Klausur Computergraphik (SS 2017)

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:	17.Juli 2017

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 **41 Punkten.**

Es wurden _____ Punkte erreicht.

Note, Handzeichen:

Aufgabe 1

Gegeben ist ein uniformer B-Spline Q(t) mit den Stützpunkten A(1,2), B(-2,1), C(0,1), D(-1,2), E(0,0) und F(-2,-1).

(a) Vervollständigen Sie den Knotenvektor

2 P. T = [4, 8, _____]

(b) Geben Sie eine Formel für das Kurvensegment an, das Q(13) enthält.
Hinweis: $M_{UBS} = \frac{1}{6} \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}$

5 P.

(c) Berechnen Sie die Tangente im Punkt Q(13)

5 P.

(d) Ein Auto hat in seinen Objektkoordinaten die Fahrrichtung in x-Richtung. Das Auto soll entlang der Kurve Q(t) fahren. Geben Sie eine Formel an, um den Winkel zu berechnen, mit dem das Auto am Punkt Q(13) gedreht werden muss, damit es am von Q(t) beschriebenen Pfad ausgerichtet ist.

2 P.

Σ2:

Aufgabe 2

Gegeben ist folgende VRML-Szene:

```
DEF T1 Transform{
  children [
    DEF S1 Shape{ geometry IndexedFaceSet{ coord Coordinate {
                                                    point [  0 0 0,
                                                             1 1 4,
                                                             0 2 0,
                                                             2 2 0,
                                                             2 0 0 ] } # Coordinate

                                          coordIndex [1 4 3 -1 1 2 0 -1  4 3 2 0 -1 0 1 4 -1 _____ ]
                                } # IndexedFaceSet
    } # S1
    DEF T2 Transform{




      children [


    ] } # T2
] } # T1
```

3 P. (a) Das IndexFaceSet soll eine Pyramide P mit quadratischer Grundfläche beschreiben. Korrigieren Sie das Feld coordIndex und ergänzen Sie ggf. fehlende Flächen.

(b) Berechnen Sie die Normale, die bzgl. der Pyramide nach außen zeigt, auf dem Dreieck, das im Feld coordIndex als erstes beschrieben wird.

3 P.

(c) Eine Pyramide P‘ soll unter Verwendung des DEF-USE-Mechanismus in die Szene eingefügt werden. Sie hat die gleiche Grundfläche wie P, die Spitze liegt bei P‘ aber in Punkt S(1, 1, -4). Ergänzen Sie dazu den Transformnode T2. Verwenden Sie nur diesen einen Transformnode.

4 P.

Σ3:

Aufgabe 3

Gegeben ist folgender Ausschnitt aus einem WebGL-Javascript, wobei die in der Lehrveranstaltung vorgestellten Hilfsfunktionen verwendet werden:
mat4() *„erzeugt eine 4x4 Einheitsmatrix“*,
transpose(M) *„transponiert die Matrix M“*,
inverse(M) *„invertiert die Matrix M“*
mult(m1, m2) *„berechnet das Matrixprodukt der Matrizen m1 und m2“*
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 $(x,y,z)^T$“*,

```
// Model-Matrix berechnen
var model = mat4();
model = mult(model, rotate(90.0, 0.0, 1.0, 0.0));
model = inverse(model);
model = mult(translate(2.0, 4.0, 6.0), model);
model = mult(rotate( -90.0, 1.0, 0.0, 0.0), model);

// View-Matrix berechnen
var view = mat4();
view = mult( view, translate (5.0, 0.0, 0.0) );
view = mult(view, rotate (180.0, 0.0, 0.0, 1.0 ));
view = mult(rotate(90.0, 1.0, 0.0, 0.0), view);
```

(a) Der Punkt P hat die Objektkoordinaten P(1,2,3). Wie lauten seine Weltkoordinaten? Geben Sie eine Formel aus nicht ausmultiplizierten Matrizen und Vektoren an.

4,5 P.

(b) Wie lauten die Kamerakoordinaten von Punkt P? Geben Sie eine Formel aus nicht ausmultiplizierten Matrizen und Vektoren an.

4,5 P.

Σ4:

(c) Geben Sie möglichst wenige VRML-Transformnodes an, welche die gleiche Transformation in Weltkoordinaten beschreiben wie die obige Matrix model.

3 P.

(d) Wie ändern sich die Weltkoordinaten von P, wenn im obigen Programm in der dritten Zeile rotate(90.0, 0.0, 1.0, 0.0) durch rotate(90.0, 0.0, 7.0, 0.0) ersetzt (Begründung angeben)?

2 P.

(e) Welcher lookAt-Befehl erzeugt die gleiche View-Matrix wie die Matrix view im obigen Programm?

4 P.

Σ5:

Aufgabe 4

Gegeben ist folgender Vertex-Shader in GLSL:

```
void main(){  // Vertex-Shader
    uniform mat4 view;              // View-Matrix
    uniform mat4 model;             // Model-Matrix
    uniform mat4 projection;        // Projektionsmatrix
    attribute vec4 position;        // die dem Vertex zugeordnete Position in Objektkoordinaten
}
```

(a) Der Vertex-Shader oben ist unvollständig, weil eine unbedingt notwendige Programmzeile fehlt. Wie lautet diese Zeile?

2 P.

(b) Schreiben Sie eine Zeile GLSL-Code, welche die 3D Koordinaten von position in Kamerakoordinaten berechnet und das Ergebnis einer Variablen p zuordnet:

vec3 p =

2 P.

(c) Erläutern Sie an obigen Beispiel, warum einige Variablen mit dem Schlüsselwort uniform deklariert wurden, während eine Variable das Schlüsselwort attribute hat.

2 P.

(d) Wie kann man den in Aufgabe 4(b) berechneten 3D-Vektor an den Fragement-Shader übergeben?

3 P.

(e) In einem Fragment-Shader wurde eine RGB-Farbe berechnet und in der Variablen vec3 color gespeichert. Um einen Nebel-Effekt zu erzielen, soll die Resultatsfarbe des Shaders eine Mischfarbe von 80% der Farbe color und 20% der Farbe Rot sein. Wie lautet der Shader-Code?

3 P.

Σ6:

Aufgabe 5

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 y = 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.

5 P.

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

2 P.

Aufgabe 6

(a) Was versteht man in der Computergraphik unter einem „Fragment"?

3 P.

Σ7:

(b) Geben Sie ein Beispiel für eine Farbe im RGB-Farbsystem, deren S-Wert im HLS-Farbsystem den Wert 0 hat und den L-Wert 0,7.

2 P.

(c) Warum filtert man in der Computergraphik Bilder häufig mit einem Tiefpass-Filter (und nicht mit einem Hochpass-Filter)?

3 P.

(d) Beschreiben Sie einen Fall, in dem der Painter-Algorithmus für die Verdeckungsrechnung nicht funktioniert? Was kann man in einem solchen Fall tun?

3 P.

(e) Erläutern Sie die Funktionsweise von Bump Mapping. Wozu dient Bump Mapping?

3 P.

Σ8: