

Übungsblatt 6: Mehr Meshes

Abgabe:

Dieses Übungsblatt ist als Gruppe zu lösen. Die Lösung ist bis **Dienstag, den 15. Juni 2010, 12:00 Uhr s.t.** über UniWorx (<http://www.pst.ifi.lmu.de/uniworx>) abzugeben.

Es werden nur die Formate PDF und Plain-Text (UTF-8) akzeptiert. Erstellen Sie für jede Aufgabe ein Unterverzeichnis nach dem Schema <Übungsblatt>-<Aufgabe>, d.h. die Lösung der ersten Aufgabe kommt in ein Verzeichnis 6-1/. Packen Sie alle Dateien in eine ZIP-Datei und laden Sie diese bei UniWorx hoch. Wenn Sie Formatierungsvorgaben nicht einhalten, werden bis zu zwei Punkte abgezogen. Lösungen müssen zumindest im CIP-Pool fehlerfrei kompilieren und laufen. Bitte geben Sie nur Quellcode ab, keine kompilierten Dateien (bitte auch keine moc_xx.*-Dateien). Es können maximal 20 Punkte erreicht werden.

Aufgabe 1: Marching Cubes (3 Punkte)

Marching Cubes ist ein Algorithmus zur Umwandlung von Voxel-Daten in Gitternetze (Meshes). Recherchieren Sie den Marching-Cubes-Algorithmus und erklären Sie seine Funktionsweise in 200-400 Wörtern. Welche Nachteile hat der Original-Algorithmus?

Tip: Marching Squares ist ein konzeptionell ähnlicher Algorithmus für den 2D-Raum. Zum Verständnis von Marching Cubes kann es helfen, erst einmal die Marching Squares zu betrachten. Geben Sie eine Datei *marchingcubes.txt* oder *marchingcubes.pdf* im Unterverzeichnis 6-1/ ab.

Aufgabe 2: Loft (12 Punkte)

Ein Loft-Objekt entsteht, wenn man ein Polygon entlang eines Pfades extrudiert. Dabei wird das Polygon immer so rotiert, dass es senkrecht auf dem Pfad steht. An den Eckpunkten des Pfades wird das Polygon so rotiert, dass es den Winkel zwischen den beiden Pfadsegmenten halbiert. Implementieren Sie eine Anwendung, die ein geloftetes 3D-Modell anzeigt. Dieses ist durch den Pfad *loftPath* und das Polygon *loftShape* definiert:

| | |
|---|--|
| <pre>GLfloat loftShape[] = { 0.5, 0, 0, 0.25, 0, -0.5, -0.25, 0, -0.5, -0.5, 0, 0, -0.25, 0, 0.5, 0.25, 0, 0.5 };</pre> | <pre>GLfloat loftPath[] = { 0, 0, 0, 0, 2, 0, -2, 6, 0, -2, 8, 0, 2, 8, 0 };</pre> |
|---|--|

Sie können davon ausgehen, dass die *loftShape* immer mit ihrem Mittelpunkt (0,0,0) am Pfad entlangläuft. Der Pfad verläuft innerhalb der XY-Ebene. Mit der Maus soll das Loft-Objekt rotiert werden können. Achten Sie darauf, dass die Enden des Loft-Objekts geschlossen sind. Geben Sie ein Projekt *loft.pro* inkl. Quellcode und Headern in einem Unterverzeichnis 6-2/ ab.

Aufgabe 3: Modelle laden mit Lib3DS (5 Punkte)

Lib3DS (<http://code.google.com/p/lib3ds/>) ist eine plattformunabhängige Bibliothek zum laden und verarbeiten von 3D-Szenen im 3DS-Format. Laden Sie lib3DS herunter und kompilieren Sie es. Unter Linux, OS X und MinGW geht dies mit der klassischen Kombination aus `./configure`, `make`, und ggf. `make install`. Unter Debian und Ubuntu kann man die Bibliothek auch mit `apt-get install lib3ds-dev` installieren. Im CIP-Pool ist dieses Paket installiert. Falls Sie lib3DS lokal installieren, müssen Sie beim Kompilieren der Anwendung das Verzeichnis zum Include-Pfad (`-I dir`) und zum Bibliothekspfad (`-L dir`) hinzufügen.

Implementieren Sie eine Anwendung, die ein einfaches 3D-Modell Ihrer Wahl aus einer `.3ds`-Datei lädt und dieses anzeigt. Mit der Maus soll das Modell rotiert werden können. Die (einheitliche) Farbe des 3D-Modells soll abhängig vom der Kameraposition sein.

Tipp: bilden Sie z.B. die Rotationen um die 3 Achsen auf die 3 Farbkanäle ab.

Ein hilfreiches Tutorial zu lib3ds gibt es unter <http://www.donkerdump.nl/node/207>.

Geben Sie ein Projekt `3ds.pro` inkl. Quellcode, Headern und 3D-Modell in einem Unterverzeichnis `6-3/` ab.

Viel Erfolg.