

Übungen zu Bildverarbeitung und Graphikprogrammierung in C++

Aufgabe 11 (H) Texture Mapping

Sie können nun 3D Objekte zeichnen, Szenen modellieren und mit Licht versehen. Jedoch erscheinen die 3D Objekte in der Szene immer noch sehr unrealistisch, da den Objekten selbst bei guter Lichtgebung lediglich Farben einer Farbpalette zugewiesen werden. Um die Szene noch realistischer wirken zu lassen, kann man Texturen verwenden.

Lesen Sie sich Kapitel 9, 'Texture Mapping', im OpenGL Programming Guide (Red Book) durch. Texturen sind im Wesentlichen Bilder, die auf die Oberfläche von Strukturen (z.B. Polygonen) abgebildet werden.

Um Bilder (Texturen) laden zu können, verwenden Sie die 'imelib' von Blatt 2. Wie schon aus einer früheren Aufgabe bekannt, können Sie mit der Klasse Bitmap eine .bmp-Datei laden und verwenden. Hinweis: 24 Bit Bitmaps werden bezüglich der Farbkanäle gemäß BGR (blau-grün-rot) und nicht wie in OpenGL verwendet (RGB) abgespeichert. Sie könnten die Farbkanäle folgendermaßen mit dem Extensionparameter `GL_BGR_EXT` beim Erzeugen der Textur vertauschen:

```
glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, mytexture.getWidth(), mytexture.getHeight(), 0, GL_BGR_EXT, GL_UNSIGNED_BYTE, mytexture.getData());
```

- Bilden Sie verschiedene Texturen Ihrer Wahl (z.B. von <http://www.texturewarehouse.com>) auf einen Tetraeder und Quader ab, die auf dem Tisch neben der Teekanne aus dem vorigen Aufgabenblatt stehen. Bitte vergessen Sie nicht die Bilder ins *SVN*-Repository zu stellen.
- Da man diese 3D Objekte skalieren kann, verwenden Sie nach Möglichkeit *mipmapping*, um bei bestimmten Skalierungsgrößen auf geringere oder höhere Auflösung der Texturen umzuschalten.
- Versuchen Sie, mit geeigneten Texturen die Teapot-Szene realistischer zu machen (z.B. Holztisch).
- Die Teekanne soll aus Glas und deshalb durchsichtig sein - realisieren Sie dies mittels *blending*.

Aufgabe 12 (H) Wassersimulation

Ziel dieses Aufgabenblocks ist es eine Durchfahrt durch eine Unterwasserlandschaft zu implementieren. Es soll außerdem möglich sein über der Wasseroberfläche zu schweben. Die Wellenbewegung der Wasseroberfläche soll durch eine einfache Methode simuliert werden. 100 prozentiger Realismus ist hier nicht erforderlich.

- a) Erstellen Sie eine blaue Oberfläche aus 40x40 Punkten mittels `glBegin(GL_TRIANGLES)`
- b) Animieren Sie auf Knopfdruck ('A','a') diese Oberfläche mit einem von Ihnen implementierten Wellengenerator.

- c) Berechnen Sie die Normalen der Oberfläche und bringen sie diese so in den Code ein, dass ein kreisendes Licht korrekt die Farbe der Wasseroberfläche manipuliert. Zeigen sie diese Senkrechten auf Knopfdruck ('N','n') an. Die Lichtquelle soll auf Knopfdruck zu sehen sein ('L','l')
- d) Laden Sie die Wassertextur mittels der Funktion aus der *'imagelib'* in den Speicher. Die Texturen finden sie auf der Website.
- e) Legen Sie die Wassertextur über die Wasseroberfläche.
- f) Auf Knopfdruck ('O','o') soll die Wasseroberfläche durchsichtig erscheinen.
- g) Auf Knopfdruck ('B','b') soll ein Meeresgrund angezeigt werden. Verwenden Sie dazu die in der Vorlesung vorgestellte Methode für Heightmapping. Erzeugen Sie mit Hilfe eines Malprogramms eine eigene Heightmap. Laden Sie die Heightmap mittels der Funktion aus der *'imagelib'* in den Speicher. Das Höhenprofil aus der Heightmap soll in einer Displayliste gespeichert werden.
- h) Simulieren Sie mit Hilfe von GL_FOG trübes Wasser. Der Nebel soll auf Knopfdruck ('F','f') ein- und ausgeschalten werden können.
- i) Um die Durchfahrt durch die Szene zu realisieren, sollen auf Tastendruck und Mouseinteraktion Rotation um die eigene Achse, Neigung des Blickes nach unten und oben, und Vorwärts- bzw. Rückwärtsbewegung möglich sein.