

Name: Tilo Weber
Matrikelnummer: s794813
Mail: s48848@beuth-hochschule.de

2. Aufgabe – WebGL Kugel

Source-Dateien der Lösung:

WebGL-Seite mit Canvas

<https://github.com/TwUni2012/realtimerendering/blob/master/Aufgabe2/aufgabe2.html>

js-Klasse sphere.js

<https://github.com/TwUni2012/realtimerendering/blob/master/Aufgabe2/lib/js/sphere.js>

Erläuterung des Lösungsansatzes:

Ein Tetraeder ist ein Objekt, welches aus vier gleichgroßen Dreiecken besteht. Klappt man diesen 3d Tetraeder auf eine 2d-Ebene entsteht ein großes Dreieck aus vier Dreiecken.

Der Mittelpunkt des Tetraeders ist auch der Mittelpunkt der Kugel. Die Kugel soll einen Radius von 1 besitzen. Damit die Eckpunkte des Tetraeders auf der Kugel liegen müssen diese normalisiert werden (Länge = 1). Anschließend werden die einzelnen Seiten des Tetraeder einzeln betrachtet. Die Seitendreiecke werden in jeweils vier gleiche große Dreiecke zerlegt. Dazu werden in einem Dreieck mit den Punkten ABC die Mittelpunkte der Seiten AB, BC und AC ermittelt. Diese neuen Punkte AB, BC und AC bilden mit den Ausgangspunkten A, B und C die vier neuen Dreiecke A-AC-AB, B-AB-BC, C-AC-BC und AB-BC-AC. Da jedoch die Dreieckspunkte AB, BC und AC noch nicht auf der Kugeloberfläche liegen, müssen diese entsprechend noch normalisiert werden. Aus den vier neu gewonnenen Dreiecken kann nun wie vorher beschrieben eine weitere Annäherung an die Kugel erfolgen.

Verwendete Ressourcen/Quellen:

<http://learningwebgl.com/blog/?p=370> – WebGL Lesson 4 – some real 3D objects

<http://www.florian-oeser.de/rtr/ue2/> --> sphere.js

.../lib/js/glMatrix-0.9.5.min.js

.../lib/js/webgl-utils.js

<http://de.wikipedia.org/wiki/Tetraeder>

Bedienungsanleitung:

Drahtgitter an/aus: w

Rekursionstiefe einstellen: +/- (von 1-10)

Kugelachse für Drehung ändern:

x-Achse: x,c

y-Achse: Pfeil hoch, Pfeil runter

z-Achse: Pfeil links, Pfeil rechts