

Echtzeitgrafik VU

WS 2009/10

„Elevator Suite“



Effekte

.Normal Mapping:

NormalMaps wurden (falls sie noch nicht vorhanden waren) mit CrazyBump erstellt und werden automatisch vom Modelloader zum Modell hinzugefügt. Ebenfalls im Modelloader werden die Tangenten des Modells berechnet und an das VBO übergeben. Im Vertex Shader werden Licht- und Viewvektoren mit Hilfe der berechneten Tangenten in den Tangentenraum transformiert und im Fragment Shader mit den Normalen aus der Normalmap für die Berechnung der Beleuchtung verwendet.

Normal Mapping kann für jedes Modell beim Laden standardmäßig aktiviert oder deaktiviert werden. Modelle mit deaktiviertem Normalmapping werden mit einfachem PPL beleuchtet.

Quellen:

http://www.ozone3d.net/tutorials/bump_mapping.php (heisst bump mapping ist aber ein nm tut ;))
<http://www.terathon.com/code/tangent.html> (für Berechnung der Tangenten)

.Bloom Shader:

Die geshadete Szene wird in ein FBO gerendert. Die daraus resultierende Textur wird auf ein Quad gemapt, das vor der Kamera platziert ist, und durch den Bloom Shader geschickt, der Mittels Gauß Kernel zuerst in x-Richtung „blurt“. Das resultierende Bild wird wieder in ein FBO gerendert und vom Bloom Shader im zweiten Durchgang in y-Richtung „geblurt“. Das so in x- und y-Richtung geblurte Bild wird schlussendlich mit dem unveränderten Originalbild geblendet.

Quelle:

http://www.gamedev.net/community/forums/topic.asp?topic_id=395186

.Wissenswertes:

Die Kamerafahrt wird von campath.xml ausgelesen und kann interaktiv erstellt werden, indem während die Demo läuft mittels „c“ die Kamerafahrt deaktiviert wird. Nun kann man sich per Maus und Tastatur durch die Szene bewegen. Mit „q“ startet man einen neuen Kamerapfad. „w“ erstellt eine neue Kameranode, die Position und Blickrichtung einspeichert (mindestens 4 Nodes werden für die Catmull Rom Spline Interpolation benötigt). Mit „e“ wird der neue Kamerapfad abgespeichert, der dann beim nächsten Start der Demo geladen wird.

.zusätzliche Quellen:

Der 3DS Modelloader basiert auf einem Tutorial von www.gametutorials.com

Bilder werden mittels DevIL geladen.

Hilfklasse für Matrizenberechnungen stammen von http://nopper.tv/opengl_3_1.html

Textfiles (shader) werden mit Hilfe einer Klasse von www.lighthouse3d.com eingelesen.

FMOD ist für den Sound verantwortlich.

TinyXML wurde für das Lesen/Schreiben des Kamerapfades verwendet.