Klausur Computergraphik (WS 2015/16)

Prüfer:	Prof. Dr. R. Dörner, Prof. Dr. C. Schulz, HS RheinMain
Bearbeitungszeit:	90 min
Zugelassene Hilfsmittel:	ein beidseitig handbeschriebenes DIN A4 Blatt, Stifte. (insbesondere Taschenrechner und eigenes Papier ist verboten)
Datum:	15. Februar 2016

Name:______________________________ Vorname: _________________________

Matr.-Nr. ____________________

____
Unterschrift

Hinweise:

- Überprüfen Sie Ihr Klausurexemplar auf Vollständigkeit (Umfang: 8 Blätter)
- Lösen die Aufgaben im dafür vorgesehenen Raum. Wenn der Platz nicht ausreicht, verwenden Sie die Rückseiten - wenn alle Rückseiten beschrieben sind, fordern Sie ein leeres Blatt bei der Aufsicht an. Schreiben Sie im vorgesehenen Raum einen Hinweis der Art "weiter siehe S. 3 Rückseite". Fehlt dieser Hinweis, ist die Lösung unleserlich oder gibt es mehrere Lösungen zu derselben Aufgabe, so werden keine Punkte vergeben.
- Wer einen Täuschungsversuch begeht oder einem Täuschungsversuch Vorschub leistet erhält die Note "nicht bestanden".
- Es darf nicht mit Bleistift geschrieben werden. Es sind nur Schreibfarben „blau“ oder „schwarz“ zulässig.
- Die Klausur ist in jedem Fall bestanden mit **40 Punkten.**

Es wurden __________ Punkte erreicht.

Note, Handzeichen:

Aufgabe 1

Gegeben sind die Punkte P_{13} bis P_{35} mit $P_n\,(2\cdot n\,/\,-2\cdot n\,/\,n)$ für n = 13, 14, ..., 35, $n \in IN$. Diese sollen Stützpunkte einer uniformen B-Spline Kurve Q sein.

(a) Der Knotenvektor lautet: T = [5, 8, …, i]. Welchen Wert hat i?

2 P.

(b) Wie viele Kurvensegmente von Q müssen alle neu gezeichnet werden, wenn man die Position von Punkt P_{34} ändert?

Wie viele Kurvensegmente von Q müssen alle neu gezeichnet werden, wenn man die Position von Punkt P_{21} ändert?

3 P.

(c) Beschreiben Sie die Form der konvexen Hülle des zehnten Kurvensegments von Q möglichst genau? Was folgt daraus für dieses Segment?

3 P.

(d) Geben Sie eine Formel an (nicht ausmultiplizierte Matrizen, keine Variablen), wie man Q(22,5) berechnet. Die Basismatrix der uniformen B-Spline-Kurven lautet dabei:

$$M_{UBS} = \frac{1}{6} \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}$$

5 P.

Σ2:

Aufgabe 2

Gegeben ist folgende VRML-Szene:

```
DEF T1 Transform{
        scaleOrientation        0 1 0 1,57
        scale                   10 1 1
        children[
              DEF T2 Transform{
                        rotation        0 1 0 1,57
                        center          1 1 1
                        children[
                                        DEF S1 Shape{ geometry Sphere{ } }
                        ] # children T2
              } # T2
              DEF S2 Shape{ geometry Sphere{ } }
              DEF T3 Transform{
                                        translation   -5 0 0
                                        center         2 2 2
                                        children[
                                                        DEF S3 Shape{ geometry Sphere{ } }
                                        ] # children T3
              } # T3
        ] # children T1
} # T1
```

(a) Zeichnen Sie den Szenengraph (nur Shape- und Transform-Nodes, keine Fields)

5 P.

(b) Geben Sie eine Formel bestehend aus (nicht ausmultiplizierten) 4x4 Matrizen an, wie man die Koordinaten (x, y, z) eines Punktes aus dem lokalen Koordinatensystem von Kugel S3 umrechnet in lokale Koordinaten (x', y', z') im Koordinatensystem von Kugel S1.

Σ3:

6 P.

(c) Wie lauten die Weltkoordinaten des P, der im Koordinatensystem von Kugel S2 die Koordinaten (1 / 2 / 3) hat?

6 P.

Σ4:

Aufgabe 3

Gegeben ist folgender Ausschnitt aus einem WebGL-Javascript, wobei die in der Lehrveranstaltung vorgestellten Hilfsfunktionen verwendet werden:
mat4() *„erzeugt eine 4x4 Einheitsmatrix"*,
mult(m1, m2) *„berechnet das Matrixprodukt der Matrizen m1 und m2"*,
transpose(m1) *„transponiert die Matrix m1"*,
inverse(m1) *„invertiert die Matrix m1"*,
rotate(alpha, [x,y,z]) *„erzeugt eine 4x4 Rotationsmatrix um die Achse* $(x,y,z)^T$ *um den Winkel alpha"*,
translate(x,y,z) *„erzeugt eine 4x4 Translationsmatrix für den Translationsvektor* $(x,y,z)^T$*"*,
projection(fov, aspect, near, far) *„erzeugt eine Projektionsmatrix"*

```
// Projektionsmatrix
var projection = perspective(60.0, 1.0, 0.01, 100.0);

// Zeile A: hier die Model-Matrix mA anlegen

var mA =




// Zeile B: hier die View-Matrix mB anlegen

var mB =




// Zeile C: hier Matrix mC anlegen, die Objektkoordinaten in Clipping-Koordinaten umrechnet

var mC =


// Zeile D: hier Matrix mD anlegen, die Normalen von Objektkoordinaten in Kamerakoordinaten
// umrechnet

var mD =
```

3 P. (a) Ergänzen Sie das Programm nach Zeile A so, dass alle Modelle zuerst um die y-Achse um 30° um den Ursprung gedreht werden und dann um -45° um die z-Achse um den Punkt P(1,2,3).

4 P. (b) Ergänzen Sie das Programm nach Zeile B so, dass entsprechend lookAt(0,4,1,1,4,0,0,-1,0) die Kamera positioniert wird (verwenden Sie dabei nur die oben angegebenen Hilfsfunktionen)

2 P. (c) Ergänzen Sie das Programm nach Zeile C so, dass eine Matrix mC angelegt wird, die Vertices von Objektkoordinaten in Clipping-Koordinaten umrechnet

2 P. (d) Ergänzen Sie das Programm nach Zeile D so, dass eine Matrix mD angelegt wird, die Normalen von Objektkoordinaten in Kamerakoordinaten umrechnet

Σ5:

(e) Wie ändert sich das Bild, wenn perspective(60.0, 1.0, 0.01, 100.0) abgeändert wird in: perspective(120.0, 2.0, 2.0, 4.0); ?

3 P.

(f) Ergänzen Sie den unten stehenden GLSL Vertex-Shader und Fragement-Shader möglichst einfach, um ein dem Gouraud-Shading möglichst entsprechendes Shading zu realisieren.

```
void main(){  // Vertex-Shader
            uniform mat4 matrix;     //  Matrix zur Umrechnung von Objekt- in Clippingkoordinaten
            attribute vec4 color;     // die dem Vertex zugeordnete Farbe
            attribute vec4 position;  //  die dem Vertex zugeordnete Position



}

void main() {  // Fragment-Shader



}
```

4 P.

(g) Beschreiben Sie, wie man die Shader bzw. das WebGL-Javascript ändern müsste, um Flat-Shading zu realisieren, wenn alle 3D Modelle nur aus Dreiecken bestehen.

3 P.

Σ6:

Aufgabe 4

Der Punkt P(-2, -1, 3) soll mit einer Kamera, die sich an Punkt A(0,-3,0) befindet, auf die Projektionsebene mit der Gleichung x = 1 perspektivisch projiziert werden. Die Bildkoordinaten P' von P sind mit der aus der Vorlesung bekannten Matrix $M_{per}(d)$ zu berechnen.

(a) Um M_{per} anwenden zu können, muss eine Standardsituation eingehalten werden:
Wo muss sich die Kamera befinden?

Wohin muss die Kamera schauen?

Wo muss sich die Projektionsebene befinden?

3 P.

(b) Wie kann man die Standardsituation für $M_{per}(d)$ erreichen?

3 P.

(c) Berechnen Sie die Bildkoordinaten von Punkt P.

3 P.

(d) Geben Sie die Ebenengleichung der verbotenen Ebene an.

2 P.

Aufgabe 5

(a) Nennen Sie zwei Vorteile von Parallelprojektion gegenüber der perspektivischen Projektion

1. ____

2 P. 2. ____

Σ7:

(b) Begründen Sie, warum in OpenGL das Clipping besonders einfach durchgeführt werden kann.

2 P.

(c) Welche Form hat das Sichtvolumen in WebGL

(i) bei perspektivischer Projektion: ____

2 P. (ii) bei Parallelprojektion: ____

(d) Was bezeichnet man in der Computergraphik mit „ambienten Licht“?

3 P.

(e) Ein Viereck mit der RGBA-Farbe (0.1, 0.2, 0.3, 0.4) ist vor einem roten Hintergrund (RGB-Farbe: (1, 0, 0)) zu sehen. Unter welcher RGB-Farbe erscheint es?

3 P.

(f) Gegeben ist folgender Ausschnitt eines GLSL – Shaders:

```
vec4 v = vec4(1.0, 2.0, 3.0, 4.0);
vec4 u = vec4(5.0, 6.0, 7.0, 8.0);
v = u.abba;
v.q = u.s;
```

3 P. Welchen Wert hat v nach Ausführung der letzten Zeile? v = (_____, ______, ______, _____)

(g) Nennen Sie zwei Unterschiede zwischen Clipping-Koordinaten und Bildkoordinaten

1. ____

____

2. ____

3 P. ____

Σ8: