PYTHAGORÄISCHE BÄUME

Wolf Bayer
Hans-und-Hilde-Coppi-Gymnasium Berlin

26. November 2009

Der holländische Ingenieur A.E. Bosman (1891-1961) war im 2. Weltkrieg zur Konstruktion von U-Booten eingesetzt. Vielleicht aus Langeweile, zur Abwechslung, konstruierte er oft statt U-Booten pythagoräische Bäume auf seinem Zeichenbrett. Seine Graphiken wurden 1957 publiziert. Wir fanden sie in dem Buch der Bremen-Gruppe um Peitgen.
Mit der wohlbekannten geometrischen Figur zum Satz des Pythagoras mit Hypothenusen-Quadrat und Katheten-Quadraten lassen sich faszinierende Figuren erzeugen, sogenannte mathematische Monster, die die Wirkung von Rekursiv-Algorithmen anschaulich demonstrieren. Sie geben einen Einblick in die Algorithmen komplexer dynamischer Systeme. Über die fraktale Dimension dieser Monster ist bisher nichts bekannt.

Diese Unterrichtsequenz bietet sich an für den Informatik-Unterricht zu Thema Rekursion. Sie kann aber auch im Rahmen von Seminarkursen zur Fraktalen Geometrie Verwendung finden.

1 Pythagoräische Ketten

Betrachten Sie Bild 1 und verifizieren Sie: Sind der Winkel α und die beiden Punkte $P_1(x_1, y_1), P_2(x_2, y_2)$ bekannt, so lassen sich alle anderen Punkte bestimmen, die Figur kann gezeichnet werden. In Bild 2 sieht man, wie die größere Kathete der Ausgangs-Figur als Hypothenuse für die Folge-Figur genommen wurde. Und deren größere Kathete wieder für die nächste Figur usw. Und man erkennt auch, dass der Algorithmus vollständig durch den Winkel α und die 5 Punkte $P_1(x_1, y_1), P_2(x_2, y_2), \ldots, P_5(x_5, y_5)$ bestimmt ist. Von P_4 und P_5 an Stelle von P_1 und P_2 aus wird jeweils die nächste Figur berechnet und gezeichnet. Der Algorithmus für die Pythagoras-Kette in Bild 2 lautet damit:

Algorithmus Pytha-Kette

Wähle Winkel α und die Punkte P_1 und P_2
WIEDERHOLE
    Bestimme P_3, P_4 so, dass P_1, P_2, P_3, P_4 ein Quadrat ist
    Zeichne das Quadrat P_1, P_2, P_3, P_4
    Bestimme P_5 mit α und Thaleskreis über $\overline{P_3 P_4}$
    Ersetze P_1 durch P_4

Ersetze P_2 durch P_5
BIS Abstand $|\overline{P_1P_2}|$ klein genug.

Mit der Top-Down-Methode präzisieren wir nun die Bausteine dieses Algorithmus.

1. Das Quadrat: Dies erzeugt man einfach: Man nehme den Vektor $\vec{P_1P_2}$, drehe ihn um 90° und addiere diesen Vektor zu den Ortsvektoren $\vec{OP_1}$ und $\vec{OP_2}$ hinzu.

$$\begin{aligned}
\vec{OP_3} = \begin{pmatrix} x_3 \\ y_3 \end{pmatrix} &= \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} (\vec{OP_2} - \vec{OP_1}) + \vec{OP_2} \\
&= \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \end{pmatrix} + \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \\
&= \begin{pmatrix} y_1 - y_2 + x_2 \\ x_2 - x_1 + y_2 \end{pmatrix} \qquad (1) \\
\vec{OP_4} = \begin{pmatrix} x_4 \\ y_4 \end{pmatrix} &= \begin{pmatrix} y_1 - y_2 + x_1 \\ x_2 - x_1 + y_1 \end{pmatrix} \qquad (2)
\end{aligned}$$

Nun kann das Quadrat gezeichnet werden, zeichne die Linien $\overline{P_1P_2}$, $\overline{P_2P_3}$, $\overline{P_3P_4}$, $\overline{P_4P_1}$.

2. Spitze: Nun ist die Spitze P_5 der Pythagorasfigur zu ermitteln. Am einfachsten ist es, den Mittelpunkt M der Linie $\overline{P_3P_4}$ zu bestimmen, den Vektor $\vec{MP_3}$ um den Winkel α zu drehen und dann diesen zum Ortsvektor von M zu addieren.

$$\begin{aligned}
\vec{OP_5} = \begin{pmatrix} x_5 \\ y_5 \end{pmatrix} &= \begin{pmatrix} cos\,\alpha & -sin\,\alpha \\ sin\,\alpha & cos\,\alpha \end{pmatrix} (\vec{OP_3} - \vec{OM}) + \vec{OM} \\
&= \begin{pmatrix} cos\,\alpha & -sin\,\alpha \\ sin\,\alpha & cos\,\alpha \end{pmatrix} \frac{1}{2}(\vec{OP_3} - \vec{OP_4}) + \frac{1}{2}(\vec{OP_3} + \vec{OP_4}) \\
&= \frac{1}{2}\left[\begin{pmatrix} cos\,\alpha & -sin\,\alpha \\ sin\,\alpha & cos\,\alpha \end{pmatrix} \begin{pmatrix} x_3 - x_4 \\ y_3 - y_4 \end{pmatrix} + \begin{pmatrix} x_3 + x_4 \\ y_3 + y_4 \end{pmatrix}\right] \\
&= \frac{1}{2}\begin{pmatrix} (x_3 - x_4)cos\,\alpha - (y_3 - y_4)sin\,\alpha + x_3 + x_4 \\ (x_3 - x_4)sin\,\alpha + (y_3 - y_4)cos\,\alpha + y_3 + y_4 \end{pmatrix} \qquad (3)
\end{aligned}$$

Jeder der Algorithmen durchläuft eine Folge von Pythagorasfiguren. Wie schon bemerkt, ist jede Figur durch die 11 Parameter $\alpha, P_1(x_1, y_1), P_2(x_2, y_2), \ldots, P_5(x_5, y_5)$ bestimmt. Um diese Werte während des Ablaufs adequat zu verwalten, kreieren wir einen Variablentyp *tfiguren* und eine Variable dieses Typs *pyfigur.* In *pyfigur* legen wir die aktuelle Werte $\alpha, P_1, P_2, \ldots, P_5$ ab. Dies bewirkt die Prozedur *figur-erzeugen;* aus den Eingabewerten $P_1(x_1, y_1), P_2(x_2, y_2)$ und dem Winkel α werden aus den Gleichungen (1), (2) und (3) alle übrigen Werte berechnet.

TYPE tfiguren = RECORD $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, x_5, y_5$,alfa END
VAR pyfigur:tfiguren

PROCEDURE $figur - erzeugen$(VAR pyfigur:tfiguren)
BEGIN pyfigur.x3:=pyfigur.x2+pyfigur.y1-pyfigur.y2 siehe (1)
pyfigur.y3:=pyfigur.y2+pyfigur.x2-pyfigur.x1
pyfigur.x4:=pyfigur.x1+pyfigur.y1-pyfigur.y2 siehe (2)
pyfigur.y4:=pyfigur.y1+pyfigur.x2-pyfigur.x1

pyfigur.x5:=0.5(cos(pyfigur.alfa)(pyfigur.x3-pyfigur.x4) siehe (3)
-sin(pyfigur.alfa)(pyfigur.y3-pyfigur.y4)
+pyfigur.x3+pyfigur.x4)
pyfigur.y5:=0.5(sin(pyfigur.alfa)(pyfigur.x3-pyfigur.x4)
+cos(pyfigur.alfa)(pyfigur.y3-pyfigur.y4)
+pyfigur.y3+pyfigur.y4)
line((round(pyfigur.x1*MF),round(pyfigur.y1*MF),
round(pyfigur.x2*MF),round(pyfigur.y2*MF))
line analog für die Strecken ; $\overline{P_2P_3}$, $\overline{P_3P_4}$, $\overline{P_4P_1}$.
END

Statt der Mehrfachanweisung *WIEDERHOLE ... BIS* in Pytha-Kette verwenden wir das Rekursionsprinzip im Hinblick auf die Bilder 3,4,5. Wenn die Strecke $\overline{P_1P_2}$ zu klein geworden ist, um noch gezeichnet werden zu können, soll die Rekursion abgebrochen werden; d.h. die Anweisungen sollen nur durchlaufen werden, wenn die Bedingung $|\overline{P_1P_2}| > 0.001$ erfüllt ist.
Um den Algorithmus zu realisieren, wählen wir den Winkel α, die Startpunkte P_1, P_2, und einen Maßstabfaktor MF für den Bildschirm.

Initialisierungen
$pyfigur.alfa := \frac{\pi}{3}$,
$pyfigur.x_1 := 1.75$, $pyfigur.y_1 := 0.6$, $pyfigur.x_2 := 2$, $pyfigur.y_2 := 0.6$
und $MF := 400$.

PROC pythakette
IF $|pyfigur.x_1 - pyfigur.x_2| > 0.001$ OR
$|pyfigur.y_1 - pyfigur.y_2| > 0.001$ {Abbruch der Rekursion}
THEN $figur - erzeugen$(VAR pyfigur:tfiguren)
$pyfigur.x_1 := pyfigur.x_4$, $pyfigur.y_1 := pyfigur.y_4$, $\{P_1 \leftarrow P_4\}$
$pyfigur.x_2 := pyfigur.x_5$, $pyfigur.y_2 := pyfigur.y_5$, $\{P_2 \leftarrow P_5\}$
pythakette {Rekursions-Aufruf}

Aufgabe 1 *Realisieren Sie den Algorithmus* pythakette *mit den angegebenen Werten. Bild 2 sollte dem Bildschirm erscheinen.*

Aufgabe 2 *Variieren Sie* α *:* $0, \frac{\pi}{8}, \frac{\pi}{6}, \frac{\pi}{5}, \frac{\pi}{4}, \frac{\pi}{2}, \dots$

2 Pythagoräische Bäume

Wir kommen zum Algorithmus für den Pythagoräischen Baum auf dem Titelblatt. Im Unterschied zu dem in Bild 2 beginnt an beiden Katheten eine Kette von Pythagoras-Figuren. Das bedeutet, unser Algorithmus wird zwei Rekursions-Aufrufe enthalten, und deshalb müssen die Werte der Punkte P_3 und P_5 für den zweiten Aufruf über den 1. Rekursions-Aufruf hinüber gerettet werden. Sie werden zwischegespeichert in Hilfvariablen u_3, v_3, u_5, v_5. Wir erweitern die PROC Pythakette zur PROC Pythabaum.

PROC pythabaum
IF $|pyfigur.x_1 - pyfigur.x_2| > 0.001$ OR
$|pyfigur.y_1 - pyfigur.y_2| > 0.001$ {Abbruch der Rekursion}
THEN $figur - erzeugen$(VAR pyfigur:tfiguren)
$u_3 := pyfigur.x_3, \quad v_3 := pyfigur.y_3$ {Werte sichern}
$u_5 := pyfigur.x_5, \quad v_5 := pyfigur.y_5$
$pyfigur.x_1 := pyfigur.x_4, \quad pyfigur.y_1 := pyfigur.y_4,$ $\{P_1 \leftarrow P_4\}$
$pyfigur.x_2 := pyfigur.x_5, \quad pyfigur.y_2 := pyfigur.y_5,$ $\{P_2 \leftarrow P_5\}$
pythabaum {1. Rekursions-Aufruf}
$pyfigur.x_1 := u_5, \quad pyfigur.y_1 := v_5,$ $\{P_1 \leftarrow P_5\}$
$pyfigur.x_2 := u_3, \quad pyfigur.y_2 := v_3,$ $\{P_2 \leftarrow P_3\}$
pythabaum {2. Rekursions-Aufruf}

Aufgabe 3 *Realisieren Sie den Algorithmus* pythabaum *mit den angegebenen Werten. Bild 3 wird auf dem Bildschirm erscheinen.*

Aufgabe 4 *Variieren Sie Parameter α. in* pythabaum.

Aufgabe 5 *Vertauschen Sie in den Wertzuweisungen zwischen den Rekursionsaufrufen die Punkte, d.h.* $P_1 \leftarrow P_3$, $P_2 \leftarrow P_5$, *so erhalten Sie ein Seepferdchen-ähnliches Fraktal.*

Aufgabe 6 Broccoli-Baum
Betrachten Sie Bild 4. Der Vektor $\vec{MP_5}$ *wurde mit dem Faktor* $\lambda := 0.3$ *verkürzt. Entwickeln Sie die Formeln* (3) *mit diesem Faktor zu Formeln (3') und fügen Sie diese ein. Startwerte:* $P_1(0.4/0.1)$, $P_2(0.7/0.1)$, $\alpha := \frac{\pi}{2}$ *und* $\lambda := 0.3$.

Aufgabe 7 *Betrachten Sie den* alternierenden Pythagoras-Baum *in Bild 5. Analysieren Sie die Logik seines Algorithmus, entwerfen und implementieren Sie ihn.*

Aufgabe 8 *Lassen Sie weitere, noch unbekannte pythagoräische Bäume wachsen.*

3 Literatur

Herrmann, Dietmar *Algorithmen für Chaos und Fraktale*
Addison-Wesley Bonn (1994) (vergriffen)

Peitgen, H.-O., Jürgens, H., Saupe, D. Bremen Gruppe
Bausteine des Chaos - Fraktale
Springer-Klett-Cotta Berlin Heidelberg Stuttgart (1992)

PYTHAGORÄISCHE BÄUME

Abbildungen

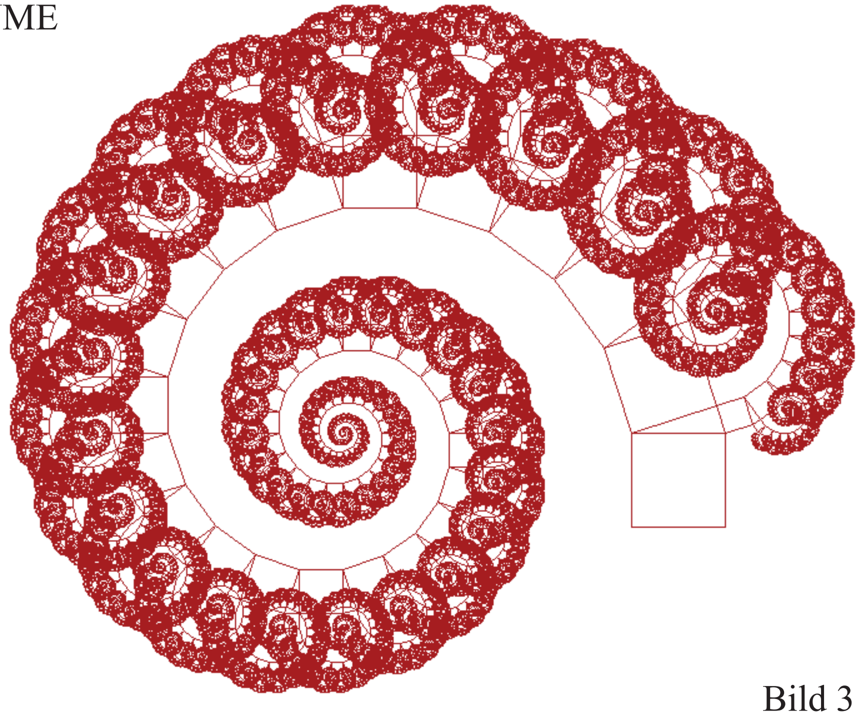

Bild 3

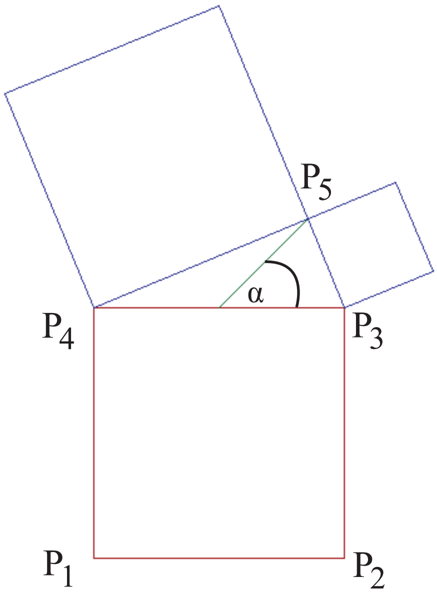

Bild 1

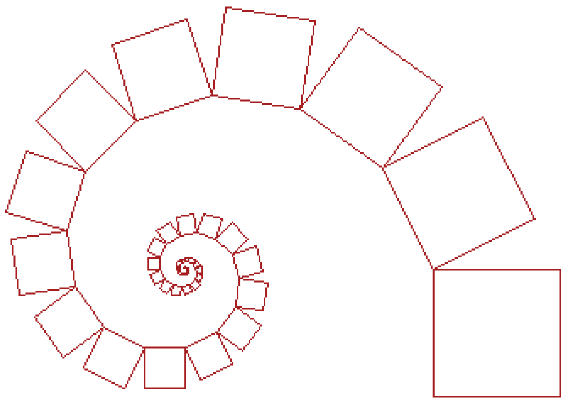

Bild 2

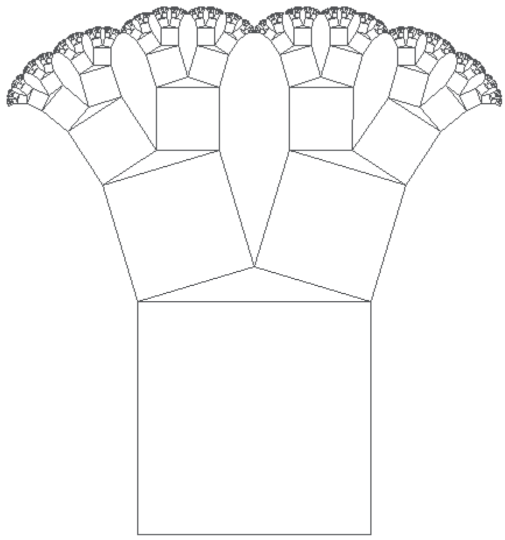

Bild 4

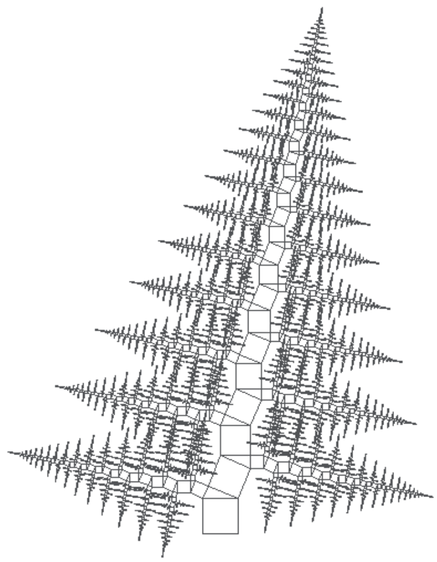

Bild 5