

Computergrafik

Fachrichtung Informatik

Dr. rer. nat. Johannes Riesterer

TINF14

Formalien

Die Klausur dauert 90 Minuten. Als Hilfsmittel sind Stift und Taschenrechner zugelassen.

Vermerken Sie bitte auf jedem Ihrer Blätter Ihre Matrikelnummer.

Aufgabe 1.**(4 Punkte)**

a) Gegeben sei die affine Basis (P, B) mit

$$B := \left\{ \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \right\}$$

und $P := \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, sowie der Punkt $Q := \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$. Berechnen Sie die Darstellung $\theta_{(P,B)}(Q)$ von Q bezüglich (P, B) .

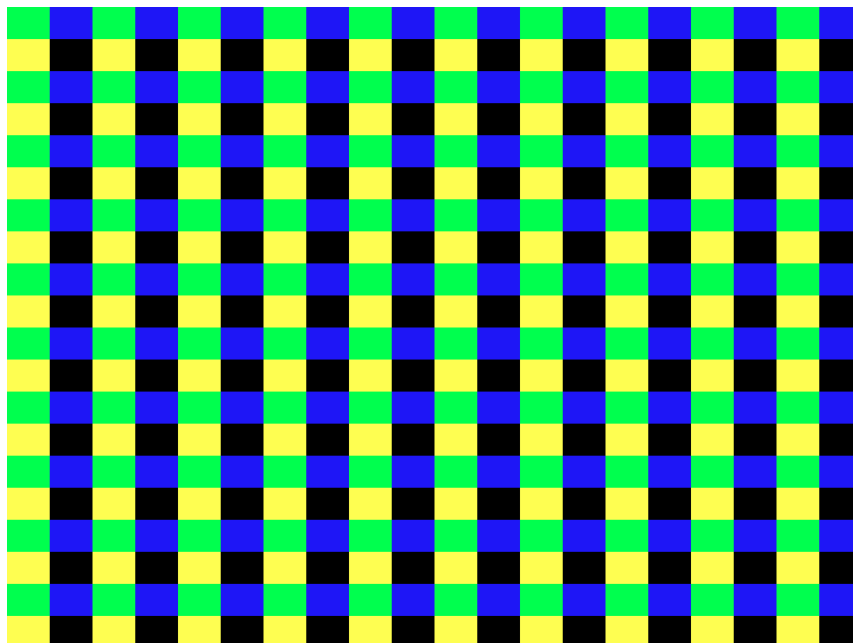
a) Berechnen Sie die Zentralprojektion des Punktes $Z := \begin{pmatrix} 4 \\ 0 \\ 4 \end{pmatrix}$ auf die Ebene parallel zur $X - Y$ -Ebene mit Augenabstand 2.

Aufgabe 2.**(4 Punkte)**

Erklären Sie schematisch die Funktionsweise eines GLSL Shaderprogrammes. Erläutern Sie dabei, an welcher Stelle der Graphikpipeline das Programm abläuft und was die Begriffe Clipping-Koordinaten, attribute, uniform und varying bedeuten.

Aufgabe 3.**(4 Punkte)**

Erweitern Sie das gegebene GLSL Shaderprogramm so, dass folgendes Bild ausgegeben wird:



aPosition ist dabei an einen Buffer gebunden, der zwei Dreiecke enthält die zusammen ein Einheitsquadrat bilden. Die Uniforms resx und resy sind auf die Grösse des GL-Fensters gesetzt.

Hinweis:

Die in GLSL eingebaute Funktion $\text{mod}(\text{float } x, \text{float } y)$ liefert als Wert die ganze Zahl x modulo y zurück. Sie wird durch $x - y * \text{floor}(x/y)$ berechnet. Die Funktion $\text{floor}(z)$ liefert hierbei die kleinste ganze Zahl, die kleiner oder gleich z ist.

Die in GLSL eingebaute Variable `gl_FragCoord` ist ein Vector, dessen Komponenten die Position im Framebuffer beinhalten. Die Farbe Gelb entsteht durch Addition von Rot und Grün. Vertex-Shader:

```

1 attribute vec2 aPosition;
2
3 uniform float resx;
4 varying float ResX;
5
6 uniform float resy;
7 varying float ResY;
8
9 void main() {
10     ResX = resx;
11     ResY = resy;
12     gl_Position = vec4(aPosition, 0.0, 1.0);
13 }

```

Fragment-Shader:

```

1 precision mediump float;
2
3 varying float ResX;
4 varying float ResY;
5
6 void main() {
7     gl_FragColor = vec4(0.0, 0.0, 0.0, 1.0);
8 }

```

Aufgabe 4.

(4 Punkte)

a) Gegeben ist ein Oberflächenpunkt $(0, 0, 0)^t$ mit Normale $(0, 1, 0)^t$, eine Lichtquelle im Punkt $(-1, 1, 0)^t$ und der Augenpunkt in $(1, 0, 0)$. Berechnen Sie die Helligkeit des Oberflächenpunktes nach dem Phongischen Beleuchtungsmodell mit dem Hardness-Faktor $h = 2$ und dem Specularity-Faktor $s = \frac{1}{4}$. Setzen Sie den Ambiente Factor $I_a = 0$ sowie alle anderen Material und Licht-Faktoren auf $I_l = I_m = 1$.

Hinweis:

Die Reflektion r des Vektors l an dem Vektor n ist gegeben durch $r = -l + 2 \langle n, l \rangle n$.

b) Erklären Sie die Funktionsweise des Shadowmap-Algorithmus zur Generierung von Verdeckungsschatten.

Aufgabe 5.

(4 Punkte)

a) Erläutern Sie die Definition der Orientierbarkeit eines Netzes.

b) Gegeben sind die Punkte $b_0 := \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$, $b_1 := \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$, $b_2 := \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix}$ so wie die Bezierkurve $B(t) := \sum_{i=0}^2 B_i^2(t) \cdot b_i$. Berechnen Sie mit Hilfe des Algorithmus von de Casteljau $B(\frac{1}{2})$.