

DREHBUCH RENDERING

Computergrafik.Online

Betreuer: Prof. Jirka Dell'Oro-Friedl
Wintersemester 2018/2019

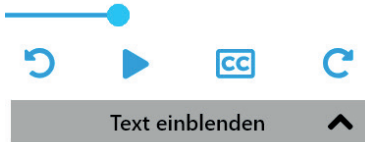
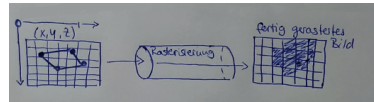
Hochschule Furtwangen University
Fakultät Digitale Medien

Version: 1.4
Letzte Änderung: 18.11.2018
Autor: Berdan Der



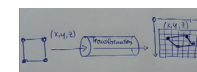
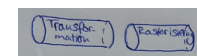
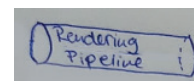
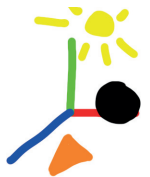
9.1 Einleitung/Anwendung	1
9.2 mathematische Grundlagen	2
9.3 Modell-Transformation	3
9.4 Augepunkt-Transformation	4
9.5 Projektions-Transformation	5
9.6 Clipping	6
9.7 Culling	7
9.8 Rasterisierung	8
9.9 Verdeckungsrechnung – z-Buffer	9
9.10 Raytracing	10
9.11 Radiosity	11
9.12 Raytracing/Radiosity - Interaktion	12
9.13 Volumengrafik	13

9.1 Einleitung/Anwendung

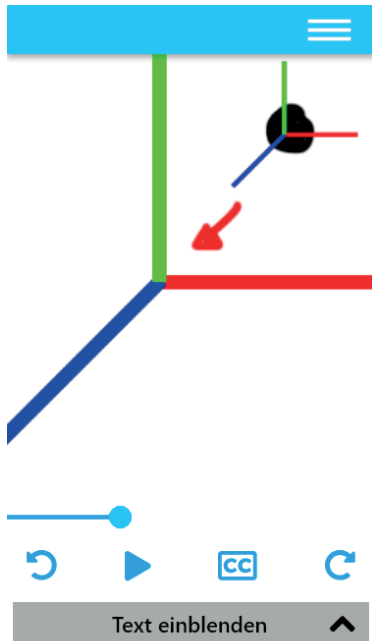


Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090101 Rendern stammt vom englischen Wort „to render“ und heißt zu deutsch etwas ausgeben. Das Rendering bezeichnet den Vorgang, ein Bild aus einer Szene zu generieren. Die Szene besteht dabei aus Mesh, Kameras und Lichtquellen.</p> <p>090102 In der Rendering-Pipeline durchläuft ein Mesh mehrere Schritte, um am Ende als ein rasterisiertes Bild dargestellt werden zu können. Hierbei sind die wichtigsten Stationen die Transformationen, bei der die Koordinaten des Meshs transformiert werden und die Rasterisierung, das dazu dient die Geometrien später auf Bildschirmen ausgeben zu können.</p> <p>090103 Des Weiteren werden aber auch nicht sichtbare Flächen entfernt und Beleuchtungsberechnungen dargestellt, um eine korrekte Beleuchtung darzustellen.</p>	<p>090101 - Rendern (dt. Bildsynthese) - Aus einer Szene wird ein Bild erzeugt</p> <p>090102 Prozess des Renderings in der Rendering Pipeline</p>	<p>090101 -Es erscheint eine Einblendung der Begrifflichkeit -Danach erscheint eine Szene mit Objekt, Kamera und Licht -Daraufhin wird das fertig gerenderte Bild angezeigt</p> <p>090102 -Es erscheint eine Rendering Pipeline, die in Transformation und Rasterisierung aufgeteilt wird. - Daraufhin läuft ein Mesh durch</p>

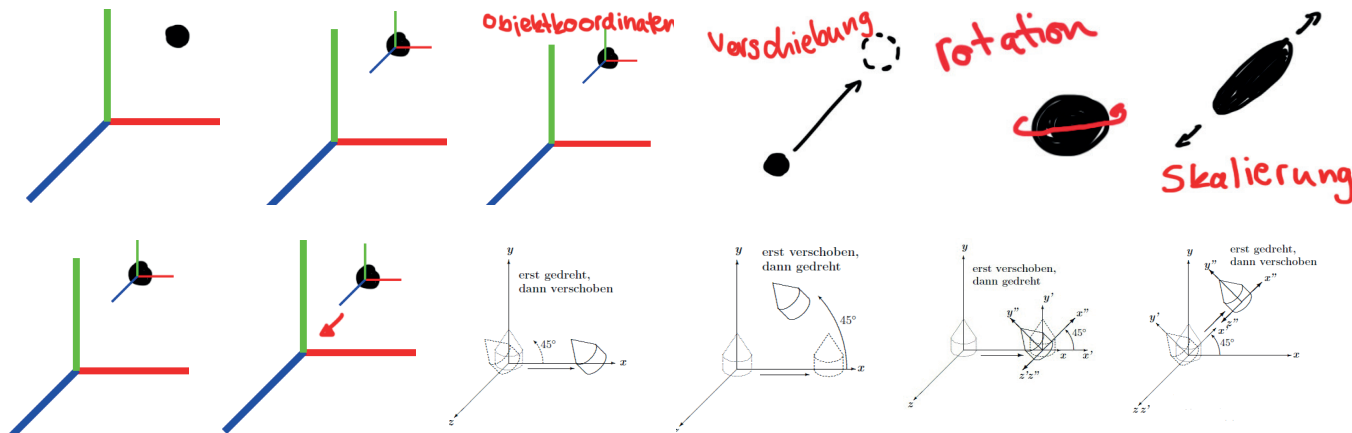
rendern
= Bildsynthese



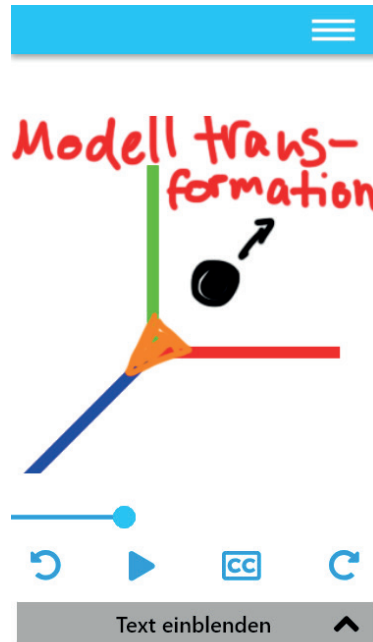
9.3 Modell Transformation



Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090301 Ein Objekt, das sich in einem dreidimensionalen Raum aufhält, wird normalerweise durch ihr lokales Koordinatensystem definiert. Dabei handelt es sich um die Objektkoordinaten, jedoch müssen sie in ein globales Koordinatensystem überführt werden.</p> <p>090302 Daher müssen Objekte durch eine Modell-Transformation an die richtige Stelle im Raum gebracht werden.</p> <p>090303 Dies wird durch Translationen, Rotationen und Skalierungen erreicht.</p> <p>090304 Eine vereinfachte Denkweise ist es, dass die lokalen Koordinatensysteme mit den Objekten gekoppelt sind. Nun werden nicht die Objekte im Koordinatensystem verschoben, sondern das Objekt mitsamt dem Koordinatensystem.</p> <p>090305 Die endgültige Position eines Objektes hängt von der Reihenfolge der Transformationen ab, und ob sich das Objekt auf lokale oder globale Koordinaten bezieht.</p> <p>090306 Wird ein Objekt, welches sich im Ursprung befindet und sich auf globale Koordinaten bezieht, um 45° gedreht und daraufhin verschoben, so befindet es sich später auf der x-Achse.</p> <p>090307 Verschiebt man es zuerst und dreht es dann um 45°, so befindet sich das Objekt auf der Winkelhalbierenden der xy-Ebene.</p> <p>090308 Betrachtet man jetzt ein Objekt, das sich auf ein lokales Koordinatensystem bezieht, so muss es erst verschoben und dann um 45° gedreht werden, dass es sich immer noch auf der x-Achse befindet.</p> <p>090309 Wird ein Objekt, das sich auf ein lokales Koordinatensystem bezieht, erst um 45° gedreht und dann verschoben, so befindet es sich auf der Winkelhalbierenden, der xy-Ebene.</p>	<p>090301 lokales Koordinatensystem = Objektkoordinaten globales Koordinatensystem = Weltkoordinaten</p> <p>090302 lok. KS \rightarrow glob. KS durch Modell-Transformation</p> <p>090303 Verschiebung (Translation) Drehung (Rotation) Vergrößerung bzw. Verkleinerung (Skalierung)</p>	<p>090301 -Es erscheint ein Koordinatensystem mit einem Objekt -Auf dem Objekt erscheint ein Objektkoordinatensystem</p> <p>090303 -Das Objekt wird verschiedenen Transformationen unterzogen</p> <p>090304 -Auf dem Objekt erscheint wieder ein Objektkoordinatensystem -das Objekt wird mitsamt Koordinatensystem verschoben</p> <p>090306 Ein Objekt erscheint im gl. Koordinatensystem und wird zuerst gedreht und dann verschoben</p> <p>090307 Ein Objekt erscheint im gl. Koordinatensystem und wird zuerst verschoben und dann gedreht</p> <p>090308 Ein Objekt erscheint im lok. Koordinatensystem und wird zuerst verschoben und dann gedreht</p> <p>090309 Ein Objekt erscheint im lok. Koordinatensystem und wird zuerst gedreht und dann verschoben</p>

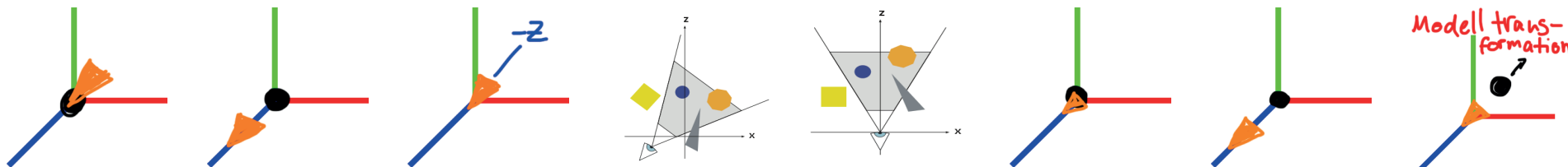


9.4 Kameratransformation

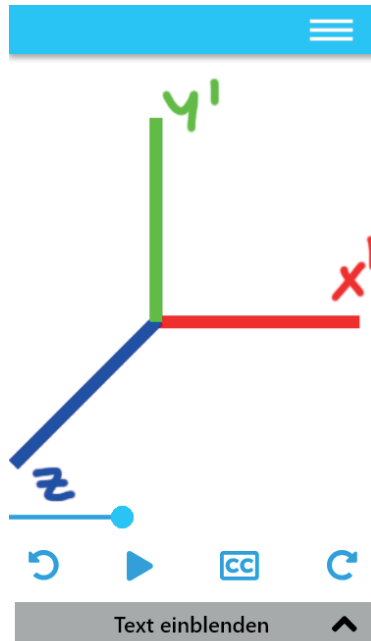


Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090401 Bei der Kameratransformation, die auf Englisch viewing transformation genannt wird, ändert man die Position und die Blickrichtung der Kamera, sodass sie auf die Objekte gerichtet ist, die später als Bild generiert werden sollen.</p> <p>090402 Am Anfang befindet sich die Kamera normalerweise im Ursprung 0,0,0. Die Blickrichtung entspricht der negativen z-Achse.</p> <p>090403 Jedoch kann es auch vorkommen dass eine Kamera mitten in der Szene angebracht ist. Um spätere Berechnungen zu vereinfachen, wird die Szene so transformiert, dass sich die Kamera im Ursprung befindet.</p> <p>090404 Liegen zum Beispiel sowohl das Objekt als auch die Kamera im Koordinatenursprung, so muss entweder die Kamera entlang der positiven z-Achse oder das Objekt entlang der negativen z-Achse verschoben werden. Zweiteres wäre eine Modell-Transformation.</p> <p>090405 Somit macht es für den Betrachter keinen Unterschied, da beide Transformationen zueinander äquivalent sind.</p>	<p>090401 Augenpunkt-Transformation = (engl.) Viewing Transformation</p> <p>Veränderung der Position und Blickrichtung der Kamera</p> <p>090405 Modell-Transformation äquivalent zu Augenpunkt-Transformation</p>	<p>090401 -Es erscheint eine Einblendung der Begrifflichkeit -Es erscheint ein Koordinatensystem mit Kamer und Objekt -Daraufhin wird die Kamera auf das Objekt gerichtet</p> <p>090402 -Es erscheint ein Koordinatensystem mit einer Kamera im Ursprung und einer negativen z-Achse</p> <p>090403 Es erscheint ein eszene mit einer willkürlich gesetzten Kamer und Objekten. diese werden dann so transformiert, dass die Kamera im Ursprung liegt.</p> <p>090404 -Ein Koordinatensystem mit Objekt und Kamer wird gezeigt, wobei Kamera und Objekt aufeinander liegen -Als erstes wird dieKamera verschoben</p> <p>090404/090405 -Zuletzt wird das Objekt verstezt.</p>

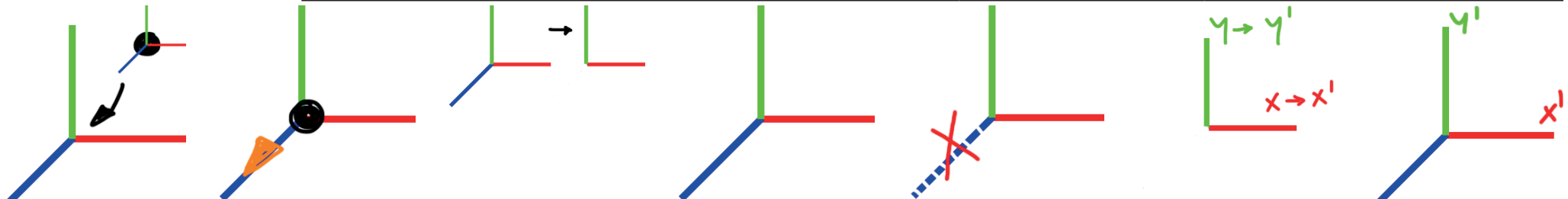
Augenpunkt-Transformation
=
Viewing Transformation



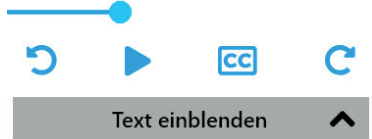
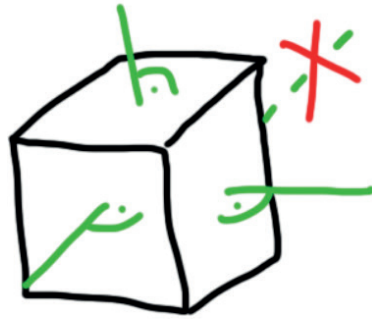
9.5 Projektions Transformation



Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090501 Nach der Modell- und der Viewing Transformation befinden sich alle Eckpunkte, welche auch Vertices genannt werden, an den gewünschten Positionen.</p>	<p>090501 Eckpunkt = Vertice</p>	<p>090501 Es erscheint eine Kurze Animation zu den vorherigen Transformationen</p>
<p>090502 Um aber ein zweidimensionales Abbild der dreidimensionalen Szene zu erhalten, muss eine Projektions-Transformation vollzogen werden.</p>	<p>090502 Projektions-Transformation = KS transformieren</p>	<p>090502 Ein dreidimensionales Koordinatensystem wird in ein zweidimensionales umgewandelt</p>
<p>090503 Um einen dreidimensionalen Punkt auf einer zweidimensionalen Fläche abbilden zu können, müsste die z-Dimension wegfallen. Des Weiteren muss die x-y-Ebene transformiert werden.</p>		<p>090503 Es erscheint wieder ein dreidimensionales KS bei welchem die z-Achse verschwindet und die übrigen umgewandelt werden</p>
<p>090504 Da man jedoch später die z-Werte bei der Verdeckungsrechnung braucht, bleiben diese erhalten und nur die x-y-Ebene wird transformiert. Somit erhält man korrekt transformierte xy Koordinaten für eine Bildschirmdarstellung und hat noch einen weiteren Wert für spätere Berechnungen.</p>	<p>090504 Vorgehensweise: z-Werte erhalten und x-y-Ebene transformieren</p>	<p>090504 die z-Achse erscheint wieder</p>
<p>090505 In der Praxis sind besonders zwei Projektions-Transformationen von großer Bedeutung: die orthograpgische und die perspektivische.</p>	<p>090505 orthograpgische und die perspektivische Projektions Transformation</p>	

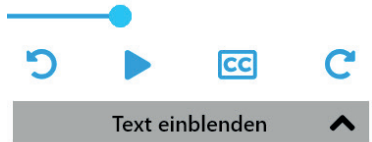
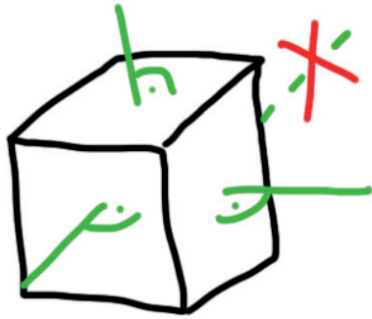


9.6 Clipping



Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090601 Beim Clipping geht es darum Flächen, die vom sichtbaren Volumen nicht mehr eingefangen werden können aus der Szene zu entfernen.</p> <p>090602 Nach der Projektionstransformation wird überprüft, ob alle Primitive vollständig im sichtbarem Bereich liegen.</p> <p>090603 Elemente die gänzlich außerhalb des Sichtfensters liegen werden komplett entfernt.</p> <p>090604 Für jede Kante des Sichtfensters wird geprüft, ob sich der Vertex eines Objekts inner- oder außerhalb der Kante befindet. Punkte die innerhalb der Grenze liegen werden in ihrer Geometrie belassen, Punkte außerhalb entfernt. An der Grenze des Sichtfensters werden neue Vertices kreiert. Dieses Verfahren wird auch Sutherland Hodgeman Clipping genannt.</p>	<p>090601 Clipping dient dazu Geometrien außerhalb des sichtbaren Volumens wegzuschneiden</p>	<p>090601-090603 -es erscheint nach und nach ein Sichtfenster mit Objekten, bei dem Objekte die gänzlich außerhalb liegen komplett entfernt werden und Objekte die teilweise im Sichtfenster liegen nur teilweise beschnitten werden</p> <p>090604 -Es erscheint ein Objekt mit Vertices und eine kante des Sichtfensters -Daraufhin werden neue Vertices berechnet und der überstehende Teil wird abgeschnitten</p>





Sprechertext

090701

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

090702

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

090703

Zur Ermittlung, ob eine Fläche sichtbar oder nicht sichtbar ist wird mit Hilfe eines Normalenvektors entschieden.

090704

Zeigt der Normalenvektor zum Beispiel in Richtung der Kamera, hat es zur Folge, dass der Betrachter die Vorderseite sieht. Ist der Normalenvektor n abgewandt handelt es sich um eine Rückseite und kann entfernt werden.

Screentext / Notizen

090701

Clipping bzw Culling dient dazu Geometrien außerhalb des sichtbaren Volumens wegzuschneiden

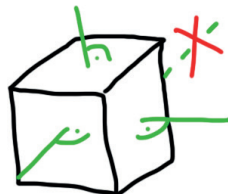
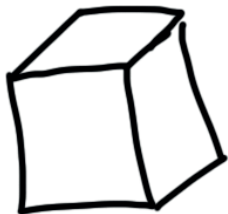
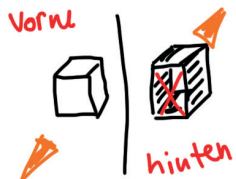
Regieanweisungen

090702

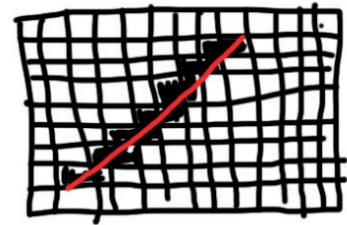
-Es erscheint ein Objekt mit Vorder- und Hinteransicht. Auf das Objekt ist eine Kamera gerichtet und der hintere wird entfernt

090803-090804

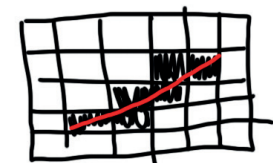
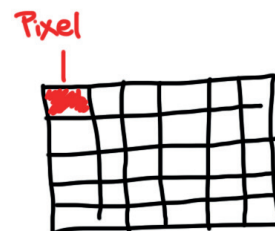
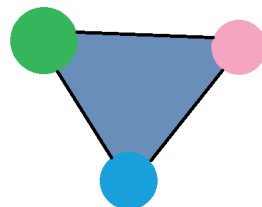
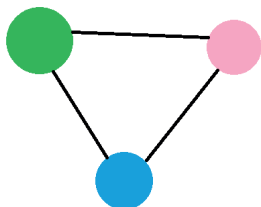
-Es erscheint ein Objekt auf dem Normalen erscheinen. die Normalen die von der Kamera abgewandt sind werden entfernt.



9.8 Rasterisierung



Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090801 Bis zu diesem Zeitpunkt liegt für jeden Vertex eines Polygons ein Farbwert vor. Um Das Bild jedoch später auf einem Monitor darstellen zu können, muss für jedes Pixel der Farbwert, durch Beleuchtungsberechnungen, berechnet werden.</p> <p>090802 Zur besseren Unterscheidbarkeit werden die Pixel in diesem Schritt Fragmente genannt. Das heißt jedes Fragment entspricht einem Pixel auf dem Bildschirm.</p> <p>090803 Bei diesem Schritt werden die Flächen in Fragmente aufgeteilt.</p>	<p>090802 Fragment = stellvertretend für Pixel</p>	<p>090801 Es erscheint Polygon, dessen Vertice Farbwerte enthalten Es erscheint ein Raster</p> <p>090802 Im Raster leuchtet eine Fläche auf, die ein Pixel darstellt. Anhand dessen wird der Begriff Fragment eingeführt</p> <p>090803 Es wird eine Linie bzw ein anderes beliebiges Objekt eingeblendet, welches den Flächen angenähert wird.</p>



9.9 Verdeckungsberechnung/z-Buffer



5	5	5	5	5	5	5	∞
5	5	5	5	5	5	∞	∞
5	5	5	5	5	∞	∞	∞
5	5	5	5	∞	∞	∞	∞
4	5	5	7	∞	∞	∞	∞
3	4	5	6	7	∞	∞	∞
2	3	4	5	6	7	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞



Sprechertext	Screen text / Notizen	Regieanweisungen
<p>090901 Bei einer Szene in der mehrere Objekte zu sehen sind, kann es dazu kommen, dass ein Objekt A vor einem Objekt B ist, oder dieses auch schneidet.</p> <p>090902 Um dies korrekt darzustellen, werden in der Computergrafik Verdeckungsberechnungen, wie z. B. der z-Buffer-Algorithmus angewandt.</p> <p>090903 Die Grundidee des z-Buffer-Algorithmuses ist es für jeden Pixel die Tiefeninformation bzw. den z-Wert zu speichern.</p> <p>090904 Es muss geprüft werden ob ein Pixel näher an der Kamera liegt als ein vorher berechneter. Dazu muss der z-Wert kleiner sein.</p> <p>090905 Falls ja, werden Farbwerte und z-Buffer für den Pixel überschrieben, andernfalls werden die alten Werte beibehalten.</p>	<p>090902 Verdeckungsberechnung durch z-Buffer-Algorithmus</p> <p>090903 z-Buffer-Algorithmus speichert für jeden Pixel z-Wert</p> <p>090905 Je kleiner der z-Wert eines Pixels, desto näher ist er am Betrachter</p>	<p>090901 -Es erscheint ein Bild, bei welchem sich Objekte überschneiden -Daraufhin erscheint das gleiche Bild nur mit falscher Verdeckungsberechnung</p> <p>090903-090905 Es wird ein Raster dargestellt. Alle Objekte werden auf dem Raster abgebildet.</p> <p>Falls der aktuell gerasterte Punkt näher am Betrachter liegt als der davor gerasterte Punkt, wird dieser durch das aktuelle ersetzt.</p> <p>Dabei wird die Distanz zum Betrachter eingetragen. Anhand dieser weiß man, welche Objekte wie überschritten und überlagert sind und wie die Objekte dargestellt werden müssen.</p>



∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞

5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5

5	5	5	5	5	5	5	∞
5	5	5	5	5	5	∞	∞
5	5	5	5	5	5	∞	∞
5	5	5	5	5	5	∞	∞
5	5	5	5	5	5	∞	∞
5	5	5	5	5	5	∞	∞
5	5	5	5	5	5	∞	∞
5	5	5	5	5	5	∞	∞

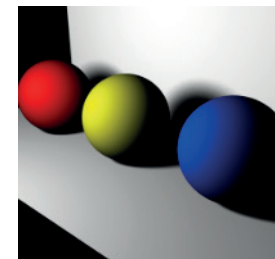
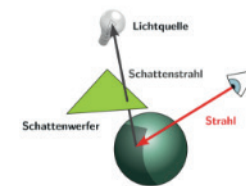
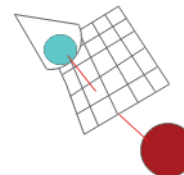
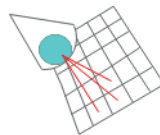
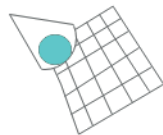
7							
6	7						
5	6	7					
4	5	6	7				
3	4	5	6	7			
2	3	4	5	6	7		

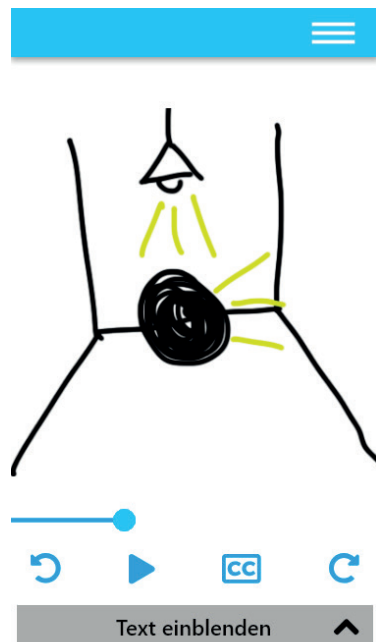
5	5	5	5	5	5	5	∞
5	5	5	5	5	5	∞	∞
5	5	5	5	5	∞	∞	∞
5	5	5	5	∞	∞	∞	∞
4	5	5	7	∞	∞	∞	∞
3	4	5	6	7	∞	∞	∞
2	3	4	5	6	7	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞

9.10 Raytracing



Sprechertext	Screentext / Notizen	Regieanweisungen
<p>091001 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.</p> <p>091102 Raytracing – zu Deutsch „Strahlen verfolgen“ – ist in erster Linie ein Algorithmus zu Verdeckungsberechnung.</p> <p>091003 Dies basiert auf dem Aussenden von Strahlen vom Betrachterblickpunkt aus. Abgesehen vom Betrachterblickpunkt ist eine Bildebene vorhanden, die in Pixel unterteilt ist und dem später gerenderten Rasterbild entspricht.</p> <p>091004 Für jeden Pixel wird ein Strahl ausgesandt. Diese werden Primärstrahl genannt.</p> <p>091005 Die Primärstrahlen ermitteln Schnittpunkte mit Primitiven. Der mit der geringsten Distanz zum Betrachter ist am nächsten.</p> <p>091006 Am Schnittpunkt des Strahls mit der Primitive wird eine Normale berechnet, die dazu dient, dass die Farbe des Oberflächenstücks berechnet werden kann.</p> <p>091007 Anschließend werden die gebrochenen bzw. die reflektierten Strahlen berechnet. Diese werden als Sekundärstrahlen bezeichnet.</p> <p>091008 Der Strahl endet, wenn er die maximale Anzahl von Schritten erreicht, auf kein weiteres Objekt oder auf eine Lichtquelle trifft.</p> <p>091009 Das Raytracing bringt den Vorteil, dass korrekte Objektspiegelungen und Schatten dargestellt werden können.</p>	<p>091002 Raytracing (dt. Strahlen verfolgen“)</p> <p>091003 -Aussendung von Strahlen vom Betrachter aus -Für jedes Rasterelement ein Strahl</p> <p>091004 Primärstrahl: Strahl von Betrachter auf Pixel</p> <p>091007 Sekundärstrahl: reflektierte/ gebrochene Strahlen</p>	<p>091001 Es wird ein Bild von einer Szene gezeigt, welches den Raytracing-Algorithmus verwendet.</p> <p>091003 Es wird ein Auge eingeblendet Es wird eine Bildebene eingeblendet</p> <p>091004 Es erscheint ein Raster. Es schießen Strahlen aus dem Auge durch jedes Rasterelement. Daraufhin wird geprüft, ob der Strahl ein Objekt trifft.</p> <p>091006 Es erscheinen Normalen Oberflächenstücke bekommen eine Farbe</p> <p>091007 Sekundärstrahlen entstehen</p> <p>091008 Ein fertig gerendertes Bild entsteht</p>





Sprechertext	Screentext / Notizen	Regieanweisungen
<p>091101 Radiosity heißt zu Deutsch Ausstrahlung. Hierbei handelt es sich um ein globales Beleuchtungsmodell, das heißt, dass sowohl das Licht, das von der Lichtquelle ausgeht, als auch das, welches von Oberflächen reflektiert wird, in die Berechnung einfließt.</p> <p>091102 Radiosity beschränkt sich dabei auf Objekte mit ideal diffusen Oberflächen.</p> <p>091103 Bei diesem Beleuchtungsmodell, werden keine Strahlen wie beim Raytracing verfolgt, sondern es findet ein Strahlungsaustausch zwischen Oberflächenstücken, den sogenannten Patches, statt.</p> <p>091104 Von jeder Fläche geht ein konstanter Lichtstrom aus, der sich aus emittierten Lichtstrom, falls es sich um eine Lichtquelle handelt, und reflektierten Lichtstrom zusammensetzt.</p> <p>091105 Radiosity hat gegenüber Raytracing den Vorteil, dass dieses Verfahren blickwinkelunabhängig ist. Dafür ist es aber sehr rechenaufwändig.</p> <p>091106 Das Radiosity Verfahren ist bestens für Innenraumszenen mit gedämpften Licht geeignet, da sanfte Beleuchtungsübergänge möglich sind.</p>	<p>091101 -radiosity = (dt.) Ausstrahlung -Objekte reflektieren Licht und werden zu auch zu Lichtquellen</p> <p>091103 Oberflächenstück = Patch</p> <p>091106 blickpunktunabhängig</p>	<p>091101 -Es erscheint ein Bild bei dem der begriff erklärt wird</p> <p>091102 Es erscheint ein Objekt</p> <p>091103-091105 Es erscheint eine Lichtquelle. Diese strahlt Strahlen aus. Objekte die von den Strahlen getroffen werden reflektieren und fangen an selbst zu einer Lichtquelle zweiter Ordnung zu werden. Darufhin erscheinen an den Objekten Begriffe, die zeigen, um welche Art von Licht es sich handelt</p> <p>091106 Es erscheint eine fertig gerenderte Szene auf Basis des Radiosity-Algorithmus</p>

radiosity
=
Ausstrahlung



9.12 Radiosity – Interaktion



- ☐ Shading
- ☐ Raytracing
- ☒ Radiosity

Anweisungen

091201

Wähle mittels der Radio Button zwischen reinem Shading, Raytracing und Radiosity aus. Betrachte die Änderungen.



Sprechertext	Screentext / Notizen	Regieanweisungen
<p>091301 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.</p> <p>091302 Die Volumengrafik basiert auf dem Strahlentransport, der beschreibt, wie sich Licht auf dem Weg durch ein Volumen verhält.</p> <p>091303 Beim Rendern einer Volumengrafik unterscheidet man vier Schritte:</p> <p><u>1. der Klassifikation:</u> Hier werden den Voxeln Materialeigenschaften gegeben</p> <p><u>2. der Interpolation:</u> Hier werden die Materialeigenschaften an Punkten zwischen den Voxeln aus den umgebenden Voxeln angenähert.</p> <p><u>3. dem Shading:</u> Beim Shading wird bestimmt, wie viel Licht von einem Voxel aus in Richtung des Betrachters reflektiert wird und welche Farbe es hat.</p> <p><u>4. der Composition:</u> Beim Compositing werden die Lichtbeiträge von Voxeln, in einer Reihe liegen, miteinander verrechnet, um einen endgültigen Bildpunkt zu erhalten.</p>	<p>091301 Volumengrafik = transparente Objekte</p> <p>Voxel = Gitterpunkt in einem dreidimensionalen Gitter.</p> <p>091303 vier Render Schritte: 1. Klassifikation 2. Interpolation 3. Shading 4. Composition</p>	<p>091301 Es wird ein Voxelgitter eingeblendet und anhanddessen ein Voxel gezeigt</p> <p>091303 Die vier Schritte werden erklärt: 1) Es werden Eigenschaften verschiedener Transparenzstufen gezeigt 2) Voxel werden am Lichtstrahl interpoliert 3) Die Voxelflächen erhalten Normalen und eine Beleuchtung 4) Die unterschiedlichen Lichtstufen einer Linie werden miteinander verrechnet</p> <p>Zum Schluss wird eine Volumengrafik eingeblendet, die sich dreht.</p>

