

# Fraktales Wachstum

Java

Alfred Nussbaumer

Mit JAVA lassen sich im Handumdrehen plattformunabhängige Applikationen oder Applets für Internetdokumente erstellen. Bereits mit einfachen Hilfsmitteln lassen sich anspruchsvolle Themen der Informatik behandeln. In diesem Beitrag sollen Wachstumsvorgänge durch zufälliges Anlagern von Teilchen besprochen werden. Entsprechende Programme verlangten in der Vergangenheit eine enorme Rechnerleistung, die heute auf handelsüblichen PCs zur Verfügung steht. Im Gegensatz zu den früher verwendeten Programmiersprachen erlaubt JAVA, solche Programme als Applets in eine HTML-Dokumentation einzubauen (vgl. [6]).

## 1. Random Walk

Leonard M. Sander und T.A. Witten haben Mitte der Achzigerjahre beschrieben, wie sich Teilchen an Flächen oder bereits festsetzenden Teilchen anlagern (siehe [1]): Entscheidend ist dabei die Idee, ein Teilchen willkürlich nach verschiedenen Richtungen taumeln zu lassen - etwa ähnlich der Brown'schen Molekularbewegung. Erst wenn es auf ein festes Objekt trifft, setzt es sich als Nachbarpartikel fest. Wir können diesen an sich räumlich ablaufenden Vorgang auf eine zweidimensionale Querschnittsfläche vereinfachen und beschreiben die aktuelle Position des Teilchens mit den beiden Koordinaten  $x$  und  $y$ . Die zufällige Taumelbewegung lässt sich realisieren, indem wir acht verschiedene, gleich wahrscheinliche Zufallszahlen erzeugen und das Teilchen davon abhängig in acht verschiedene Richtungen weiterwandern lassen:

```
int taumel = (int) (zufallszahl.nextDouble()*8);
// zufällige Ganzzahl zwischen 0 und 7
if (taumel == 0) x+=1; // Schritt nach rechts
if (taumel == 1) y+=1; // Schritt nach unten
if (taumel == 2) x-=1; // Schritt nach links
if (taumel == 3) y-=1; // Schritt nach oben
if (taumel == 4) { x+=1; y+=1; } // diagonal nach rechts unten
if (taumel == 5) { x+=1; y-=1; } // diagonal nach rechts oben
if (taumel == 6) { x-=1; y-=1; } // diagonal nach links oben
if (taumel == 7) { x-=1; y+=1; } // diagonal nach links unten
```

Bereits mit einfachen Hilfsmitteln der Java-Programmierung lässt sich dieser Vorgang simulieren.

## 2. Anlagerung von Teilchen

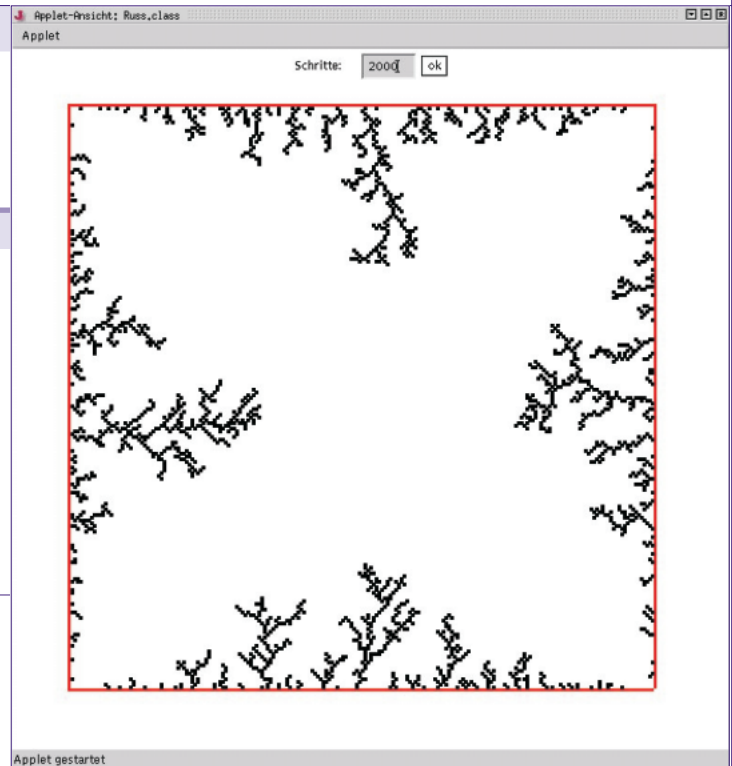
In der Mitte der Querschnittsfläche eines quadratischen Raumes werden Russteilchen freigesetzt, die sich in der schon beschriebenen Taumelbewegung nach allen Richtungen über die Fläche bewegen. Trifft ein Teilchen die Wand oder ein festsetzendes Teilchen haftet es fest, und das nächste Teilchen wird freigesetzt. Die Information über die Position der Gefäßwand und der bereits angelagerten Teilchen wird in einem zweidimensionalen Array (`baum[][]`) gespeichert.

Interessanterweise erhalten wir fraktale Strukturen. Dabei ist allerdings zu beachten, dass die Bewegung und Anlagerung der Teilchen im schwerelosen Raum simuliert wurde.

## 3. Programmcode (Russ.java)

Wie schon erwähnt, kann das angegebene Applet mit einfachen Hilfsmitteln realisiert werden. Im folgenden Beispiel wird das Layout mit Ein- und Ausgabe in der Methode `init()` erzeugt; die Methode `actionPerformed()` wird so überschrieben, dass die Eingabe in das Textfeld `eingabe` beim Drücken der Schaltfläche `ok` als Ganzzahl in die Variable `teilchenzahl` übernommen und die Methode `paint()` neu aufgerufen wird. In der Methode `paint()` steht der schon beschriebene Algorithmus zur Taumelbewegung und zur Anlagerung der Teilchen:

```
import java.awt.*;
import java.applet.*;
```



Russ

```
import java.awt.event.*;
import java.util.Random;
public class Russ extends Applet
implements ActionListener {
    Random zufallszahl;
    int teilchenzahl;
    Button ok;
    TextField eingabe;
    public void init () {
        setBackground(Color.white);
        setLayout(new FlowLayout());
        add(new Label("Anzahl der Teilchen: "));
        eingabe = new TextField("1000", 5);
        add(eingabe);
        ok = new Button("ok");
        add(ok);
        ok.addActionListener(this);
        teilchenzahl = Integer.parseInt(eingabe.getText());
    }
    public void actionPerformed (ActionEvent ae) {
        if (ae.getSource() == ok) {
            teilchenzahl = Integer.parseInt(eingabe.getText());
            repaint();
        }
    }
    public void paint(Graphics bs) {
        int x;
        int y;
        int baum[][] = new int[250][250];
        for (int i=0; i<250; i++)
            for (int j=0; j<250; j++)
                baum[i][j]=0;
        bs.setColor(Color.white);
        bs.fillRect(0,50,250,230);
        // Quadrat (= Begrenzung) festlegen und ausgeben
        bs.setColor(Color.red);
        x = 20;
        for (int i=0; i<210; i++) {
            y = i + 20;
            baum[x][y] = 1;
            bs.fillRect(2*x,2*y,2,2);
        }
        y = 20;
        for (int i=0; i<210; i++) {
            x = i + 20;
            baum[x][y] = 1;
            bs.fillRect(2*x,2*y,2,2);
        }
        x = 230;
        for (int i=0; i<210; i++) {
            y = i + 20;
            baum[x][y] = 1;
            bs.fillRect(2*x,2*y,2,2);
        }
        y = 230;
```

```

for (int i=0; i<210; i++) {
    x = i + 20;
    baum[x][y] = 1;
    bs.fillRect(2*x,2*y,2,2);
}
// Teilchen erzeugen, taumeln lassen, die
// Position des angelagerten Teilchens speichern
// und ausgeben ...
bs.setColor(Color.black);
zufallszahl = new Random();
// Schleife über die festgelegte Teilchenzahl
for (int i=1; i<teilchenzahl; i++) {
    boolean erledigt = false;
    // Startposition in der Mitte des Quadrates
    x=125;
    y=125;
    // Schleife für Taumelbewegung
    while (!erledigt) { int taumel =
        (int)(zufallszahl.nextDouble()*8);
        if (taumel == 0) x+=1;
        if (taumel == 1) y+=1;
        if (taumel == 2) x-=1;
        if (taumel == 3) y-=1;
        if (taumel == 4) { x+=1; y+=1; }
        if (taumel == 5) { x+=1; y-=1; }
        if (taumel == 6) { x-=1; y-=1; }
        if (taumel == 7) { x-=1; y+=1; }

        // Testen, ob das Nachbarfeld ein Wandelement
        // oder ein angelagertes Teilchen enthält
        if (baum[x-1][y+1]==1 || baum[x-1][y]==1 ||
            baum[x-1][y-1]==1 || baum[x][y+1]==1 ||
            baum[x][y-1]==1 || baum[x+1][y+1]==1 ||
            baum[x+1][y]==1 || baum[x+1][y-1]==1){
            baum[x][y]=1;
            bs.fillRect(2*x,2*y,3,3);
            erledigt = true;
            // "Taumel"-Schleife verlassen
        }
    } // Ende der "Taumel"-Schleife
} // Ende der Schleife über die Teilchenzahl
} // Ende paint()
    
```

Das Applet wird mit dem <APPLET>-Tag in die HTML-Datei eingefügt:

```

<APPLET CODEBASE="CLASSES" CODE="RUSS.CLASS"
    WIDTH="500" HEIGHT="500">
</APPLET>
    
```

Der Attributwert des Attributes CODEBASE gibt das Verzeichnis an, in dem die Klassendatei des Applets gespeichert ist; das Attribut CODE enthält den Klassennamen.

#### 4. Wachstumsverhalten (Wachstum.java)

Warum bilden die Teilchen beim Anlagern keine volle Fläche aus? Warum bilden sich keine geschlossenen Äste? Wir untersuchen dazu die Anlagerung von Teilchen an eine Gerade:

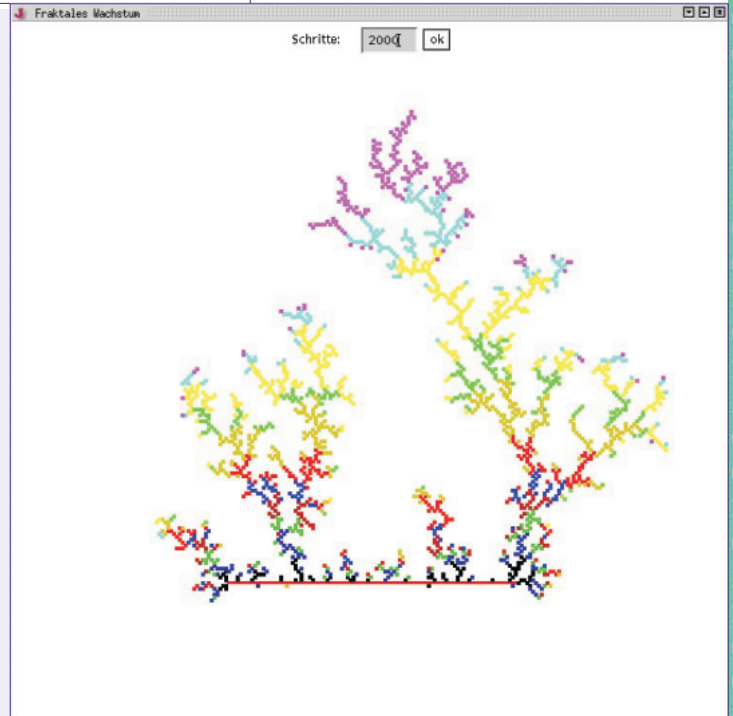
Im obigen Beispiel wurde der zeitliche Ablauf bei der Anlagerung der Teilchen mit Hilfe verschiedener Farben dargestellt. Dazu wird nach einer bestimmten Anzahl von Teilchen die Zeichenfarbe gewechselt:

```

for (int i=1; i<teilchenzahl; i++) {
    boolean erledigt = false;
    x = 125;
    y = 25;
    if (i>teilchenzahl*0.05) bs.setColor(dblue);
    if (i>teilchenzahl*0.1) bs.setColor(dgreen);
    if (i>teilchenzahl*0.15) bs.setColor(dred);
    if (i>teilchenzahl*0.2) bs.setColor(Color.blue);
    if (i>teilchenzahl*0.25) bs.setColor(Color.red);
    if (i>teilchenzahl*0.35) bs.setColor(dyellow);
    if (i>teilchenzahl*0.5) bs.setColor(Color.green);
    if (i>teilchenzahl*0.6) bs.setColor(Color.yellow);
    if (i>teilchenzahl*0.8) bs.setColor(Color.cyan);
    if (i>teilchenzahl*0.9) bs.setColor(Color.magenta);
    while (!erledigt) {
        ...
    }
}
    
```

Dabei sind die Grenzen für die einzelnen Farben willkürlich gewählt. Jedenfalls ist beispielsweise zu erkennen, dass längere Äste bei der Anlagerung bevorzugt werden, während Teilchen kaum mehr in "Täler" gelangen.

Erreicht ein Teilchen beim Taumeln den Rand der Ausgabefläche, so kann es entweder eliminiert werden (vgl. [2]), oder man schließt die Fläche zu einem Ring: Kann die x-Koordinate bei-



#### Wachstum

spielsweise im Intervall [20;230] variieren, weist man ihr den Wert 230 zu, wenn sie die linke Grenze 20 unterschreitet und umgekehrt.

```

if (x<20) x=230;
if (x>230) x=20;
if (y<20) y=20;
if (y>250) y=250;
    
```

Nach oben oder nach unten taumelnde Teilchen werden einfach an den Grenzen y=20 bzw. y=250 zurück gehalten. Auf diese Weise erhält man ausgewogene Fraktale, auch wenn alle Teilchen von einer gleichbleibenden Startposition ausgehen.

#### 5. Fraktales Korallenwachstum (Koralle.java)

Im letzten Beispiel lassen wir Teilchen an einem "Kondensationskeim" oder "Kristallisationskeim" anlagern:

Um zu erreichen, dass die Teilchen sich möglichst gleich wahrscheinlich an allen Ästen anlagern, bilden wir aus der quadratischen Fläche einen Torus:

```

if (x<20) x=230;
if (x>230) x=20;
if (y<20) y=230;
if (y>230) y=20;
    
```

Als Startposition wurde für das oben angegebene Fraktal der feste Punkt (125/25) gewählt.

```

for (int i=1; i<teilchenzahl; i++) {
    boolean erledigt = false;
    x = 125;
    y = 25;
    ...
}
    
```

Um das zufällige Anlagern von Teilchen noch zu verbessern wird beispielsweise vorgeschlagen, die Teilchen in einer beliebigen Richtung aber festen Entfernung vom Kristallisationskeim, also auf einer Kreislinie, starten zu lassen. Die Startposition auf dieser Kreislinie wird durch einen Zufallswinkel zwischen 0° und 360° bestimmt:

```

for (int i=1; i<teilchenzahl; i++) {
    boolean erledigt = false;
    int winkel = (int) (zufallszahl.nextDouble()*360);
    x = (int) (100 * Math.cos(winkel)+125);
    y = (int) (100 * Math.sin(winkel)+125);
    ...
}
    
```

Interessanterweise verbessert dies das Aussehen des Fraktals nur geringfügig, wenn die Teilchen auf einem Torus taumeln.

# Elektronik-Händler

<b>Holz Elektronik</b>	<b>1030</b>
Wien Hohlweggasse 30/2 Tel.: 01-799604-40 Fax: 799604-444 <a href="http://www.holz-elektronik.de/">http://www.holz-elektronik.de/</a> <b>Infineon-Bauteile</b>	
<b>Print-Technik</b>	<b>1062</b>
Wien Stumpergasse 34 Tel.: 01-5973423 Fax: 5973423-8 print@gps.at <a href="http://www.gps.at/">http://www.gps.at/</a>	
<b>Technotronic</b>	<b>1070</b>
Wien Zieglergasse 27 Tel.: 01-5236204 Fax: 5236205 ver- kauf@technotronic.at <a href="http://www.technotronic.at/">http://www.technotronic.at/</a>	
<b>Gepex</b>	<b>1090</b>
Wien Nußdorfer Straße 27 electronic Tel.: 01-3198881 Fax: 3198881-23 office@gepex.at <a href="http://www.gepex.at/">http://www.gepex.at/</a>	
<b>Funktechnik Kuso</b>	<b>1100</b>
Wien Waldgasse 26 Tel.: 01-6043040 Fax: 6045575	
<b>Technotronic</b>	<b>1100</b>
Wien Laxenburger Straße 74 Tel.: 01-6029525 Fax: 6064794 verkauf@technotronic.at <a href="http://www.technotronic.at/">http://www.technotronic.at/</a>	
<b>Wien Schall</b>	<b>1120</b>
Wien Krichbaumgasse 25 Tel.: 01-81155-202 Fax: 81155-180 org@wienschall.com <a href="http://www.wienschall.com/">http://www.wienschall.com/</a>	
<b>EBV Elektronik</b>	<b>1150</b>
Wien Diefenbachgasse 35/1 Tel.: 01-89152-0 Fax: 89152-30 <a href="http://www.ebv.com">http://www.ebv.com</a> <b>Infineon-Bauteile</b>	
<b>Silica</b>	<b>1150</b>
Wien Diefenbachgasse 35-2 Tel.: 01-86642-300 Fax: 86642-350 silica.wien@avnet.com <a href="http://www.silica.com">http://www.silica.com</a> <b>Infineon-Bauteile</b>	
<b>Distrelec</b>	<b>1200</b>
Wien Leithastrasse 25 Tel.: 01-3341010 Fax: 3341010-99 info-at@distrelec.com <a href="http://www.distrelec.at/">http://www.distrelec.at/</a> <b>Versand</b>	
<b>Computerkabel Kaminek</b>	<b>1210</b>
Wien Leopoldauerstraße 20 und 24 Tel.: 01-2700000 office@kaminek.at <a href="http://www.kaminek.at/">http://www.kaminek.at/</a>	
<b>Technotronic</b>	<b>1210</b>
Wien Floridsdorfer Hauptstraße 23 Tel.: 01-2706520 Fax: 2706817 verkauf@technotronic.at <a href="http://www.technotronic.at/">http://www.technotronic.at/</a>	
<b>Conrad Electronic</b>	<b>1220</b>
Wien Gewerbeparkstraße 12 Tel.: 01-7347990-0 Fax: 7347990-14 filiale.wien2@conrad.at <a href="http://www.conrad.at/">http://www.conrad.at/</a>	
<b>Conrad Electronic</b>	<b>2334</b>
Vösendorf SCS-Nordring 2 Tel.: 01-6985500-0 Fax: 6985500-14 filiale.wien@conrad.at <a href="http://www.conrad.at/">http://www.conrad.at/</a>	
<b>RS-Components</b>	<b>3950</b>
Gmünd Albrechtserstraße 11 Tel.: 02852-505, 53 765, 53 2 23 Fax: 53 223, 54 650 verkauf@rs-components.at <a href="http://www.rs-components.at/">http://www.rs-components.at/</a> <b>Versand</b>	
<b>IGS Electronic</b>	<b>4041</b>
Linz Pfeiferstraße 7 Tel.: 0732-733128 Fax: 736040 info@igs-electronic.at <a href="http://www.igs-electronic.at/">http://www.igs-electronic.at/</a>	
<b>Conrad Electronic</b>	<b>4600</b>
Wels Durisolstraße 2, Postfach 293 Tel.: 07242-203040 Fax: 203044 <a href="http://www.conrad.at/">http://www.conrad.at/</a> <b>Versand</b>	
<b>Gretter Technica Versand</b>	<b>4943</b>
Greinberg Postfach 3 Tel.: 07723-21421 Fax: 21421-40 servi- ce@gretter.at <a href="http://www.gretter.at/">http://www.gretter.at/</a> <b>Versand</b>	
<b>Wien Schall</b>	<b>5020</b>
Salzburg Untersbergstrasse 2a Tel.: 0662-844305 Fax: 844305-22 g.steiner@wienschall.com <a href="http://www.wienschall.com/">http://www.wienschall.com/</a>	
<b>Conrad Electronic</b>	<b>8054</b>
Graz Kärntner Straße 228 Tel.: 0316-286464-0 Fax: 286464-10 filiale.graz@conrad.at <a href="http://www.conrad.at/">http://www.conrad.at/</a>	

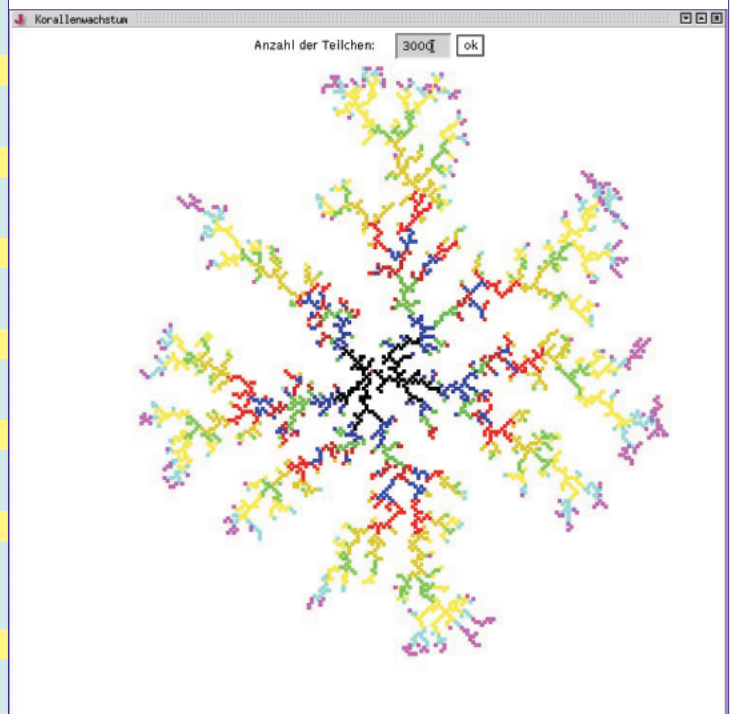
## 6. Erweiterungen

Die angegebenen Beispiele können im Rahmen der Unterrichtsarbeit, für Haus- oder Projektarbeiten in vielfältiger Weise erweitert werden:

- Der Programmcode soll hinsichtlich des Layouts des Java-Applets verbessert und in Bezug auf Benutzerhilfe und Fehlerbehandlung erweitert werden.
- Eigene Beobachtungen sollen gemeinsam mit entsprechenden Applets in HTML-Dokumentationen zusammengefasst werden.
- Das fraktale Wachstum (auch "diffusion-limited aggregation", "diffusionsbegrenzte Aggregation" genannt) soll hinsichtlich der technischen Relevanz untersucht werden.

## 7. Literatur, Weblinks

- [1] Spektrum der Wissenschaft, März 1987, Leonard M. Sander, "Fraktales Wachstum"
- [2] A.K.Dewdney, "Computer-Kurzweil" in "Spektrum der Wissenschaft", Februar 1989, S. 8
- [3] <http://apricot.polyu.edu.hk/~lam/dla/dla.html> (Ausgezeichnetes Applet zu "Korallenwachstum" mittels diffusionsbegrenzter Aggregation)
- [4] <http://www.swin.edu.au/chem/bio/fractals/dla01.htm> (Beschreibung diffusionsbegrenzter Aggregation und Untersuchung der fraktalen Dimension)
- [5] <http://www.uche.de/~ecotopia/dla/>, Beschreibung und zahlreiche Links zur DLA (diffusion-limited aggregation)
- [6] <http://www.gymmelk.ac.at/~nus/informatik/wpj/JAVA/index.php?kat=appt&teil=russ> (Applet zur Simulation der Anlagerung von Russteilchen an Gefäßwände, Codebeispiel)



Koralle