

Klausur Computergraphik (SS 2020)

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: 22. Juli 2020

Name: _____	Vorname: _____
Matr.-Nr. _____	
	_____ Unterschrift

Hinweise:

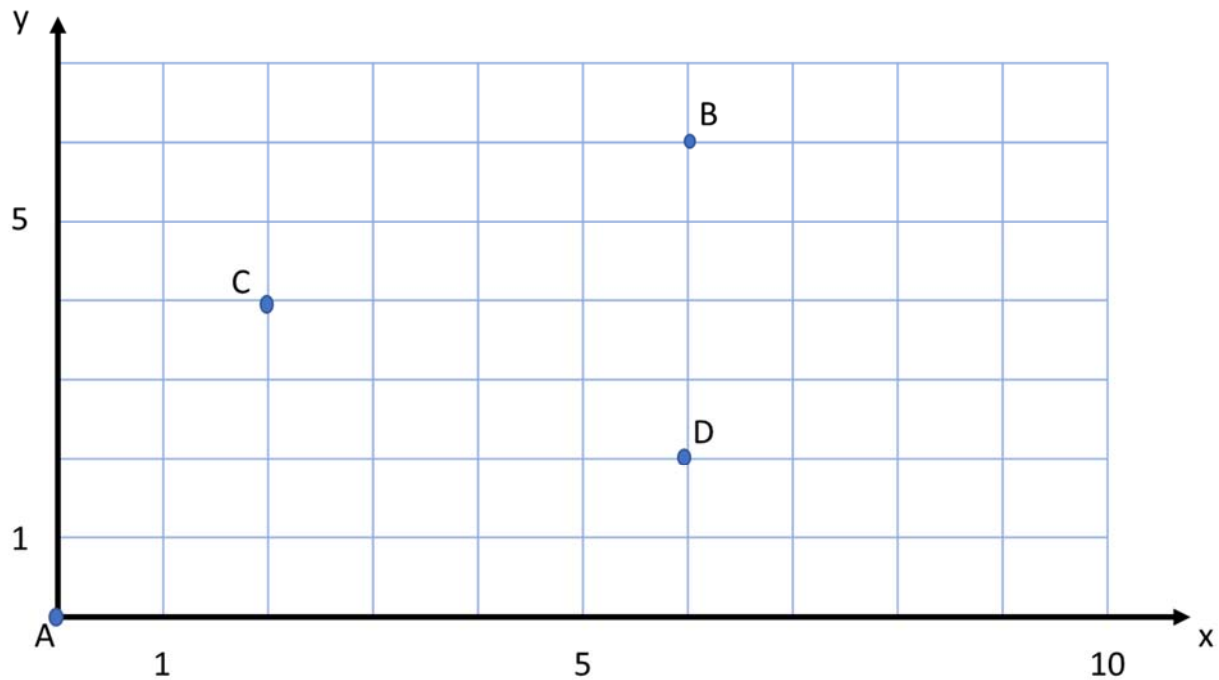
- Überprüfen Sie Ihr Klausurexemplar auf Vollständigkeit (Umfang: 8 Blätter)
- Lösen die Aufgaben im dafür vorgesehenen Raum. Wenn der Platz nicht ausreicht, verwenden Sie die Rückseiten - wenn alle Rückseiten beschrieben sind, fordern Sie ein leeres Blatt bei der Aufsicht an. Schreiben Sie im vorgesehenen 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 **49 Punkten**.

Es wurden _____ Punkte erreicht.

Note, Handzeichen:

Aufgabe 1

Gegeben ist die Bezier-Kurve $Q(t)$, $t \in [0,1]$ mit den Stützpunkten A, B, C, D.



(a) Bestimmen Sie zeichnerisch den Punkt $Q(0,5)$ mit dem Algorithmus von deCasteljau.

4 P. $Q(0,5) = (\quad , \quad)$

4 P. (b) Skizzieren Sie die $Q(t)$

(c) An dem Kurvenpunkt D soll eine weitere Bezier-Kurve $R(t)$, $t \in [0,1]$ angeschlossen werden, so dass ein C^1 -stetiger Übergang entsteht und die Kurve in Punkt B mit Tangente $(-1,1)^T$ endet. Wie lauten die Stützpunkte von R?

R – erster Stützpunkt: (\quad , \quad) , R – zweiter Stützpunkt: (\quad , \quad)

4 P. R – dritter Stützpunkt: (\quad , \quad) , R – vierter Stützpunkt: (\quad , \quad)

(d) Der aus Q und R gebildete Bezier-Spline soll durch drei Bezier-Kurven S_1 , S_2 und S_3 ersetzt werden, welche die gleiche Kurvenform haben. Wie lauten die Stützpunkte von S_1 , S_2 und S_3 ?

S_1 – erster Stützpunkt: (\quad , \quad) , S_1 – zweiter Stützpunkt: (\quad , \quad)

S_1 – dritter Stützpunkt: (\quad , \quad) , S_1 – vierter Stützpunkt: (\quad , \quad)

S_2 – erster Stützpunkt: (\quad , \quad) , S_2 – zweiter Stützpunkt: (\quad , \quad)

S_2 – dritter Stützpunkt: (\quad , \quad) , S_2 – vierter Stützpunkt: (\quad , \quad)

S_3 – erster Stützpunkt: (\quad , \quad) , S_3 – zweiter Stützpunkt: (\quad , \quad)

10 P. S_3 – dritter Stützpunkt: (\quad , \quad) , S_3 – vierter Stützpunkt: (\quad , \quad)

Aufgabe 2

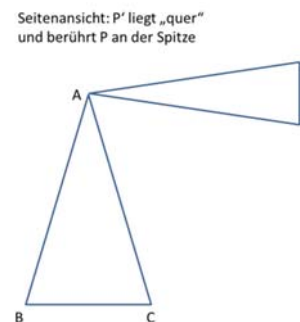
Gegeben ist folgende VRML-Szene:

```
DEF T1 Transform{
  children [
    DEF S1 Shape{ geometry IndexedFaceSet{ coord Coordinate {
                                                    point [ 0 0 0,
                                                            1 1 4,
                                                            2 0 0,
                                                            2 2 0,
                                                            0 2 0 ] } # Coordinate
                                                    coordIndex [ 4 0 2 3 -1 1 0 4 -1 1 0 2 -1 1 4 3 -1 _____ ]
                                                    } # IndexedFaceSet
    } # S1
    DEF V1 Viewpoint{ orientation 2 2 0 1.57
                      position 2 0 -1
    } # V1
    DEF T2 Transform{

      children [

        ] } # T2
    ] } # T1
```

- 5 P.
- Das IndexFaceSet soll eine Pyramide P mit quadratischer Grundfläche beschreiben. Korrigieren Sie das Feld coordIndex und ergänzen Sie ggf. fehlende Flächen.
 - Berechnen Sie die Normale, die bzgl. der Pyramide nach außen zeigt, auf dem durch die Eckpunkte A(1, 1, 4), B(0,2,0) und C(2,2,0) definierten Dreieck D.



- 3 P.
- 4 P.
- Die Spitze P soll eine Pyramide P' berühren, welche die gleiche Höhe, aber eine nur halb so große quadratische Grundfläche wie P hat. Die Grundflächen von P und P' stehen senkrecht zueinander und die Grundfläche von P' befindet sich auf der Seite von Punkt C (vgl. Skizze oben). Ergänzen Sie dazu den Transformnode T2. Benutzen Sie dabei den DEF-USE-Mechanismus.

- (d) Wie weit ist die Spitze der Pyramide P von der Position der Kamera V1 in Kamerakoordinaten entfernt?

3 P.

- (e) Wie lauten die Koordinaten des View-Up-Vektors in Kamerakoordinaten?

2 P.

- (f) Geben Sie eine Formel, die nicht ausmultiplizierte 4×4 Matrizen enthält, zur Berechnung des View-Up-Vektors in Weltkoordinaten an.

5 P.

Aufgabe 3

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 vec3 lightDirection; // Richtung einer direktionalen Lichtquelle in Weltkoordinaten
    uniform vec3 lightIntensity; // Intensität der Punktlichtquelle (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?

2 P.

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

2 P.

vec3 p =

- (c) Warum ist „uniform vec3 lightDirection;“ der Deklaration „attribute vec3 lightDirection;“ vorzuziehen?

3 P.

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

3 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.

- (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 übergeben werden, um diesen Lichtanteil berechnen zu können?

4 P.

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

3 P.

Aufgabe 4

Gegeben ist ein Quaternion $\hat{q} = (\cos 30^\circ, \frac{\sqrt{2}}{2} \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix})$

- (a) Wie lautet das zu \hat{q} gehörige Einheitsquaternion (Hinweis: $\sin 30^\circ = \frac{1}{2}$)?

3 P.

(b) Welche Rotation R wird durch das zu \hat{q} gehörige Einheitsquaternion repräsentiert?

3 P.

(c) Wie lauten die Eulerwinkel (Konvention: Reihenfolge x, y, z), welche die gleiche Rotation wie R erzeugen (es kann auch eine Reihe von Eulerwinkeln angegeben werden)?

3 P.

(d) Geben Sie eine Formel aus nicht ausmultiplizierten 4×4 Matrizen an, mit denen die Drehung R in homogenen Koordinaten berechnet werden kann

4 P.

Aufgabe 5

(a) Erklären Sie den Begriff „Gimbal Lock“. Wie kann man einen Gimbal Lock vermeiden?

3 P.

(b) Was versteht man in der Computergrafik unter „BRDF“?

3 P.

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

3 P.

(d) Nennen Sie zwei Unterschiede zwischen der Reflektionsgleichung und der Renderinggleichung.

1. _____

2. _____

3 P.

(e) Was versteht man unter „View Frustum Culling“?

3 P.

(f) Warum kann man mit dem Phong-Beleuchtungsmodell keinen Schatten berechnen?

3 P.