

Alibaba

Echtzeitgraphik VU
186.140
WS 2007/2008

Emre Izgöl	0126571	532	e0126571@student.tuwien.ac.at
Karl Grosse	0225662	932	e0225662@student.tuwien.ac.at

Effekte:

Anbei nur eine Auflistung der Effekte:

- Shadow Volumes

Quelle: <http://www.gamedev.net/reference/articles/article1873.asp>

Stencil Shadow Volumes inklusive Selfshadowing (siehe Traumsequenz Voodoo Lounge) wurden von uns nur mittels zfail Algorithmus implementiert. Der Effekt wird nur von der CPU errechnet, eine GPU Unterstützung fehlt. Es kommt kein Shader zum Einsatz, der die Schatten weicher machen würde.

- Environment Bump Mapping (mittels Cubemap)

Quelle: Folien der Vorlesung

Mittels FBO erstellen wir an den Plätzen unserer Geometrie die Cubemap, welche nicht mehr upgedatet wird. Zu sehen ist dieser Effekt in Traumszene Nr. 1 (Tikki Tik). Zusätzlich kommt bei der Beleuchtung der Objektes unser Normal Mapping Shader zum Einsatz um die Spiegelung zu bumpen.

- Per Pixel Lighting

Quelle: OpenGL Shading Language Course

http://www.typhoonlabs.com/tutorials/glsl/Chapter_3.pdf

Hearn & Baker: Computer Graphics with OpenGL

Im Grunde genommen gibt es hier wohl nicht viel zu sagen.

- Normal Mapping

Quelle: Folien der Vorlesung

Die Tangents unserer Modelle berechnen wir uns im Modeloader zu Programmstart. Anschließend binden wir sie an das Color Array um im Shader darauf zugreifen zu können (sorry uns ist nichts besseres eingefallen auf die Schnelle). Im Shader transformieren wir anschließend den Viewvektor und den Lightvektor in den Tangent Space um die Beleuchtung zu errechnen. Die Normalmaps wurden mittels Nvidia Photoshop Plugin erstellt.

- Bloom

Quelle: http://www.gamasutra.com/features/20040526/james_pfv.htm

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

Blooming wurde anhand des zweiten Link realisiert. Zuerst rendern wir die Szene in ein FBO, bluren im Shader (mit Gauss – Filter 7x7) zuerst horizontal, texturieren damit ein Quad und rendern wieder in ein FBO, bluren im Shader diesmal vertikal, erhöhen jeweils die Helligkeit, texturieren damit anschließend

einen Quad (welcher vor der Kamera positioniert ist) und blenden diesen mit der Szene.

– Ivory

Quelle: OpenGL Orange Book

Ein unseren Bedürfnissen angepaßter „Elfenbeinshader“ um die Nelsons Column und die Tikki Tiks am Ende weiß zu machen.

Die Implementierung ist in C++ und OpenGL. Als Shading Language verwenden wir GLSL.

Hardware: ATI X1950XT (oder höher), AMD Athlon 64 X2 Dual Core Prozessor 4600+ oder höher.

Bei der Implementierung wurden keine Tiere verletzt.