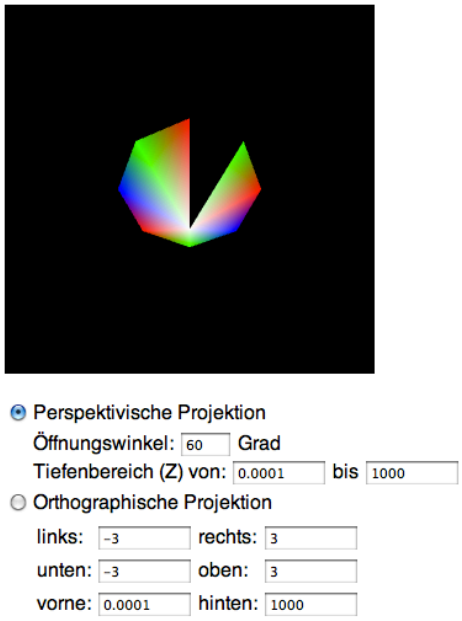
Prof. Dr.-Ing. Hartmut Schirmacher
Fachbereich Informatik und Medien

Computergrafik 2 / Aufgabe 4:
WebGL Geometrie und Transformation

Laden Sie das Material zur Aufgabe 4 von der Vorlesungs-Webseite herunter und studieren Sie die Struktur und Funktionsweise der bereitgestellten WebGL-Applikation. Einige der verwendeten Objekte und Konzepte wurden bereits in der Vorlesung vorgestellt.



Installieren Sie einen WebGL-fähigen Browser auf Ihrem System und testen Sie, ob beim Betrachten von index.html eine Seite vergleichbar mit der hier dargestellten erscheint.

Testen Sie, ob Maus-Dragging die Rotation des Objekts im Viewer bewirkt.

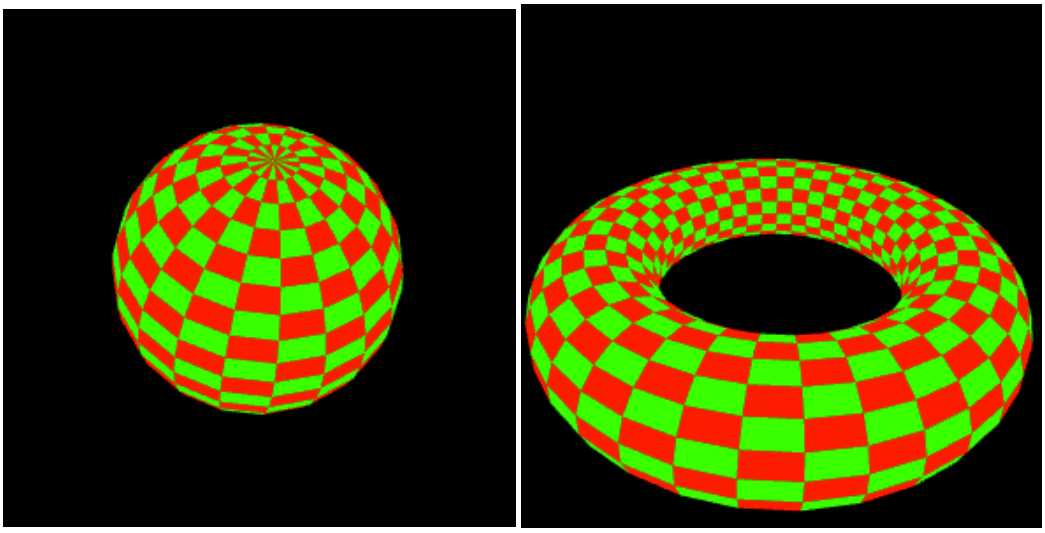
Sollten Ihre Tests fehlschlagen, versuchen Sie es bitte einmal mit den Browser *Chrome* von Google, gegen den das Beispiel getestet wurde.

Aufgabe 4.1: Würfel

Erweitern Sie die Applikation in shapes.js um ein geometrisches Objekt „Cube“ analog zu dem vorhandenen Objekt „Triangle“ bzw. „TriangleFan“.

- Der Würfel soll um den Ursprung zentriert sein und in allen Dimensionen die Kantenlänge L haben, die als Konstruktor-Parameter übergeben wird.
- Definieren Sie im Konstruktor des Cube-Objekts die notwendigen Vertices und die enstsprechenden TRIANGLES-Primitive. Dabei ist es erlaubt, den gleichen Eckpunkt mehrfach zu definieren (z.B. für jede Seite des Würfels neu).
- Weisen Sie verschiedenen Ecken des Quaders verschiedene Farbwerte zu.

Erzeugen Sie eine Szene mit einem Würfel und demonstrieren und dokumentieren Sie das Ergebnis.



3D-Geometrie im Viewer mittels WebGL: Kugel und Torus

Aufgabe 4.2: Kugel oder Torus

Erweitern Sie die Applikation in shapes.js um ein weiteres geometrisches Objekt, wahlweise eine Kugel oder einen Torus.

- Das Objekt soll um den Ursprung zentriert sein
- Der Radius der Kugel bzw. die beiden Radii des Torus sollen als Konstruktor-Parameter spezifiziert werden
- Um die Vertices für das Objekt zu erzeugen, unterteilen Sie die Oberfläche der Kugel / des Thorus in NxM Abschnitte, welche aus je zwei Dreiecken bestehen. N und M sollen im Konstruktor angegeben werden.
- Geben Sie dem Konstruktor zwei Farbwerte mit, und weisen Sie den Vertices die Farben so zu, dass die Farbe für die Unterteilungs-Abschnitte des Objekts wie bei einem Schachbrett alterniert (siehe Abbildung auf dieser Seite).

Definieren Sie einige Test-Szenen, in denen Sie eine Kugel bzw. einen Torus mit unterschiedlicher Unterteilungs-Auflösung platzieren. Testen und dokumentieren Sie diese Ergebnisse geeignet.

Für eine sehr gute Note implementieren Sie sowohl Kugel als auch Torus und erweitern Sie die HTML-Seite der Anwendung um ein Formular, welches die Auswahl eines geometrischen Objekts (Dreieck, Würfel, Kugel) sowie die Angabe der Tesselierungs-Parameter N und M zulässt . Gehen Sie dabei analog zu dem bereits vorhandenen Formular „cameraParameters“ vor.

Hilfestellung zu Kugel und Torus (siehe wikipedia.de):

- die parametrische Form der Kugelgleichung mit Radius r ist
  - x = r*sin(u)*cos(v);
  - y = r*sin(u)sin(v);
  - z = r*cos(u)
  - für 0 <= u < Pi und 0 <=v <= 2*Pi

- die parametrische Form des Torus mit den Radien R und r ist:
  - x = (R + r*cos(u)) * cos(v)
  - y = (R + r*cos(u)) * sin(v)
  - z = r*sin(u)
  - für 0 <= u,v <= 2*Pi

Aufgabe 4.1 – Transformationen

Fügen Sie den geometrischen Objekten eine Modelltransformation hinzu, so dass Sie auf einfache Weise mehrere Objekte in einer Szene platzieren können.

- Fügen Sie dem Objekt *VertexBasedShape* eine Transformationsmatrix vom Typ mat4 hinzu (siehe z.B. *Scene.transform* und *glmatrix.js*)
- Fügen Sie dem Objekt *VertexBasedShape* eine Methode *setUniforms* hinzu. Diese Methode setzt in den Shadern die Uniform-Variable *nodeTransform* auf die Werte der Transformationsmatrix des geometrischen Objekts (vgl. *Scene.draw()*)
- Ändern Sie *Scene.draw()* so, dass vor dem Malen eines jeden Objekts dessen *setUniforms*-Methode aufgerufen wird
- Ändern Sie die Shader so, dass die zusätzliche Matrix *nodeTransform* vor der eigentlichen *modelViewMatrix* angewendet wird.

Definieren Sie eine geeignete Testszene mit mehreren Objekten, die an verschiedenen Orten und verschieden groß sind. Ändern Sie ggf. die initialen Werte der Kamera, damit die Szene beim Aufruf der Webseite gut dargestellt wird.

Abgabe

Die Abgabe erfolgt bis spätestens zum **19.12.2011** für die Montagsübungen und bis zum **21.12.2011** für die Mittwochsübungen. Bitte demonstrieren und erläutern Sie die Lösung spätestens am Abgabetag in der Übung und schicken Sie die Quellen am gleichen Tag spätestens um 24 Uhr an `hschirmacher@beuth-hochschule.de`. Erstellen Sie für die Abgabe ein Archiv des gesamten lauffähigen Projekts, in dem nur die Quellen und generierten Bilder enthalten sind.

Wenn Sie die o.a. Zeiten nicht einhalten, gilt die Übung als verspätet abgegeben; dies wird mit einem Abschlag auf die Note belegt. Bitte beachten Sie die Hinweise auf dem Handout der ersten Vorlesung.