

# Echtzeitgraphik

Christof Furxer  
Peter Vorlauffer

9805680 532  
9727036 532

WS 07/08

# Tribbles

Leider haben wir aus Zeitmangel noch keine Kamerafahrten und Objectbewegungen einbauen können (kommt noch). Die Tribbles lassen ihre Haare zum beat der Musik erzittern, und ein directional light bewegt sich durch die Szene. Momentan kann man die Kamera selbst bedienen:

Esc = exit

q = up

e = down

w,a,s,d = move

Mouse = look around

f = fullscreen

m = change camera mode

## Effekte:

### Fur rendering

Shaderfile: fur.vert und fur.frag

Die Texturen der einzelnen layer werden durch zufällig verteilte Punkte erzeugt, wobei bei jedem Start der Verteilung der selbe random-seed Wert gesetzt wird, um immer dieselben Punkte zu treffen. Mit zunehmender Distanz werden weniger Punkte gesetzt, um so die Länge der Haare zu variieren, außerdem nimmt auch der Alphawert ab und die Farbintensität zu. Weiters wird das Prinzip der sich wiederholenden Zufallswerte auch für die Farbe der einzelnen Haare verwendet. So können zufällige Farbabweichungen für einzelne Haare gewählt werden.

Um allen Tribbles dieselbe Gravityrichtung übergeben zu können, wird für jeden pro frame seine ModelMatrix berechnet, damit die World-gravity transformiert und an den shader übergeben. Vertices werden mit zunehmender Haarlänge immer stärker in Richtung der gravity verschoben. Die Bewegungsrichtung und -geschwindigkeit eines Tribbles beeinflussen Richtung und Stärke der Haarbiegung. Inspiration haben wir uns hier geholt: <http://www.xbdev.net/directx3dx/specialX/Fur/index.php>, jedoch nicht das Interfurshadowing implementiert, da das unserer Ansicht nach keine optische Verbesserung bringt (nur doppelten rendering Aufwand). Schönere Effekte lassen sich mit Varianz der Haarfarbe, Abhängigkeit der Farbe und des Alphawerts von der Haarlänge und linearem Filtering der Haartexturen erzeugen.

### Normal mapping

Eine RGB sinus-txtur wird generiert und als Normalmap an den Shader übergeben.

Die resultierenden Werte werden als Höheninformation genutzt und es wird ein per-pixel-lighting durchgeführt.

Quelle: wie in der VO besprochen.

### Shadow mapping

Mit Hilfe eines Framebufferobjects wird die Tiefeninformation aus Sicht des directional Lights in eine Textur gerendert. Wenn der alphawert ein bestimmtes limit (0.1) überschreitet, wird der tatsächliche Tiefenwert geschrieben, ansonsten 1.0. Um Löcher in der Mitte der Tribbles zu vermeiden, werden im Pixelshader aller Shadowreceiver nicht nur das Tiefen-texel an der projizierten Texcoord, sondern auch 8 Nachbarn berücksichtigt. Je nach Anzahl der beschatteten Nachbarn wird die Stärke der Beschattung variiert.

Quelle: wie in der VO besprochen.

### **Physic(Bullet)**

Fürs herumspringen der Tribbles sorgt das Physiclibrary Bullet. Der Mittelwert der letzten 20 Geschwindigkeitsvektoren bestimmt die Krümmung der Haare (biegen sich entgegengesetzt zur Bewegungsrichtung).

Verwendete libs:

Opengl, GLSL, glut, glew

Für Matrizen- und Vectorberechnungen verwenden wir folgende Klassen von David Eberlys Wild Magic engine:

[Wm4Matrix4.h](#)

[Wm4Matrix3.h](#)

[Wm4Vector3.h](#)

[Wm4Quaternion.h](#)

[Wm4BSplineCurve3.h](#)