

# Praktikum Computergraphik

## – Übungsblatt 1 – Lehrstuhl für Computergraphik und Multimediasysteme Julian Bader

### 1. Themen

Modellierung, Transformation, Beleuchtung, Texturierung

### 2. Übersicht

In dieser Aufgabe sollen Sie sich zunächst mit der Modellierung und Darstellung von Objekten in *OpenGL* auseinandersetzen. Hierzu soll eine einfache Szene mit Hilfe des *OpenGL* Framework modelliert werden, die einen bewaldeten Planeten darstellt, auf dem zufällig zwei Typen von Bäumen verteilt sind. Der Planet soll von einer Sonne umkreist werden, die ihn beleuchtet. Außerdem soll der Planet von einer Skybox mit Sternentextur umschlossen sein.

### 3. Aufgabe

Modellieren Sie unter Verwendung des zur Verfügung gestellten Rahmen-Programms zunächst folgende Grundprimitive in *OpenGL*:

- Kugel (Planet, Krone der Laubbäume)
- Zylinder (Baumstamm)
- Kegel (Krone der Tannenbäume)

Verwenden Sie kein *GLU* oder *GLUT*, sondern modellieren Sie die Primitive selbst durch geeignete Aufrufe von `glVertex3f` und der Verwendung passender *OpenGL*-Primitive. Hierbei soll der Detailgrad (d.h. die Anzahl der Polygone) der Körper einstellbar sein.

Definieren Sie anschließend die zwei genannten Baumtypen, indem Sie die geometrischen Primitive mittels Transformationen entsprechend anordnen und verteilen Sie diese abschließend zufällig auf der Planetenoberfläche. Überlegen Sie sich hierfür eine geeignete Transformations-Reihenfolge. Färben Sie die Primitive entsprechend ihrer Verwendung in geeigneten Farbtönen ein.

Ist die Szene fertig modelliert, soll diese noch mittels *Smooth Shading* und *Flat Shading* beleuchtet werden. Erweitern Sie hierzu ihre Geometriedefinition um geeignete, analytische (kein Kreuzprodukt) Oberflächen-Normalen und ihre Szene um eine *OpenGL* Lichtquelle. Platzieren Sie des Weiteren eine kleine gelbe Kugel in ihrer Szene, welche die Lichtquellen-Position repräsentiert und lassen Sie die Lichtquelle abschließend um den Planeten kreisen. Verwenden Sie hierzu die Funktionalität eines `QTimer`, um die

Animation zu implementieren. Genauere Informationen zur Verwendung der `QTimer`-Klasse finden Sie in der Dokumentation von `Qt` (siehe Link auf unserer Praktikums-Website).

Der Planet soll texturiert werden. Verwenden Sie dafür die mitgelieferte Textur. Für das *Texture Mapping* benötigt die Kugel Texturkoordinaten. Erweitern Sie hierzu die Geometriedefinition entsprechend. Verwenden Sie auch hier keine vorgefertigten Funktionen zum Generieren der Texturkoordinaten.

Der Planet soll von einer Skybox mit Sternentextur umschlossen sein. Erstellen Sie hierzu einen texturierten Würfel, der stets relativ zur Kamera zentriert ist. Verwenden Sie auch hier die mitgelieferten Texturen.

Erweitern Sie das Qt-Hauptfenster durch einen Einstellungsdialog für folgende Programm-Parameter:

- Animate Sun (boolean)
- Smooth Shading (boolean)
- Number Broadleaf Trees (int)
- Number Fir Trees (int)
- Planet Radius (float)
- Tree Size (float)
- Tree Resolution (int)
- Planet Resolution (int)
- Timer Interval (int)

Verwenden Sie hierfür den Qt-Designer und die für vorgesehene `Settings.ui` Datei. Der angepasste Build-Schritt zum Aufruf des `uic`-Kompilers erfolgt analog zum `moc`-Kompileraufruf (siehe Eigenschaften der `MainWindow.h.qt`).

#### 4. Bewertung

Bei der Bewertung werden folgende Kriterien berücksichtigt:

1. Die Aufgaben sind gemäß der Aufgabenstellung gelöst.
2. Das gesamte Projekt unterliegt einer **objektorientierten** Denkweise, genügt den Grundlagen der **modularen** Gestaltung und ist übersichtlich editiert.
3. Der Quellcode ist in **englischer Sprache kommentiert** und enthält sämtliche Hilfen, die zum Verständnis der Programmstruktur bzw. der verwendeten Algorithmen notwendig sind.
4. Das Programm lässt sich ohne Fehlermeldungen und Warnungen kompilieren.