



Aufgabenblatt 6: Schatten

In diesem Aufgabenblatt implementieren Sie den Schatten-Volumen-Algorithmus.

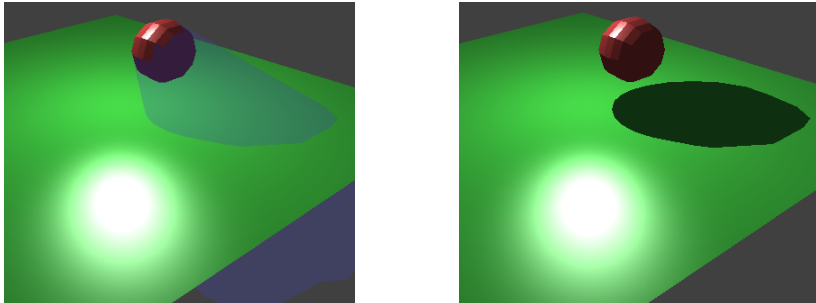


Abbildung 1: Rendering der Schattenvolumen (links) und Schattendarstellung (rechts) in einer Szene bestehend aus einer Kugel und einem Quadrat.

Aufgabe 5.1: Silhouetten-Kanten

Schwerpunkte: Finden der Silhouetten-Kanten in einem Dreiecksnetz

Aufgabe: Ergänzen Sie die Klasse `TriangleMesh` um eine Methode zum Finden aller Silhouettenkanten zu einer Lichtquellenposition. Möglicherweise ist es einfacher, wenn Sie die Implementierung der Halbkantendatenstruktur verwenden, weil dort die Kanten schon explizit repräsentiert sind. Achten Sie aber darauf, dass dort jede Kante durch zwei Halbkanten repräsentiert wird. Für die Silhouetten-Kanten dürfen die Kanten nicht doppelt verwendet werden. An jeder Kante liegen genau zwei Dreiecke an. Eine Silhouettenkante erkennen Sie, indem Sie für beide Dreiecke prüfen, ob die Lichtquelle in Normalenrichtung liegt oder nicht. Stellen Sie dazu für die beiden Dreiecksebenen die Hesse'sche Normalform auf und setzen Sie die Lichtquellenposition ein. Für eine Silhouettenkante ist einer der Werte positiv und einer negativ. Die Position der Lichtquelle können Sie im Wurzelknoten abfragen.

Aufgabe 5.2: Schattenpolygone erstellen

Schwerpunkte: Erzeugen von Schattenpolygonen

Aufgabe: Implementieren Sie nun die Methode

`void createShadowPolygons(Vector3 lightPosition, float extend, ITriangleMesh shadowPolygonMesh).`

Der erste Parameter ist die Position der Lichtquelle (`lightPosition`), der zweite Parameter gibt die Länge des Polygons an und der dritte Parameter ist ein `TriangleMesh`-Objekt, das sie mit Ihren Schattenpolygonen befüllen müssen. Für jede Kante der Silhouette müssen Sie ein Schattenpolygon, also zwei Dreiecke, erzeugen. Die äußeren Kanten jedes Schattenpolygons verlaufen in der gleichen Richtung wie der Vektor von der Lichtquellenposition zu den Eckpunkten der Kante. Achten Sie darauf, dass die Normalen der Schattenpolygone (ebenso wie bei den regulären Meshes) nach außen zeigen.

Aufgabe 5.3: Schattenpolygone zeichnen

Für das Zeichnen der Schattenpolygone finden Sie auf EMIL zusammen mit diesem Aufgabenblatt eine Version des Knotens `TriangleMeshNode` (sowohl für das Java als auch für das C#-Framework). Damit können sowohl reguläre Meshes gezeichnet werden als auch solche, die Schatten werfen.

Hinweise: Das Zeichnen der Schattenpolygone und damit verbunden das Zeichnen der Schatten ist bereits im Framework vorgegeben. Dazu gibt es drei `RenderModi`, die in der Klasse `INode` als Enum vorgegeben sind. Bisher haben wir den Modus „REGULAR“ verwendet. Zum Debuggen der Schattenpolygone dient der Modus „DEBUG_SHADOW_VOLUME“. Die finalen Schatten werden mit dem Modus „SHADOW_VOLUME“ gezeichnet. Zwischen den drei Modi können Sie wechseln indem Sie entweder den Modus in Ihrer

Anwendungsklasse setzen (Aufruf des Basisklassenkonstruktors von Scene) oder indem Sie zur Laufzeit mit Tastenkürzeln umschalten:

- „R“: reguläres Zeichnen
- „D“: Schattenvolumen (zum Debugging)
- „S“: Schatten