

Lothar Limbeck, Studium der Betriebswirtschaft, Kunst und Pädagogik. Derzeit Studienrat z.A. in Essen.
Kahrstraße 59, 4300 Essen 1

Reiner Schneeberger, Studium der Betriebswirtschaft.
von 1976 bis 78 Lehrbeauftragter für Computergrafik an der Universität München.
Seit 1979 Mitglied des Vorstands der GCC.
Kanalstraße 15, 8043 Unterföhring

CIP-Kurztitelaufnahme der Deutschen Bibliothek

Limbeck, Lothar:
Computergrafik: e. Lehr- u. Lernbuch / Lothar
Limbeck; Reiner Schneeberger. – München: E.
Reinhardt, 1979.
ISBN 3-497-00905-9
NE: Schneeberger, Reiner:

ISBN 3-497-00905-9

© by Ernst Reinhardt, GmbH & Co., Verlag, München 1979
Alle Rechte vorbehalten. Ohne schriftliche Genehmigung der
Ernst Reinhardt, GmbH & Co., München, ist es nicht gestattet,
dieses Buch, ganz oder auszugsweise in irgendeiner Form,
zu vervielfältigen, zu speichern oder in andere Sprachen zu
übersetzen.

Satz: Satzstudio Froberg, Freigericht
Druck: Pressedruck, Augsburg
Aufbindung: Grimm & Bleicher, München
Printed in Germany

Kunstinformatik, die Anfänge ...

Als Grundlage meiner in den Jahren 1976-1978 und 1981 am Lehrstuhl für Didaktik der Bildenden Künste an der Universität München (Prof. Hans Daucher) geleisteten Arbeit wählte ich die „Generative Computergrafik“ von Georg Nees. Am Beispiel der Programme SNEBRW und SNERKT lässt sich zeigen wie leicht die Nutzung „des Computers“ den Studenten der Kunsterziehung gemacht wurde. Seinerzeit nutzen wir eine TR440 von Telefunken am LRZ München im Stapelbetrieb mit Lochkarten auf denen die Aufrufe der Routinen eingestanzt wurden. Die in der Sprache FORTRAN verfassten Programme wurden im Jahre 1982 von Dietrich M. Scheringer auf eine Digital Equipment PDP 11 an der Fachhochschule Bielefeld übertragen und für den Dialogbetrieb angepasst. Das von ihm als MARION bezeichnete System diente dort über viele Jahre der Lehre (Prof. Jäger, Prof. Staudt).

Reiner Schneeberger, im September 2010

Die Brownsche Bewegung als Einstieg in die Kunstdidaktik

Die Frage *was ist der Anteil des Computers an der Computerkunst?* beschäftigt mich seit ich das Buch „Generative Computergraphik“ (1969) von Georg Nees in meiner Gymnasialzeit gelesen hatte. Sodann entwickelte ich meinen eigenen Zufallszahlengenerator. So stieß ich auf die Brownsche Bewegung und nahm diese zum Anlass der Programmierung einer Routine: SNE BRW. Wird der Zufall dabei „kontrolliert“ entstehen ganz nette Sachen ... nun ja 30 Jahre später ist das anerkannte Kunst.

Hier nun die Buchauszüge:



3.3.1.7. Die Routine SNEBRW

SNEBRW lehnt sich an das Modell der Brown'schen Molekularbewegung an. Dabei wird die Spur eines Teilchens simuliert und als Linienzug dargestellt. Da aufgrund des Modells die Spur eines Teilchens von den zufällig erfolgenden Zusammenstößen mit anderen Molekülen abhängt, können mit SNEBRW theoretisch alle überhaupt möglichen Bilder generiert werden. Es ist unmöglich die Spur auch nur annähernd vorherzusagen.

Natürlich ist diese Art von Zufälligkeit kein echtes Gestalten mehr, deshalb wurden in SNEBRW eine Reihe von zufallseinschrän-

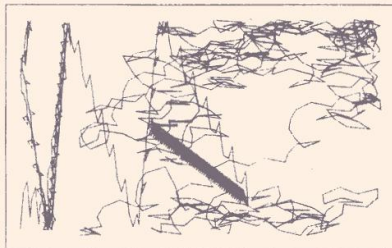


Abb. 47:
CALL BLATT (80., 50.)
CALL SNEBRW (15., 77., 3., 47., 600., 5., 2., 0., 0., 999.)
CALL SNEBRW (3., 50., 4., 47., 500., 1., 4., 0., 0., 4.)

kenden Gestaltungsparametern aufgenommen. Erst sie ergänzen die zufallsgesteuerten Superpositionen um die verschiedensten Gestaltungsideen, wie Erzeugen von Einschlußlinien, Fluß und Geländestrukturen oder Schwärzen von Teilbereichen mit Zufallsmustern.

Mit der Routine SNEBRW kann ein guter Eindruck von der Rolle des Zufalls in der Computergraphik und Computerkunst gewonnen werden*.

Es empfiehlt sich die mit SNEBRW erstellten Graphiken testweise auf einem Graphikbildschirm auszugeben.

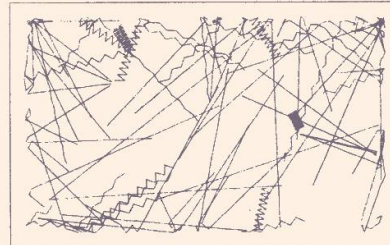
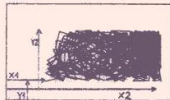


Abb. 48:
CALL BLATT (80., 50.)
DO 1 I = 1, 50*
SX = SNERAD (5., 75., 999)*
SY = SNERAD (1., 45., 999.)
P = SNERAD (1., 3., 999.)
1 CALL SNEBRW (3., 77., 3., 47., 25., 2., 2., SX, SY, P)
* Hinweis:
Die DO-Anweisung dient zum Wiederholen der nachfolgenden Anweisungen. siehe 3.5.1.
SNERAD liefert einen Zufallswert innerhalb des angegebenen Intervalls. siehe 3.3.2.2.

* Parameter PERIOD

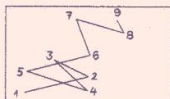
Die Routine SNEBRW benötigt 10 Angaben:

1. Teilbereich in dem der Linienzug liegen soll



X1, X2 (= Bereichsgrenzen in x-Richtung)
Y1, Y2 (= Bereichsgrenzen in y-Richtung)
i.B. X1 = 8. mm X2 = 32. mm
Y1 = 4. mm Y2 = 14.4 mm

2. Anzahl von Punkten des Linienzugs



ANZPKT (= Anzahl der Punkte)
dieser Wert gibt an aus wieviel Punkten der Linienzug (Spur) bestehen soll.
i.B. ANZPKT = 9.

3. Maximale Abstände von Nachbarpunkten



AMPLX (= Max. Abstand in x-Richtung)
AMPLY (= Max. Abstand in y-Richtung)
i.B. AMPLX = 9.6 mm
AMPLY = 4.8 mm

4. Beginn der Bewegung



STARTX (= Startwert in x-Richtung)
STARTY (= Startwert in y-Richtung)
i.B. STARTX = 11.2 mm
STARTY = 4. mm

5. Figur der Bewegung



PERIOD (= Sequenzwiederholung, Periode)
diese Zahl gibt an nach wieviel Punkten sich der Linienzug wiederholen soll. Trifft dieser auf eine Bereichs-

grenze (X1, X2, Y1, Y2) wird er „zurückgelenkt“.

i.B. PERIOD = 3.

Die Aufruffolge:
CALL SNEBRW (X1, X2, Y1, Y2, ANZPKT, AMPLX, AMPLY, STARTX, STARTY, PERIOD)

3.3.1.8. Die Routinen SNEBXS und SNEBYS

Die Routinen SNEBXS und SNEBYS sind Erweiterungen der Routine SNEBRW. Mit Ihnen ist es möglich die mit SNEBRW erzeugten Linienzüge als Ganzes oder Teile davon um einen konstanten oder „dynamisch“ veränderlichen Abstand in x-Richtung (SNEBXS) oder in y-Richtung (SNEBYS) zu verschieben. Da die Spur von SNEBRW relativ stark zufallsbedingt ist kann durch die Wiederholung des Linienzugs mit den Routinen SNEBXS und SNEBYS ein ordnender Effekt erreicht werden.

Primär wurden die Routinen SNEBXS und SNEBYS aus didaktischen Überlegungen für den Nichtprogrammierer aufgenommen, sie bieten jedoch auch für den Programmierer möglicherweise vereinfachte Anwendungen.

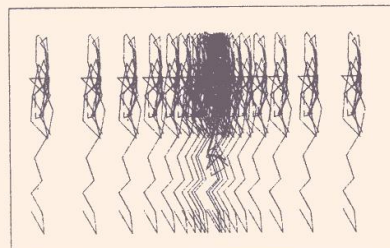


Abb. 49:
CALL BLATT (80., 50.)
CALL SNEBXS (40., 45., 4., 45., 250., 4., 5., 0., 9., 4., 3., 77., 99., 1., 50., 2., -1.2)

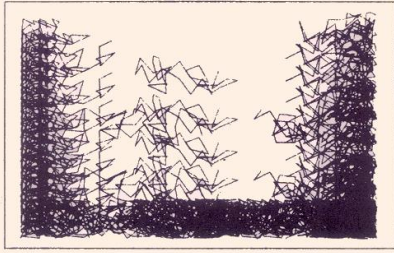
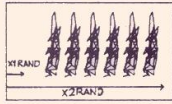


Abb. 50:
CALL BLATT (80., 50.)
DO 4 I = 1,10
SX = SNERAD (3., 76., 999.)
V = SNERAD (2., 4., 999.)
R = SNERAD (1., 2., 999.)
4 CALL SNEBYS (3., 76., 2., 10., 200., 4., 4., SX, 0., 999., 0., 48.,
99., 1., 35., V, R)

Die Routinen SNEBXS und SNEBYS benötigen neben den Parametern der Routine SNEBRW zusätzlich folgende 7 Parameter:

6. Bereichsgrenzen der Gesamtgraphik für SNEBXS



X1RAND (= Bereichsbeginn in x-Richtung)
X2RAND (= Bereichsende in x-Richtung)

i.B. X1RAND = 3.2 mm
X2RAND = 32. mm

analog die Bereichsgrenzen für SNEBYS:

Y1RAND (= Bereichsbeginn in y-Richtung)
Y2RAND (= Bereichsende in y-Richtung)

149

Wie bei der Routine SNEBRW wird ein Linienzug aufgebaut, lediglich verläuft er bei SNERKT als waagrecht und senkrecht wechselnde Strichfolge und gleicht somit einer Treppenform, oder einem sich überschneidenden Labyrinth. Da der Weg des Linienzugs unmöglich vorherbestimmt werden kann, werden mit SNERKT, genau wie mit SNEBRW, theoretisch alle überhaupt denkbaren Graphiken möglich. Jedoch ist durch die Rechteckgenerator Eigenschaft von SNERKT der Gestaltungsrahmen insoweit auf die horizontale und vertikale Dimension eingengt.

Gerät der Linienzug bei SNERKT außerhalb des angegebenen Gestaltungsbereichs, so wird er – im Unterschied zu SNEBRW – nicht „zurückgerichtet“ sondern „abgeschnitten“. Deshalb werden bei mit SNERKT erstellten Graphiken meist zusätzliche FORTRAN Befehlsfolgen eingesetzt, um die Routine mit veränderten Parametern zu wiederholen.

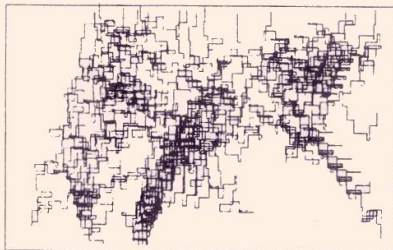
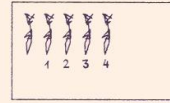


Abb. 51:
CALL BLATT (80., 50.)
DO 2 I = 1,40
SX = SNERAD (5., 75., 999.)
SY = SNERAD (5., 45., 999.)
2 CALL SNERKT (3., 77., 2., 48., 100., 3., 3., SX, SY, 20.)

* siehe: Georg Nees „generative Computergraphik“ Siemens Verlag

151

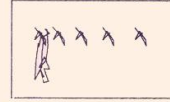
7. Zahl der Verschiebungen (Wiederholungen)



ANZAHL (= Max. Zahl der Verschiebungen)

i.B. ANZAHL = 4.

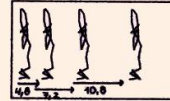
8. Abgrenzen auf einen Teil (Ausschnitt) des Linienzugs



APUKTE (= Anfangspunkt zum Verschieben)
EPUKTE (= Endpunkt zum Verschieben)

i.B. APUKTE = 1.
EPUKTE = 7.

9. Angabe der Verschiebesteuerung



VERSCH, RICHTST (=Verschiebesteuerung)

die Verschiebung wird berechnet aus:

$VERSCH \leftarrow VERSCH * RICHTST$

i.B. $VERSCH = 3.2 \text{ mm}$
 $RICHTST = 1.5 \text{ mm}$

d.h. $4.8 = 3.2 * 1.5$
 $7.2 = 4.8 * 1.5$
 $10.8 = 7.2 * 1.5 \text{ u.s.f.}$

Die Aufruffolgen:

CALL SNEBXS (X1, X2, Y1, Y2, ANZPKT, AMPLX, AMPLY, STARTX, STARTY, PERIOD, X1RAND, X2RAND, ANZAHL, APUKTE, EPUKTE, VERSCH, RICHTST)

CALL SNEBYS (X1, X2, Y1, Y2, ANZPKT, AMPLX, AMPLY, STARTX, STARTY, PERIOD, Y1RAND, Y2RAND, ANZAHL, APUKTE, EPUKTE, VERSCH, RICHTST)

3.3.1.9. Die Routine SNERKT

Mit der Routine SNERKT wird in der SNE COMP ART ein Rechteckgenerator zur Verfügung gestellt, der über die von Georg Nees* bekannten Funktionen hinaus Gestaltungsmöglichkeiten bietet.

150

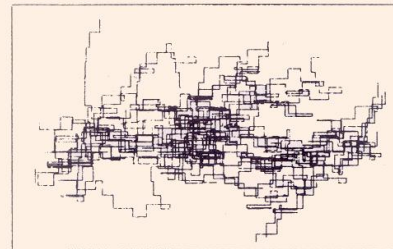
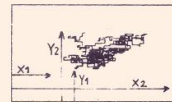


Abb. 52:
CALL BLATT (80., 50.)
DO 1 I = 1,30
SX = SNERAD (5., 75., 99.)
1 CALL SNERKT (3., 77., 3., 47., 50., 5., 3., SX, 20., 999.)

Die Routine SNERKT benötigt 10 Angaben:

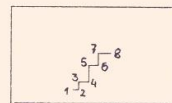
1. Teilbereich in dem der Linienzug liegen soll



X1, X2 (= Bereichsgrenzen in x-Richtung)
Y1, Y2 (= Bereichsgrenzen in y-Richtung)

i.B. X1 = 8. mm X2 = 32. mm
Y1 = 6.4 mm Y2 = 14.4 mm

2. Maximale Anzahl von Punkten des Linienzugs

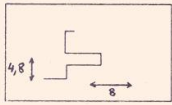


ANZPKT (= Max. Anzahl der Punkte)
dieser Wert gibt an aus wieviel Punkten der Linienzug höchstens bestehen soll.

i.B. ANZPKT = 8.

152

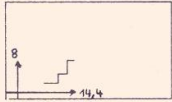
3. Maximale Abstände von Nachbarpunkten



AMPLX (= Max. Abstand in x-Richtung)
 AMPLY (= Max. Abstand in y-Richtung)

i.B. AMPLX = 8. mm
 AMPLY = 4.8 mm

4. Beginn der Bewegung



STARTX (= Startwert in x-Richtung)
 STARTY (= Startwert in y-Richtung)

i.B. STARTX = 14.4 mm
 STARTY = 8. mm

5. Figur der Bewegung



PERIOD (= Sequenzwiederholung, Periode)

dieser Wert gibt an nach wieviel Punkten sich der Linienzug wiederholen soll.

i.B. PERIOD = 4.

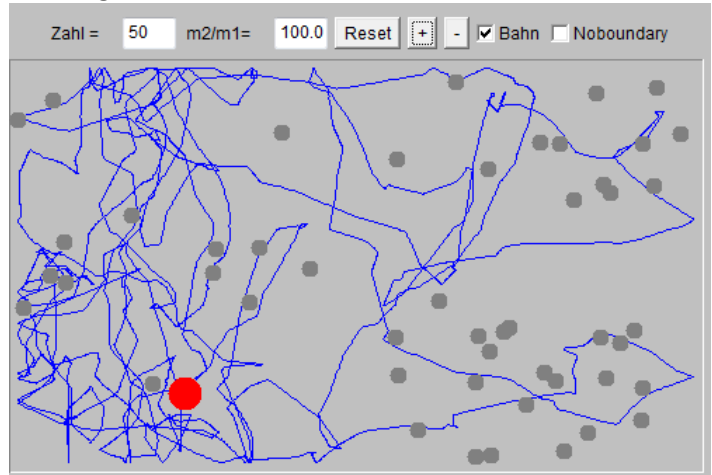
Die Aufruffolge:
 CALL SNERKT (X1, X2, Y1, Y2, ANZPKT, AMPLX, AMPLY, STARTX, STARTY, PERIOD)



2010: Wenn Sie auf Ihrem PC Java installiert haben: <http://java.com>, dann können Sie leicht eine Brownsche Bewegung im Browser simulieren. Einfach „googlen“ nach:

Java-Applet von Fu-Kwun Hwang Brownsche Molekularbewegung

Das Programmfenster sieht so aus:



Die Produktion von Kunst basiert hier (1976) auf Parametern die an die Routine SNExxx („die Kunstmaschine“) zur pflichtgemäßen Anwendung übergeben werden. Kunst entsteht durch „Auswahl“ eines Realisats und Benennung dessen als „Softwarekunst“.

30 Jahre später blicke ich auf diese Werke in einer Simulationswelt die ich und andere „Identitätseinheiten“ bewohnen: <http://simulacron-1.com>

