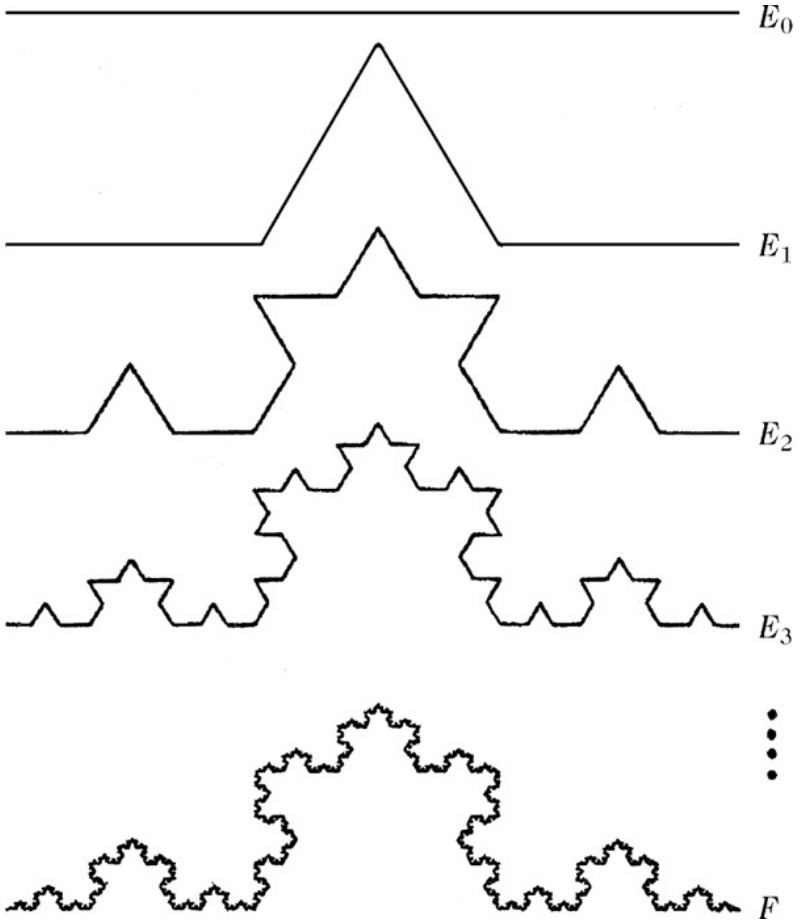
Burning-Ship Fraktal (A217)

Implementierung eines optimierten Fraktal-Generators.

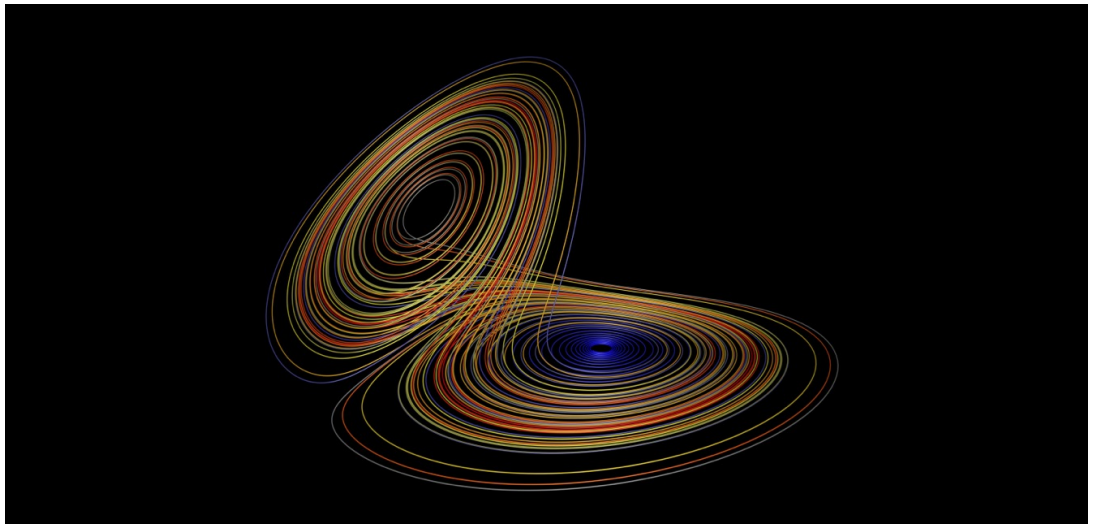
Was sind Fraktale?

- Eine (Zahlen-) Menge
- Feinstruktur auf beliebigen Skalen
- Erzeugt durch Rekursion
- Meistens selbstähnlich
- Besitzen eine ***fraktale*** Dimension
- Begriff ist geprägt von
  Benoit B. Mandelbrot



Wie wird ein Fraktal auf dem Computer generiert?

- Darstellung hängt von **Eigenschaften des Fraktals** ab.
- **Methoden**:
  - Iterated Function Systems (IFS)
  - Lindenmayer-System
  - Differenzialgleichungen (Strange Attractors)
  - **Escape-Time**



Definition der Mandelbrotmenge

$$z_0 = 0, \boldsymbol{z_{n+1}} = \boldsymbol{z_n^2} + \boldsymbol{c}$$

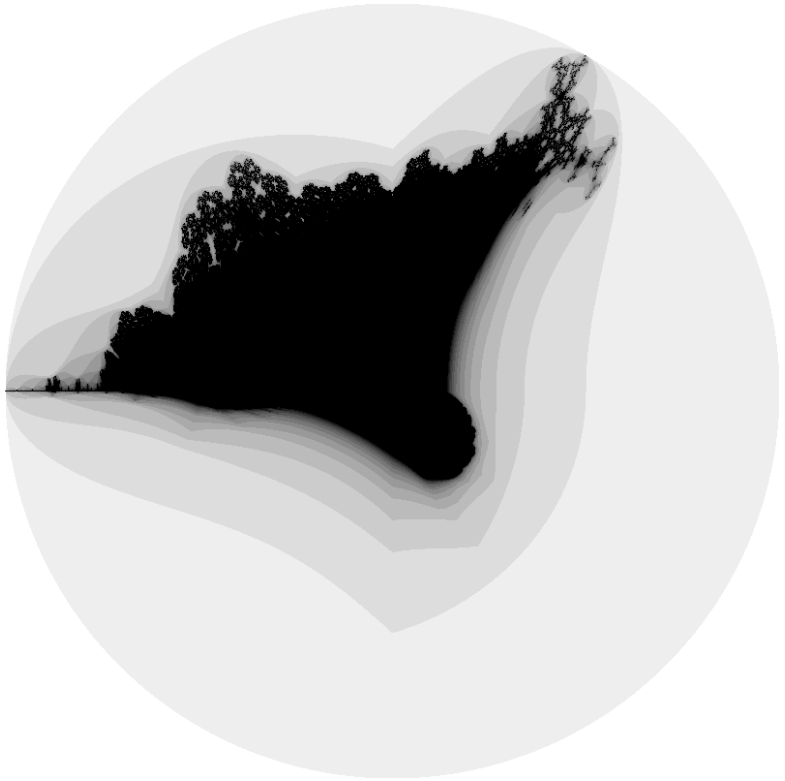
$$c \in \mathbb{M} \Leftrightarrow \lim_{n\to\infty} \sup |z_n| \leq 2$$

- Eine **kompakte** Menge der Komplexen Zahlen.
- Beobachtung:
  Es gibt **zwei** Klassen von Zahlen.
  Divergente und **Beschränkte**.

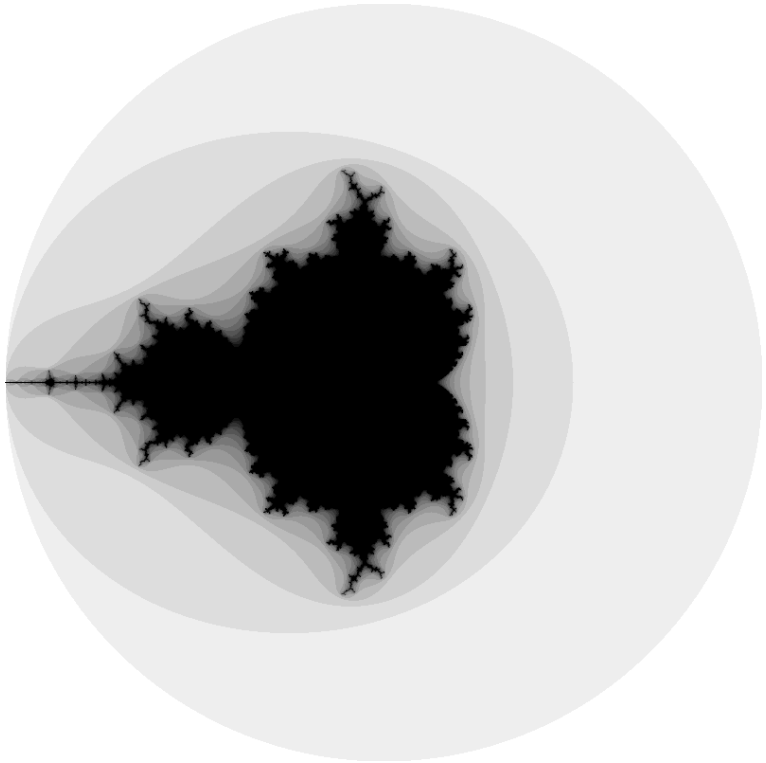
z1	z2	z3	z4
1	2	5	26
-1	0	-1	0
-2	2	2	2

Burning-Ship vs. Mandelbrot

$$z_{n+1} = (|Re(z_n)| + i * |Im(z_n)|)^2 + c$$



$$z_{n+1} = z_n^2 + c$$



Iterationsvorschrift des Burning-Ship Fraktals (1)

$$z_{n+1} = (|\Re(z_n)| + i * |\Im(z_n)|)^2 + c$$

$$z_{n+1} = |\Re(z_n)|^2 + 2i|\Re(z_n)||\Im(z_n)| + i^2|\Im(z_n)|^2 + c$$

$$z_{n+1} = \Re(z_n)^2 - \Im(z_n)^2 + i * 2|\Re(z_n) * \Im(z_n)| + c$$

$$z_{n+1} = (\Re(z_n)^2 - \Im(z_n)^2 + \Re(c)) + i * (2|\Re(z_n) * \Im(z_n)| + \Im(c))$$

Iterationsvorschrift des Burning-Ship Fraktals (2)

Require: $x \in [-2, 2] \land y \in [-2, 2]$

1: **function** BURNING SHIP$(x, y, width, height, res, n, p)$
2: **for** $h \in [0, height]$ **do**
3: **for** $w \in [0, width]$ **do**
4: $\Im(c) \leftarrow (h \rightarrow [-1, 1] \times res) + (y \times res)$
5: $\Re(c) \leftarrow (w \rightarrow [-1, 1] \times res) + (x \times res)$

6: $p_{wh} \leftarrow escape(c, n)$
7: **end for**
8: **end for**
9: **end function**

1: **procedure** ESCAPE(c, n)
2: $\Re(z_0) \leftarrow 0$
3: $\Im(z_0) \leftarrow 0$

4: $i \leftarrow 0$
5: **while** $i < n$ **do**
6: $\Re(z_{n+1}) \leftarrow \Re(z_n)^2 - \Im(z_n)^2 + \Re(c)$
7: $\Im(z_{n+1}) \leftarrow 2 \mid \Re(z_n) * \Im(z_n) \mid + \Im(c)$

8: **if** $\mid\mid z_{n+1} \mid\mid > 2$ **then break**
9: **end if**

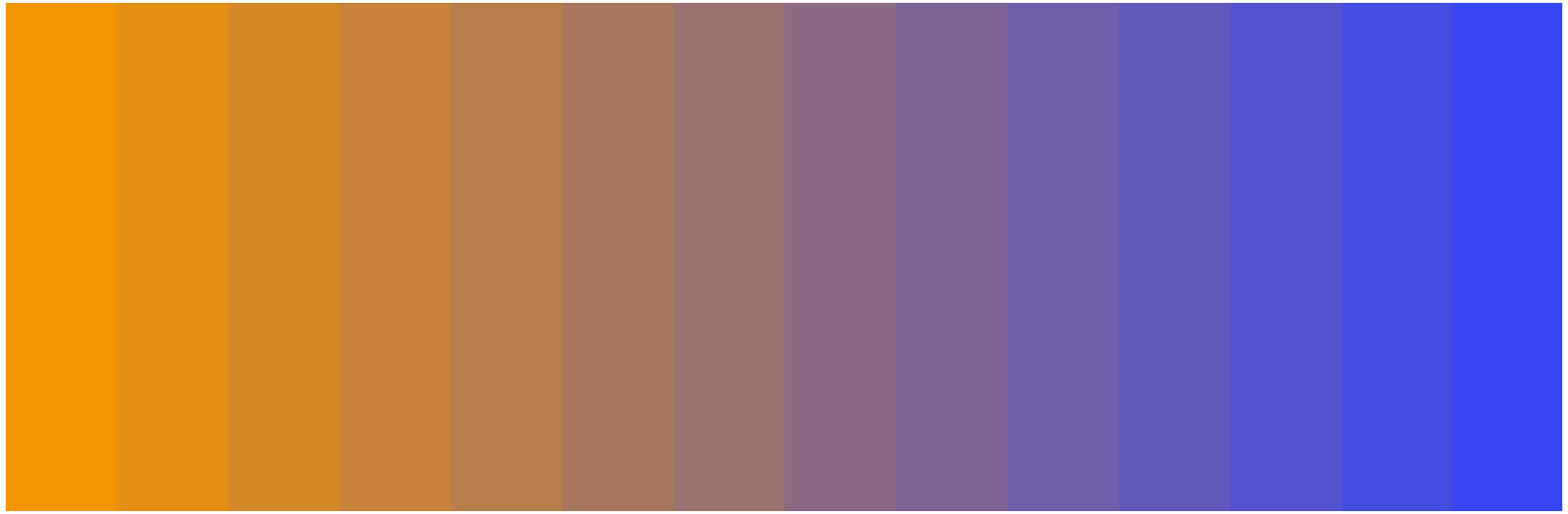
10: $i \leftarrow i + 1$
11: **end while**
return $i \times n$
12: **end procedure**

Implementierungen

- **V0 (1 Pixel)**
- **V1 (SIMD) (4 Pixel)**
- **V2 (AVX) (8 Pixel)**

Färbung

- Vordefinierte Farben Tabelle
- Skalierung des Iterationsschrittes zu [0,15]
  - **0=Weiß**
  - **15=Schwarz**



Genaugkeit

d	r	s	n	SIMD	AVX
800,800	1.0	0,0	40	14.116094%	14.116094%
800,800	1.0	0,0	30	0.270625%	0.270625%
800,800	1.0	0,0	50	12.767344%	12.767344%
1000,1000	1.0	0.0	40	13.974400%	13.974400%
800,800	4.0	0,0	40	16.977344%	16.977969%
800,800	0.1	0,0	40	0%	0%
800,800	0.009	-1.77001035,-0.05000005	40	48.806250%	48.881719%

Tabelle 1: Fehlerquotioent der Implementierungen

Performanzanalyse