Klausur Computergraphik (WS 2019/20)

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: 17. Februar 2020

Name:______________________________ Vorname: ________________________

Matr.-Nr. ____________________

MUSTERLÖSUNG

____
Unterschrift

Hinweise:

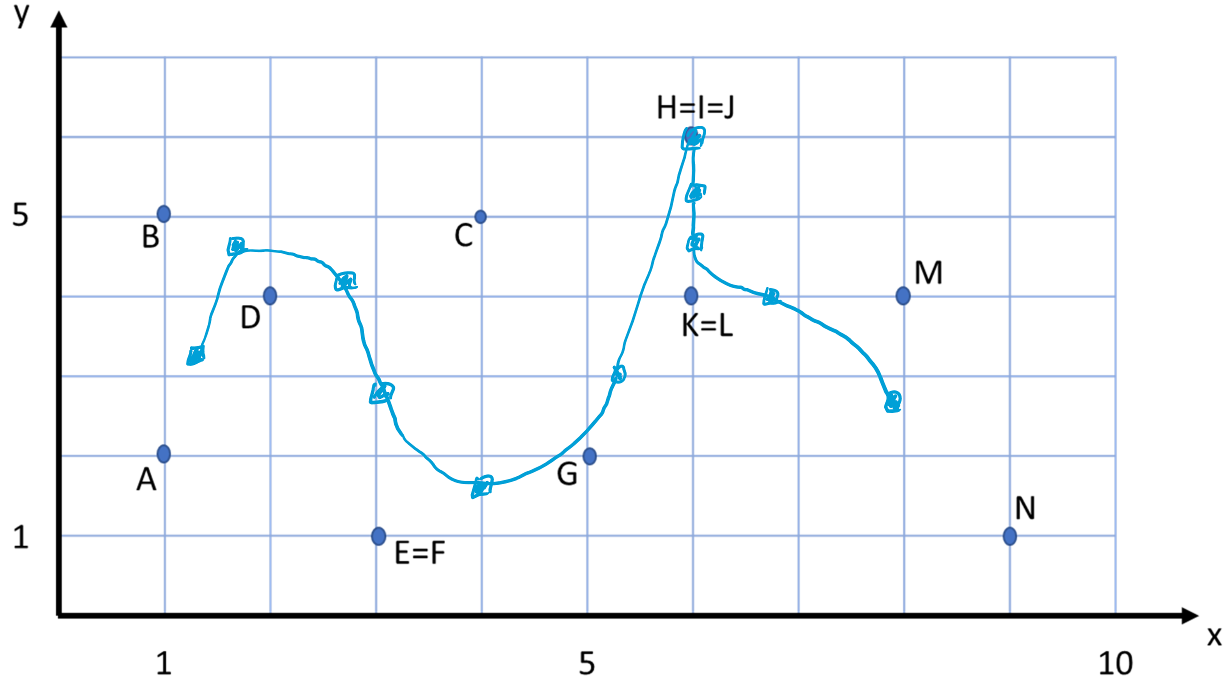
- Überprüfen Sie Ihr Klausurexemplar auf Vollständigkeit (Umfang: 8 Blätter)
- Lösen die Aufgaben im dafür vorgesehenen Raum. Wenn der Platz nicht ausreicht, verwenden Sie die Rückseiten - wenn alle Rückseiten beschrieben sind, fordern Sie ein leeres Blatt bei der Aufsicht an. Schreiben Sie im vorgesehenen 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 **40 Punkten.**

Es wurden __________ Punkte erreicht.

Note, Handzeichen:

Aufgabe 1

Gegeben ist eine kubische B-Spline-Kurve Q(t) mit den Stützpunkten A, B, C, …, N (siehe Zeichnung) und dem (unvollständigen) Knotenvektor T = [10, 13, …]. Die Basismatrix für ein Kurvensegment einer uniformen, kubischen B-Spline Kurve sei $M_{B\text{-Spline}}$.



(a) Vervollständigen Sie den Knotenvektor T so, dass Q(t) ein uniformer B-Spline wird

2 P. T = [10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43]

6 P. (b) Skizzieren Sie Q(t) in der obigen Zeichnung, heben Sie dabei die Knoten hervor.

(c) In welchem Bereich liegt das erste Kurvensegment von Q(t)?

2 P. Dreieck ABC (= konvexe Hülle A, B, C, D)

(d) Geben Sie eine Formel für die Berechnung von Q(24) an (setzen Sie so weit möglich die aktuellen Zahlen ein – der Punkt muss nicht berechnet werden).

3 P. $$Q(24) = Q_{5.Seg}\left(\frac{2}{3}\right) = \left[\left(\frac{2}{3}\right)^3 \left(\frac{2}{3}\right)^2 \ \frac{2}{3} \ 1\right] \cdot M_{B\text{-}Spline} \cdot \begin{bmatrix} 3 & 1 \\ 3 & 1 \\ 5 & 2 \\ 6 & 6 \end{bmatrix}$$

(e) Geben Sie eine Formel für die Berechnung des Endpunktes von Q(t) an (setzen Sie so weit möglich die aktuellen Zahlen ein – der Punkt muss nicht berechnet werden).

$$Q(43) = Q_{11.Seg.}(1) = [1 \ 1 \ 1 \ 1] \cdot M_{B\text{-}Spline} \cdot \begin{bmatrix} 6 & 4 \\ 6 & 4 \\ 8 & 4 \\ 9 & 1 \end{bmatrix}$$

3 P.

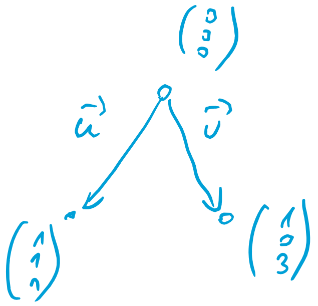
Σ2:

Aufgabe 2

Gegeben ist folgende VRML-Szene:

```
DEF T1 Transform{
  translation 1 2 1
  children [
    DEF S1 Shape{ geometry IndexedFaceSet{ coord Coordinate {
                                              point [  0 0 0, # Punkt A
                                                       1 1 1  # Punkt B
                                                       1 0 3, # Punkt C
                                                    } # Coordinate
                                     coordIndex [ 1 2 0 -1 ]
                         } # IndexedFaceSet
    } # S1
   DEF V1 Viewpoint{
                      position     1 2 3
                      orientation  0 0 1 3.14
   } # V1
 ] } # T1
```

(a) Berechnen Sie die Normale auf die im IndexedFaceSet-Node definierte Fläche und geben Sie deren Koordinaten in Weltkoordinaten an.



$$\vec{n}^{(S1)} = \vec{u} \times \vec{v} = \begin{pmatrix}1\\1\\1\end{pmatrix} \times \begin{pmatrix}1\\0\\3\end{pmatrix} = \begin{pmatrix}3\\-2\\-1\end{pmatrix}$$

$$\vec{n}^{(S1)} = \vec{n}^{(Welt)} = \begin{pmatrix}3\\-2\\-1\end{pmatrix}$$

3 P.

Translation ändert Vektor nicht

(b) Berechnen Sie die Koordinaten von Punkt B in Kamerakoordinaten mit V1 als Kamera.

$$\vec{p}^{(V1)} = M_{V1}^{-1} \cdot \vec{p}^{(S1)} = [T(1,2,3) \cdot R_z(180°)]^{-1} \cdot \vec{p}^{(S1)}$$

$$= R_z(-180°) \cdot T(-1,-2,-3) \cdot \vec{p}^{(S1)}$$

$$= \begin{bmatrix}-1&0&0&0\\0&-1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix} \cdot \begin{bmatrix}1&0&0&-1\\0&1&0&-2\\0&0&1&-3\\0&0&0&1\end{bmatrix} \cdot \begin{pmatrix}1\\1\\1\\1\end{pmatrix} = \begin{pmatrix}0\\1\\-2\\1\end{pmatrix}$$

4 P.

$$\Rightarrow p^{(V1)}(0, 1, -2)$$

(c) Durch welchen lookAt-Befehl kann man die äußere Orientierung der Kamera V1 beschreiben?

3 P. lookAt(2 , 4 , 4 , 2 , 4 , 2 , 0 , -1 , 0)

$2 < 4$

Σ3:

Aufgabe 3

Gegeben ist folgender Ausschnitt aus einem WebGL-Javascript, wobei die in der Lehrveranstaltung vorgestellten Hilfsfunktionen verwendet werden sowie die Konstante PI mit Wert π:

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 $(x,y,z)^T$“*,
scale(sx,sy,sz) *„erzeugt eine 4x4 Skalierungsmatrix für die Skalierungsfaktoren (s_x,s_y,s_z)“*,
perspective(fov, aspect, near, far) *„erzeugt eine Projektionsmatrix“*

```
// Projektionsmatrix
var projection = perspective(60.0, 1.0, 0.5, 200.0);

// Zeile A: hier die Model-Matrix mA anlegen
var mA = mult(translate(1, 1, 3), rotate(4 * 180 / Pi, [2, 2, 4]));
mA = mult(mA, rotate(3 * 180 / PI, [1, 1, 3]));
mA = mult(mA, Scale(1, 1, 3));
mA = mult(mA, rotate(-3 * 180 / Pi, [1, 1, 3]));
mA = mult(mA, translate(-1, -1, -3));

// Zeile B: hier die View-Matrix mB anlegen
var mB = mult(rotate(90, [0, 1, 0]), translate(0, 0, -3));

// Zeile C: hier Matrix mC anlegen, die Objektkoordinaten in Clipping-Koordinaten umrechnet
var mC = mult(projection, mB);
mC' = mult(mC', mA);

// Zeile D: hier Matrix mD anlegen, die Normalen von Objektkoordinaten in Kamerakoordinaten
// umrechnet
var mD = mult(mB, mA);
mD = transpose(invert(mD));
```

(a) Ergänzen Sie das Programm nach Zeile A so, dass alle Modelle auf die gleiche Weise transformiert werden, wie es in folgendem VRML-Transform Node spezifiziert wird:

```
Transform{      scaleOrientation    1 1 3 3
                center              1 1 3
                scale               1 1 3
                rotation            2 2 4 4
                children [ # ...
]}
```

4 P.

Σ4:

(b) Ergänzen Sie das Programm nach Zeile B so, dass die Kamera so orientiert und platziert wird, wie es in folgendem VRML-Viewpoint-Node spezifiziert wird (verwenden Sie dabei nur die oben angegebenen Hilfsfunktionen):

```
Viewpoint{     orientation    0 1 0 -1.57
               position       0 0 3
}
```

3 P.

(c) Mit welchem lookAt-Befehl würde man die selbe Kameraposition spezifizieren wie mit dem VRML-Viewpoint-Node aus Teilaufgabe (b)?

x > 0

3 P. lookAt(0 , 0 , 3 , x , 0 , 3 , 0 , 1 , 0)

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

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

(f) Wie ändert sich das Bild, wenn perspective(90.0, 1.0, 0.5, 100.0) abgeändert wird in: perspective(30.0, 1.0, 0.5, 5.0); ?

1.) FOV wird kleiner ⇒ weniger Objekte links u. rechts sichtbar

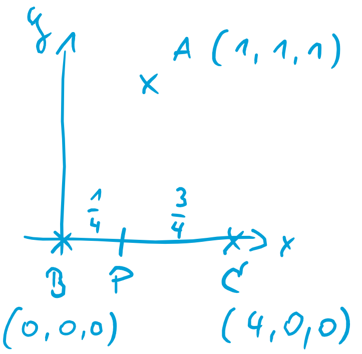
2 P.

2.) Far Plane näher ⇒ Objekte mit Entfernung ∈ [5, 100] vom Kamera nicht mehr auf dem Bild

Aufgabe 4

Gegeben ist ein Dreieck D mit den Koordinaten A(1, 1, 1), B(0, 0, 0) und C(4, 0, 0). Die Fläche liegt wie z.B. in VRML üblich im Umlaufsinn links. Am Punkt (6, 3, 2) befindet sich der Augpunkt der Kamera K. Alle Koordinaten sind in Weltkoordinaten angegeben.

(a) Eine Beleuchtungsrechnung ergibt, dass an Punkt B die diffuse Reflektion die Farbe (60°; ½; 1) und an Punkt C die Farbe (undefined; ½; 0) vorliegt (Farbangaben sind im HLS-Farbsystem). Wie lautet die Farbe (im RGB-Farbsystem) am Punkt P(1; 0; 0), wenn Gouraud-Shading verwendet wird?



HLS ≙ RGB

$$F_B = (60°, \tfrac{1}{2}, 1) \;\hat{=}\; (1, 1, 0)$$

$$F_C = (\text{undef}, \tfrac{1}{2}, 0) \;\hat{=}\; (\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2})$$

$$F_P = \tfrac{1}{4} \cdot F_C + \tfrac{3}{4} \cdot F_B$$

$$= (\tfrac{1}{8}, \tfrac{1}{8}, \tfrac{1}{8}) + (\tfrac{3}{4}, \tfrac{3}{4}, 0)$$

$$= (\tfrac{7}{8}, \tfrac{7}{8}, \tfrac{1}{8})$$

5 P.

Σ5:

(b) Berechnen Sie, ob der Punkt A durch Backface-Culling entfernt wird oder nicht.

$$\vec{n} = \vec{c} \times \vec{a} = \begin{pmatrix} 4 \\ 0 \\ 0 \end{pmatrix} \times \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -4 \\ 4 \end{pmatrix}$$

$$\langle \vec{n}, \vec{k} - \vec{a} \rangle = \left\langle \begin{pmatrix} 0 \\ -4 \\ 4 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \\ 2 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\rangle$$

$$= \left\langle \begin{pmatrix} 0 \\ -4 \\ 4 \end{pmatrix}, \begin{pmatrix} 5 \\ 2 \\ 1 \end{pmatrix} \right\rangle = -4 < 0$$

$\Rightarrow$ A wird entfernt

4 P.

Aufgabe 5

Der Punkt P(3, 3, 1) soll mit einer Kamera, die sich an Punkt A(1, 1,-3) befindet, auf die Projektionsebene mit der Gleichung z = -5 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? $(0, 0, d)$, $d > 0$

Wohin muss die Kamera schauen? $-z$ - Richtung

3 P. Wo muss sich die Projektionsebene befinden?

$z = 0$

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

$T(-1, -1, 5)$

2 P.

(c) Berechnen Sie die Bildkoordinaten von Punkt P.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{2} & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} 3 \\ 3 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 6 \\ -2 \end{pmatrix} \xrightarrow{\text{persp. Div.}} \begin{pmatrix} -1 \\ -1 \\ -3 \end{pmatrix}$$

4 P.

$\Rightarrow P'(-1, -1)$

Σ6:

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

$z = 2$

2 P.

Aufgabe 6

(a) Nennen Sie zwei innere Parameter einer virtuellen Kamera:

1. Field of View

2 P.

2. Aspect Ratio

(b) Nennen Sie einen Vorteil den Rastergrafiken gegenüber Vektorgrafiken bieten.

gleiche Speichergröße unabhängig vom Bildinhalt

2 P.

(c) Nennen Sie einen Vorteil des HLS-Farbmodells gegenüber dem RGB-Farbmodell.

Orientierung an menschlicher Wahrnehmung, daher Farbe für Menschen einfacher zu spezifizieren

2 P.

(d) Was versteht man in der Computergrafik unter einem Billboard?

Rechteck mit einer Textur, das immer senkrecht zum Betrachter ausgerichtet wird

2 P.

Σ7:

(e) 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.rara;
v.q = u.s;
```

2 P. Welchen Wert hat v nach Ausführung der letzten Zeile? v = (5, 8, 5, 5)

(f) Nennen Sie drei Möglichkeiten ein Texturmapping anzugeben:

1. Abbildungsvorschrift
2. Projektion mit Hüllkörper
3. Texturkoordinaten

3 P.

(g) Wozu dient der Painter-Algorithmus?

Verdeckungsrechnung

2 P.

(h) Nennen Sie zwei Beispiele für Per-Fragment-Ops in der Render-Pipeline

1. Depth – Test
2. Alpha – Test

2 P.

(i) Welches Problem des z-Buffer-Algorithmus kann durch Verwendung von Nebel in einer 3D Szene gemindert werden?

mit wachsendem z wird Auflösung durch Gleitkommadarstellung geringer

=> Risiko steigt, dass Objekte mit unterschiedlicher Entfernung den gleichen z-Wert erhalten

=> in diesem Fall kann Flackern auftreten.

Nebel kaschiert das Flackern

3 P.

Σ8: