

186.166 Entwurf und Programmierung einer Rendering-Engine, Technischer Bericht - TerraGen

Bernhard Rainer 0828592

8. März 2016

Übersicht

Ziel dieser Aufgabe ist es, eine interaktive Rendering-Engine zu programmieren, die es dem User ermöglicht selbst Geometrie zu erstellen und aktiv verändern. In diesem Beispiel ist ein interaktiver Terrain-Renderer implementiert, mit dem der User zur Laufzeit verschiedene Terrains, basierend auf einer Heightmap, generieren und verändern kann. Da es sich bei Terrains zumeist um großflächige Areale handelt werden hier einige Techniken angewendet, die die Skalierung und Detaillierung der Geometrie unterstützen.

Geometrie

Das Basis-Geometrie-Objekt ist ein simples 10x10 Grid mit 200 Triangles, eine Einheit entspricht einem Meter. Wird das Terrain skaliert, wird dieses Grid wiederholt an nebeneinander liegenden Positionen gezeichnet, sodass eine homogene Fläche entsteht. Anhand der Vertex-Position im Grid, sowie der relativen Position der Grids zueinander, können Texturkoordinaten einfach berechnet und skaliert werden.

Tessellierung

Eine Grid-Auflösung von einem Meter ist für detaillierte Landschaften oftmals nicht ausreichend. Daher wird, relativ zur Kamera, das Terrain in feinere Dreiecke unterteilt. Diesen Vorgang heißt Tessellierung. Hierzu werden dynamisch Tessellierungslevel, basierend auf der Distanz des Dreiecks zur Kamera, berechnet. Diese Level entscheiden, wie oft das Dreieck unterteilt wird. Diese feinere Unterteilung erlaubt ein feineres Sampling der Höhentextur (Heightmap).

Heightmap

Die Heightmap bestimmt pro Vertex die Höhe der Geometrie entlang der y-Achse. Hierzu wird die Textur in regelmäßigen Abständen ausgelesen, sodass ein kontinuierlicher Terrain-Verlauf entsteht. Wird das Terrain skaliert, verringern

sich die Abstände und die Auflösung wird höher. Selbiges trifft ebenfalls auf die Tessellierung zu.

Die Heightmap wird mit Hilfe der libnoise-Bibliothek [1] erstellt. Hierzu wird Perlin-Noise von verschiedenen Stärken genutzt um sowohl eine grobe Struktur, als auch eine feinere Detaillierung der Landschaft zu generieren.

Erstellung einer Heightmap

- Terrain Frequency: bestimmt die grobe Struktur des Terrains
- Persistence: bestimmt, wie schnell sich die Basisfrequenz zwischen den Oktaven verringert
- Bump Frequency: bestimmt die Frequenz der Detaillierung des Terrains
- Bias: Reduziert oder erhöht Erhebungen
- Scale: Skalierung der feinen Struktur

Zusätzlich kann eine beliebige Textur als Heightmap eingelesen werden.

Texturierung und Schattierung

Für eine realistische Darstellung sind Texturierung und Schattierung unabdingbar. Hierzu ist es notwendig, den Normalvektor pro Vertex, basierend auf den Höhendaten, zu berechnen. Wie in [2] beschrieben, wird hierzu ein 4-Sample-Cross-Filter verwendet. Die Differenzen der Höhenwerte in x- bzw. z-Richtung der jeweiligen Vor- und Nachfolger ergeben die einzelnen Werte der Normalen. Diese Normale wird für die Schattierung benutzt.

In diesem einfachen Programm wird zwischen einer Stein-Textur für steile Abhänge, sowie einer Gras-Textur für flache Ebenen unterschieden. Basierend auf der Neigung der Normalen wird zwischen den zwei Texturen unterschieden, oder interpoliert. Die Neigung berechnet sich aus Eins minus y-Komponente der Normalen.

Navigation

Die Kamera kann mittels der Maus bewegt werden, solange die rechte Maustaste gedrückt ist. Das Mausexplorer bewegt die Kamera vorwärts/rückwärts. Alternativ kann die Kamera ebenfalls mit den WASD-Tasten bewegt werden.

Evaluierung und Performance

Die Einteilung in verschiedene Detailstufen in diesem Beispiel ist einfach gehalten. Basierend auf der Distanz zur Kamera werden die Tessellierungs-Level in vier verschiedene Stufen unterteilt. Daher sind sogenannte Popping-Artefakte, ein sprunghafter Übergang zwischen zwei Stufen, zu erkennen. Dennoch erhöht der höhere Detailgrad merklich den Realismus des Terrains.

Abhängig von der genutzten Hardware lässt sich das Grid problemlos bis zu hundert mal pro Dimension replizieren, sodass die Anwendung immer noch

interaktiv bleibt. Da es sich hierbei immer um das selbe Geometrieobjekt handelt, wäre hier ein potenzieller Optimierungsschritt die Verwendung von Instanced Rendering. Der Performance-Unterschied zwischen tesselierter und normaler Geometrie beläuft sich bei 100x100 Gridwiederholungen, also insgesamt zwei Millionen Triangles (untesseliert), auf durchschnittlich 1-2 ms, gemessen auf einer AMD Radeon R9 270. Der Unterschied ist also äußerst gering.

Literatur

- [1] libnoise. <http://libnoise.sourceforge.net/>.
- [2] Johan Andersson. Terrain rendering in frostbite using procedural shader splatting. *ACM SIGGRAPH courses*, pages 42–43, 2007.