Ludwig-Maximilians-Universität München
LFE Medieninformatik
Prof. Dr. Andreas Butz
Alina Hang, Dominikus Baur

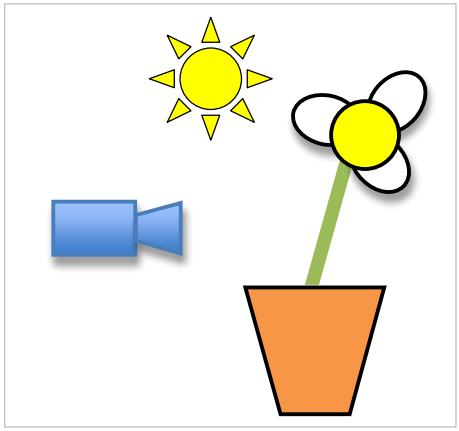
Übungsblatt 10: Klausurvorbereitung

Abgabe:
Dieses spezielle Übungsblatt hat keine Abgabefrist und wird auch nicht korrigiert. Die Lösung gibt es wie immer auf der Homepage der Vorlesung oder in den Übungen vom 25. bis 27.7.

Inhalt:
Dieses Übungsblatt enthält drei Aufgaben die in dieser Form auch in der Klausur vorkommen könnten. Dabei bezieht sich diese Klausurähnlichkeit allerdings nur auf die Komplexität und den Typ der Aufgabe – die Themen werden nicht unbedingt in der Klausur behandelt. Die Klausur wird ca. 5 – 10 Aufgaben dieses Kalibers enthalten. Der Stoff der Klausur umfasst die gesamte Vorlesung und Übung.

Aufgabe 1: Kamerakontrolle und Projektionen

In interaktiven 3D-Umgebungen wird die Kontrolle der Kamera oft dem Benutzer überlassen, manchmal aber auch automatisch gesteuert.



a) In der nebenstehenden (skizzierten) Szene wird die Transformation der Kamera über folgende WebGL-Befehle bestimmt (die Variable *tick* zählt dabei von 0 bis 1 und ihr Wert wird bei jedem Durchlauf um 0.1 erhöht):

```
mat4.identity(mvMatrix);
var c0 = [0, 0, -10];
var c1 = [20, 0, -5];
var c2 = [0, 0, 10];
var cpos = new Array();
cpos.push((1 - t) * (1 - t) * c0[0] + 2 * t * (1 - t) * c1[0]
+ t * t * c2[0]);
cpos.push((1 - t) * (1 - t) * c0[1] + 2 * t * (1 - t) * c1[1]
+ t * t * c2[1]);
cpos.push((1 - t) * (1 - t) * c0[2] + 2 * t * (1 - t) * c1[2]
+ t * t * c2[2]);
mat4.lookAt(cpos, [0, 0, 0], [0, 1, 0], mvMatrix);
```

Zeichnen Sie den Pfad und die Blickrichtung der Kamera in die nachstehende Skizze ein (die initiale Position und Blickrichtung findet sich bereits in der Skizze). Berechnen Sie mindestens die Werte t = 0, t = 0,1, t = 0,5, t = 0,7 und t = 1:

Wie nennt man die Kategorie von Kurven die die Kamerabahn beschreibt?

b) Erstellen Sie einen Szenengraph für die obenstehende Szene. Der Szenengraph soll dabei jedes sichtbare Primitiv, Beleuchtung und die Kamera enthalten.

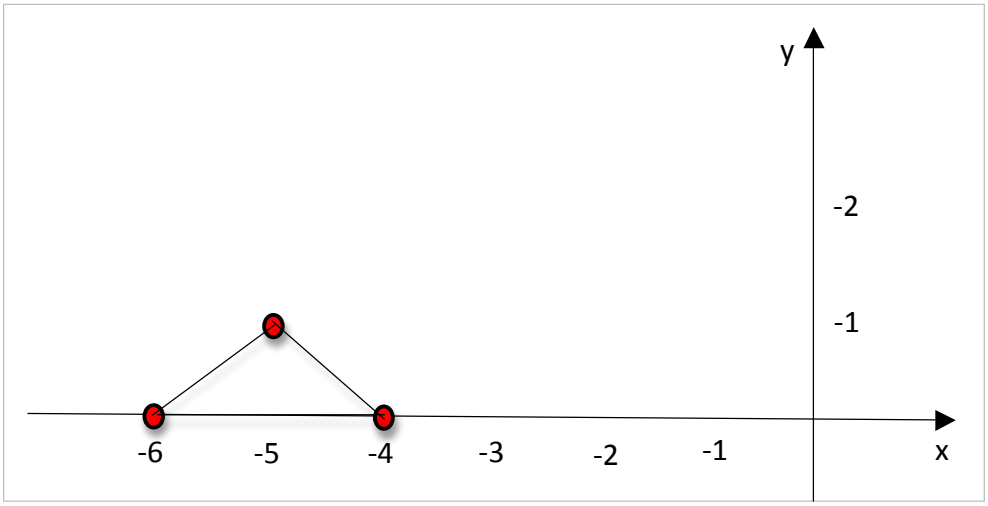
c) Welche Unterschiede im sich ergebenden Bild entstehen wenn die Projektion von perspektivisch auf orthographisch umgestellt wird?

<u>Aufgabe 2:</u> Rotationsobjekte und Voxel

In der Vorlesung haben Sie verschiedene Arten von algorithmisch generierten Polygonmeshes kennengelernt. Eine Form dieser Meshes sind Rotationsobjekte. Diese basieren auf einem zweidimensionalen Grundkörper der entlang eines Kreises geführt wird. In dieser Aufgabe gehen wir von einem Dreieck aus, dessen Eckpunkte sich anfänglich bei (-6, 0, 0), (-5, 1, 0) und (-4, 0, 0) befinden. Dieses soll auf einem Kreis mit dem Radius 5 um den Ursprung herum gedreht werden.

a) Schreiben Sie einen Algorithmus in Javascript oder Pseudocode der ausgehend von dem Grundkörper ein Rotationsobjekt erstellt (das entstehende Objekt soll das Dreieck an mindestens vier Stellen entlang des Rings zeichnen). Generieren Sie dabei einen Vertex- und einen Indexbuffer für die entstehenden Dreiecke. Achten Sie darauf, dass der Ring am Ende abschließt.

b) Die Umgebung wird jetzt auf ein Voxelsystem umgestellt. Skizzieren Sie (Sie müssen keinen Algorithmus verwenden) die untenstehende Scheibe des Rotationsobjekts in einer 2D-Voxelumsetzung. Die Voxel haben dabei eine Kantenlänge von 0.5 Einheiten.



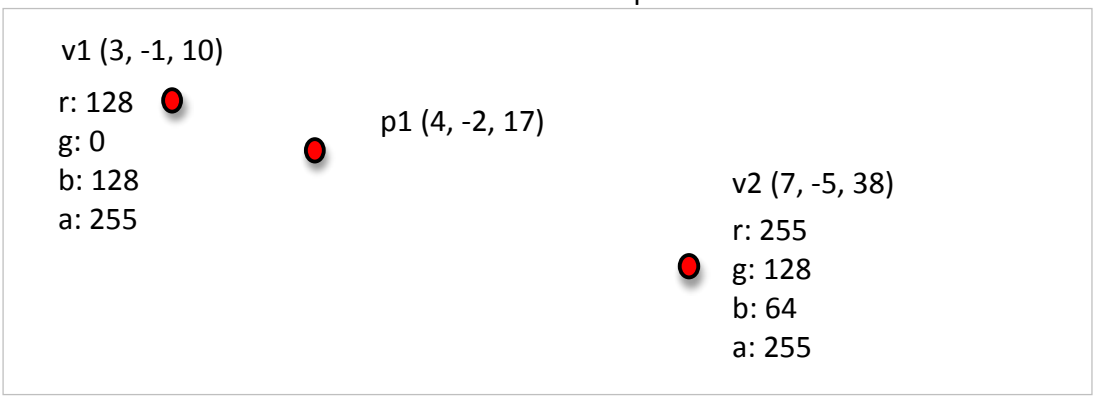
c) Welchen Typ von Verfahren würde man bei der Umwandlung von Vertices in Voxel anwenden? Nennen Sie einen beispielhaften Algorithmus der in der Vorlesung behandelt wurde.

Aufgabe 3: Shading

Shader erlauben fortgeschrittene Arten der Vertex- und Pixelmanipulation in OpenGL und WebGL.

a) Skizzieren Sie in welcher Reihenfolge Vertex- und Fragmentshader angewendet werden und durch welche besonderen Variablen sich Werte zwischen beiden übergeben lassen. Wie heißen diese Variablen?

b) In dieser simplen Szene werden die Farben eines Pixels durch die Farben der jeweiligen Vertices erstellt. Wie heißt das interne Verfahren das abläuft um die Werte zwischen zwei Vertices zu berechnen? Welche Farbe hat Punkt p1?



c) In dem nachfolgenden Codegerüst wird eine Textur an den Fragmentshader übergeben. Füllen Sie die main-Methode des Shaders aus, so dass das Polygon mit der invertierten Textur texturiert wird
Anmerkung: Die Variable *vTextureCoord* enthält dabei die Texturkoordinaten und *uSampler* die Textur.

```
#ifdef GL_ES
precision highp float;
#endif

varying vec2 vTextureCoord;
uniform sampler2D uSampler;

void main(void) {
```