

Gruppe littlecrew

Stefan Ring, 9725446
Adrian Gligor, 9926374

Demobeschreibung:

Unsere Demo besteht aus zwei Szenen, die an der klassischen Demo "Second Reality" angelehnt sind.

In der ersten Szene hüpfen zwei Polyeder über ein Schachbrett. Wenn es dieses berührt, verformt es sich (es schwabbelt). Weiters tritt nach einiger Zeit ein zweites Polyeder aus dem Hintergrund hervor, das erste hüpfen jedoch durch, als ob das zweite Polyeder keine Substanz hätte. Auf einmal wird es fest und vom ersten Polyeder mitgerissen, wobei es stark schwabbelt wenn es oben oder unten an dieses anstößt.

In unserer Szene sind die Polyeder wie im original transparent, und reflektieren sich im Schachbrett. Der Schwabbeleffekt ist mittels eines Vertex Shader Programmes realisiert.

In der zweiten Szene, wird eine Linse über ein Bild bewegt und dieses dabei vergrößert und verzerrt. Die Linse ist dabei als reines Pixel-Shader-Programm implementiert. Das Bild zoomt stark aus, wobei es sich an den Rändern wiederholt (Kachel). Die Linse berücksichtigt die Kachelung. Nach einiger Zeit fängt die Linse an sich um die eigene Achse zu drehen.

Implementierung:

Für das Programmieren des Vertex Shaders wurde das NeHe Tutorial 47 verwendet, das Schwabbeln mittels einer gedämpften Schwingungsgleichung implementiert. Der Transparenzeffekt wird mit einer einfachen Multipass-Technik erreicht: statt echtem Depth-Sorting wird hier die Konvexität ausgenutzt und mittels Backface-Culling abwechselnd hintere (Backfaces) und vordere (Frontfaces) gezeichnet. Die Reflexion wird mittels eines zusätzlichen Rendschrittes gezeichnet und wird mittels Stencil Buffer auf den Bereich des Schachbrettes eingeschränkt. Weiters wurde ein Nvidia-Paper über Reflexionen mittels des Stencil-Buffers als hilfreich empfunden.

Für die Programmierung des Pixel Shaders wurden ebenfalls das NeHe Tutorial 47 sowie die Cg Dokumentation von NVidia herangezogen. Der Effekt ist etwas "geschummelt", da nicht von dem ausgegangen wird, was schon im Buffer ist. Stattdessen werden die Texturkoordinaten so gesetzt, dass - abgesehen von der Linsenverzerrung - die Hintergrundtextur nochmal über sich selbst gezeichnet wird. Der Pixelshader verwendet 3 Texturen: eine für den Hintergrund, eine für die Offsets der Verzerrung, und eine für das bläuliche Aussehen der Linse. Für das "Drehen" der Linse ist danach noch ein Vertex-Programm dazugekommen, weil eine richtige Drehung durch die perspektivische Verzerrung seltsam ausgesehen hätte.

Grafiken und Musik wurden zum größten Teil von 2nd Reality entlehnt. Die Musik wird mit der MOD-Player Library BASS abgespielt

Quellen:

NeHe Tutorial 47: <http://nehe.gamedev.net/data/lessons/lesson.asp?lesson=47>

Nvidia-Paper: http://developer.nvidia.com/object/reflections_shadows_stencil_buffers.html

Second Reality: http://www.mindcandydvd.com/demos/oldskool/Second_Reality.zip

BASS: <http://www.un4seen.com/bass.html>