# Computer Graphics – WebGL, Teil 5

## 1 Solid Cube

Als nächstes möchten wir einen ausgefüllten, farbigen Würfel zeichnen. Den Vertex Buffer und den Element Buffer vom Drahtgittermodell könnten wir zwar dazu verwenden, aber dann könnten wir Farben nur an den einzelnen Ecken angeben, was zu einem komischen Farbverlauf führen würde (Abb. 1).

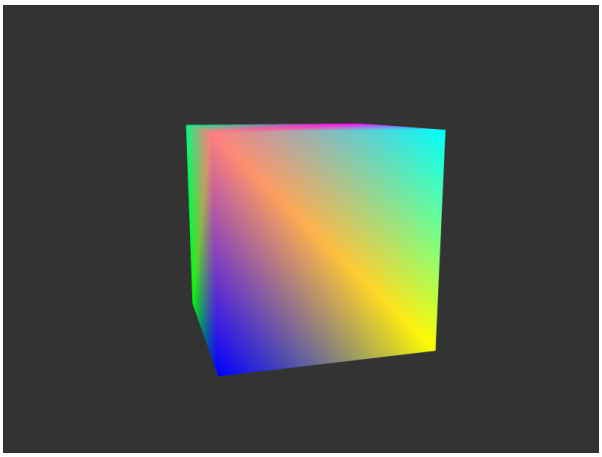

Abbildung 1: Würfel mit gemeinsamen, farbigen Eckpunkten

.

Wenn wir Flächen in einer Farbe zeichnen möchten, müssen wir deshalb jeden Vertex dreimal angeben, einmal für jede Fläche, in der er vorkommt. So können wir dann pro Vertex und Fläche eine Farbe zuordnen (Abb. 2). Dies werden wir später auch weiter so benutzen, um Normalenvektoren zu den Seiten anzugeben, was wir dann für die Beleuchtung brauchen. Generell werden separate Vertices dann eingefügt, wenn eine Kante modelliert werden soll. Ist an dem Vertex keine Kante vorhanden, zum Beispiel bei einem eigentlich runden Objekt, gibt es den Vertex nur einmal.

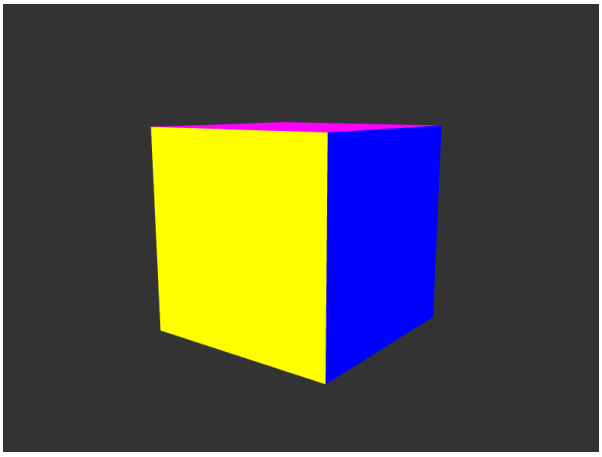

Abbildung 2: Würfel mit farbigen Seiten.

.

## 2 Backface Culling und z-Buffer

WebGL stellt zwei Methoden zur Verfügung, um zu verhindern, dass nicht sichtbare Flächen gezeichnet werden: Backface Culling und z-Buffer.

Beim Backface Culling werden die hinten liegenden und damit verdeckten Flächen nicht gezeichnet. Dazu muss WebGL jedoch wissen, welche Flächen hinten liegen. Dies geschieht mit der Reihenfolge, in der die Eckpunkte (Vertices) der Dreiecke spezifiziert sind. Die Konvention ist, dass die Ecken von aussen gesehen im Gegenuhrzeigersinn angegeben werden. WebGL kann dann aus den 3 Eckpunkten einen Normalenvektor berechnen und anhand dessen entscheiden, ob die Fläche gezeichnet werden soll oder nicht. Backface Culling muss entsprechend eingeschaltet werden mittels `gl.enable`, die weiteren unten aufgeführten Befehle entsprechen der Default-Konfiguration von WebGL.

```
gl.frontFace (gl.CCW);        // defines how the front face is drawn
gl.cullFace (gl.BACK);        // defines which face should be culled
gl.enable (gl.CULL_FACE);     // enables culling
```

### Aufgabe 1

Zeichnen sie einen ausgefüllten Würfel mit verschiedenfarbigen Seiten. Der Code für den Würfel soll in einem eigenen JavaScript-File analog der Lösung zum Drahtgitter Würfel abgelegt werden.

Backface Culling allein funktioniert nicht mehr, wenn sich mehrere Objekte überlagern. WebGL verwendet deshalb zusätzlich den Z-Buffer-Algorithmus, um herauszufinden, welche Pixel eines Objekts von Pixeln eines anderen Objekts verdeckt werden. Dazu wird die Tiefe des Pixels zusätzlich zu seiner Farbe gespeichert und dann beim Zeichnen ein Tiefenvergleich durchgeführt. Um diese Funktion zu verwenden, muss der Vergleich mittels `gl.enable(gl.DEPTH_TEST)` aktiviert werden und vor dem Zeichnen der Tiefenbuffer mit `gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT)` gelöscht werden.

```
// in init
gl.enable (gl.DEPTH_TEST);
...
// in draw
gl.clear (gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT)
```

# 3 Texture Mapping für 3D Objekte

## Aufgabe 2

Implementieren sie Texture Mapping für die Seiten des Würfels wie in Abb. 3 dargestellt. Dabei müssen, wie im 2D-Beispiel zu Texture Mapping, an jedem Vertex die Textur-Koordinaten angegeben werden.

Um Texture Mapping im Fragment Shader ein- und ausschalten zu können, empfiehlt es sich, eine uniform-Variable einzuführen, die dann im Javascript Programm gesetzt und im Shader abgefragt werden kann.

```
precision mediump float;
varying vec4 vColor;
varying vec2 vTextureCoord;

uniform sampler2D uSampler;
uniform int uEnableTexture;

void main ()
{
    if (uEnableTexture == 0) {
        gl_FragColor = vColor;
    }
    else {
       gl_FragColor = texture2D (uSampler, vTextureCoord);
    }
}
```

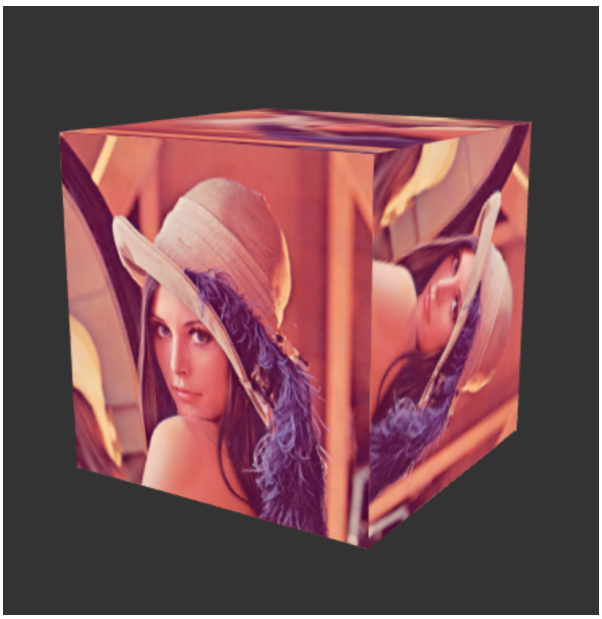

Abbildung 3: Würfel mit Texture Mapping

.

### Aufgabe 3

Stellen sie zwei um ihre eigene Achse rotierende Würfel, einen farbigen Würfel sowie einen Würfel mit Texture Mapping, dar.

### Aufgabe 4* (optional)

Stellen Sie im gleichen Fenster vier verschiedene Ansichten des Würfels dar. Benutzen Sie dazu `gl.viewport`.

### Aufgabe 5* (optional)

Definieren und zeichnen Sie eine Kugel.