Karlsruher Institut für Technologie

Institut für Visualisierung und Datenanalyse
Lehrstuhl für Computergrafik
Prof. Dr.-Ing. Carsten Dachsbacher

Hauptklausur Computergrafik
WS 2017/2018

6. März 2018

Kleben Sie hier **vor Bearbeitung der Klausur** den Aufkleber auf.

Beachten Sie:

- Trennen Sie vorsichtig die dreistellige Nummer von Ihrem Aufkleber ab. Sie sollten sie gut aufheben, um später Ihre Note zu erfahren.
- Die Klausur umfasst 23 Seiten (11 Blätter) mit 9 Aufgaben.
- Es sind **keine Hilfsmittel** zugelassen.
- Sie haben **90 Minuten** Bearbeitungszeit.
- Schreiben Sie Ihre Matrikelnummer oben auf jedes bearbeitete Aufgabenblatt.
- Schreiben Sie Ihre Lösungen auf die Aufgabenblätter. Bei Bedarf können Sie weiteres Papier anfordern.
- Wenn Sie bei einer Multiple-Choice-Frage eine falsche Antwort angekreuzt haben und diesen Fehler korrigieren möchten, füllen Sie das betreffende Kästchen ganz aus:

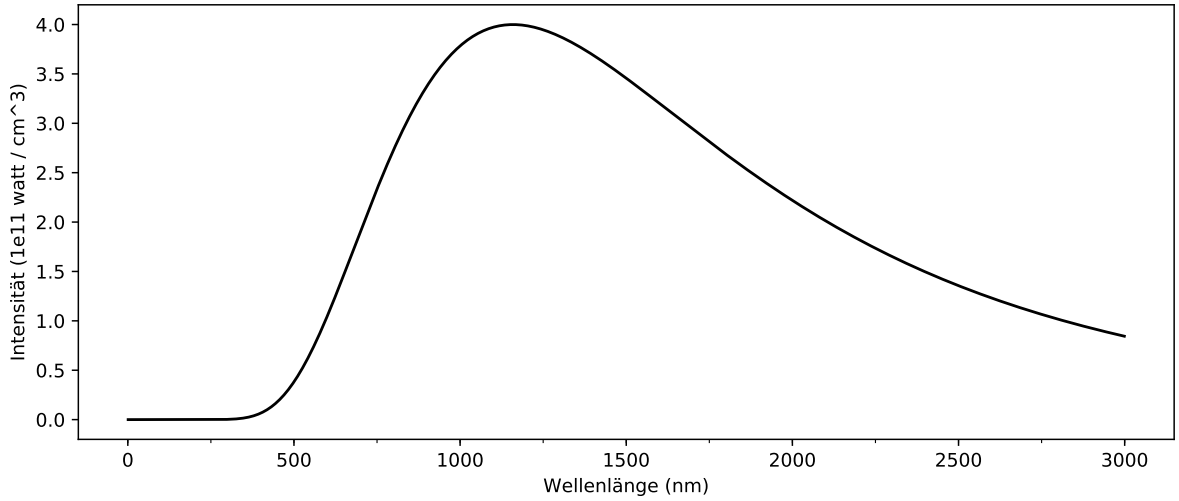
- Wir akzeptieren auch englische Antworten.

Aufgabe	1	2	3	4	5	6	7	8	9	**Gesamt**
Erreichte Punkte										
Erreichbare Punkte	11	27	23	22	8	16	20	22	31	**180**

Note

Aufgabe 1: Farben (11 Punkte)

a) Im folgenden Diagramm ist das Emissionsspektrum eines Schwarzkörpers bei 2500K abgebildet:



i) Markieren Sie in der Zeichnung den für Menschen sichtbaren Wellenlängenbereich! **(2 Punkte)**

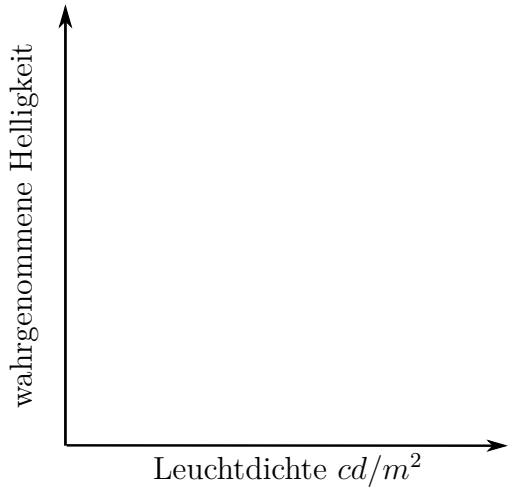
ii) Welche Farbe hat der Körper für einen menschlichen Betrachter? **(2 Punkte)**

b) Sie schicken von der Grafikkarte den Intensitätswert von 0.5 an einen Bildschirm und messen an dem Bildschirm eine Ausgabe mit einem Viertel (0.25) der Maximalintensität. Bestimmen Sie den Gammawert γ des Bildschirms und geben Sie den Rechenweg an! **(4 Punkte)**

Matrikelnummer: _____

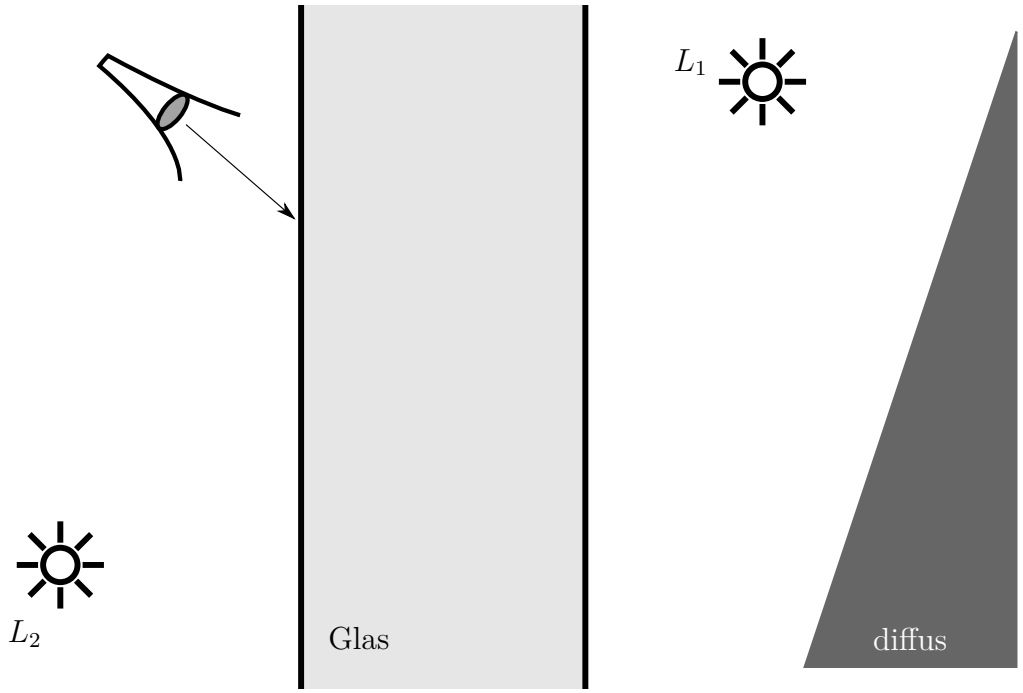
c) Nennen Sie eine gesättigte Farbe, die kein Teil des Regenbogens sein kann und begründen Sie Ihre Antwort kurz! **(3 Punkte)**

d) Skizzieren Sie die von einem menschlichen Betrachter subjektiv empfundene Helligkeit abhängig von der Leuchtdichte (Intensität des Reizes) in dem gegebenen Koordinatensystem! Wie verhält sich die empfundene Helligkeit mathematisch zur Leuchtdichte? **(2 Punkte)**



Aufgabe 2: Raytracing und das Phong-Beleuchtungsmodell (27 Punkte)

a) Die Abbildung zeigt eine Kamera, einen Primärstrahl und zwei Punktlichtquellen in einer Szene. In der Mitte befindet sich ein perfekt transmittierender Glasblock, rechts eine diffuse Oberfläche. In beiden Aufgabenteilen soll ein Whitted-Style Raytracer eingesetzt werden, der Dispersion umsetzt.



i) In diesem Aufgabenteil wird die Szene nur von L_1 und nicht von L_2 beleuchtet. Zeichnen Sie alle Sekundärstrahlen ein, die der Whitted-Style Raytracer ausgehend von dem gegebenen Primärstrahl erzeugen würde! Beachten Sie hierbei die Trennung der Strahlen für R,G,B und beschriften Sie die Strahlen entsprechend. Gehen Sie davon aus, dass für die Brechungsindizes des Glaskörpers $1 < \eta_R < \eta_G < \eta_B$ gilt und dass der Brechungsindex der umgebenden Luft 1 ist. Zeichnen Sie nur Strahlen ein, die einen Beitrag zum Ergebnis liefern können! **(8 Punkte)**

ii) Wie würde das resultierende Bild aussehen, wenn die Szene von L_2 und nicht von L_1 beleuchtet werden würde? Begründen Sie Ihre Antwort! **(4 Punkte)**

Matrikelnummer: _____

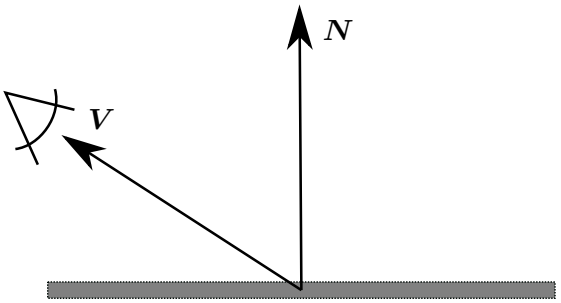
b) Das Phong-Beleuchtungsmodell beschreibt die Intensität I des reflektierten Lichts in Abhängigkeit der Materialkonstanten k_a, k_d, k_s, n, der Richtung zur Lichtquelle $\boldsymbol{L}$, der Richtung zur Kamera $\boldsymbol{V}$, der Oberflächennormale $\boldsymbol{N}$ und der Lichtintensität I_L mit der folgenden Gleichung:

$$I = k_a I_L + k_d I_L(\boldsymbol{N} \cdot \boldsymbol{L}) + k_s I_L(\boldsymbol{R_L} \cdot \boldsymbol{V})^n,$$

wobei

$$\boldsymbol{R_L} = -\boldsymbol{L} + 2\boldsymbol{N}(\boldsymbol{L} \cdot \boldsymbol{N}).$$

i) Zeichnen Sie in die gegebene Skizze einen Lichtvektor $\boldsymbol{L}$ und Reflexionsvektor $\boldsymbol{R_L}$ ein, für die das Skalarprodukt $\boldsymbol{R_L} \cdot \boldsymbol{V}$ begrenzt (geclampt) werden muss! **(3 Punkte)**



ii) Wie verändert sich das Glanzlicht, wenn Sie den Phong-Exponenten erhöhen? **(2 Punkte)**

Aufgabe 3: Transformationen (23 Punkte)

a) Geben Sie die 4×4-Transformationsmatrix für ein rechtshändiges Koordinatensystem an, die einen Punkt $\boldsymbol{p} \in (\mathbb{R}^4)$ in homogenen Koordinaten

i) um einen Vektor $\boldsymbol{t} = (1, 3, 1)^T$ verschiebt und mit $\boldsymbol{s} = (2, 3, 4)^T$ skaliert: **(2 Punkte)**

$M(\boldsymbol{t}, \boldsymbol{s}) =$

ii) um den Winkel θ gegen den Uhrzeigersinn um die Achse $\boldsymbol{r} = (0, 0, 1)^T$ rotiert: **(4 Punkte)**

$R(\theta, \boldsymbol{r}) =$

Matrikelnummer: _____

b) Vervollständigen Sie die folgende GLSL-Funktion `rotate`, welche einen Punkt $\boldsymbol{p} \in \mathbb{R}^3$ um den Winkel θ um eine beliebige Achse $\boldsymbol{r} = (r_x, r_y, r_z)^T$ rotiert! Sie können davon ausgehen, dass $||\boldsymbol{r}|| = 1$, $r_x \neq 0, r_y \neq 0$ und $r_z \neq 0$. Sie können die Funktion `rotate_z` benutzen. **(10 Punkte)**

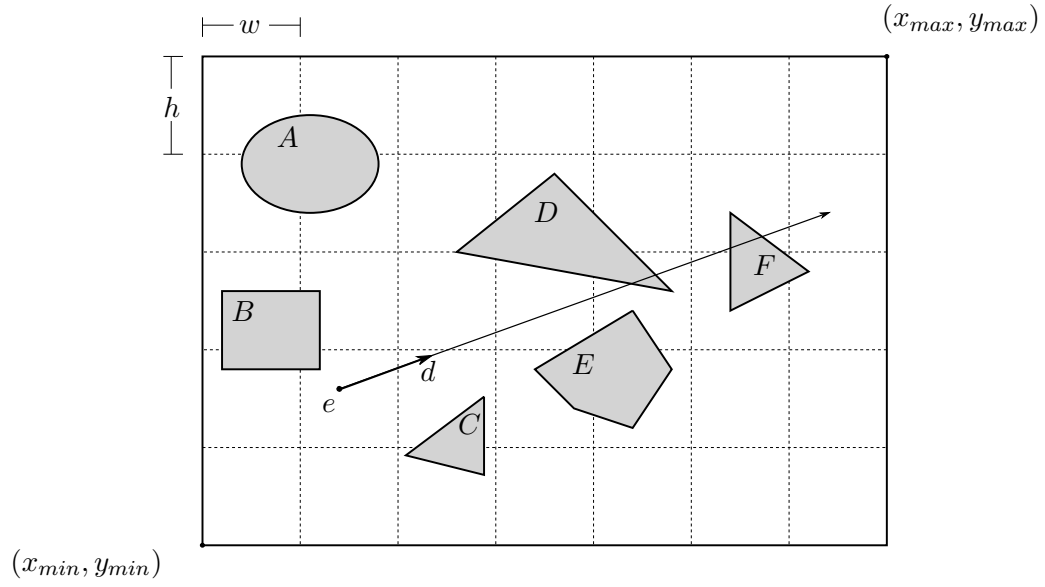
```
mat3 rotate_z(float theta); // gibt eine 3x3-Matrix zurück,
                            // die um die Achse (0,0,1) rotiert

vec3 rotate(vec3 p, float theta, vec3 r)
{




}
```

Aufgabe 4: Beschleunigungsstrukturen (22 Punkte)

In der Vorlesung wurden Datenstrukturen vorgestellt, die beim Raytracing die Anzahl der Schnittberechnungen von Strahlen mit Geometrie einer Szene reduzieren können.

a) Warum sind hierarchische Datenstrukturen in der Lage, die Anzahl der Schnitttests beim Raytracing zu reduzieren? **(3 Punkte)**

b) Das in der Abbildung gezeigte reguläre Gitter wird für die Beschleunigung eines Strahlschnitts benutzt. Ziel ist es, den nächstliegenden Schnittpunkt zu finden. Gehen Sie davon aus, dass kein Mailboxing verwendet wird. Notieren Sie alle mit Primitiven durchgeführten Schnitttests in der Reihenfolge, in der sie stattfinden! **(4 Punkte)**



Matrikelnummer: _____

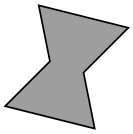
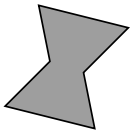
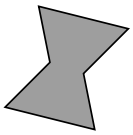
c) Ein reguläres Gitter soll für den Strahl $\boldsymbol{r}(t) = \boldsymbol{e} + t\,\boldsymbol{d}$ mit $\boldsymbol{e} = (e_x, e_y)$ im Gitter und $\boldsymbol{d} = (d_x, d_y)$ traversiert werden. Für den gesamten Aufgabenteil gelte $d_{x,y} > 0$ und $||\boldsymbol{d}|| = 1$. Die Ausmaße des Gitters sind durch die Ecke links unten (x_{min}, y_{min}) und rechts oben (x_{max}, y_{max}) gegeben. Das Gitter hat eine Zellenbreite von w und Zellenhöhe von h. Der Index einer Zelle entlang der x- bzw. y-Achse wird durch i bzw. j beschrieben.

i) Bestimmen Sie, wie in der Vorlesung besprochen, die Startzelle (i, j), die zuerst vom Strahl traversiert wird! **(5 Punkte)** ☐

ii) Initialisieren Sie für die Startzelle (i, j) die beiden Distanzen t_{next_x} und t_{next_y} entlang des Strahls, bei denen beim Traversieren in die nächste Zelle in x- bzw. y-Richtung gewechselt wird! **(6 Punkte)** ☐

iii) Im nächsten Schritt soll die nächste zu testende Zelle bestimmt werden. Geben Sie in Pseudocode an, wie i, j, t_{next_x} und t_{next_y} inkrementell aktualisiert werden! **(6 Punkte)** ☐

d) Skizzieren Sie um die Primitive in der folgenden Abbildung drei verschiedene Hüllkörper, die Sie in der Vorlesung kennengelernt haben. Wie bezeichnet man diese Hüllkörper? Nennen Sie je einen Vor- und Nachteil des Hüllkörpers im Kontext von Beschleunigungsstrukturen! **(6 Punkte)**

Skizze	**Name**	**Vor- und Nachteil**
		
		
		

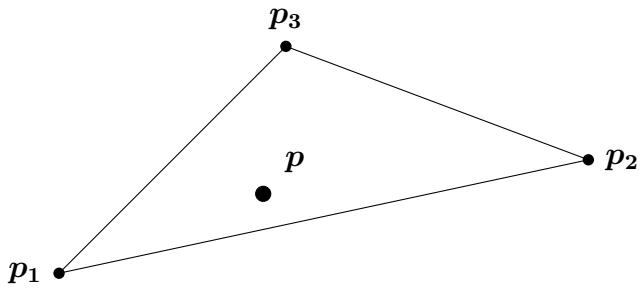
e) Kreuzen Sie jeweils die zutreffenden Kästchen an! Sie erhalten für jede vollständig richtige Zeile 1,5 Punkte. **(6 Punkte)**

Aussage	**BVH**	**Octree**	**kD-Baum**	**Gitter**
Die Datenstruktur partitioniert Objekte.	☐	☐	☐	☐
Der Aufwand für die Traversierung der Datenstruktur ist im Durchschnitt (Average Case) logarithmisch in der Anzahl der Primitive.	☐	☐	☐	☐
Die Datenstruktur eignet sich für Szenen mit großer Ausdehnung und ungleichmäßig verteilter Geometrie.	☐	☐	☐	☐
Für den Aufbau der Datenstruktur kann die Surface Area Heuristic (SAH) eingesetzt werden.	☐	☐	☐	☐

Matrikelnummer: _____

Aufgabe 5: Texturen und baryzentrische Koordinaten (8 Punkte)

a) In der folgenden Abbildung ist ein Dreieck mit Eckpunkten $\boldsymbol{p_1}, \boldsymbol{p_2}, \boldsymbol{p_3} \in \mathbb{R}^3$ dargestellt. Das Dreieck besitzt die zugehörigen Texturkoordinaten $\boldsymbol{t_1}, \boldsymbol{t_2}, \boldsymbol{t_3} \in \mathbb{R}^2$ und hat einen Flächeninhalt von $\Delta(\boldsymbol{p_1}, \boldsymbol{p_2}, \boldsymbol{p_3})$.



i) Geben Sie die Formeln zur Bestimmung der baryzentrischen Koordinaten $\lambda_1, \lambda_2, \lambda_3$ für den Punkt $\boldsymbol{p}$ an! **(4 Punkte)**

ii) Bestimmmen Sie die Texturkoordinate $\boldsymbol{t}$ am Punkt $\boldsymbol{p}$! **(2 Punkte)**

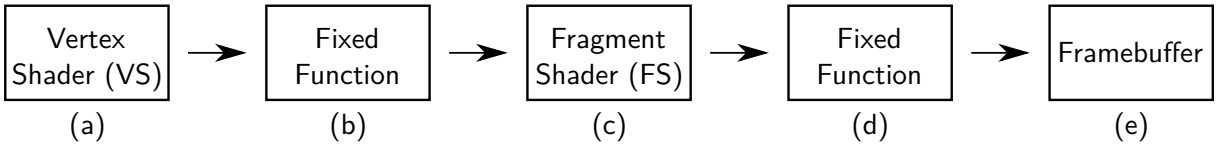
b) Wenn texturierte Oberflächen gezeichnet werden, können Verkleinerung (Minification) und Vergrößerung (Magnification) der Textur auftreten. Erklären Sie diese beiden Fälle und geben Sie für jeden Fall eine geeignete Filtermethode für 2D-Texturen an! **(6 Punkte)**

c) Welches Problem der isotropen Texturfilterung löst die anisotrope Texturfilterung? **(3 Punkte)**

d) Texturen werden nicht nur als (diffuse) Farbtexturen eingesetzt. Nennen Sie drei weitere Anwendungen bzw. Techniken und beschreiben Sie jeweils kurz, welche Bedeutung die Texelwerte dabei haben, z.B. welcher Term des Phong-Beleuchtungsmodells dabei beeinflusst wird. **(6 Punkte)**

Matrikelnummer: _____

Aufgabe 6: OpenGL - Pipeline (16 Punkte)



Ein indiziertes Dreiecksnetz wird mit OpenGL gezeichnet. Pro Vertex werden eine Position und Normale in Modellkoordinaten gespeichert. Es soll Phong-Shading mit dem Phong-Beleuchtungsmodell umgesetzt werden. Weiterhin ist eine Punktlichtquelle in Weltkoordinaten definiert.

a) Nennen Sie die Uniform-, Ein-, und Ausgabevariablen inklusive der Transformationsmatrizen, die der Vertex Shader (a) und der Fragment Shader (c) verwenden! Geben Sie für alle Variablen die entsprechenden Koordinatensysteme an. Geben Sie insbesondere für die Transformationsmatrizen an, von welchem in welches Koordinatensystem sie jeweils transformieren. Sie müssen die Materialparameter, die für die Auswertung des Phong-Beleuchtungsmodells benötigt werden, *nicht* angeben. **(12 Punkte)**

	Bedeutung und ggf. Koordinatensystem
VS uniform:	
VS in:	
VS out / FS in:	
FS uniform:	
FS out:	

b) In welchem Shader wird das eigentliche Beleuchtungsmodell ausgewertet? **(2 Punkte)**

c) In welchem Block in der Pipeline (a,b,c,d,e) werden üblicherweise die Primitive geclippt? Begründen Sie, warum der Schritt an dieser Stelle erfolgt! **(4 Punkte)**

d) Ist es im Allgemeinen möglich, bei einem Triangle-Strip Back-Face-Culling im Vertex Shader umzusetzen? Begründen Sie Ihre Antwort (kurz)! **(4 Punkte)**

Matrikelnummer: _____

Aufgabe 7: OpenGL-Blending (20 Punkte)

OpenGL stellt folgende Funktionen zur Konfiguration des Blendings bereit:

```
void glBlendFunc(GLenum sfactor, GLenum dfactor);
void glBlendEquation(GLenum mode);
```

a) Gegeben ist die folgende Sequenz von OpenGL-Aufrufen, die das Blending konfiguriert:

```
glEnable(GL_BLEND);
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
glBlendEquation(GL_FUNC_ADD);
```

Geben Sie den Namen dieses Blending-Verfahrens an! Geben Sie zudem als Ausdruck an, wie OpenGL den endgültigen Farbwert *final* des Framebuffers berechnet! Verwenden Sie eine an GLSL angelehnte Darstellung mit den Symbolen *src* und *dst*, wobei Sie jeweils die Swizzles *r,g,b* und *a* verwenden können, um auf Komponenten zuzugreifen! Beispiel: *final = src * dst.a* **(2 Punkte)**

b) Sie sollen in dieser Aufgabe das Ergebnis eines Blending-Vorgangs ausrechnen. Die Blending-Konfiguration wurde mit den folgenden Befehlen gesetzt:

```
glBlendFunc(GL_ONE, GL_SRC1_COLOR);
glBlendEquation(GL_FUNC_ADD);
```

Dazu haben Sie folgende Informationen:

- SRC_COLOR = vec4(0.1, 0.2, 0.3, 0.4)
- SRC1_COLOR = vec4(0.4, 0.5, 0.6, 1.0)
- Der Inhalt des Framebuffers ist an jeder Stelle vec4(0.1, 0.6, 0.4, 1.0)

Welchen Wert hat der Framebuffer an der Stelle des Fragments, nachdem ein Zeichenbefehl mit der obigen Konfiguration ausgeführt wurde? Geben Sie den Rechenweg mit an! **(3 Punkte)**

c) In einer Szene mit einem Partikelsystem soll ein Teil der Partikel mit dem Blending-Verfahren aus a), ein zweiter Teil mit additivem Blending und ein dritter Teil mit gewichtetem additivem Blending mit Gewicht *src.a* gezeichnet werden. Alle Partikel sollen in einem einzigen Zeichenaufruf (Draw-Call) gezeichnet werden, d.h. mit einer einzigen Blending-Konfiguration.

Zunächst soll dafür vormultipliziertes Alpha-Blending konfiguriert werden. Geben Sie die OpenGL-Aufrufe zum Setzen der richtigen Konfiguration an! Sie können aus den folgenden Konstanten wählen und in den Aufgaben deren Nummern eintragen:

0	GL_ZERO	5	GL_ONE_MINUS_DST_ALPHA
1	GL_ONE	6	GL_SRC_COLOR
2	GL_SRC_ALPHA	7	GL_ONE_MINUS_SRC_COLOR
3	GL_ONE_MINUS_SRC_ALPHA	8	GL_DST_COLOR
4	GL_DST_ALPHA	9	GL_ONE_MINUS_DST_COLOR

A	GL_FUNC_ADD	C	GL_FUNC_MIN
B	GL_FUNC_SUBSTRACT	D	GL_FUNC_MAX

Vervollständigen Sie außerdem die folgenden Fragment Shader-Funktionen, die den RGBA-Wert des zu zeichnenden Fragments erhalten und den an das Blending weiterzugebenden RGBA-Wert bestimmen! **(9 Punkte)**

OpenGL-Aufrufe zum Setzen der Konfiguration:

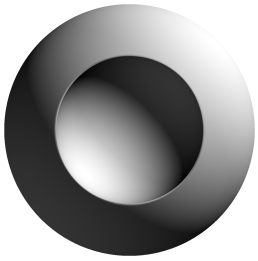
```
glBlendFunc( ____________________________, ____________________________)

glBlendEquation(____________________________);
```

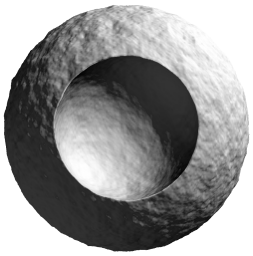
Fragment Shader-Funktionen:

```
vec4 blendingSourceColor_MethodA(vec4 src)
{
    return ____________________________________________;
}
vec4 blendingSourceColor_Additive(vec4 src)
{
    return ____________________________________________;
}
vec4 blendingSourceColor_WeightedAdditive(vec4 src)
{
    return ____________________________________________;
}
```

d) Ist das Ergebnisbild abhängig von der Sortierung der Partikel? Begründen Sie Ihre Antwort! **(3 Punkte)**

Aufgabe 8: Prozedurale Modellierung - Fragment Shader (22 Punkte)





Ein Objekt soll mithilfe von Distanzfunktionen in einem OpenGL-Fragment Shader mit Sphere Tracing gezeichnet werden. Das Objekt besteht aus einer Kugel, aus der eine kleinere Kugel ausgeschnitten ist (Abb. links). Zur Variation der Oberfläche wird eine Turbulenzfunktion auf die Distanzfunktion angewendet (Abb. rechts).

a) Implementieren Sie die Distanzfunktion `D_sphere` einer Kugel, die für einen Punkt $\boldsymbol{p}$ die Distanz zur Kugel mit Mittelpunkt $\boldsymbol{c}$ und Radius r angibt! **(3 Punkte)**

```
float D_sphere(vec3 p, vec3 c, float r)
{




}
```

b) Implementieren Sie die Distanzfunktion `D` des Objekts, das entsteht, wenn aus einer Kugel mit $\boldsymbol{c_0} = (0, 0, 0)$, $r_0 = 1$ eine andere Kugel mit $\boldsymbol{c_1} = (0, 0, 0.8)$, $r_1 = 0.5$ ausgeschnitten wird! **(4 Punkte)**

```
float D(vec3 p)
{




}
```

Matrikelnummer: _____

c) Gegeben ist eine Funktion `noise`, die einem Punkt $\boldsymbol{p}$ einen Rauschwert zuordnet. Implementieren Sie damit eine additive Turbulenzfunktion mit 4 Oktaven, in der sich sukzessive die Frequenz verdoppelt und die Amplitude halbiert! **(6 Punkte)**

```
float noise(vec3 p); // Rauschfunktion
float turbulence(vec3 p)
{




}
```

d) Erklären Sie kurz, wie Sie zu einem Punkt $\boldsymbol{p}$ auf der Oberfläche des Objekts, welches durch eine Distanzfunktion `D` beschrieben wird, die Oberflächennormale bestimmen können! **(4 Punkte)**

e) Vervollständigen Sie den folgenden Fragment Shader, der für jeden Pixel einmal aufgerufen wird und das Objekt mithilfe von Sphere Tracing zeichnet! Alle benötigten Funktionen und Konstanten sind aufgelistet. Falls nach `MAX_IT` Iterationen kein Oberflächenpunkt mit einer Distanz kleiner als `EPS` gefunden wurde, soll das Fragment verworfen werden. Andernfalls soll für den Punkt ein Farbwert mit der Funktion `shade` berechnet und in `color` geschrieben werden. **(10 Punkte)**

```
#define EPS 0.001 // Schwellwert für Schnittberechnung
#define MAX_IT 64 // Maximale Anzahl von Schritten

out vec4 color; // Ausgabe-Farbe für das Fragment

vec3 shade(vec3 p, vec3 n); // Shading mit Position und Normale
vec3 D(vec3 p);
vec3 D_n(vec3 p); // Berechnet Normale für Oberflächenpunkt p

... // weitere Funktionen

void main()
{
    vec3 p = ray_origin();    // Strahlursprung
    vec3 d = ray_direction(); // normalisierte Strahlrichtung









}
```

Matrikelnummer: _____

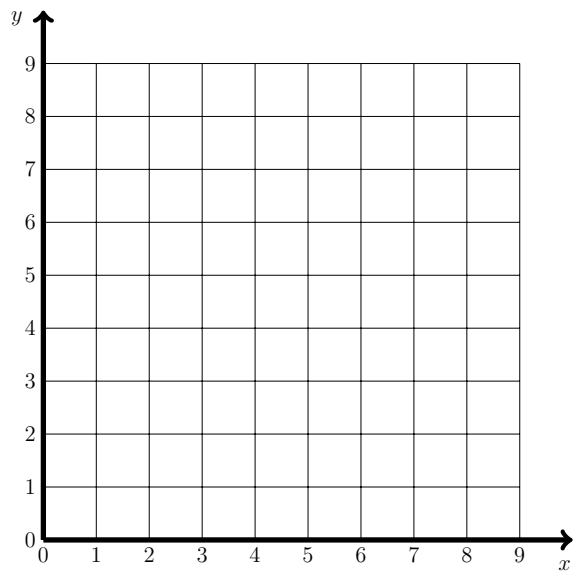
Aufgabe 9: Bézierkurven (31 Punkte)

a) Welche Eigenschaft von Bézierkurven macht man sich zunutze, wenn man eine affine Transformation M auf die Bézierkurve $\boldsymbol{b}(u) = \sum_{i=0}^{N} B_i^n(u)\boldsymbol{b}_i$ anwenden möchte? **(2 Punkte)**

b) Aufgrund welcher Eigenschaft von Bézierkurven kann der Punkt $\boldsymbol{x} = (8, 6)^T$ nicht Teil der kubischen Béziererkurve sein, die durch die folgenden 4 Kontrollpunkte definiert ist? **(3 Punkte)**

$$\boldsymbol{b}_0 = \begin{pmatrix} 3 \\ 1 \end{pmatrix},\ \boldsymbol{b}_1 = \begin{pmatrix} 8 \\ 4 \end{pmatrix},\ \boldsymbol{b}_2 = \begin{pmatrix} 6 \\ 8 \end{pmatrix},\ \boldsymbol{b}_3 = \begin{pmatrix} 2 \\ 7 \end{pmatrix}.$$

Hilfsskizze (wird nicht bewertet):



c) Die beiden kubischen Bézierkurven $\boldsymbol{b}(u)$ und $\boldsymbol{d}(u)$ sind durch folgende Kontrollpunkte definiert und in der Abbildung eingezeichnet:

$$\boldsymbol{b}_0 = \begin{pmatrix}1\\1\end{pmatrix},\ \boldsymbol{b}_1 = \begin{pmatrix}2\\5\end{pmatrix},\ \boldsymbol{b}_2 = \begin{pmatrix}3\\6\end{pmatrix},\ \boldsymbol{b}_3 = \begin{pmatrix}4\\4\end{pmatrix},$$

$$\boldsymbol{d}_0 = \begin{pmatrix}9\\4\end{pmatrix},\ \boldsymbol{d}_1 = \begin{pmatrix}10\\3\end{pmatrix},\ \boldsymbol{d}_2 = \begin{pmatrix}11\\3\end{pmatrix},\ \boldsymbol{d}_3 = \begin{pmatrix}12\\7\end{pmatrix}.$$

Die beiden Bézierkurven sollen C^1-stetig durch eine dritte Bézierkurve $\boldsymbol{c}(u)$ mit den Kontrollpunkten $\boldsymbol{c}_i$ verbunden werden. Zeichnen Sie eine solche Bézierkurve sowie ihre Kontrollpunkte $\boldsymbol{c}_1$ und $\boldsymbol{c}_2$ in die Abbildung ein! **(6 Punkte)**.