

LEHRSTUHL FÜR RECHNERARCHITEKTUR UND PARALLELE SYSTEME

**Aspekte der systemnahen Programmierung
bei der Spieleentwicklung**Gruppe 132 – Abgabe zu Aufgabe A214
Wintersemester 2020/21

Mohammed Attia

Patrick Zimmermann

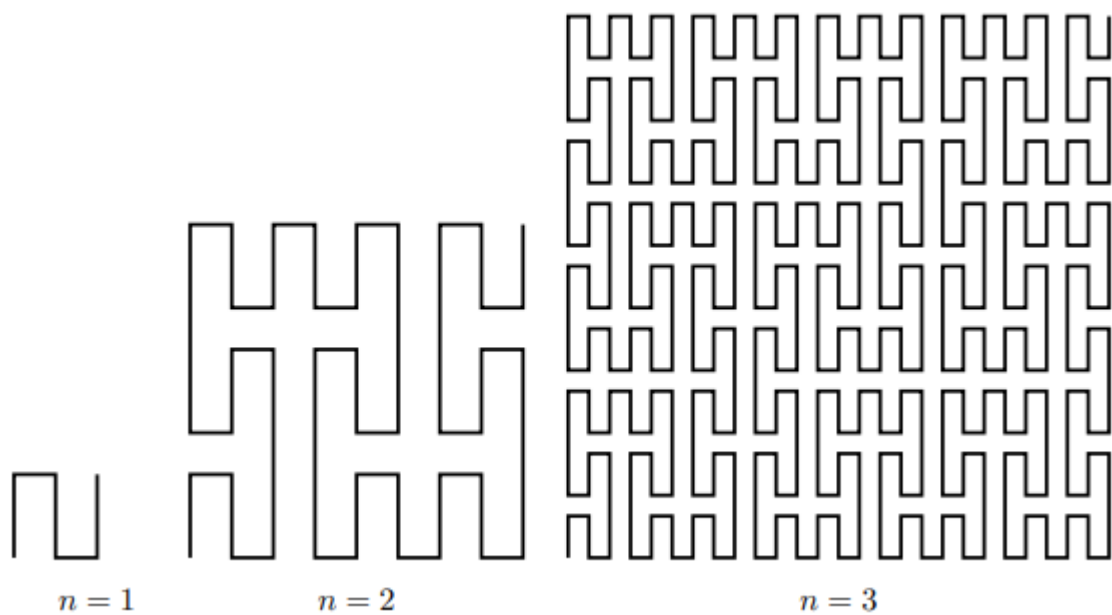
Thomas Torggler

1 Einleitung

Raumfüllende Kurven bilden die Brücke zwischen Kunst und mathematischer Geometrie. In der Mathematik werden sie gemeinhin benutzt um ein n -dimensionales Problem in ein eindimensionales zu konvertieren. Eine solche Kurve beschreibt essentiell einen linearen Pfad durch n -dimensionale Räume. Giuseppe Peano war der Erste, der eine solche Kurve 1890 definierte. Um einen n -dimensionalen Raum in die Dimension $n-1$ zu konvertieren, lässt sich eine stetig surjektive Funktion $f(x)$ erstellen, so dass gilt: $\forall x \in \mathbb{R}^{n-1} \quad y \in \mathbb{R}^n$. Für einen Beweis siehe (Quelle?). Hier wollen wir uns auf die sogenannten Peano-Kurven beschränken. Für eine solche Kurve definieren wir ein Intervall $I = [0,1]$, sowie $f : I \rightarrow I^2$. Dann ist die Peano-Kurve: $\lim_{x \rightarrow \infty} f(x)$, mit $x \in I$. Sie entspricht dem Grenzwert einer Folge von Funktionen $f(x)$ und lässt sich mit der Bedingung, dass sich die Kurve nicht überschneiden darf, folgendermaßen konstruieren:

Man unterteile eine Fläche in 9 Quadrate. Jedes dieser Quadrate soll nun durch eine Kurve besucht werden. Dadurch durchläuft die Kurve die Quadrate in Form eines „S“. In einem Iterationsschritt lässt sich eines der 9 Quadrate in weitere 9 Quadrate unterteilen, die wiederum auf selbe Art verbunden werden, wie in Abb.1 gezeigt.

Im Folgenden Definieren wir $n \in \mathbb{N}$ als Grad der Kurve. Wir beschreiben nun unseren Ansatz einen iterativen Algorithmus zu finden, um die eben beschriebene Peano-Kurve darzustellen.

Abbildung 1: Peano-Kurve mit $n = \{1,2,3\}$, [?]

2 Lösungsansatz

Der Aufbau der Peano-Kurve ermöglicht es die Kurve des aktuellen Grades mit einer Variation der gespiegelten und originalen Kurve des vorherigen Grades zu zeichnen. Diese Observation war der Grundstein unseres iterativen Algorithmus. Anfangs wird dabei die Startkurve hardcodiert in einem Integer Array abgespeichert wobei die Zahlen von 1 bis 4 jeweils einer der vier Richtungen entsprechen. Falls der eingegebene Grad $n > 1$ ist, wird aufsteigend über alle Grade bis inklusive n iteriert wobei während jedem Schritt die Kurve des vorherigen Grades in der entsprechenden Reihenfolge entweder im Originalzustand oder gespiegelt acht mal in den Array eingefügt wird, sodass die vollständige Kurve des aktuellen Grades entsteht.

Zu beachten ist dabei jedoch, dass durch das Spiegeln und Einfügen von Kurven deren Richtung noch inkorrekt sein könnte, dafür gibt es aber ähnlich zur Spiegelungsmethode eine Methode zum Umkehren der Richtung. Nach diesen Schritten sind die neu erstellten Kurven jedoch noch nicht miteinander verbunden. Da sich das Muster der ursprünglichen Kurve durch alle Grade wiederholt, kann nach jedem Schritt ein hardcodierter Pfad zur nächsten Kurve eingefügt werden. Nachdem über alle Grade iteriert wurde, läuft der Algorithmus den vollständigen Richtung Array durch, verändert dabei bei jedem Schritt je nach Richtungsangabe entweder die x oder y -Koordinate und speichert diese dann in die Ausgabeliste der Peano-Methode.

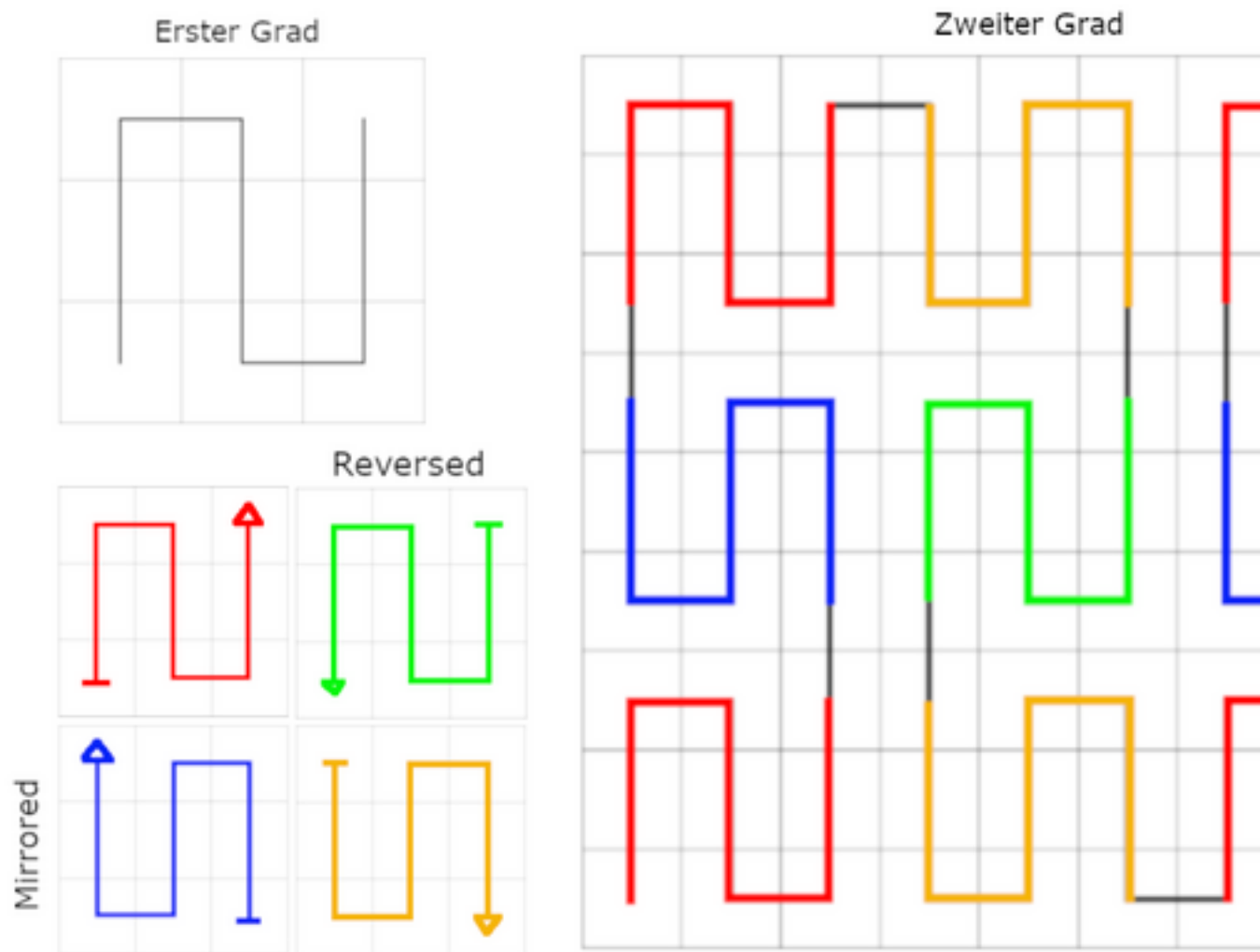


Abbildung 2: Peano-Kurve mit $n = 1$, $n = 2$

3 Korrektheit

Die Genauigkeit unseres Algorithmus ist abhängig von dem Grad n , dass die Anzahl an Iterationen bestimmt, wie in den Abb.1 beschrieben. Das geht auch aus der beschriebenen mathematischen Definition hervor, da die Peano-Kurve an sich ein Grenzwert ist. Ansonsten ist die Kurve wie ebenfalls schon oben beschrieben definiert, weshalb nur die Korrektheit unseres Algorithmus entscheidend ist. Grundsätzlich lässt sich die Korrektheit einer Kurve, besonders bei der von uns behandelten, nachweisen indem man ihre graphischen Darstellungen vergleicht. Da das Theoretisch aber nicht möglich ist, folgt auch eine Implementierung in Assamblen und C. Wir wollen dennoch unseren Ansatz auf Korrektheit prüfen. Nachdem die Kurve immer wieder die gleichen (elementaren) Bestandteile verwendet, müssen diese und deren Permutationen richtig berechnet werden. Im speziellen sind das die Kurven mit $n = 1$ und $n = 2$. So lässt sich induktiv die Korrektheit beweisen, da wir, wie im vorherigen Kapitel beschrieben, zum berechnen eine Kurve n -ten Grades nur die Kurve selbst sowie die gespiegelte und invertierte Versionen der Kurve vom Grad $n-1$ benutzen. Beginnen wir mit $n = 1$. Hier ist die Kurve vordefiniert, dementsprechend gibt es nichts zu zeigen. Da auf diesem Muster alle anderen Kurven mit $n > 1$ basieren, ist sie in unserem Algorithmus ebenfalls als Basis fest gegeben. Da die Funktion wie definiert surjektiv ist, gilt: $\forall n > 1 \in \mathbb{N}, \exists y \in \mathbb{R}^n$ mit $x \in [0, 1]: f(x) = y$. Dadurch lassen sich Kurven mit höherem Grad konstruieren. Sei nun $n > 1$ und beliebig gewählt. Betrachten wir nun Kurven mit $n \geq 2$. Hierzu müssen wir die Permutationen der vorhergehenden Kurve untersuchen, die wir zur Konstruktion der Kurve mit Grad n benötigen. Die gespiegelte Permutation ist im Kern nichts anderes als die normale Peano-Kurve mit anderer Reihenfolge der Koordinatenberechnung, also von $p = (1, 1)$ nach $p_2 = (0, 0)$. Dies hat zur Folge, dass sich außer der Reihenfolge der Punkte an der grundlegenden Kurve nichts ändert. Somit ist die Spiegelung induktiv eben so korrekt wie die zugrundeliegende Kurve niedrigeren Grades. Bei der Invertierung wird die Kurve ähnlich zur Achsenspiegelung ebenfalls die vertikale Richtung geändert, jedoch wird weiterhin auch die horizontale Richtung geändert. Die Kurve wird in dem mittleren Punkt der Kurve gespiegelt. Da die Kurve immer 9^n Felder besitzt, existiert immer ein solcher Mittelpunkt. Dadurch wird die vorausgehende Kurve notwendigerweise verändert. Diese zusätzliche Operation ist ebenfalls elementare Spiegelungen und somit direkt auf die Koordinatenberechnung Einfluss haben ohne die Kurve anders zu verändern, ändert sich der charakteristische Verlauf der Kurve nicht.

4 Performanzanalyse

5 Zusammenfassung und Ausblick
