Hochschule Darmstadt
Fachbereich Informatik

Chaos und Fraktale

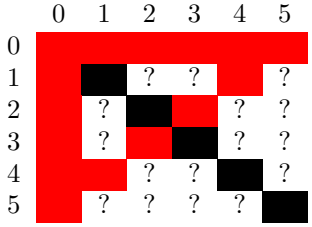
# Praktikum

| | |
|---|---|
| Semester: | SoSe 2017 |
| Laboranten: | Ken Hasenbank<br>Artur Schmidt |
| Datum: | 03.11.2017 |

# 1 Aufgabe 1

Die in Aufgabe 3 verwendete Formel (a + b) gibt, wenn genau einer der Eingabewerte ungerade ist auch ein ungerades Ergebnis aus. Werden zwei gerade oder zwei ungerade Werte eingegeben, so wird das Ergebnis zwangsläufig gerade.
Die hier angegebene Vorschrift Prüft die beiden Pixel auf ungleichheit. Betrachtet man nun den Wert ,,1“ als ungerade Zahl und den Wert ,,0“ als gerade, so ergibt sich mit beiden Vorschriften die gleiche Logik.

# 2 Aufgabe 2




In der Diagonalen (0, 0) bis (n, n) treffen immer zwei gleiche Zahlen aufeinander, sodass bei einer Bitweisen Verundung ,,&“ wieder diese Zahl herauskommt. Da nur die Zahl ,,0“ Rot wird ist in dieser diagonalen jeder Pixel (mit Ausnahme des ersten) Schwarz.
In der Diagonalen (0, n - 1) bis (n - 1, 0) - wobei n eine Zweierpotenz ist - sind die Binärdarstellungen der beiden Zahlen immer genaue Binärkomplementäre, wodurch bei der Bitweisen Verundung ,,&“ immer eine ,,0“ herauskommt. Dadurch sind hier alle Pixel Rot.

# 3 Aufgabe 3

## a)

Betrachtet man den gezeigten ,,Menger-Teppich“ genau, so fällt einem auf, dass er genau 8-mal in sich selbst enthalten ist und so die Seiten dreiteilt.

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 8 | | 4 |
| 7 | 6 | 5 |

Nach kurzen probieren mit den 4 Eckpunkten und dem Code aus Aufgabe 1, sind wir auf die Idee gekommen statt immer den Mittelpunkt als neuen Punkt zu wählen die Distanz zu dritteln (wie oben aufgefallen). Dabei sind in den Bereichen 2, 4, 6 und 8 Allerdings Freiräume entstanden. Also haben wir zu den 4 Eckpunkten die Mittelpunkte berechnet und als zusätzliche Randpunkte zur Verfügung gestellt.

## b)

Das Problem bei dieser Aufgabe ist, das rasante Wachstum des Pascalschen Dreiecks. Die Zahlen werden so groß, dass sie nicht mehr in den Datentyp Integer „passen“und es zu einem Überlauf kommt. Da beim Überlauf das Vorzeichenbit gesetzt wird, ergibt die Summe zweier positiver großer Zahlen ein negatives Ergebnis.
Um einen Überlauf zu vermeiden, wurde nicht die Summe im Array gespeichert, sondern der entsprechende Modulowert. D.h. es wurde die Summe gebildet, diese Modulo einer natürlichen

zahl gerechnet und das Ergebnis dann im Array abgelegt. Das hatte zum Ergebnis, dass die Zahlenwerte innerhalb des Arrays niemals größer wurden als der Wert mit dem Modulo gerechnet wurde.

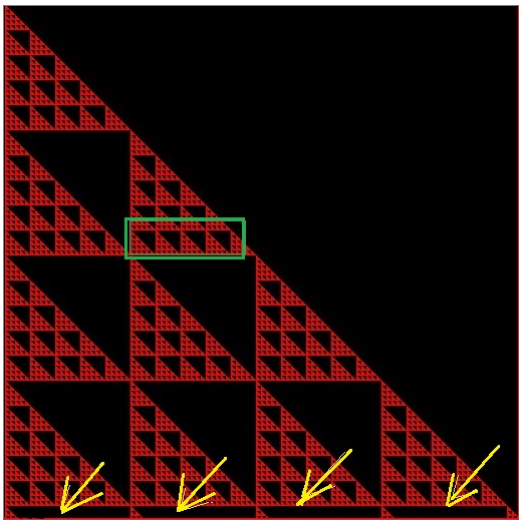


Abbildung 3.1: Ausgabe mit Modulo 5

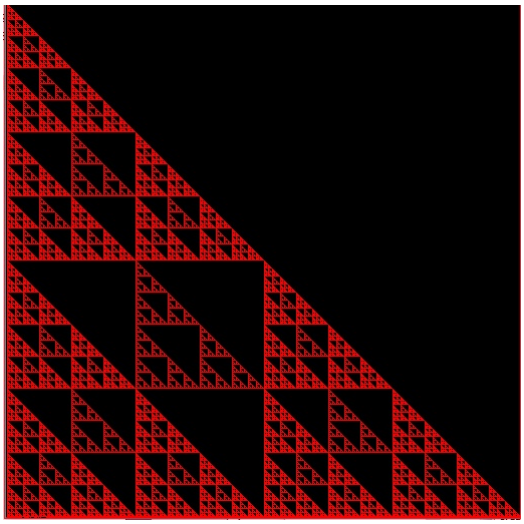


Abbildung 3.2: Ausgabe mit Modulo 4

Es fällt beim Vergleich der Ausgaben zunächst auf, dass bei Primzahlen am unteren Rand nur Teile von Dreiecken vorhanden sind (Siehe gelbe Pfeile in Abbildung 3.1). Wohingegen bei Zahlen die keine Primzahl sind, alle entstandenen Dreiecke vollständig im Bild sind. Eine weitere Besonderheit bei der Ausgabe mit Primzahlen ist: Über den Dreiecken sind so viele kleine Dreiecke in Reihe wie die um eins verringerte Zahl mit der Modulo gerechnet wurde. Das grünes Rechteck

in Abbildung 3.1. Zuletzt ist anzumerken, dass die Ausgabe bei Primzahlen geordneter wirkt, wohingegen sie bei allen anderen Zahlenwerte deutlich fragmentierter wirkt.

## c)

Werden wir machen!