

Programmierübung 1: Dreiecksgitter und Transformationen

Computergrafik, Herbst 2012

Abgabedatum: Donnerstag 4. Oktober, 11:59

In dieser Übung sammeln Sie praktische Erfahrung mit 3D Objekten aus Dreiecksgittern und 3D Transformationen. Wir stellen auf Ilias zusätzlich zur Übungsbeschreibung Java Code zur Verfügung, der Ihnen den Einstieg in die 3D Programmierung erleichtern soll. Ihre Lösung muss auf diesem Code aufgebaut sein, weil wir diese Grundlage auch in den weiteren Übungen verwenden werden. Diese Übung muss bis Donnerstag 4. Oktober, 11:59 Uhr auf Ilias abgegeben werden. Die Korrektur der Übungen findet gemäss Einschreibeliste direkt mit den Vorlesungsassistenten statt.

Starten des Java Code im ExWi Pool

Kopieren Sie zuerst den vorgegebenen Java Code in ein temporäres Verzeichnis und entpacken Sie das .zip Archiv. Starten Sie dann die **Eclipse Entwicklungsumgebung**. Verwenden Sie den Standard Workspace, oder legen Sie falls gewünscht einen neuen Workspace an. Importieren Sie nun den Java Code in diesen Workspace, indem Sie den Menüpunkt "File → Import → Existing project into workspace → Next" und dann den temporären Ordner mit dem Java Code wählen. Nun sollten Sie das Projekt "simple" ausführen können. Machen Sie sich mittels der JavaDoc Dokumentation mit dem Aufbau des Codes bekannt.

Installieren des Java Code auf einem eigenen Rechner

Wenn Sie auf Ihrem eigenen Rechner entwickeln wollen, müssen Sie einige Zusatzsoftware installieren. Der Java Code verwendet **JOGL** als Schnittstelle zwischen Java und **OpenGL**. Holen Sie sich eine **aktuelle Version von JOGL hier**. Ausserdem benötigen Sie eine **aktuelle Version von gluegen** als zusätzliche Schnittstelle zu OpenGL. Für ein 64-bit Windows System wählen Sie zum Beispiel die Datei "jogl-2.0-*-windows-amd64.7z". Kopieren Sie die JOGL und gluegen Dateien in Ordner Ihrer Wahl und entpacken Sie die Archive.

Starten Sie Eclipse und gehen Sie vor wie oben, um den Code in einen Workspace zu importieren. Nun muss Eclipse angepasst werden, um die JOGL und gluegen Libraries zu verwenden. Die Projekte "simple" und "jrtr" sind so eingerichtet, dass sogenannte "User Libraries" in Eclipse verwendet werden. Diese User Libraries heissen "jogl" und "j3d", und sie müssen nun eingerichtet werden. Dazu wählen Sie "Window → Preferences" in Eclipse

und gehen zu "Java → Build Path → User Libraries". Klicken Sie auf "new" und erstellen Sie eine neue User Library "jogl". Dann wählen Sie "Add external JARs...". Hier navigieren Sie zu Ihren JOGL .jar Dateien und wählen "jogl-all.jar". Wählen Sie noch einmal "Add external JARs..." und fügen Sie auch "gluegen-rt.jar" hinzu, indem Sie den entsprechenden Pfad zu Ihrer Datei angeben.

Zusätzlich müssen Sie die "Native library location" setzen, damit die JOGL und gluegen native libraries gefunden werden können. Dazu klicken Sie auf das "+" Symbol jeweils neben den JOGL und gluegen .jar Dateien, die Sie eben hinzugefügt haben. Nun wählen Sie "Native library location" und dann "Edit". Hier geben Sie den Pfad zum "lib" Verzeichnis Ihrer JOGL, bzw. gluegen, Dateien an.

Der Java Code benutzt weiter Matrix- und Vektorfunktionen aus der Java3D Bibliothek, die [hier erhältlich](#) ist. Es ist am einfachsten, die binary Version mit dem Installer einzurichten. Erstellen Sie in Eclipse eine weitere User Library mit dem Name "j3d" durch gleiches Vorgehen wie oben und fügen Sie "vecmath.jar" als externes JAR hinzu. Es ist nicht nötig, eine "Native library location" zu setzen.

1 Zylinder (3 Punkte)

Schreiben Sie eine Funktion, welche ein Dreiecksgitter für einen Zylinder generiert. Die Funktion sollte Arrays für Eckpunkte, Farben, und Indizes zurückliefern, welche dann ähnlich verwendet werden wie für den Würfel im bestehenden Java Code. Die Funktion sollte als Eingabeparameter die Auflösung des Dreiecksgitters annehmen. Dieser Parameter gibt an, wie viele Segmente verwendet werden, um den Zylinder darzustellen. Die Ober- und Unterseite des Zylinders sollte mit einer Scheibe geschlossen werden. Verwenden Sie ein Farbschema, das den Dreiecken verschiedene Farben zuordnet.

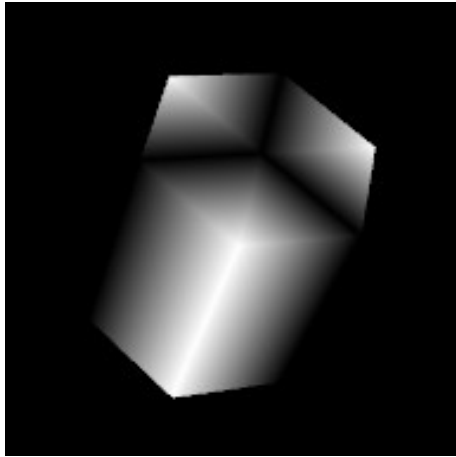
Achtung: Beachten Sie, dass die Reihenfolge der Dreieckseckpunkte eine Rolle spielt. Die Eckpunkte müssen im Gegenuhrzeigersinn geordnet sein, wenn die Fläche von aussen betrachtet wird. Diese Orientierung ist als "front facing" definiert. Wenn die Eckpunkte im Uhrzeigersinn geordnet sind, wird das Dreieck nicht gezeichnet. Diese Orientierung ist als "back facing" definiert.

2 Torus (3 Punkte)

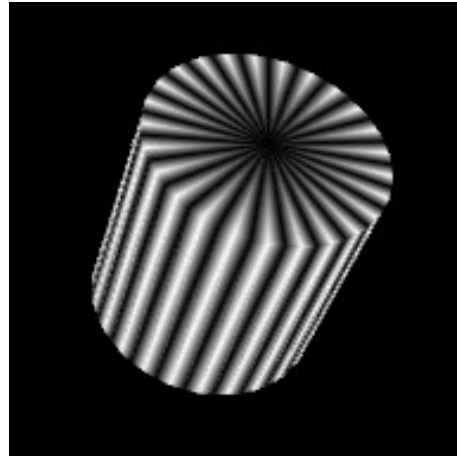
Schreiben Sie eine Funktion, welche einen Torus erzeugt, ähnlich wie der Zylinder oben. Diese [Wikipedia Seite](#) enthält nützliche Informationen über die mathematische Darstellung eines Torus.

3 Animation (4 Punkte)

Konstruieren Sie ein animiertes Objekt, das aus mindestens drei Teilen besteht. Verwenden Sie Zylinder, Tori, und Würfel. In Ihrer Animation sollen sich mindestens drei Teile relativ zu den anderen bewegen. Mögliche Beispiele sind:



Zylinder mit 6 Segmenten



Zylinder mit 50 Segmenten

- Ein Fahrzeug mit drehenden Rädern, das auf einer Ebene im Kreis fährt. Die Räder drehen relativ zum Fahrzeug, welches sich relativ zur Ebene bewegt.
- Ein Flugzeug mit drehendem Propeller, das in einem Kreis über eine Ebene fliegt.
- Ein Helikopter mit drehenden Rotoren, der in einem Kreis über eine Ebene fliegt.

Modellieren Sie die Objekte, indem Sie die Grundobjekte (Zylinder, Tori, Würfel) in die gewünschte Form skalieren und transformieren.