



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, 29. August 2015

Inhaltsverzeichnis

1	Motivation	1
2	Ergebnis	2
	2.1 Rand Detektion.....	2
	2.2 Ramer Douglas Peucker Algorithmus	4
	2.3 Ear clipping.....	5
3	Probleme und Ausblick	6
	Literaturverzeichnis	7

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

[illegible]

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

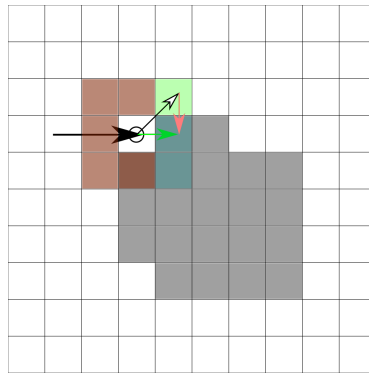


Abb. 2.2: Gefundener nächster Punkt
 Roter Felder: Ausgeschlossene Kandidaten
 Hell Grünes Feld: Gefundener Rand

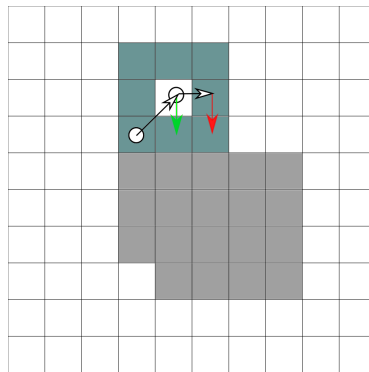


Abb. 2.3: nächster Punkt

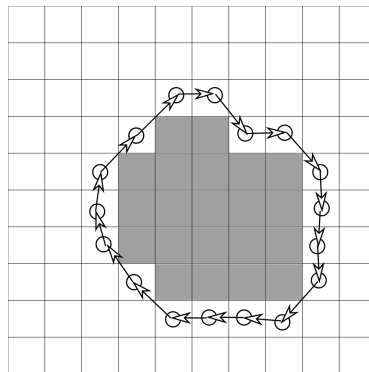


Abb. 2.4: Rand Vollständig

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]

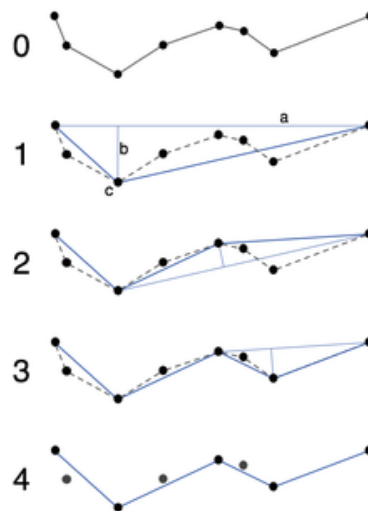


Abb. 2.5: Linienglättung nach dem Douglas-Peucker-Algorithmus

2.3 Ear clipping

Ear clipping ist ein sehr einfach umzusetzender Triangulierungs Algorithmus von Polygonen.

Die Idee ist jedes Ohr vom Polygon abzuschneiden bis das ganze Polygon in einzelne 3 ecke zerlegt ist. [Ebe] [del]

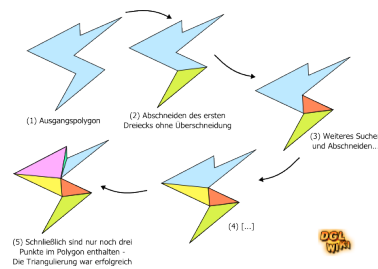


Abb. 2.6: Ablauf des Ear Clipping Verfahrens

Probleme und Ausblick

Der Umgesetzte Triangulierungs Algorithmus ist ein $O(n^2)$ dieser kann durch den Monton Polygon Triangulierung Algorithmus ausgetauscht werden wodurch die die Triangulierung nur noch $O(n * \log(n))$ ist.

Die Triangulierung könnte auch noch erweitert werden um Löcher im Sprite zu beachten.

Eine weitere Verbesserungen wäre die Implementierung eines Gauß Algorithmus um einen größeren Rahmen um den Sprite zu Ziehen.

Zu guter Letzt könnte man noch eine Option bieten die Polygone in Konvexe Polygone zu zerlegen, um die generierten Meshes leichter in 2D Physik Engine zu nutzen.

Literaturverzeichnis

- del. *Ear Clipping Triangulierung.*
wiki.delphigl.com/index.php/Ear_Clipping_Triangulierung.
- Ebe. EBERLY, DAVID: *Triangulation by Ear Clipping.*
www.geometrictools.com/Documentation/TriangulationByEarClipping.pdf.
- Wik. *Wikipedia - Ramer-Douglas-Peucker algorithm.*
en.wikipedia.org/wiki/Ramer?Douglas?Peucker_algorithm.