

Fractal Island

Echtzeitgrafik, 2. Abgabe

Christian Folie, 0225810 / 932
Meinrad Recheis, 0026003 / 532

Wir haben unser Projekt in C# mit Managed Direct X realisiert. Als Grund-Engine haben wir das Aardvark - Toolkit aus der VO "Entwurf und Programmierung einer Rendering Engine" verwendet, welches die grundlegende DirectX Initialisierung übernimmt und einige Datenstrukturen bereitstellt. Eine einfach Navigation ist über die Maus möglich, es kann ran und weg gezoomt werden.

Effekte

Wasser - Spiegelung

Die Landschaft und alle anderen 3D-Objekte werden gespiegelt und in eine Reflection-Textur gerendert. Diese wird als projektive Textur auf ein Quad aufgebracht und soll das reflektierende Wasser darstellen.

Referenzen

GameDev.net:

<http://www.gamedev.net/reference/programming/features/vcp/default.asp>

CodeProject.com:

<http://www.codeproject.com/useritems/DirectXMirror.asp>

Dot3 BumpMapping

In der Mitte befindet sich ein Quad welcher ein über Dot3 generiertes BumpMap besitzt, welches abhängig von einem Licht-Vektor dynamisch generiert wird. Das Licht kreist um den Quad herum und vermittelt so einen Eindruck einer rauerer Oberfläche.

Als NormalMap wurde eine zufällig generierte Textur verwendet.

Referenzen

Two-Kings.de

<http://www.two-kings.de/tutorials/dxgraphics/dxgraphics14.html>

CodeSampler.Com

http://www.codesampler.com/dx9src/dx9src_4.htm#dx9_dot3_bump_mapping

Weitere Besonderheiten

Terraingenerierung

Das Terrain wird als Fraktales Terrain mit dem MidPoint-Displacement Algorithmus generiert und mittels einiger Basistexturen höhenabhängig texturiert.

L-System Bäume

Die dargestellten "Bäume" werden über ein L-System mit vorgegebener Grammatik Zufall gesteuert generiert.

Verwendete Bibliotheken

DirectX SDK (Oktober 06)
Aardvark SDK
IronPython