

Programmierübung 1: Dreiecksgitter und Transformationen

Computergrafik, Herbst 2011

Abgabedatum: Donnerstag 6. Oktober, 11:59

In dieser Übung sammeln Sie praktische Erfahrung mit 3D Objekten aus Dreiecksgittern und 3D Transformationen. Wir stellen 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 6. 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**. Kopieren Sie die JOGL Dateien in einen Ordner Ihrer Wahl. 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.

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 Library zu verwenden. In Eclipse wählen Sie "Window → Preferences" und gehen zu "Java → Build Path → User Libraries". Klicken Sie auf "jogl" und dann "Add JARs...". Hier navigieren Sie zu den JOGL .jar Dateien. Wählen Sie die folgenden JAR-Dateien

- jogl.all.jar

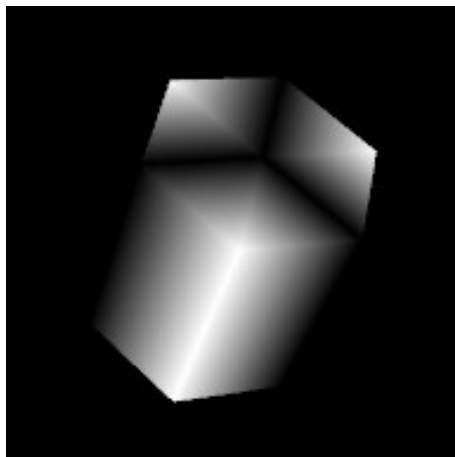
- `nativewindow.all.jar`
- `gluegen-rt.jar`

Zusätzlich müssen Sie die "Native library location" setzen, damit die JOGL native library gefunden werden kann. Dazu klicken Sie auf das "+" Symbol neben den JOGL .jar Dateien, die Sie eben hinzugefügt haben. Nun wählen Sie "Native library location" und dann "Edit". Hier geben Sie denselben Pfad wie vorher zu den JOGL Dateien. Eventuell müssen Sie den Pfad zur Java3D Bibliothek durch ähnliches Vorgehen anpassen.

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.



Zylinder mit 6 Segmenten



Zylinder mit 50 Segmenten

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:

- 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.