

Kapitel 9 Rendering

0901 Einleitung

Rendern stammt vom englischen Wort „to render“ und heißt zu deutsch „etwas ausgeben“. Das Rendering bezeichnet den Vorgang, ein Bild zu generieren. Beim Rendern einer 3D-Szene werden im Wesentlichen Meshes, Lichtquellen und Kameras berücksichtigt. Daraufhin erfolgt eine Projektion der Szene auf die Projektionsebene.

In der Rendering-Pipeline durchläuft ein Mesh mehrere Schritte, um am Ende als ein rasterisiertes Bild dargestellt werden zu können. Hierbei sind die wichtigsten Stationen die Umwandlung der Koordinaten des Meshes und die Rasterisierung.

Des Weiteren werden aber auch Sichtbarkeits- und Beleuchtungsberechnungen durchgeführt, Texturen gemappt und spezielle Effekte dargestellt.

0902 Modell-Transformation

Ein Mesh, wird normalerweise durch sein lokales Koordinatensystem definiert. Ohne Transformationen würden alle Objekte im Weltursprung liegen.

Durch Translationen, Rotationen und Skalierungen wird ein Mesh an die gewünschte Stelle im Raum gebracht.

Die endgültige Position eines Meshes hängt von der Reihenfolge der Transformationen in der Szenenhierarchie ab.

Um diese Transformationsverkettung aufzulösen werden die Vertexkoordinaten des Meshes in das Weltkoordinatensystem übertragen.

Von nun an liegen die Koordinaten nicht mehr als transformierte lokale Koordinaten, sondern in Weltkoordinaten vor.

0903 Kamera-Transformation

Bei der Kamera-Transformation, die auf Englisch viewing transformation genannt wird, werden die Kamera und alle Meshes so transformiert, dass die Kamera im Ursprung liegt.

Die Ausrichtung der Kamera verläuft nach der Transformation entlang der z-Achse.

Nach diesem Schritt liegen die Weltkoordinaten des Meshes nun als Kamerakoordinaten vor.

0904 Projektions-Transformation

Nach der Modell- und der Kamera-Transformation liegen alle Koordinaten der Meshes so vor, dass sie gut weiterverarbeitet werden können.

Bei der Projektions-Transformation wird das View Frustum näher betrachtet. Bei der perspektivische Projektion hat das Frustum die Form eines Pyramidenstumpfes und besitzt eine Far Clipping Plane und eine Near Clipping Plane, die auch Projektionsebene genannt wird.

Meshes außerhalb des Frustums werden nicht dargestellt.

Als nächstes wird das Frustum mitsamt der Inhalte derart transformiert, dass ein Quader entsteht.

Nach diesem Schritt liegen die Koordinaten im sogenannten Clip-Space vor.

Die Meshes sind nun perspektivisch verzerrt. Bei der orthografischen Projektion werden die Meshes nicht verzerrt, da das Kamera-Frustum von Beginn an ein Quader ist.

0905 Projektions-Transformation– Interaktion

Wähle zwischen perspektivischer und orthografischer Projektion aus und betrachte die Änderungen.

0906 Clipping

Beim Clipping geht es darum Flächen, die vom sichtbaren Volumen nicht mehr eingefangen werden können,
aus der Szene zu entfernen.

Nach der Projektionstransformation wird überprüft, welche Meshes vollständig im sichtbarem Bereich liegen.

Meshes, die teilweise im Frustum liegen, werden geschnitten und in außen- und innenliegende Meshes aufgeteilt.

Meshes die gänzlich außerhalb des Frustums liegen werden komplett entfernt.

0907 Backface-Culling

Beim Culling geht es darum Flächen, die vom Betrachter nicht wahrgenommen werden können aus der Szene zu entfernen.

Durch das Backface-Culling werden die Polygone aus der Szene entfernt, die vom Betrachter abgewandt sind.

Ob eine Fläche sichtbar oder nicht sichtbar ist wird mit Hilfe des Normalenvektors entschieden.

Ein Normalenvektor ist ein Vektor, der senkrecht auf der zugehörigen Fläche steht.

Zeigt der Normalenvektor zum Beispiel in Richtung der Kamera, hat es zur Folge, dass der Betrachter die Vorderseite sieht. Ist der Normalenvektor n von der Kamera abgewandt heißt das, dass es sich bei der Fläche um eine Rückseite eines Meshes handelt.

Damit kann diese Fläche entfernt werden.

0908 Rasterisierung

Zunächst wird im Rasterisierungsschritt die Projektionsebene gerastert.

Die Meshes im Frustum werden nun auf diesem Raster dargestellt.

Die Rasterisierung der Projektionsebene liefert allerdings nicht unbedingt die endgültigen Pixel des fertigen Bildes.

Das liegt daran, dass noch Transparenzen und mögliche Verdeckungen berechnet werden müssen.

Deswegen werden die Rasterelemente während dem Rasterisierungsschritt auch Fragmente genannt.

0909 Verdeckungsrechnung – z-Buffer

Bei einer Szene in der mehrere Meshes zu sehen sind, kann es dazu kommen, dass ein Mesh von einem anderen verdeckt wird, oder dieses auch schneidet.

Um dies korrekt darzustellen, werden in der Computergrafik Verdeckungsrechnungen, wie z. B. der z-Buffer-Algorithmus angewandt.

Die Grundidee des z-Buffer-Algorithmus ist es für jeden Pixel die Tiefeninformation bzw. den z-Wert zu speichern.

Es muss geprüft werden ob ein Pixel näher an der Kamera liegt als ein vorher berechneter. Dazu muss der z-Wert kleiner sein.

Falls dies so ist, werden Farbwerte und z-Buffer für den Pixel überschrieben , andernfalls werden die alten Werte beibehalten.

0910 Raytracing

Beim realistischen Rendern liegt das Hauptaugenmerk auf der physikalischen Korrektheit der Darstellung des gerenderten Bildes.

Dafür sollte jedoch eine höhere Rechenzeiten in Kauf genommen werden.

Raytracing – zu Deutsch „Strahlen verfolgen“ – ist in erster Linie ein Algorithmus zu Verdeckungsrechnung.

Dieser basiert auf dem Aussenden von Strahlen vom Betrachterblickpunkt. Des Weiteren ist eine Bildebene vorhanden, die in Pixel unterteilt ist und dem später gerenderten Rasterbild entspricht.

Für jeden Pixel wird ein Strahl ausgesandt. Diese werden Primärstrahl genannt.

Die Primärstrahlen ermitteln Schnittpunkte mit Flächen.

Anschließend werden die gebrochenen bzw. die reflektierten Strahlen berechnet. Diese werden als Sekundärstrahlen bezeichnet.

Der Strahl endet, wenn er die maximale Anzahl von Schritten erreicht, auf kein weiteres Objekt oder auf eine Lichtquelle trifft.

Das Raytracing bringt den Vorteil, dass korrekte Objektspiegelungen und Schatten dargestellt werden können.

0911 Raytracing – Interaktion

Wähle zwischen reinem Shading und Raytracing aus und betrachte die Änderungen.

0912 Volumengrafik

Volumengrafiken sind in der Lage transparente Objekte und Objekte ohne scharfe Abgrenzungen, wie z. B. Wolken, zu modellieren. Diese bestehen aus Voxeln. Voxel bezeichnet einen Gitterpunkt in einem dreidimensionalen Gitter. Dies entspricht einem Pixel in einem 2D-Bild, einer Rastergrafik. Der Begriff Voxel ist eine Analogie zum Pixel leitet sich aus den Begriffen „volume“ und „element“ ab.

Die Volumengrafik basiert auf dem Strahlentransport, der beschreibt, wie sich Licht auf dem Weg durch ein Volumen verhält.

Beim Rendern einer Volumengrafik unterscheidet man vier Schritte:

1. der Klassifikation: Hier werden den Voxeln Materialeigenschaften gegeben.

Bei der Erzeugung des Voxels besitzt dieser zunächst nur eine Eigenschaft. Weitere müssen bei der Klassifikation vom Benutzer vorgegeben werden.

Eine Eigenschaft könnte zum Beispiel sein, wie sehr das Voxel spiegeln soll.

2. der Interpolation: Da es sich bei Voxeln um Punkte handelt, ist es unwahrscheinlich, dass sie von einem Strahl getroffen werden. Deswegen werden die Materialeigenschaften an Punkten zwischen den Voxeln aus benachbarten Voxeln angenähert.

3. dem Shading: Beim Shading wird bestimmt, wie viel Licht von einem Voxel aus in Richtung des Betrachters ausgeht und welche Farbe es hat.

4. der Composition: Beim Durchqueren des Lichts durch Voxel ändert sich die Farbe und die Intensität. Bis der Lichtstrahl auf die Bildebene fällt, kann dieser mehrere Voxel durchqueren. Die letzten Eigenschaften des Strahles, färben den Pixel auf der Bildebene.