Übung zu Computergraphik I

– Übungsblatt 13 –

Lehrstuhl für Computergraphik und Multimediasysteme

Andreas Görlitz, Jan Mußmann

Abgabe: Bis spätestens Dienstag 22. Januar 2019, 10 Uhr
Besprechung: Dienstag 29. Januar 2019 und Mittwoch 30. Januar 2019

Hinweise: Schriftliche Aufgaben bitte mit Name, Matrikelnummer und Übungsgruppe beschriften und zusammengeheftet in den Briefkasten vor Büro H-A 7107 werfen. Die Programmieraufgaben müssen per E-Mail **an Jan Mußmann** eingereicht werden. Geben Sie dabei bitte immer Ihren **Namen**, Ihre **Matrikelnummer**, sowie Ihre **Übungsgruppe (Di. / Mi.)** an. Geben Sie nur die von Ihnen **geänderten Dateien** ab.

Aufgabe 1 Kay-Kajiya (3 Punkte)

Für das Raytracing-Verfahren ist der Schnitt eines Strahls mit einer Geometrie ein wesentlicher Aspekt. Zur Beschleunigung wird im Kay-Kayjia-Ansatz der Schnitt mit einer achsenparallelen, die Geometrie umfassende Bounding-Box gebildet. Wird die Bounding-Box vom Strahl nicht geschnitten, so ist ein weiterer Test nicht notwendig.

Gegeben seien eine Bounding-Box $[x_{\text{min}}, x_{\text{max}}] \times [y_{\text{min}}, y_{\text{max}}] \times [z_{\text{min}}, z_{\text{max}}]$ und ein Strahl g. Der Strahl g führt durch den Anfangspunkt $\mathbf{V} = (0, 2, -4)^{\mathsf{T}}$ mit Richtung $\vec{\mathbf{r}} = (0, 1, -5)^{\mathsf{T}}$. Die Bounding-Box ist durch folgende Grenzen definiert:

$$\begin{aligned} x_{\text{min}} &= -1, \quad x_{\text{max}} = 2 \\ y_{\text{min}} &= -1, \quad y_{\text{max}} = 2 \\ z_{\text{min}} &= -1, \quad z_{\text{max}} = 3 \end{aligned}$$

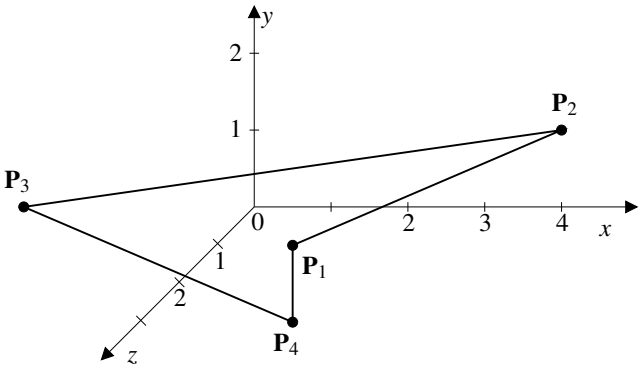
Führen Sie einen Schnitt zwischen Gerade und Bounding-Box durch, indem Sie nacheinander – wie in der Vorlesung – die drei Koordinatenachsen x, y, z betrachten.

Aufgabe 2 Point-in-Polygon-Test (4 Punkte)

Im Raycasting-Verfahren wird für jeden Bildpunkt ein Strahl mit den Objekten der Szene geschnitten. Im Folgenden soll ein Point-in-Polygon-Test durchgeführt werden, um zu prüfen, ob ein Strahl ein Polygon mit vier Eckpunkten trifft.
Das Polygon sei definiert durch die folgenden vier Punkte, die in dieser Reihenfolge miteinander verbunden sind:

$$\mathbf{P}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \mathbf{P}_2 = \begin{pmatrix} 4 \\ 1 \\ 0 \end{pmatrix}, \mathbf{P}_3 = \begin{pmatrix} -2 \\ 1 \\ 2 \end{pmatrix}, \mathbf{P}_4 = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$$

Wir betrachten einen Strahl g in z-Richtung, ausgehend vom Punkt $\mathbf{V} = (-2, 0, 0)^{\mathsf{T}}$.



2.1 Bestimmen Sie den Schnittpunkt $\mathbf{S}$ der Polygonebene mit dem Strahl g. Hinweis: Berechnen Sie hierfür den Normalenvektor $\hat{\mathbf{n}}$ der Polygonebene.

2.2 Projizieren Sie die Punkte $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4$ und den Schnittpunkt $\mathbf{S}$ entlang der treibenden Achse des Normalenvektors $\hat{\mathbf{n}}$.

2.3 Führen Sie eine Verschiebung der Punkte durch, so dass der Schnittpunkt $\mathbf{S}$ auf den Ursprungspunkt verschoben wird.

2.4 Zeichnen Sie die transformierten Punkte in eine 2D-Skizze ein, und prüfen Sie graphisch und rechnerisch, ob $\mathbf{S}$ im Polygon enthalten ist. Hinweis: Betrachten Sie die Schnittpunkte entlang der x-Achse.

2.5 Für welche Polygonkante(n) reicht es aus, eine Trivial-Reject-Prüfung durchzuführen?

Aufgabe 3 Raycasting (Bonusaufgabe 5 Punkte)

In dieser Aufgabe sollen Sie einen Raycaster erweitern. Laden Sie hierzu zunächst das Programmgerüst `ueb13.zip` von der Übungswebseite herunter. Im Programm wird bereits das Objekt erzeugt und es stehen folgende Steuerungsmöglichkeiten zur Verfügung:

1. Mit gedrückter linker Maustaste rotieren Sie das Objekt um seinen Mittelpunkt.
2. Mit gedrückter rechter Maustaste zoomen Sie in die Szene.
3. Mit gedrückter mittlerer Maustaste verschieben Sie Ihren Fokus auf die Szene.
4. Mittels Leertaste kann die Ansicht auf die Initialeinstellung zurückgesetzt werden.

Abbildung 1 zeigt einen Screenshot des vollständigen implementierten Raycasters.

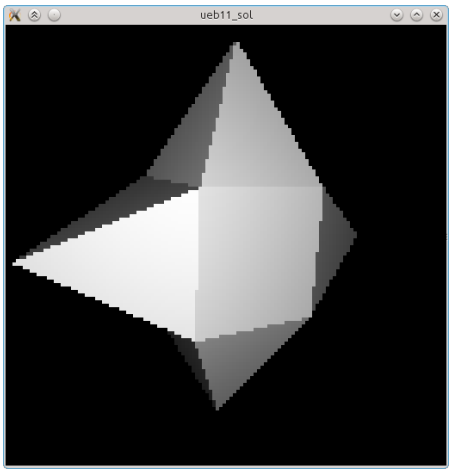


Abbildung 1: *Rendering eines sternenförmigen Objektes mittels Raycasting.*

Nur Strahlen, die ein Dreieck treffen, erzeugen einen sichtbaren Anteil im Bild. Hierzu wird für jeden Strahl die Methode `Raycaster::pointInPolygonTest()` aufgerufen. Implementieren Sie in dieser Funktion die Schritte für den Point-In-Polygon Test aus Aufgabe 2 auf Basis von Dreiecksprimitiven. Als Parameter bekommen Sie ein Dreieck `tri` vom Typ `Triangle` und den Schnittpunkt mit der Dreiecksebene `iPoint` vom Typ `vec3` übergeben. Geben Sie `true` im Falle dass der Schnittpunkt im Dreieck liegt oder andernfalls `false` zurück.

Sofern Sie alles richtig implementiert haben, sollte Ihr Ergebnis dem der Abbildung entsprechen. Im Urzustand bleibt das Fenster leider schwarz.