Klausur Computergraphik (SS 2019)

Prüfer:	Prof. Dr. R. Dörner, HS RheinMain
Bearbeitungszeit:	90 min
Zugelassene Hilfsmittel:	ein beidseitig handbeschriebenes DIN A4 Blatt, Stifte. (insbesondere Taschenrechner und eigenes Papier ist verboten)
Datum:	15. Juli 2019

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.
- Starten Sie mit der Bearbeitung der Klausur nur, wenn Sie prüfungsfähig sind.
- Die Klausur ist in jedem Fall bestanden mit **38 Punkten.**

Es wurden _____ Punkte erreicht.

Note, Handzeichen:

Aufgabe 1

Gegeben ist eine Bézierkurve Q(t) mit $t \in [0,1]$ mit den Stützpunkten A(-1, 3, 0), B(0, 0, 0), C(1, 0, 3) und D(-1, 0, 5).

Außerdem ist der Vektor $\vec{v} = \begin{pmatrix} -1 \\ 0 \\ -5 \end{pmatrix}$ gegeben.

Hinweis: Die Basismatrix der Bézierkurven lautet: $M_{Bezier} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$

(a) Kann der Punkt P(0, 0, 10) auf der Kurve liegen? (Begründen Sie Ihre Antwort)

2 P.

(b) Wie lautet die Tangente der Kurve an Punkt D?

2 P.

(c) Geben Sie eine Formel für die Berechnung von Punkten auf der Kurve Q an.

3 P.

(d) Aus der Kurve wird eine Fläche F erzeugt, indem sie entlang einer Strecke (beschrieben durch den Vektor $\vec{v}$) nach hinten verschoben wird. Geben Sie eine Formel für die Berechnung von Punkten auf der Fläche F an.

3 P.

Σ2:

Aufgabe 2

Gegeben ist folgende VRML-Szene:

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

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




      children [


    ] } # T2
] } # T1
```

4 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 die Eckpunkte A(2, 2, 4), B(4,4,0) und C(0,4,0) enthält.

3 P.

(c) Die Spitze P soll eine Pyramide P‘ berühren, welche die gleiche Höhe, aber eine nur halb so große quadratische Grundfläche wie P hat. Der Abstand der Grundflächen von P und P‘ soll der doppelten Pyramidenhöhe entsprechen. Ergänzen Sie dazu den Transformnode T2. Benutzen Sie dabei den DEF-USE-Mechanismus.

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“*

Außerdem werden die folgenden neuen Hilfsfunktionen verwendet:
rotateX_NEW(alpha, M) *„erzeugt eine 4x4 Matrix R für die Rotation um die x-Achse um Winkel alpha und berechnet R * M“*,
rotateY_NEW(alpha, M) *„erzeugt eine 4x4 Matrix R für die Rotation um die y-Achse um Winkel alpha und berechnet R * M“*,
translate_NEW(x,y,z, M) *„erzeugt eine 4x4 Translationsmatrix T für den Translationsvektor $(x,y,z)^T$ und berechnet T * M “*,

```
// Model-Matrix berechnen
var model = mat4();
model = rotateY_NEW(- 90.0, model);
model = inverse(model);
model = rotateX_NEW(- 90.0, model);
model = translate_NEW(-1.0, -1.0, -1.0, model);

// View-Matrix berechnen
var view = mat4();
view = rotateX_NEW(90.0, view);
view = translate_NEW(0.0, 0.0, 3.0, view);
view = rotateY_NEW(-90.0, view);
```

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

4 P.

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

4 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 Objektkoordinaten von P, wenn im obigen Programm die Zeile inverse(model) durch transpose(model) ersetzt wird (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
    uniform vec4 lightDir;          // Richtung einer direktionalen Lichtquelle in Weltkoordinaten
    uniform vec3 lightIntensity;    // Intensität der direktionalen Lichtquelle (für RGB)
    attribute vec3 diffuseColor;    // die dem Vertex zugeordnete Farbe (für RGB)
    attribute vec4 normal;          // die dem Vertex zugeordnete Normale in Weltkoordinaten
    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?

3 P.

(b) Schreiben Sie eine Zeile GLSL-Code, welche die 3D Koordinaten von position einer Variablen p zuordnet:

vec3 p =

2 P.

(c) Welchen Nachteil gibt es, wenn statt „uniform vec4 lightDir;“ die Zeile „attribute vec4 lightDir;“ im Shader steht?

2 P.

(d) Im Vertex-Shader soll eine Phong-Beleuchtungsrechnung durchgeführt werden und zwar nur für den diffusen Teil. Die Formel lautet $C_{diff} = I_{diff} \cdot R_{diff} \cdot \cos \alpha$. Was ist die Bedeutung des Winkels α und wie kann der $\cos \alpha$ berechnet werden?

2 P.

(e) Welcher Code ist in den Vertex-Shader dazu aufzunehmen? Hinweis: Denken Sie daran, dass das berechnete Ergebnis auch der weiteren Renderpipeline zur Verfügung stehen soll.

4 P.

Σ6:

(f) Welche Codezeilen müssen im Shader ergänzt werden, wenn neben dem diffusen Licht auch das ambiente Licht bei der Beleuchtungsrechnung berücksichtigt werden soll?

2 P.

(g) Welcher Lichtanteil aus dem Phong-Modell ist bisher nicht berücksichtigt? Welche Informationen müssen generell übergeben werden, um diesen Lichtanteil berechnen zu können?

4 P.

(h) Statt eines direktionalen Lichtes soll eine Punktlichtquelle für die Beleuchtung verwendet werden. Wie ändert sich dadurch der Vertex-Shader?

3 P.

Aufgabe 5

(a) Was besagt das Nyquist-Theorem?

3 P.

Σ7:

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

2 P.

(c) In welchem Wertebereich liegen normalisierte Gerätekoordinaten in OpenGL?

2 P.

(d) Beschreiben Sie den Painter-Algorithmus für die Verdeckungsrechnung und geben Sie ein Beispiel an, bei dem der Algorithmus nicht anwendbar ist.

3 P.

(e) Nennen Sie eine Anwendung von Quaternionen in der Computergraphik.

2 P.

(f) Warum kann man mit Pipeline-Rendering keine Spiegelungen berechnen?

3 P.

Σ8: