



HOCHSCHULE TRIER

Trier University of Applied Sciences

Informatik - Computer Science

Alpha zu Mesh Tool

English Title

Jeremias Boos

Hausarbeit zur Vorlesung Tool- und Pluginprogrammierung

Betreuer: Prof. Dr.-Ing. Christof Rezk-Salama

Trier, 28. August 2015

Inhaltsverzeichnis

1	Motivation	1
2	Ergebnis	2
	2.1 Rand Detektion.....	2
	2.2 Ramer Douglas Peucker Algorithmus	4
	Literaturverzeichnis	5

Motivation

Wenn Sprites mit einem großen Transparenten Flächen in Spielen verwendet werden, wird geht viel Rechenzeit für die Berechnung und Verwerfung von Absolute transparenten Pixeln drauf. In solchen Fällen kann es Besserer sein einen Angepassten Mesh zur Textur zu verwenden anstelle eines einfachen Quadrates. Außerdem kann der Mesh als Kollisionsgeometrie benutzt werden wenn per Pixel Kollision zu aufwendig ist.

2.1 Rand Detektion

A diagram of a chessboard showing a knight's possible moves. The knight is positioned in the center square (d5). Arrows point to the squares it can move to: b6, c6, e6, f6, b4, c4, e4, and f4. The squares b6, c6, e6, and f6 are highlighted in light blue, while b4, c4, e4, and f4 are highlighted in light green. The squares d6, e5, f5, g5, h5, g4, f4, e4, d4, c4, b4, c3, d3, e3, f3, g3, h3, g2, f2, e2, d2, c2, b2, c1, d1, e1, f1, g1, h1, g0, f0, e0, d0, c0, b0, a0, b0, c0, d0, e0, f0, g0, h0, g1, f1, e1, d1, c1, b1, a1, b1, c1, d1, e1, f1, g1, h1, g2, f2, e2, d2, c2, b2, a2, b2, c2, d2, e2, f2, g2, h2, g3, f3, e3, d3, c3, b3, a3, b3, c3, d3, e3, f3, g3, h3, g4, f4, e4, d4, c4, b4, a4, b4, c4, d4, e4, f4, g4, h4, g5, f5, e5, d5, c5, b5, a5, b5, c5, d5, e5, f5, g5, h5, g6, f6, e6, d6, c6, b6, a6, b6, c6, d6, e6, f6, g6, h6, g7, f7, e7, d7, c7, b7, a7, b7, c7, d7, e7, f7, g7, h7, g8, f8, e8, d8, c8, b8, a8, b8, c8, d8, e8, f8, g8, h8 are shaded in gray.

Abb. 2.1: Rand Anfang
Schwarzer Pfeil: Pixel scann
Grüner Pfeil: Pixel über Schwellwert
Schwarzer Weißer Pfeil: Potenzieller Pfad
Roter Pfeil: Rechts vom Potenziellen Pixel
Grüne Felder: Potenzieller Pfad punkte
Grünes Feld gelbe Umrandung: Nachbar scann

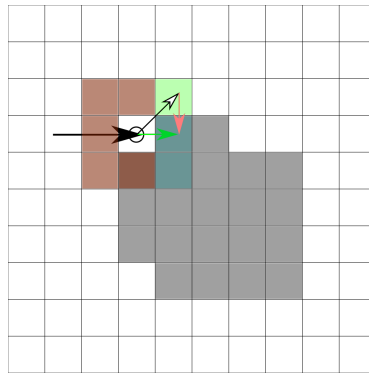


Abb. 2.2: Gefundener nächster Punkt
 Roter Felder: Ausgeschlossene Kandidaten
 Hell Grünes Feld: Gefundener Rand

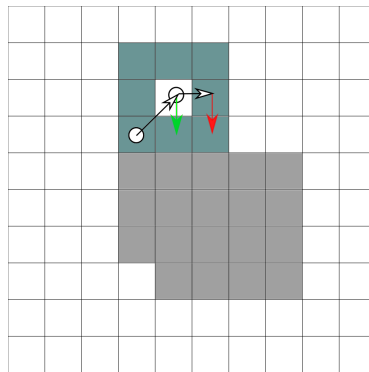


Abb. 2.3: nächster Punkt

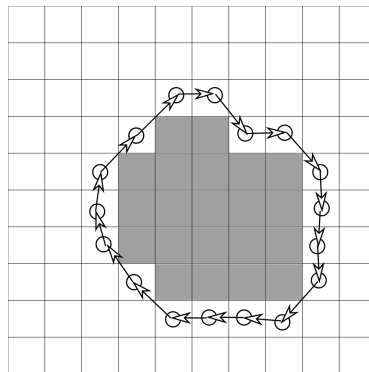


Abb. 2.4: Rand Vollständig

2.2 Ramer Douglas Peucker Algorithmus

Der Pfad besteht nun aus allen Pixeln die den Rand darstellen. Da dies auch einige Geraden beinhaltet lässt sich der Rand gut vereinfachen um später in der Triangulierung 3-ecke zu sparen.

Zur Vereinfachung wird der "Ramer Douglas Peucker Algorithmus" verwendet. Dieser lässt sich leicht implementieren und arbeitet zuverlässig.

Der Algorithmus sucht rekursive den entferntesten Punkt zwischen Anfang und ende. Wenn der Punkt von der Linie weiter als der Schwellwert entfernt ist wird eine Spaltung der Linie durchgeführt. Wenn das Maximum innerhalb des Schwellwertes ist werden Anfangs- und Endpunkt direkt mit einander verbunden. [Wik]

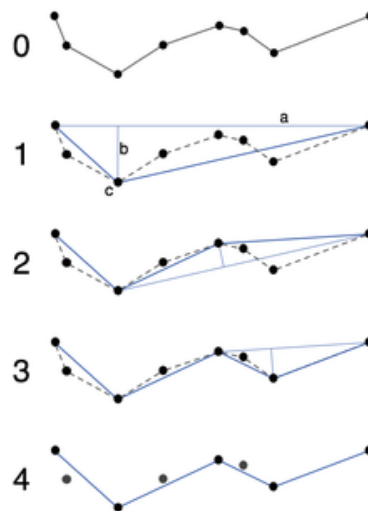


Abb. 2.5: Linienglättung nach dem Douglas-Peucker-Algorithmus

Literaturverzeichnis

Wik. *Wikipedia - Ramer?Douglas?Peucker algorithm.*
`en.wikipedia.org/wiki/Ramer?Douglas?Peucker_algorithm.`