

Gedächtnisprotokoll

Interaktive Computergrafik

24. Februar 2021

Dieses Gedächtnisprotokoll basiert auf dem Ersttermin des Moduls Interaktive Computergrafik aus dem Wintersemester 20/21 von Frank Steinicke. Die Klausur fand als open-book Take-Home-Exam statt und die Bearbeitung erfolgte im Rahmen eines Moodle Tests. Es gab randomisierte Aufgaben, das heißt jeder hatte eine andere Zusammenstellung von Fragen. Es waren alle Materialien aus den Vorlesungen und Übungen als Hilfsmittel erlaubt. Es gab insgesamt 60 Punkte zu erreichen, wir hatten 70 Minuten Zeit (60min Bearbeitungszeit + 10min Puffer für technische Probleme).

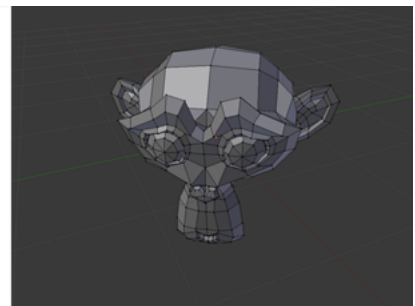
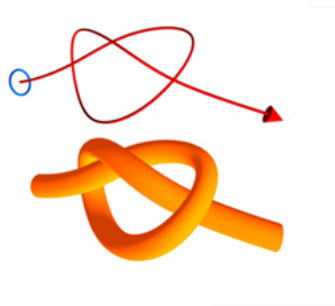
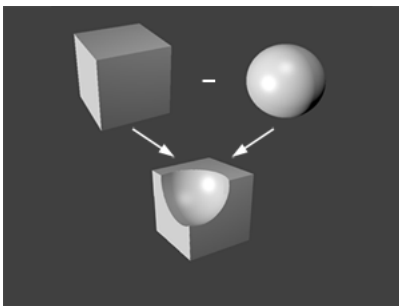
1 Frage

Wie viel Speicher in Bytes benötigt ein untexturiertes 3D-Objekt aus 140 Vertices und 59 Polygonen (Dreiecken) in indizierter Darstellung?

Gehen Sie davon aus, dass jeder Vertex Position und Normale hat, deren einzelne Komponenten jeweils 4 Bytes benötigen. Nehmen Sie weiter an, dass ein Index je zwei Byte benötigt. (**2 Punkte**)

2 Frage

Ordnen Sie jeder Abbildung die zugehörige Objekt-Darstellungsform zu. (**2 Punkte**)



- Constructive Solid Geometry (CSG)
- Boundary Representation (B-Rep)
- Extrusionskörper

3 Frage

Berechnen Sie folgende Transformation mit dem gegebenen Vektor und der gegebenen Matrix. (**3 Punkte**)

$$\begin{pmatrix} 9 & 8 & 7 \\ 4 & 6 & 4 \\ 2 & 3 & 5 \end{pmatrix} \cdot \begin{pmatrix} 5 \\ 5 \\ 3 \end{pmatrix}$$

4 Frage

Implementieren Sie den De Casteljau Algorithmus. (4 Punkte)

```
/*
Lineare Interpolation zwischen zwei 2-dimensionalen Punkten
@param [x0, y0] Der erste Punkt zur Interpolation
@param [x1, y1] Der zweite Punkt zur Interpolation
@param t Der Interpolationsfaktor, dieser darf nur zwischen 0 und 1 liegen
@return Gibt den interpolierten 2-dimensionalen Punkt zurueck,
*/
function lerp([x0, y0], [x1, y1], t){
    cont s = 1 - t;
    return [x0 * s + x1 * t, y0 * s + y1 * t];
}

/*
Berechnung eines Punktes auf einer Bezierkurve
@param controllPoints Ein Array aus mindestens zwei 2-dimensionalen Punkten
mit ihren jeweiligen xy-Koordinaten. Bspw.: [[1,2],[3,4],[5,6],[7,8]]
@param t Der Interpolationsfaktor zwischen 0 und 1.
@return Gibt den 2-dimensionalen Punkt auf der Bezierkurve zurueck.
*/
function DeCasteljau(controllPoints, t){
    return controllPoint[0];
}
```

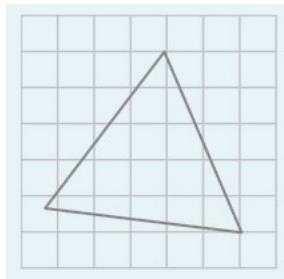
5 Frage

In welcher Reihenfolge gibt der Painters-Algorithmus Polygone auf dem Bildschirm aus? (2 Punkte)

- ☐ Von links nach rechts
- ☐ In keiner spezifischen
- ☐ Von rechts nach links
- ☐ Von vorne nach hinten
- ☐ Von hinten nach vorne

6 Frage

Erklären Sie anhand des im Folgenden dargestellten Polygons die 3 Hauptschritte des Scanline-Algorithmus zur Rasterisierung von Polygonen. (3 Punkte)



7 Frage

Durchlaufen Fragmente, welche aufgrund ihrer Einträge im z-Buffer von anderen Fragmenten überdeckt werden, den Fragment-Shader?

- ☐ Dies ist von WebGL gesteuert und implementationsabhängig
- ☐ Nein, das Fragment wird vorher verworfen
- ☐ Ja, dies ist so durch den Standard spezifiziert
- ☐ Nur nur bei Flat Shading
- ☐ Dies hängt von anderen Werten ab

8 Frage

Erläutern Sie kurz, wie der Bresenham Algorithmus verwendet werden kann, um die Kontur einer 3-dimensionalen Kugel zu rasterisieren. Beschreiben Sie dazu kurz, wie der Bresenham Algorithmus angewendet oder modifiziert werden muss, um die gewünschte Kontur zu erhalten. Geben Sie dazu auch die wesentlichen Schritte des Algorithmus kurz in Stichworten an. (4 Punkte)

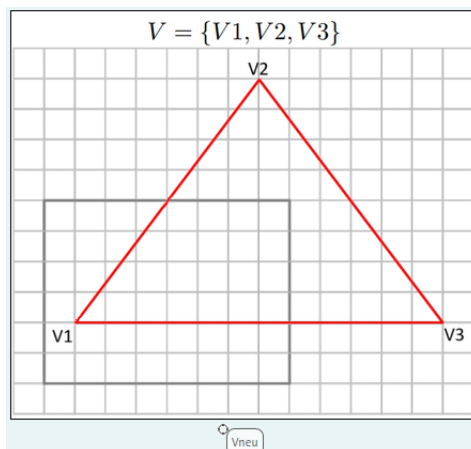
9 Frage

Gegeben sei ein Würfel, welcher von einer Kamera mit einem horizontalen und vertikalen Field of View von 60° aufgenommen wird. Wie verändert sich das Kamerabild, wenn sich das Field of View bei gleicher Distanz zwischen Kamera und Würfel auf 30° verringert? (2 Punkte)

- ☐ Der Würfel erscheint entfernter
- ☐ Das Kamerabild verändert sich nicht
- ☐ Der Würfel wird kleiner
- ☐ Der Würfel erscheint näher

10 Frage

Nutzen Sie den Sutherland-Hodgman-Algorithmus, um das Dreieck in der Abbildung an der **oberen** Kante des **grau** eingezeichneten Bereichs zu clippen. Platzieren Sie dazu die korrekte Anzahl an neuen Vertices (V_{neu}) an den entsprechenden Positionen im Bild. (2 Punkte)



11 Frage

Prüfen Sie durch eine Berechnung ob die Flächennormale $[\frac{1}{3} \quad \frac{2}{3} \quad \frac{2}{3}]$ von einer Kamera mit der Blickrichtung $[\frac{2}{3} \quad -\frac{2}{3} \quad -\frac{1}{3}]$ verworfen wird.

Geben Sie als Antwort **Wahr** an, wenn die Fläche durch Back Face Culling entfernt wird, andernfalls **Falsch**. (2 Punkte)

12 Frage

Erklären Sie das Prinzip von Frustum Culling. Fassen Sie außerdem kurz zusammen, welche mathematischen Operationen (konzeptionell) notwendig wären, um Frustum Culling umzusetzen. (3 Punkte)

13 Frage

Kreuzen Sie an, in welchem Raum das Phong-Beleuchtungsmodell **nicht** berechnet werden kann. (2 Punkte)

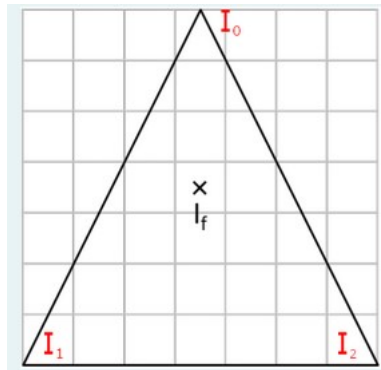
- ☐ Clip-Koordinaten
- ☐ Welt-Koordinaten
- ☐ Sicht-Koordinaten
- ☐ Model-Koordinaten

14 Frage

Gegeben sei das folgende Dreieck mit den folgenden Intensitäten für die drei Eckpunkte:

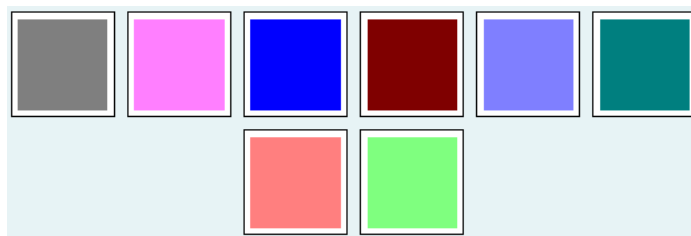
$$I_0 = (0.4, 0.1, 0.2), I_1 = (0.2, 0.6, 0.1), I_2 = (0.5, 0.3, 0.6)$$

Berechnen Sie die Intensitäten des Fragmentes I_f unter Flat Shading (2 Punkte)



15 Frage

Geben Sie an, welcher normalisierten Farbe im RGB-Farbraum die Normale (1,0,0) im Tangentenraum entspricht. (2 Punkte)

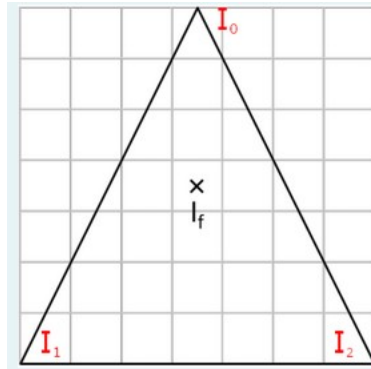


16 Frage

Gegeben sei das folgende Dreieck mit den folgenden Intensitäten für die drei Eckpunkte:

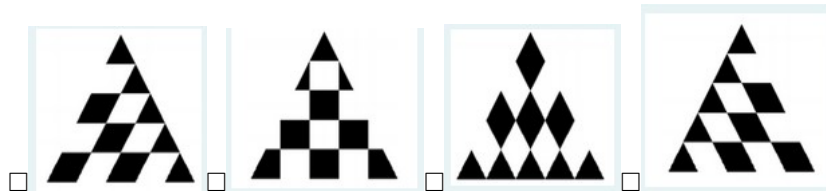
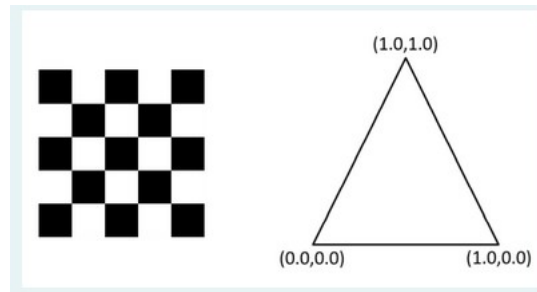
$$I_0 = (0.0, 0.1, 0.5), I_1 = (0.5, 0.0, 0.7), I_2 = (0.4, 0.5, 0.8)$$

Berechnen Sie die Intensitäten des Fragmentes I_f unter Gouraud Shading (3 Punkte)



17 Frage

Welche Texturierung ergibt sich, wenn die links dargestellte Textur auf ein Dreieck mit den rechts vorgegebenen (s,t)-Texturkoordinaten gemappt wird? (3 Punkte)



18 Frage

Ordnen Sie die aufgeführten Schritte des Shadow-Mapping Algorithmus in die korrekte Reihenfolge. (2 Punkte)

- Rendere Tiefenbild aus Sicht der Lichtquelle in eine Textur
- Schattiere Fragment, wenn der gespeicherte Wert kleiner als der Abstand ist
- Rendere Bild aus Sicht der Kamera. Vergleiche dabei Abstand von Fragment zur Lichtquelle mit dem in der Shadow-Map gespeicherten Wert.
- Normiere Tiefenwerte auf 0 bis 1

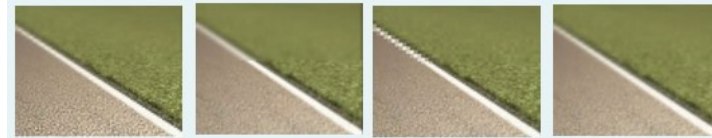
19 Frage

Beschreiben Sie in eigenen Worten kurz, wie es zu Moiré Mustern kommt und wie diese verhindert werden können. (3 Punkte)

20 Frage

Ordnen Sie die entsprechenden Texture Sampling Methoden den jeweiligen Bilder zu. (4 Punkte)

Hinweis: Achten Sie insbesondere auf den Übergang von Straße zur Grünfläche, sowie die Straße selbst



- Bilinear Filtering
- Nearest Neighbor Filtering
- Anisotropic Filtering
- Trilinear Filtering

21 Frage

Was sagt der Qualifier "in" über eine Variable in einem WebGL Vertex-Shader aus? (2 Punkte)

- ☐ Dass die Variable vom Vertex- zum Fragment-Shader übergeben und dabei interpoliert wird
- ☐ Dass die Variable konstant ist
- ☐ Dass die Variable einem Vertex Buffer Objekt entnommen wird
- ☐ Dass die Variable über alle Vertices und Fragmente konstant bleibt

22 Frage

Beschriften Sie folgendes Schaubild der **vereinfachten** WebGL-Rendering-Pipeline mit den richtigen Begriffen und beschreiben Sie anhand der drei Stufen kurz den schematischen Ablauf eines WebGL-Programms. (2 Punkte)

Anwendungsprogramm $\rightarrow (a) \rightarrow (b) \rightarrow (c) \rightarrow$ Framebuffer

- Vertex-Verarbeitung
- Model-Transformation
- Clipping
- Fragment-Verarbeitung
- Rasterisierung
- Texturierung
- Beleuchtung
- Tessellation

23 Frage

Schreiben Sie einen Fragment Shader, welcher die Farbe aller Fragmente auf **Weiß** setzt. (2 Punkte)

```
precision mediump float;
```

```
in vec4 vfColor;  
out vec4 fColor;
```

```
void main(){  
  
}
```

24 Frage

Schreiben Sie einen Fragment Shader, welcher den Grauwert für alle Fragmente bestimmt. (3 Punkte)

```
precision mediump float;
```

```
in vec4 vfColor;  
out vec4 fColor;
```

```
void main(){  
  
}
```