



Demo Beschreibung

Team: ByteStorm
Echtzeitgrafik

Teammitglieder (Name, MatrNr, E-Mail.):

1. Murat Sari, 126198, murat_sari@gmx.at
2. Stefan Tomitsch, 126195, TTec@gmx.net

Gedruckt am: 23.01.2006 02:41:00



1.) DOT3 BumpMapping:

1.1.) Quellen:

- http://gpwiki.org/index.php/OpenGL:Tutorials:GLSL_Bump_Mapping
- http://www.blacksmith-studios.dk/projects/downloads/bumpmapping_using_cg.php
- <http://www.3dkingdoms.com/tutorial.htm>
- <http://www.terathon.com/code/tangent.html>
- <http://www.gamedev.net/reference/articles/article1903.asp>

1.2.) Effekt:

Es wurde TangentSpace Bumpmapping verwendet die Berechnung der Tangenten übernimmt NVMeshMender. Die TBN Matrixen werden im eignen File Format abgelegt und werden nur einmalig in unserem Exporter vorberechnet.

Die NormalMap vom Roboter und Terrain, wurde mittels dem Nvidia NormalMap-Plugin für Photoshop erzeugt. Das 2te Model welches sich auf dem Terrain befindet, ist vom Spiel Doom3 (ebenfalls Texturen und Normalmaps). Die TBN Matrizen werden über die Texcoords übergeben.

2.) Soft ShadowMapping:

2.1.) Quellen:

- http://download.developer.nvidia.com/developer/SDK/Individual_Samples/DEMOS/Direct3D9/src/HLSL_SoftShadows/docs/HLSL_SoftShadows.pdf
- <http://www.delphi3d.net/articles/viewarticle.php?article=phong.htm>
- <http://www.lighthouse3d.com/opengl/glsl/>



2.2.) Effekt:

Die Shadowmap wird mit einer Auflösung von 768x768 in den Framebuffer gerendert und mittels `glCopyTexSubImage2D` in eine Textur kopiert. Beim Rendern der Shadowmap wird nur die Geometrie mittels der fixed-function Pipeline gezeichnet. Der Schatteneffekt selbst wurde in GLSL implementiert. Der Schatten wurde zusätzlich mit einem percentage-closer Filter versehen.

Der Lichtkegel wird mit einer Lightmap erzeugt, die aus Sicht der Lichtquelle auf die Landschaft projiziert wird. Die Beleuchtung wurde nach dem Blinn-Phong Modell berechnet.

3.) Allgemein:

Das Bumpmapping wird auf das Terrain und die beiden Modelle die sich auf dem Terrain befinden angewandt. Der 2te Effekt der implementiert wurde ist Shadowmapping. Um die Effekt gut sehen zu können, haben wir eine rotierende Lichtquelle über die Modelle platziert. Die Skybox im Hintergrund sorgt dafür, dass ein wenig Weltraumfeeling erzeugt wird. Das ganze wurde mittels GLSL Shader implementiert. Wir haben die beiden Shader kombiniert und bumpmappen nur die Teile, die sich nicht im Schatten befinden.

3.1.) Input:

Als Input haben wir beschlossen ein eignes Model Format zu verwenden welches wir während der Implementierung entwickelt haben. Wir wollten die TBN - Matrixen nicht bei jeden Start neu berechnen diese werden jetzt von unseren Modelexporter berechnet und in die Datei gespeichert. Der Exporter wurde in MFC/OpenGL entwickelt und kann zurzeit nur Milkshape Modelle importieren und in unser Format speichern.

Das Terrain kann man auch im Exporter erzeugen und als Model speichern um so Zeit zu sparen.



3.2.) Klassen:

- CCGLoader: Dient zum laden von CG Programmen wird im Programm nicht mehr verwendet(Leiche)
- GLSLLoader: Dient zum laden von GLSL Programmen.
- Modul Kamera: Dient zur Steuerung der Kamera.
- Modul Math: Die Mathematik Klassen zur Vector berechnung wurde von PaulsProject geliehen :)
- Modul Model:
 - CModel: Mit dieser Klasse kann man nur Modelle im bs3d Format Speicher/Laden man kann die BoundingBox anzeigen oder die Normal Vektoren und die Sub-Meshes zeichnen.
 - CMesh: Jedes Model besteht aus n verschiedenen SubMeshes diese Klasse dient zum zeichnen dieser unter verwendung von Vertex Arrays/VertexBufferObjects je nach wahl.
- Modul Texture Manager: Dient zum laden von Texturen und zum binden von Texturen. Zum laden der Bilder wird Devil verwendet.
- Modul SkyBox: Dient zum zeichnen der Skybox.
- Modul Material Struct: Dient zum laden der Material Eigenschaften. Jedes SubMesh verfügt über ein Material-Object.
- Modul BoundingBox: Hier werden die Daten der BoundingBox gespeichert.
- Modul Log System: Mit dieser Klasse werden Fehler gelogt und in ein XML File gespeichert, das man sich im IE ansehen kann :)
- Modul Window: Dient zur Erzeugung eines Fensters mittels WinAPI.



- Modul XML Parser:
 - COptions: Dient zum Laden der Options.XML
 - Modul Base: CXML Ist die Basis Klasse die das Laden von XML Files unter Verwendung von TinyXML ermöglicht.
- Modul Main Func:
 - CMain: Hier befindet sich die GameLoop wo die Modelle gezeichnet werden und die Shader angewendet werden.