

# Übung zu Computergraphik I

## – Übungsblatt 7 –

### Lehrstuhl für Computergraphik und Multimediasysteme

Hendrik Hochstetter, Bianca Kretz, Rene Winchenbach

**Abgabe:** Für Studenten mit 5 LP verpflichtend bis spätestens 30. Mai 2014, 10 Uhr.

**Besprechung:** Mittwoch 4. Juni 2014 und Freitag 6. Juni 2014

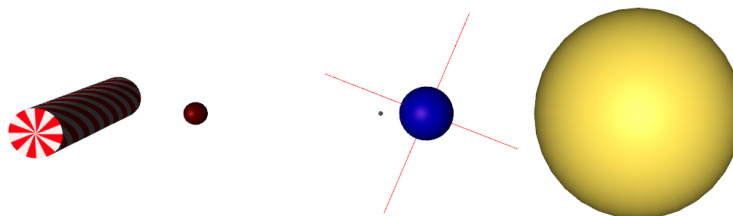
**Hinweise:** Schriftliche Übungen bitte zusammengeheftet in den Pappkarton vor Büro H-A 7115/1 werfen. Programmieraufgaben bitte per Mail mit Name und Matrikelnummer an Ihren jeweiligen Tutor senden. Geben Sie bei dabei nur Ihre modifizierte Quelltextdatei als Anhang ab.

#### Aufgabe 1 Szenengraphen und Transformationshierarchien (1 Punkt)

In der folgenden Aufgabe sollen Sie ein sehr einfaches Sonnensystem, bestehend aus vier Planeten, zunächst als Szenengraph modellieren und anschließend nachprogrammieren. Laden Sie hierzu das Programmgerüst `ueb07.zip` von der Übungsseite herunter.

Das Modellsonnensystem soll aus einer zentralen Sonne im Ursprung des Weltkoordinatensystems bestehen, um die ein erdähnlicher Planet und ein marsähnlicher Planet kreisen. Außerdem kreist um die Erde ein kleiner Mond. Alle Kreisbahnen sollen in derselben Ebene liegen. Die Erde soll sich um ihre eigene y-Achse drehen, wobei diese um  $23,44^\circ$  gegenüber der Ebene der Kreisbahnen geneigt ist. Die Sonne soll das größte Objekt des Systems darstellen, es folgen Erde, Mars und Mond. Zur Darstellung von Kugeln kann die Funktion `drawSphere` verwendet, die als Parameter den Kugelradius und einen RGB-Farbvektor (vom Typ `float[3]`) erwartet. Die Entsprechenden Farben der Planeten sind im Programmcode bereits global vordefiniert.

- 1.1 Beschreiben Sie das Sonnensystem durch einen gerichteten azyklischen Graphen (DAG).
- 1.2 Erweitern Sie das gegebene Programmgerüst, sodass ein Sonnensystem wie in Abb. 1 zu sehen ist. Verwenden Sie zur hierarchischen Beschreibung des Sonnensystems die OpenGL-Befehle `glPushMatrx ()` und `glPopMatrix ()`.

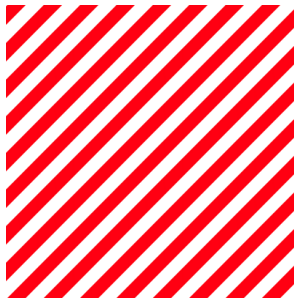


**Abbildung 1:** Modellsonnensystem aus Sonne, Erde, Mars, Mond und natürlich mit Candy-Stick für Aufgabe 2. Zur Veranschaulichung sind die lokale x- und y-Achse der Erde mit eingezeichnet.

**Aufgabe 2 Texturierung (1 Punkt)**

In der folgenden Aufgabe soll das Programmgerüst aus Aufgabe 1 um ein texturiertes Objekt erweitert werden. In Abbildung 1 ist am linken Rand ein Candy-Stick zu sehen. Die Geometrie des Candy-Sticks entspricht der eines Zylinders. Die rot-weißen Streifen sollen mittels der Textur aus Abbildung 2 realisiert werden. In der Methode `drawCylinder` wird bereits die Geometrie eines Zylinders erzeugt. Ebenso wird bereits eine Textur geladen und gebunden, sodass nur die Texturkoordinaten gesetzt werden müssen.

Geben Sie für jeden Vertex des Zylinders Texturkoordinaten an. Die Textur soll den Zylinder genau einmal komplett umwickeln. Es sollen keine sichtbaren Nahtstellen entstehen.



**Abbildung 2:** *Streifen-Textur zur Modellierung des Candy-Sticks aus Aufgabe 2.*

**Hinweis:** Stellen Sie beim Ausführen des Programms sicher, dass sich die Textur `stripes.png` in Ihrem Arbeitsverzeichnis befindet.

**Aufgabe 3 Bildspeicherung (1 Punkt)**

In der folgenden Aufgabe sollen Sie sich ein paar Gedanken zum Thema Bildspeicherung machen.

- 3.1 Beschreiben Sie die Unterschiede zwischen Vektor- und Pixelgraphiken.
- 3.2 Wie gut ist das JPEG-Verfahren geeignet, die Textur aus Abbildung 2 zu speichern? Wie würde sich das JPEG-Verfahren verhalten, wenn die Streifen vertikal verliefen und jeweils 8 Pixel breit wären? Begründen Sie Ihre Antworten.
- 3.3 Kann die Textur mit Hilfe einer Vektorgraphik-Beschreibung besser gespeichert werden? Begründen Sie Ihre Antwort.