeiten
 - Bearbeitungswindows
 - Bereiche
 - Colorcycling
 - Dauerhaft stauchen
 - Duplizieren
-

- Ein-/Ausgabe
- Farben
- Farbnummer
- FSH
- Invertieren
- Kopieren
- Löschen
- Name
- Palettenwindow
- RGB

Plasma:

- Granulation
- Parameter
- Proportion
-

 Theorie

- Zufallsreihe

Position zeigen

Proportion

Preview

Quit

Redo

Systeminfo

Taskpriorität

Theoretisches zu:

-

 Bifurkation

-

 Dynamisches System

-

 Juliamenge

-

 Mandelbrotmenge

-

 Lyapunov-Raum

-

 Plasma

 Undo

Windows auslagern

Zoom
