

## 3D Programmierpraktikum

### Abgabetermin:

Die Lösung zu diesem Übungsblatt ist bis zum 04. Juni 2007 abzugeben.

### Inhalt:

In diesem Übungsblatt vertiefen Sie Ihre bereits erworbenen Kenntnisse der Matrizen in *OpenGL*. Dazu werden hierarchische Transformationen betrachtet. Diese sind z.B. bei Szenegraphen zu finden. Zusätzlich erlernen Sie die Grundtechniken der Texturierung in *OpenGL*.

### Aufgabe 12 (H) Der Szenegraph

Der Szenegraph ist eine in der Computergraphik häufig verwendete hierarchische Darstellungsform von 3D-Szenen. Es handelt sich dabei um einen DAG (Directed Acyclic Graph). In diesem Graph kann man (mindestens) drei Typen von Knoten unterscheiden. Gruppenknoten dienen der Zusammenfassung mehrerer Kinder zur Hierarchischen Organisation von Objekten, Transformationsknoten beschreiben geometrische Transformationen, also z.B. Rotationen oder Translationen, und Strukturknoten beschreiben geometrische Objekte, z.B. ein Tisch oder Stühle.

Beim Rendern durchläuft man den Graphen nun mit einer Tiefensuche, wobei die Kinder von Gruppenknoten in fester Ordnung abgearbeitet werden (von links nach rechts). Gruppenknoten sollten „state separating“ sein d.h. Material und Beleuchtungs Statusänderungen sollten sich *nicht* auf die Kindknoten von *benachbarten* Gruppenknoten auswirken.

- a) Erstellen Sie einen Szenegraphen der Aufgabe 11 des letzten Übungsblattes.

Geht man in einem Szenegraphen „von unten nach oben“, so entspricht dies der Ausführung von invertierten Transformationen. Hierdurch ist es z.B. möglich, die Position eines Stuhls relativ zum Tisch durch folgende Berechnung zu bestimmen:

$$\mathbf{T}_{\text{gesucht}} = \mathbf{T}_{\text{Tisch}}^{-1}$$

- b) Berechnen Sie analytisch die Linksinverse  $\mathbf{H}^{-1,L}$

$$\mathbf{H}^{-1,L} = \begin{pmatrix} \mathbf{R}^* & t^* \\ 0 & 1 \end{pmatrix} \text{ mit } \mathbf{H}^{-1,L} \cdot \begin{pmatrix} \mathbf{R}_x & t_x \\ 0 & 1 \end{pmatrix} = \mathbf{I}$$

in Abhängigkeit von den Parametern  $\mathbf{R}_x$  und  $t_x$  einer homogenen Transformationsmatrix.  
*Tipp:* Da  $\mathbf{R}_x$  eine orthonormale Rotationsmatrix ist, gilt  $\mathbf{R}_x^{-1} = \mathbf{R}_x^T$ .

### Aufgabe 13 (P) Bewegung durch eine texturierte Welt

In dieser Aufgabe erstellen Sie einen virtuell begehbaren Raum. Die Kontrolle für den Benutzer soll sich an bekannten 3D-Spielen orientieren, die die Szene aus der *ich*-Perspektive darstellen.

- a) Laden Sie sich die Datei `texture.zip` von der Praktikums-Homepage herunter und verwenden Sie die Datei `texture.h` in dieser Aufgabe. Dieser Header ermöglicht das Einlesen von Bitmap-Dateien sowohl unter Windows als auch unter Linux. Bezüglich der Bitmaps gelten zunächst folgende Regeln:
  - Die Pixelgröße und das Seitenverhältnis des Bitmaps spielt keine Rolle.
  - Die Farbtiefe beträgt 24 Bit.
- b) Erstellen Sie einen Raum, der mindestens drei verschiedene Texturen enthält (mind. eine für die Wand, sowie jeweils unterschiedliche Texturen für Boden und Decke). Geben Sie diesem Raum eine hinreichende Größe, so dass Sie sich später in ihm bewegen können.
- c) Erstellen sie beliebige Objekte um den Raum zu möblieren (es kann sich auch um abstrakte Objekte handeln), versehen sie auch diese Objekte mit Texturen. Verwenden sie für statische Objekte Displaylists.
- d) Erstellen Sie eine 2D-Anzeige, die zu jeder Zeit die aktuellen Bilder pro Sekunde (*frames per second*) ausgibt. Dies kann an einer beliebigen Stelle des Bildschirms geschehen, sollte aber zu jederzeit sichtbar sein.
- e) Implementieren Sie die Kontrollen für den *Betrachter*. Diese sollen folgende Eigenschaften haben:
  - **Laufen des Betrachters:** Durch Drücken der Pfeiltasten läuft der Betrachter. Dabei geht sie/er vorwärts bzw. rückwärts beim Drücken der Rauf- bzw. der Runter-Taste. Beim Betätigen der Links- bzw. der Rechts-Taste dreht sich der Betrachter *nicht*, sondern sie/er macht einen Seitwärtsschritt in die jeweilige Richtung. Zusätzlich soll beim Laufen (in jede Richtung) eine sinus-förmige Auf- und Abbewegung des Körpers implementiert werden, um eine realistischere Bewegung zu simulieren.
  - **Sicht des Betrachters:** Bei einer Bewegung der Maus nach links bzw. rechts *dreht* sich der Betrachter nach links bzw. rechts. Bei der Bewegung hoch bzw. runter sieht sie/er nach oben bzw. unten. Achten Sie bei der Bewegung in y-Richtung darauf, dass der Benutzer seinen Kopf meist nicht 360° in dieser Richtung drehen kann. Achten Sie darauf, dass der Mauszeiger die Grenzen des Bildschirms nicht erreichen kann, da eine weitere Drehung sonst nicht möglich ist. Betrachten Sie hierzu die Funktion `void glutWarpPointer(int, int)`, die ein Zurücksetzen des Zeigers ermöglicht.
  - **Qualität der Texturen:** Durch Drücken der Taste *F* soll die Qualität der Texturen verändert werden, d.h. die drei Filterstufen sollen durchgeschaltet werden.

### Aufgabe 14 (P) Terrain Erzeugung

In dieser Aufgabe sollen sie eine Landschaft erzeugen durch die man sich als Betrachter bewegen kann (erneut *Ich*-Perspektive).

- a) Erzeugen sie sich eine Height-Map mittels eines Malprogramms (Graustufen Bild). Laden sie dieses Bild mittels des zur Verfügung gestellten Codes in den Speicher.

- b) Erzeugen sie aus der Height-Map eine Landschaft, verwenden sie gegebenenfalls eine zweite Textur um die Landschaft einzufärben. Speichern sie die erzeugte Kontur in einer Displaylist.
- c) Fügen sie einen Betrachter hinzu sie können entweder eine Steuerung analog zu Aufgabe 13 verwenden (achten sie darauf, das der Betrachter nicht unter dem Boden läuft), oder einen „fliegenden“ Betrachter implementieren.

### **Aufgabe 15 (T) Konzeption des Teamprojekts**

NächsteWoche sollen die Projektaufgaben beginnen. Dazu werden in dieser Aufgabe die Projektvorschläge konkretisiert und Arbeitsziele definiert. Erstellen Sie zusammen mit Ihrem Projektteam eine Präsentation (ca. 5 Folien, Inhalt: siehe Slides 23.5.07), in der Sie die Zielsetzung sowie die geplante Herangehensweise beschreiben. Verwenden Sie dazu die PowerPoint-Vorlage<sup>1</sup>. Erstellen Sie außerdem eine Seite im Wiki, auf der Sie einen Projektplan und eine Projektbeschreibung mittels geeigneter Mockups (Screenshots, Zeichnungen etc.) erstellen. Zusätzlich sollen die Anforderungen und eine Funktionalitätsbeschreibung auf dieser Seite abgelegt werden. Diese Seite gilt als zusätzliche Diskussionsgrundlage für die erste Besprechung.

---

<sup>1</sup>[http://www.medien.ifi.lmu.de/lehre/ss07/3dp/Slide\\_Template.pot](http://www.medien.ifi.lmu.de/lehre/ss07/3dp/Slide_Template.pot)