

# Projektdokumentation Real-Time Rendering

Manuel EGELE (0025546, 881), Martin SZYDLOWSKI (0025313,881)

27. Januar 2004

Bezugnehmend auf das Dokument "Projektvorschlag Real-Time Rendering" können wir folgendes bekannt geben:

## 1 Status Quo

Wie geplant, wandelt der Tod durch die Bibliothek in der Unseen University, wie es sich gehört sind darin die Bücher in den regalen für den gewöhnlichen Betrachter nicht zu erblicken. (keine Angst die Zauberer in Ausbildung sehen sie aber). In der acht-eckigen Bücherei sind acht Fackeln aufgestellt. Der Flammeneffekt wird durch einen fragment-shader implementiert der zeitabhängig eine vorberechneten tileable-3d-distorsion-map und eine 1d flame color ramp kombiniert, die dann jeweils auf 2 orthogonal stehende planes gemappt werden. (looks gr8). Die Verwendung von Lightmaps war nach anfänglichen Schwierigkeiten (3ds exportiert nur Texturkoordinaten des 1. Kanals ins 3DS Format) kein Problem. Es gibt mehrere Sets von Lightmaps und es wird mittel eines Fragment Shaders (ARB\_fragment\_program) immer zwischen 2 von ihnen zeitabhängig linear intepoliert und das Ergebnis dann mit der Fragmentfarbe multipliziert. Die Reihenfolge der Lightmaps ist zufallsbestimmt, um einen möglichst realistischen Eindruck eines von Fackeln erleuchteten Raumes zu erzeugen. Für die Shadow Maps werden Depth Textures in pBuffer gerendert. Allerdings hat sie die Implementation des tatsächlichen Shadow Mappings als recht schwierig erwiesen und ist in der abgegebenen Version nicht verfügbar. Der Grund liegt vor allem in der von ATI Grafikkarten (inclusive 9800PRO) nicht unterstützen WGL\_NV\_render\_depth\_texture Extension die den Prozess wesentlich erleichtern würde. So mussten wir einen Fragment Shader erstellen, der Tiefenwerte in den RGBA buffer kopiert. Was fehlt, ist der Lookup in der Depth Texture um die Sichtbarkeit eines Fragments von der Lichtquelle aus zu bestimmen. Allerdings können die Inhalte der pBuffer am Bildschirm ausgegeben werden (siehe Steuerung). Der Demo-modus wurde bisher auch noch nicht implementiert, aber möglicherweise könnte das bis zur Präsentation behoben werden. Im Moment kann man den Tod bzw. die Kamera interaktiv steuern. Die Tastenbelegung sei der folgenden Liste zu entnehmen:

**cursor**tasten vorwärts/rückwärts, links/rechts drehen

**A** run

**S** strafe left

**F** strafe right

**F1** toggle Fullscreen;

**F2** toggle Wireframe;

**F3** toggle Freeflight;

**F4** toggle FPS

**F5** toggle Draw pBuffers on Screen

**F6** toggle Rendering to pBuffers

**F7** toggle Draw Player

**F8** toggle Render Flames

**F9** toggle Stenciled Depth Texture

**F12** = screenshot

**1 - 6** auflösung switchen (640x480 - 1600x1200)

**Im Freeflight modus:**

**E** = move up

**D** = move down

**PgUp** = cam pitch up

**PgDown** = cam pitch down

## 2 Geplante Effekte

- Lightmaps [100%]
- Real-time shadows [75%]
- Shader für Flammeneffekte[100%]
- (Bump mapping)[0%]