

# Echtzeitgrafik WS 04/05:

## Hello Kitty

Alexander Kusternig

0026571

[alexander.kusternig@gmx.at](mailto:alexander.kusternig@gmx.at)

*Mit dem Normal Mapping bin ich leider nicht mehr fertig geworden, da steckt noch irgendwo ein Bug, sodass es nicht sowohl auf ATI- wie auch auf nVidia-Karten läuft.*

### Projektbeschreibung

In dem Programm sieht man eine Katze auf einer Terrasse sitzen. Dabei wird sie von mehreren Schmetterlingen umschwirrt, während langsam die Zeit vergeht.

### Einstellungen

Über die Ini-Datei „hellokitty.ini“ lassen sich neben Auflösung und Bildwiederholrate auch einige Parameter im Programm steuern:

- **ButterflyCount:** Die Anzahl der Schmetterlinge in der Szene.
- **BlurredShadow:** Damit kann gesteuert werden ob der Schatten scharf oder leicht verschwommen dargestellt werden soll. Dieser Effekt wird erzielt, indem die Textur, in die die Schatten gerendert wurden, mehrmals leicht versetzt übereinandergeblendet wird.
- **RenderShadow:** Gibt an, ob überhaupt Schatten gerendert werden sollen.
- **FurShells:** Die Anzahl der Schichten aus Fell, die über dem Katzenbasismodel gerendert werden sollen.
- **RenderLawn:** Hier kann der dreidimensionale Rasen ausgeschaltet werden
- **RenderEnvironment:** Damit können alle Umgebungsmodels (Haus, Zaun, ...) deaktiviert werden.

### Interaktion:

- **Linke Maustaste:** Solange die linke Maustaste gedrückt ist kann man die Kamera um die Katze herumbewegen.
- **Mausrad:** Mit dem Mausrad kann gezoomt werden.
- **F:** Damit kann das Fur Rendering an- und ausgeschaltet werden.
- **B:** Das Rendering des Basismodels der Katze wird an- oder ausgeschaltet.
- **C:** Die Lichtquelle kann zwischen dem normalen Sonnenlicht und einer Lichtquelle, die aus Kamerarichtung kommt, umgeschaltet werden. Dadurch lässt sich der Normal Mapping Effekt besser erkennen.
- **F5:** Das Programm erzeugt einen Screenshot im BMP Format und speichert ihn in den Programmordner.
- **P:** Damit können Post-Processing Fragment Programs (shaders/post processing/\*.txt) aktiviert und deaktiviert werden. Diese Fragment Programs arbeiten auf einer Texture, die aus dem Framebuffer erzeugt wird.

- **Plus, Minus:** Mit diesen Tasten kann der aktive Post Processing Effekt durchgeschaltet werden.
- **A, S:** Uhrzeit ändern.

### Effekte:

- **Fur Shading:** Um den Fell-Effekt zu erzielen wird das Katzenmodel 16 mal in „Shells“ gerendert, die immer leicht größer sind als ihr Vorgänger. Dazu wird ein Vertex Program (shaders/furshells.txt) verwendet. Das eigentliche Fragment Program, das für jedes Shell angewendet wird (shaders/fur.txt), benutzt mehrere Texturen um den gewünschten Effekt zu erzielen. Als 1. Textur wird eine RGB Basistextur benutzt um die einzelnen Haare einzufärben. Als 2. Textur eine „rules“ Textur, die die Regeln vorgibt, nach denen das Fell an dieser Stelle gerendert werden soll. Dabei stehen im R und G Kanal dieser Textur die Richtung, in die das Fell fallen soll und im B Kanal die Länge des Fells an diesem Texel. Die 3. Textur enthält schließlich die Information, wo sich die einzelnen Haare befinden und wie lang diese sind, wobei hier im Prinzip nur der Alphakanal relevant ist. Das Gras benutzt einen ähnlichen Effekt, also ebenfalls Rendern in mehreren übereinanderliegenden Schichten, allerdings mit einem weitaus simpleren Fragment Program (shaders/lawn.txt).
- **Normal Mapping:** Der Steinboden benutzt eine Normal Map, diese wird im Object Space im Fragment Program (shaders/normalmap.fp.txt) berechnet, wobei nur Diffuse und Ambient Color relevant sind. Die Normal Map wurde mit dem nVidia NormalMapFilter Tool für Adobe Photoshop generiert. Da ich die Texturkanäle direkt als Normale verwende, da der Boden immer gleich orientiert ist, vertausche ich den G und B Kanal, da in der Normal Map der B-Kanal nach oben zeigt, während es bei mir der G-Kanal ist.
- **Sphere Mapping:** Sowohl das Glitzern der Augen der Katze wie auch der Glanz der Schmetterlingsflügel wie auch die Reflektion im Glas wurde mit Sphere Maps erzeugt.
- **pBuffers:** Die Schatten werden in einem separaten Renderingschritt in einen 1024x1024 pBuffer gerendert. Dabei wird die Kamera über der Szene positioniert, blickt gerade nach unten und die Ausdehnung der Schatten wird über ein Vertex Program (shaders/shadow.txt) bestimmt, das als Eingabeparameter die Lichtrichtung nimmt. Anschließend wird die daraus erzeugte Schattentextur mehrfach über den Boden geblendet, um einen etwas weicheren Schatten zu erzeugen. Im Prinzip handelt es sich hier also um eine Lightmap.
- **Render to Texture:** Mit glCopyTexSubImage2D wird der Frame Buffer in eine Textur geschrieben, die für die Post Processing Effekte benutzt wird.

### Benutzte Tools:

- Milkshape 3D
- Katzenmodel von [www.3dcafe.com](http://www.3dcafe.com)
- NormalMapFilter von <http://developer.nvidia.com>
- Transmogrifying Textures von <http://developer.nvidia.com>
- Adobe Photoshop