

LFrac

COLLABORATORS

	<i>TITLE :</i> LFrac		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		June 25, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	LFrac	1
1.1	LFrac Dokumentation	1
1.2	LFrac Copyright und Vertrieb	1
1.3	LFrac Adresse des Autors	2
1.4	Zweck und Funktionsweise von LFracs	2
1.5	Ausführliche Einführung zu LFracs	3
1.6	Wie Sie LFracs zum Laufen bringen	5
1.7	Erklärungen zu den Menüs	6
1.8	Erklärungen zu den Requestern	7
1.9	LFrac Programm-Geschichte	8
1.10	Literaturhinweise zu LFracs	8

Chapter 1

LFracS

1.1 LFracs Dokumentation

Willkommen zum LFracs V1.0 User Manual

Bitte wählen Sie aus der nachfolgenden Gliederung die gewünschten Informationen und folgen Sie den Querverweisen.

Copyright und Vertrieb

Adresse des Autors

Einführung

Installation

Menüs

Requester

Programm-Geschichte

Literatur

(c) Alexander Welisch

1.2 LFracs Copyright und Vertrieb

LFracS (das ausführbare Programm, Quelltext und Anleitung) ↔
ist
Copyright © 1996 Alexander Welisch.
Alle Rechte vorbehalten.

LFracS ist FREeware!

LFracS darf frei weitergegeben werden. Der Quellcode liegt dem Programmpaket bei und darf nach Belieben für den persönlichen Gebrauch verändert werden. Eine solche veränderte Version darf allerdings NICHT weitergegeben werden.

Da LFracs kostenlos ist, gibt es KEINE GARANTIE für den einwandfreien Betrieb. Der Autor ist nicht verantwortlich für irgendeinen Verlust oder Schaden, der durch die Benutzung von LFracs entsteht. Sie benutzen LFracs auf eigene Gefahr.

LFracs darf nicht verkauft werden; erlaubt ist lediglich eine Kopiergebühr von nicht mehr als 5 DM. Außerdem darf LFracs nicht ohne vorherige Zustimmung seitens des Autors zusammen mit einem kommerziellen Produkt vertrieben werden. LFracs darf ausschließlich weitergegeben werden, wenn die gesamte Anleitung vorhanden ist und nicht verändert wurde; außerdem sollte der Quellcode dabei sein.

Siehe auch:

Adresse des Autors

.

1.3 LFracs Adresse des Autors

Bitte schicken Sie Kommentare, Vorschläge, Fehlerberichte, Lobeshymnen, Geld usw. an:

Alexander Welisch
Kolpingweg 15
94086 Griesbach
DEUTSCHLAND

ODER EMAIL: welisch@kirk.fmi.uni-passau.de

1.4 Zweck und Funktionsweise von LFracs

Bevor Sie weiterlesen, sollten Sie das Programm bereits auf ←
Diskette (oder
Festplatte) fertig installiert haben (s.
LFracs installieren
).

Starten Sie nun LFracs (von einer Shell oder der Workbench aus).

Zu Beginn erscheint der Eingabe-Requester, in dem Sie einfach den "LADEN"-Knopf betätigen und im anschließend erscheinenden Datei-Requester die Datei "Dragon.code" auswählen. Verlassen Sie darauf den Eingabe-Requester mit dem "OK"-Knopf.

Darauf wählen Sie im nun sichtbaren Zeichnen-Requester das "Tiefe"-Feld aus und geben Sie eine Zahl zwischen 4 und 12 ein. Um die übrigen Felder brauchen Sie sich jetzt noch nicht zu kümmern.

Drücken Sie wiederum den "Ok"-Knopf und das Fraktal "Dragon" wird zuerst im Hintergrund bis zur gewünschten Tiefe berechnet und danach gezeichnet.

Sie können jederzeit (auch während ein Fraktal gezeichnet wird) durch Auswahl eines Menü-Punkts entweder ein anderes Fraktal wählen (Menü-Punkt "Neu" => Eingabe-Requester), oder das Fraktal in einer anderen Tiefe betrachten (Menü-Punkt "Zeichnen" => Zeichnen-Requester).

Der Zeichenvorgang kann jederzeit durch Drücken der rechten Maustaste gestoppt werden.

Wenn Sie jetzt ein bißchen rumexperimentieren (andere "#?.code"-Dateien im Eingabe-Requester laden oder andere Werte für die "Tiefe" eingeben), werden Sie schnell das Prinzip von LFracs verstehen (vielleicht sogar ohne die

ausführliche Einführung
lesen zu müssen).

1.5 Ausführliche Einführung zu LFracs

LFracS ist ein Programm zum Zeichnen von L-Fraktalen (Line-Fractals).

L-Fraktale sind Fraktale, die durch Rekursive Weiterentwicklung eines "Ursprungs-Codes" (im folgenden "Axiom" genannt) in Form einer Zeichenkette erzeugt werden können. Diese Zeichenkette kann dann in ein Bild übersetzt (gezeichnet) werden.

Ein Beispiel:

Angenommen, wir geben das Axiom "f" ein, so bedeutet diese Zeichenkette für den Computer: "Zeichne eine waagerechte Linie der Länge L von links nach rechts".

Das Programm übersetzt also das Axiom in Zeichenbefehle und zwar auf eine ganz bestimmte Weise, die Kennern der Programmiersprache LOGO bekannt vorkommen wird:

Stell Dir eine Schildkröte auf einem großen Blatt Papier vor. Nehmen wir einmal an, unsere Schildkröte kann nur geradeaus gehen (keine Kurven), stehenbleiben, oder sich auf der Stelle um die eigene Achse drehen. Das heißt, wenn die Schildkröte einen anderen Weg als eine gerade Linie beschreiben will, muß sie zuerst stehenbleiben, sich dann auf der Stelle drehen (z.B. um 90°) und danach wieder geradeaus weitergehen (zu Anfang steht die Schildkröte). (zu ←

Außerdem tut diese Schildkröte genau das, was man ihr sagt; angenommen Du sagst zu ihr "Gehe geradeaus!", dann geht sie eine bestimmte Strecke (z.B. 1 Meter – das muß Du vorher festlegen) geradeaus und bleibt dann stehen. Sagst Du "Drehe dich nach links", so dreht sich die Schildkröte um die eigene Achse um einen ebenfalls vorher vereinbarten Winkel nach links und wartet dann auf weitere Anweisungen.

Das besagte Axiom besteht aus Befehlen für die Schildkröte:

(Seien $L=1m$ und $w=30^\circ$)
 "f" bedeutet "Gehe die Strecke L geradeaus"
 "-" bedeutet "Drehe dich um w Grad nach links"
 "+" bedeutet "Drehe dich um w Grad nach rechts"

"f-f+f" bedeutet: "Gehe 1m geradeaus, drehe dich dann 30 Grad nach links, gehe darauf wieder 1m geradeaus, drehe dich um 30 Grad nach rechts gehe 1m geradeaus und bleibe dann stehen."

Würde man den Weg der Schildkröte nachzeichnen, würde das ungefähr so aussehen:

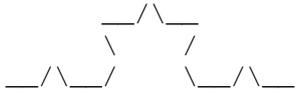


Im folgenden wollen wir vom Axiom $a = a[0] = "f-f++f-f"$ \Leftrightarrow $_/__$ ausgehen:

Ersetzt man jedes "f" in a durch a selbst, so erhält man eine neue Zeichenkette

$a[1] = "f-f++f-f - f-f++f-f ++ f-f++f-f - f-f++f-f"$

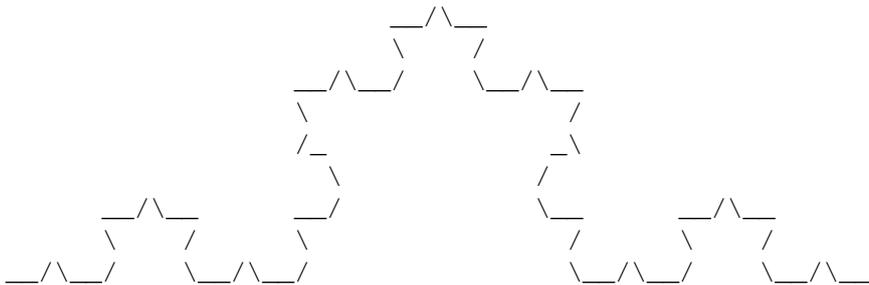
als Grafik übersetzt gibt das:



Ersetzt man in $a[1]$ wiederum jedes "f" durch a, so gibt das

$a[2] = "f-f++f-f - f-f++f-f ++ f-f++f-f - f-f++f-f -$
 $f-f++f-f - f-f++f-f ++ f-f++f-f - f-f++f-f ++$
 $f-f++f-f - f-f++f-f ++ f-f++f-f - f-f++f-f -$
 $f-f++f-f - f-f++f-f ++ f-f++f-f - f-f++f-f"$

also



Würde man dieses Verfahren unendlich oft fortsetzen, erhielte man mit $a[\text{unendlich}]$ ein echtes Fraktal. Dieses Programm kann aber nur endliche $a[0..\text{endliche Zahl}]$ erzeugen, also Näherungs-Fraktale, was allerdings nicht weiter auffällt, da ein Bildschirm-Pixel ab $a[10]$ (wenn $a[10]$ auf dem Bildschirm vollständig sichtbar gezeichnet werden soll) sowieso größer als ein Vorwärtsschritt der Schildkröte ist.

Geht man von einem anderen Axiom a aus, erhält man auch andere Bilder. Außerdem ist es auch möglich, andere Buchstaben als "f" zu verwenden, z.B.:

$a = a[0] = "xf++y"$

mit den Regeln:

Ersetze "f" durch "ff"

Ersetze "x" durch "x++ff-f"

Ersetze "y" durch "y++ff-f"

dadurch erhält man nach einmaliger Ersetzung

$a[1] = "x++ff-f ff ++ y++ff-f"$

und die Schildkröte würde folgenden Weg gehen:

```

  /
 /
 \
  \
   \
  /
 /
 \

```

(Achtung: nur "f" steht für "gehe einen Schritt vorwärts"; "x" und "y" haben für die Schildkröte keine Bedeutung, sondern nur für die Erzeugung der Befehls-Zeichenkette!)

Außerdem gibt es außer "f", "+" und "-" noch die Sonderzeichen

"(" = "Merke dir deine jetzige Position"
 ")" = "Gehe zur zuletzt gemerkten Position zurück (ohne daß der Rückweg mitgezeichnet wird) und fahre von dort mit der Zeichenketten-Ausführung fort"

Mit Hilfe dieser Sonderzeichen ist es möglich, auch Baumartige Fraktale zu erzeugen, z.B.:

```

a = a[0] = "f(ffx)(-ffx)+ffx"
f : "f"           (heißt "Ersetze `f` durch `f`")
x : "f(ffx)(-ffx)+ffx"

```

a[0] als Bild (mit Startwinkel 90°):

```

  \ | /
   \|/
    |

```

dann ist

```

a[1] = "f(ff f(ffx)(-ffx)+ffx )(-ff f(ffx)(-ffx)+ffx )+ff
        f(ffx)(-ffx)+ffx"

```

und als Bild:

```

  \ | \ | / | /
   \| \| \| / \| /
    \ | /
     \|/
      |

```

1.6 Wie Sie LFracs zum Laufen bringen

LFracs besteht aus folgenden Dateien und Verzeichnissen (ggfs. komprimiert in einem Lha-Archiv mit Namen LFracs.lha):

```

LFracs (Dir) ____ LFracs (Programm)
|
|__ Docs (Dir) ____ LF.deutsch.guide (Dokumentation in deutsch)
|                   |__ LF.english.guide (Dokumentation in englisch)
|
|__ Catalogs (Dir) ____ deutsch (Dir) ____ LFracs.catalog

```

```

|
|__ FracsCodes (Dir)  ___ Dragon.code
|                       |__ Hilbert.code
|                       |__ Peano.code
|                       |__ #?.code
|                       ...
|
|__ FracsPictures (Dir) ___ Dragon10.lfpic
|                       |__ Dragon12.lfpic
|                       |__ #?.lfpic
|                       ...

```

LFracS.readme
LFracS-Source.lha

Zum Installieren von LFracs erzeugen Sie einfach ein (Unter)Verzeichnis dieser Form auf einer Diskette (oder auch Festplatte). Dies geschieht automatisch beim Entpacken des Lha-Archivs LFracs.lha.

Wichtig ist das Vorhandensein der Unterverzeichnisse "FracCodes" und "FracPictures"; die Dateien "#?.code" und "#?.lfpic" sind nur Beispieldateien, sie sind für die korrekte Funktion von LFracs nicht notwendig.

Der Quelltext zu LFracs befindet sich im Archiv LFracs-Source.lha (wenn Sie die Quellen nicht interessieren können Sie sie löschen).

1.7 Erklärungen zu den Menüs

Die Menüzeile von LFracs enthält folgende Einträge:

Projekt	Bild
Neu	Zeichnen
Beenden	Laden
	Speichern
	Drucken

Erklärungen zu den einzelnen Menü-Punkten:

Neu: Dieser Menüpunkt ruft den "Eingabe"-Requester auf, der auch sofort nach dem Starten von LFracs erscheint (s.

Requester
).

Hier kann man ein neues Fraktal eingeben, laden oder speichern.

Beenden: Beendet das Programm nach einer Sicherheitsabfrage.

Zeichnen: Öffnet den "Zeichnen"-Requester (s.

Requester
).

Hier kann man sich das im Eingabe-Requester gewählte Fraktal nach Eingabe einiger Parameter (Schrittweite, Zeichen-Startpunkt, Zeichen-Tiefe) auf den Bildschirm zeichnen lassen.

Laden: Lädt ein zuvor abgespeichertes Fraktal-Bild.

Speichern: Speichert das momentan auf dem Bildschirm gezeigte Fraktal

als Bilddatei (wichtig für schöne Fraktal-Bilder, für deren Neu-Berechnung und Neu-Zeichnung der Computer sehr lange arbeiten müßte).

Bemerkungen: - Als "Fraktal-Bild" wird das in einer bestimmten Tiefe fertig berechnete und gezeichnete Abbild eines Fraktals bezeichnet.
 - Mit "Fraktal" ist der, das Fraktal-Bild erzeugende, Buchstaben-Code gemeint, der im Eingabe-Requester als Axiom mit den zugehörigen Expansions-Codes eingegeben wurde.

1.8 Erklärungen zu den Requestern

Gleich beim Start von LFracs erscheint der Eingabe-Requester:

Hier kann man ein neues Fraktal selbst eingeben (Axiom- u. Code-Eingabefelder), speichern (SPEICHERN-Taste) oder vordefinierte Fraktale laden (LADEN-Taste).

Axiom- u. Code-Eingabefelder: Hier kann man selbst ein neues Fraktal eingeben. Um es später abspeichern zu können, sollte man ihm einen Namen geben (Name-Feld). Danach muß die Grundstruktur des Fraktals (entspricht a[0] in

ausf. Einführung

) definiert werden (z.B. "f++f-f" oder "xf+y" im Axiom-Feld).

Für jedes im Axiom neu eingeführte Zeichen (z.B. "x" oder "y") kann in einem der darauffolgenden Code-Felder ein Ersetzungscode eingegeben werden (bei der Berechnung von a[1] wird dann dieses Zeichen durch den Ersetzungscode ersetzt, z.B. "x:y++f-f").

In den Code-Feldern sind die gleichen Zeichen wie im Axiom-Feld erlaubt.

Auch für das Zeichen "f" (= "gehe einen Schritt vorwärts") kann man einen Ersetzungscode definieren (z.B. "f:ff"); der Default-Wert ist "f:f", dies gilt auch für alle neu eingeführten Zeichen wie "x" oder "y" (d.h. falls für "x" in keinem Code-Feld ein Ersetzungscode angegeben wurde, wird "x" immer durch "x" ersetzt).

Vordefinierte Zeichen: "+", "-", "(", ")", "f"

Durch Benutzer definierbare Zeichen: "a", "b", ..., "e", "f", "g", ..., "x", "y", "z"

SPEICHERN-Taste: Sollte erst betätigt werden, wenn Namens-Feld, Axiom-Feld und ggfs die Code-Felder korrekt ausgefüllt sind!

Der Eingabe-Requester ist auch durch den Menü-Punkt "Neu" erreichbar.

Nach Ausfüllen des Eingabe-Requesters, bzw. nach Betätigung der OK-Taste im Eingabe-Requester erscheint der Zeichnen-Requester:

Hier wird definiert, in welcher Tiefe das Fraktal berechnet und gezeichnet werden soll (Tiefe-Feld), an welcher Bildschirmposition mit dem Zeichnen begonnen werden soll (Startposition), in welcher Größe gezeichnet wird (Schrittweite-Feld), wie stark sich die Schildkröte beim Ausführen der Zeichen "+" bzw. "-" drehen muß und in welche Richtung sie zu Beginn ihrer Reise schauen soll (Startwinkel).

Die voreingestellten Werte sind für die meisten Fraktale sinnvoll.
Typische Werte für "Drehwinkel": 24 bei Baumartigen Fraktalen
30 oder 60 für Flocke- und Dreisi-Fraktal
90 für Dragon-, Peano-, Hilbert, und
Quasi-Fraktal
xxx andere Winkel sind immer einen Versuch
wert

1.9 LFracs Programm-Geschichte

v0.0 bis v0.6: (nicht veröffentlicht)

- v0.1: Erste Implementierung auf Amiga500 mit 512kB und zwei Diskettenlaufwerken. Zur Speicherersparnis werden Ergebnisse in Paketen zu je 1024 Buchstaben (Code-Länge) auf Diskette zwischengespeichert.
- v0.2: Hinzufügen einer Grafischen Benutzeroberfläche (Gadgets und Requester)
- v0.3: Gravierende Performance-Optimierungen; Zwischenspeicherung in Paketen zu je 32000 Buchstaben. Erzeugungs-Codes können jetzt geladen und gespeichert werden.
- v0.4: Modularisierung des Programms. Anpassung an AmigaOS2.04 und Implementation auf Amiga1200 mit 60MB-Festplatte und 1MB Fast-Ram.
- v0.5: Halbierung des Speicherbedarfs zur Code-Erzeugung; dadurch wurde es möglich, auf das Zwischenspeichern gänzlich zu verzichten.
- v0.6: Jetzt können fertige Fraktale gespeichert, geladen und gedruckt werden. Außerdem paßt sich das Programm an verschiedene Bildschirmmodi (Proportionen!) an und besitzt nun eine fontsensitive Benutzeroberfläche.

- v1.0: Erste offizielle Programm-Veröffentlichung
LFracs wurde vollkommen lokalisiert und im Design überarbeitet. Nun liegen dem Programm Catalog-Dateien in Deutsch und eine englische Anleitung bei.

1.10 Literaturhinweise zu LFracs

Die Idee zu LFracs habe ich aus dem Buch

Peitgen u. Saupe, "The Beauty of Fractals", Springer-Verlag, New York 1988