

DREHBUCH RENDERING

Computergrafik.Online

**Betreuer: Prof. Jirka Dell'Oro-Friedl
Sommersemester 2018**

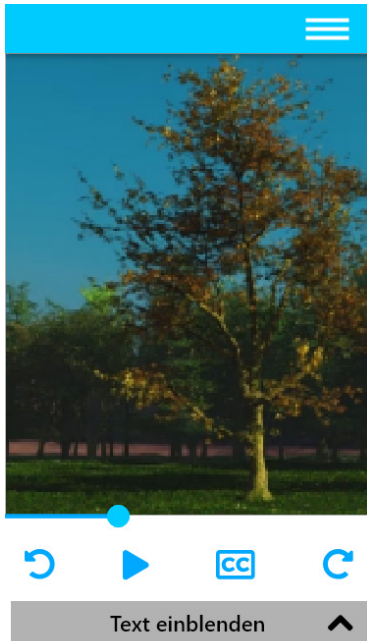
**Hochschule Furtwangen University
Fakultät Digitale Medien**

**Version: 1.2
Letzte Änderung: 22.07.2018
Autor: Berdan Der**

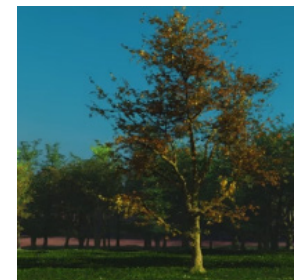
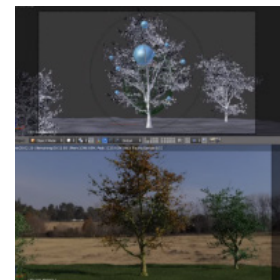
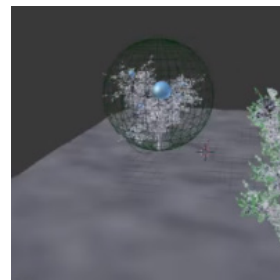
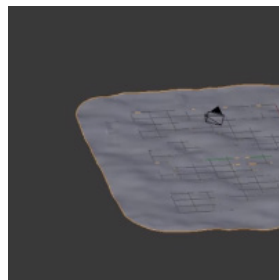
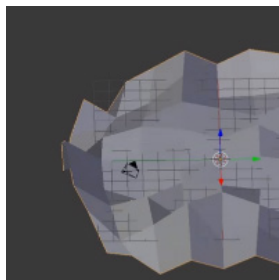
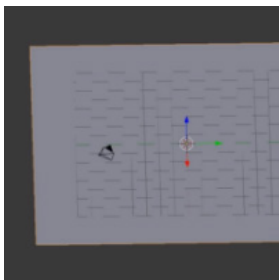


1.1 Einleitung	1
1.2 Geometrie und Primitive	2
1.2.1 Geometrie und Primitive – Interaktion	3
1.3 Stufen der Rendering Pipeline	4
1.3.1 Modell Transformation	5
1.3.1.1 Modell Transformation – Interaktion	6
1.3.2 Viewing Transformation	7
1.3.3 Per-Vertex Transformation	8
1.3.4 Projektions Transformation	9
1.3.5 Culling und Clipping	10
1.3.6 Rasterisierung	11
1.3.7 Texturierung	12
1.3.8 Blending	13
1.4 Echtzeit Rendering	14
1.4 Echtzeit Rendering – Interaktion	15
1.5 Realistisches Rendern	
1.5.1 Raytracing	16
1.5.2 Radiosity	17
1.5.2 Radiosity – Interaktion	18
1.6 Volumengrafik	19

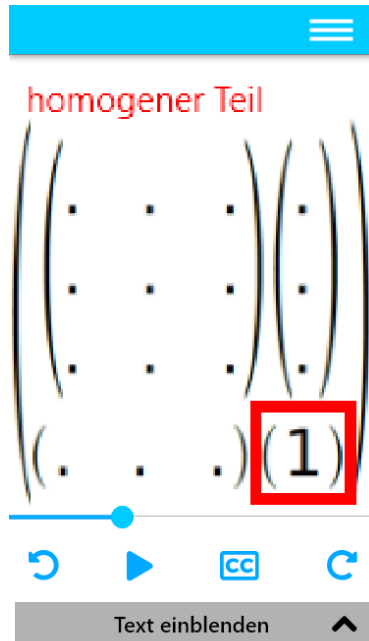
1.1 Einleitung



Sprechertext	Screentext / Notizen	Regieanweisungen
<p>120101 Rendern heißt zu Deutsch Bildsynthese und bezeichnet den Vorgang aus Rohdaten ein Bild zu erzeugen.</p> <p>120102 Dabei wird das 3D-Objekt in ein rasterisiertes Bild umgewandelt und zum Schluss auf einem Ausgabegerät dargestellt.</p> <p>120103 Die Aufgaben eines Renderers sind es: die Flächenposition, deren Ausrichtung, die Lichtsituation und die Farbe zu berechnen bzw. zu ermitteln</p>	<p>120101 - Rendern (dt. Bildsynthese) - Aus Rohdaten wird ein Bild erzeugt</p> <p>120103 Aufgaben Renderers: - Flächenposition, - Ausrichtung, - Lichtsituation, - Farbe ermitteln/berechnen</p>	<p>Es wird zu Anfang eine Szene gebaut, die daraufhin gerendert wird. Das alles geschieht während dem Sprechertext als Timelaps</p>



1.2 Geometrie und Primitive



Sprechertext	Screentext / Notizen	Regieanweisungen
<p>120201 Die Rendering-Pipeline arbeitet mit Geometrien, die aus einer Vielzahl von Primitiven bestehen können.</p> <p>120202 Aus diesen einfachen Formen lassen sich komplexere Objekte darstellen.</p> <p>120203 Punkte selbst werden als homogene Koordinate dargestellt.</p> <p>120204 Diese enthält Koordinaten zur Rotation, Skalierung, Scherung und Verschiebung.</p> <p>120205 Des Weiteren besitzt sie einen Projektions- und einen homogenen Teil.</p> <p>120206 Verändere mittels der Leiste die Anzahl der Polygone und beobachte, wie sich das Objekt verändert.</p>	<p>120201 - Rendering-Pipeline arbeitet mit Geometrien</p>	<p>120201 Es werden verschiedene Geometrien eingeblendet</p> <p>120202 Aus diesen Polygonen wird ein komplexeres Objekt gebaut. Dieses Objekt bekommt nach und nach mehr Polygone</p> <p>120203 Es wird eine homogene Koordinate eingeblendet.</p> <p>Daraufhin werden Teile der homogenen Koordinate rot markiert und es wird gezeigt, wofür sie zuständig sind.</p>



1.2.1 Geometrie und Primitive – Interaktion



Anweisungen

12020101

Verändere mittels der Leiste die Anzahl der Polygone und beobachte, wie sich das Objekt verändert.

1.3 Stufen der Rendering Pipeline

	Sprechertext	Screentext / Notizen	Regieanweisungen
<div> <div>☰</div> <div> 1. Modell-Transformation 2. Viewing-Transformation 3. Per-Vertex-Transformation 4. Projektions Transformation 5. Texturierung 6. Blending </div> <div> <div> <div></div> <div>↺</div> <div>▶</div> <div>CC</div> <div>↻</div> </div> <div>Text einblenden</div> <div>⬆</div> </div> </div>	<p>120301 Die Rendering-Pipeline besteht aus mehreren Schritten.</p> <p>120302 Der erste Schritt ist die Modell-Transformation.</p> <p>Dann die Viewing-Transformation, auf welche die Per-Vertex-Transformation folgt.</p> <p>Als vierter Schritt wird die Projektions Transformation durchgeführt.</p> <p>Den Schluss macht die Texturierung und das Blending</p>	<p>Rendering-Pipeline-Schritte:</p> <ol style="list-style-type: none"> 1. Modell-Transformation. 2. Viewing-Transformation 3. Per-Vertex-Transformation 4. Projektions Transformation 5. Texturierung 6. Blending 	<p>Die Schritte der Pipeline werden nach und nach aufgelistet.</p>

1. Modell-Transformation

1. Modell-Transformation
2. Viewing-Transformation

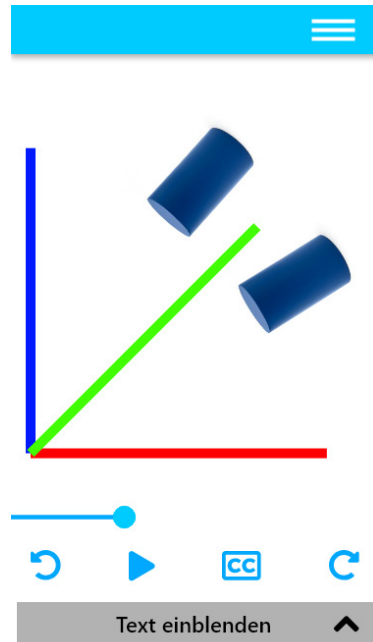
1. Modell-Transformation
2. Viewing-Transformation
3. Per-Vertex-Transformation

1. Modell-Transformation
2. Viewing-Transformation
3. Per-Vertex-Transformation
4. Projektions Transformation

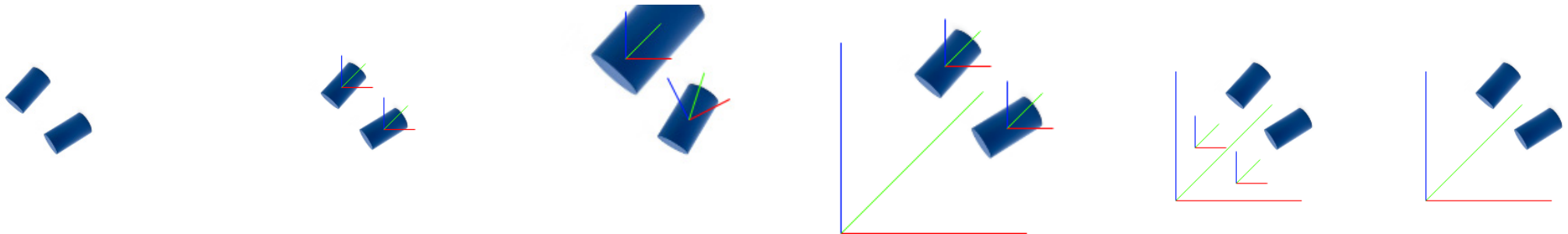
1. Modell-Transformation
2. Viewing-Transformation
3. Per-Vertex-Transformation
4. Projektions Transformation
5. Texturierung

1. Modell-Transformation
2. Viewing-Transformation
3. Per-Vertex-Transformation
4. Projektions Transformation
5. Texturierung
6. Blending

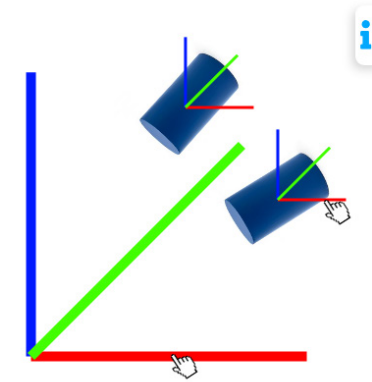
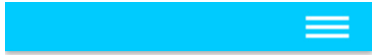
1.3.1 Modell Transformation



Sprechertext	Screentext / Notizen	Regieanweisungen
<p>12030101 3D-Szenen bestehen meist aus mehreren Objekten, die ein lokales Koordinaten besitzen.</p> <p>12030102 Dies ermöglicht die einfache Handhabung einfacher Transformationen.</p> <p>12030103 Jedoch erfolgen alle weiteren Schritte der Rendering-Pipeline in einem globalen Koordinatensystem. Deswegen müssen alle Objekte in diese überführt werden.</p> <p>12030104 Wechsle zwischen den Koordinatensystemen mittels der Radio Buttons und führe verschiedene Transformationen durch.</p>	<p>12030101 Objekte besitzen lokale Koordinatensysteme</p> <p>12030103 Objekte werden in globales Koordinatensystem überführt</p>	<p>12030101 Es werden mehrere Objekte eingeblendet. Diese erhalten dann ein lokales Koordinatensystem (KS).</p> <p>12030102 Die Objekte werden daraufhin an diesem KS einfach transformiert.</p> <p>12030103 Es erscheint ein globales KS. Die Objekte werden nun in dieses überführt (Die Achsen des lok. KS wandern auf die des globalen)</p>



1.3.1.1 Modell Transformation – Interaktion

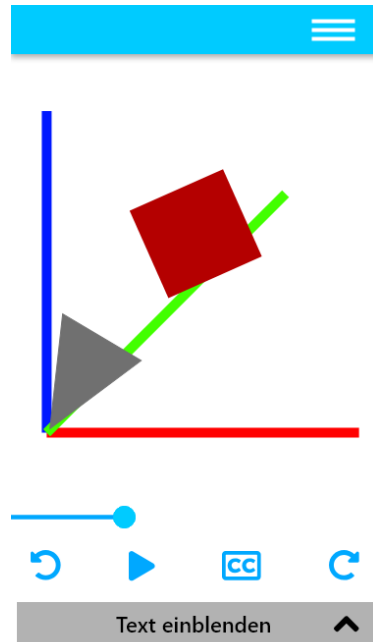


Anweisungen

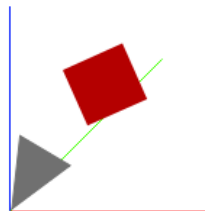
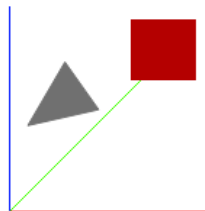
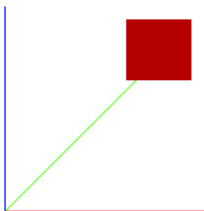
1203010101

Wechsle zwischen den Koordinatensystemen mittels der Radio Buttons und führe verschiedene Transformationen durch.

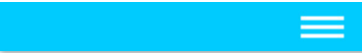
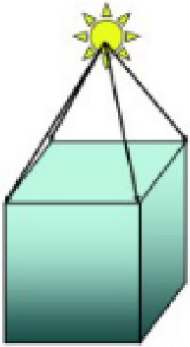

1.3.2 Viewing Transformation



Sprechertext	Screentext / Notizen	Regieanweisungen
<p>12030201 Um am Ende ein Bild ausgeben zu können wird in der Szene eine Kamera benötigt.</p> <p>12030202 Die Kamera wird zunächst als ein Objekt in das globale Koordinatensystem gesetzt. Dadurch entstehen aber komplexe Berechnungen in der Projektionsmatrix bei der Projektionstransformationen.</p> <p>12030203 Um diesen Komplikationen zu entgehen, setzt man die Kamera in den Ursprung der Szene. Dies vereinfacht die Matrize und spätere Berechnungen.</p>	<p>12030203 Kamera wird in den Ursprung des globalen Koordinatensystems gesetzt</p>	<p>12030201 Es wird eine Szene eingeblendet.</p> <p>12030202 Darauf folgt eine Kamera, die in der Szene platziert wird.</p> <p>12030203 Die Kamera wird in den Ursprung des globalen KS gesetzt.</p>

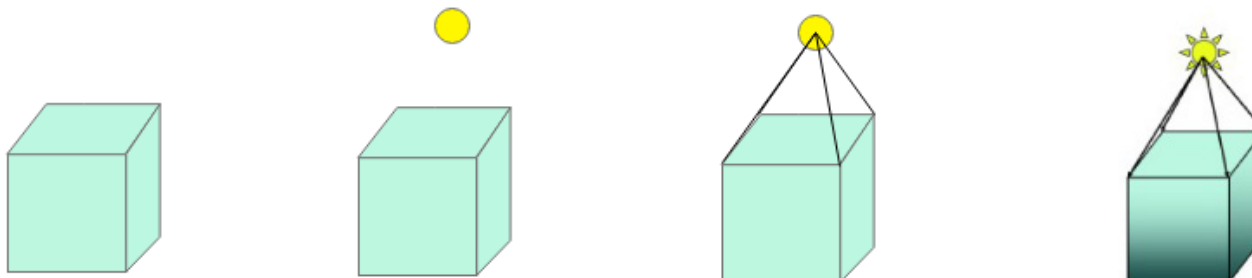


1.3.3 Per-Vertex Transformation

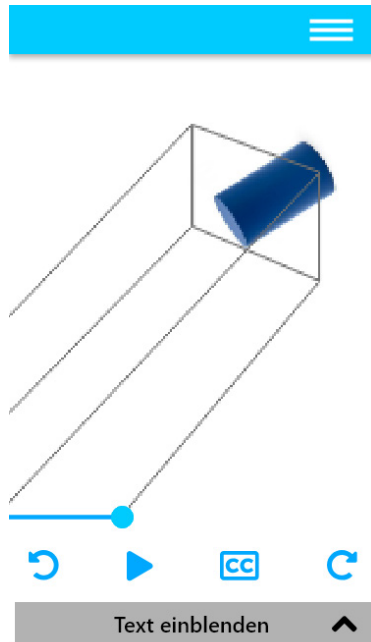




Text einblenden

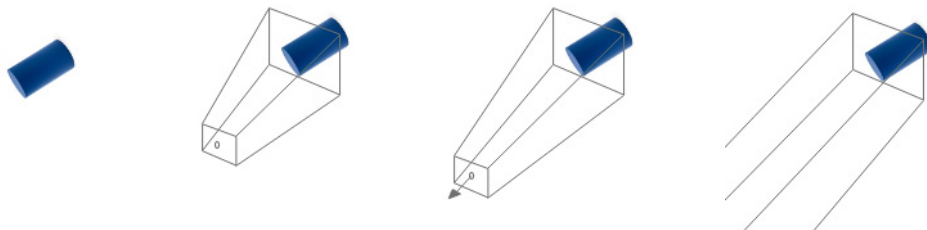
Sprechertext	Screentext / Notizen	Regieanweisungen
<p>12030301 Bei der Per-Vertex-Transformation wird zunächst eine Lichtquelle in das Koordinatensystem gesetzt.</p> <p>12030302 Daraufhin wird die Beleuchtung, anhand der Winkel des einfallenden Lichtes und der Oberflächenbeschaffenheit der Objekte berechnet.</p> <p>12030303 Bei der Rasterung werden später die Farbwerte des Objektes berechnet.</p>	<p>12030302 Beleuchtung wird anhand des einfallenden Lichts und der Oberflächenbeschaffenheit berechnet</p>	<p>12030301 Es erscheint ein Würfel in der Szene. Diesem folgt eine Lichtquelle.</p> <p>12030302 von der Lichtquelle aus werden Strahlen an den Würfel geschossen</p> <p>Daraufhin erscheint der Würfel in korrekter Beleuchtung.</p>



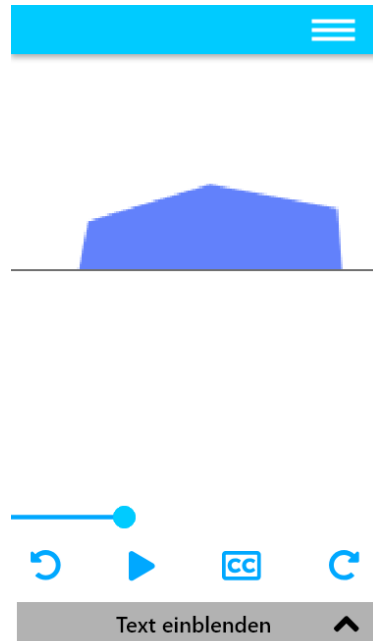
1.3.4 Projektions Transformation



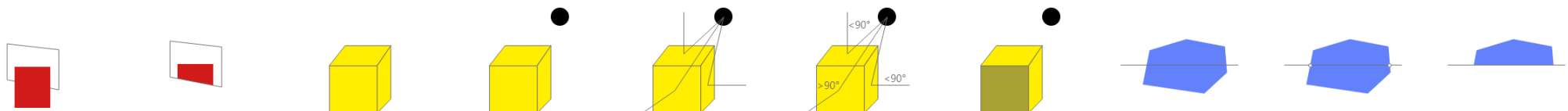
Sprechertext	Screentext / Notizen	Regieanweisungen
<p>12030401 Der View Space, also der Bereich, den die Kamera einfängt, hat immernoch die Form einer abgeschnittenen Pyramide.</p> <p>12030402 Für die nächsten Schritte sollte der View Space jedoch die Form eines Quaders annehmen.</p> <p>12030403 Dies erreicht man dadurch, dass man die Kamera ins Unendliche verschiebt. Dies erleichtert im Folgenden z. B. das Clipping.</p> <p>12030404 Verändere den View Space durch das Ziehen an der Kamera und schau, wie sich das Bild verändert.</p>	<p>12030401 am Anfang: View Space = abgeschnittene Pyramide</p> <p>12030403 Kamera wird ins Unendliche verschoben: View Space = Quader</p>	<p>12030401 Es wird ein Objekt in der Szene eingeblendet. Daraufhin wird der View Space eingeblendet.</p> <p>12030403 Die Kamera wird ins Unendliche verschoben und man sieht, wie sich der View Space ändert.</p> <p>Am Ende ist der View Space einem Quader ähnlich.</p>



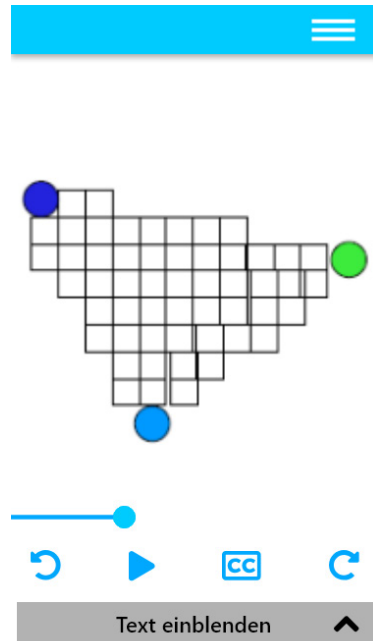
1.3.5 Culling und Clipping



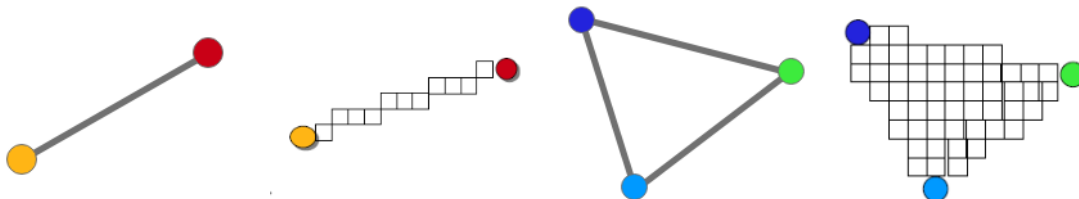
Sprechertext	Screentext / Notizen	Regieanweisungen
<p>12030501 Beim Clipping und Culling geht es darum, Elemente, die vom View Space nicht mehr eingefangen werden könne aus der Szene zu entfernen.</p> <p>12030502 Durch das Backface-Culling werden die Polygone aus der Szene entfernt, die vom Betrachter abgewandt sind.</p> <p>12030503 Eine mögliche Vorgehensweise wäre es die Winkel zwischen den Normalen der Flächen und einem Vektor der Kamera zu berechnen.</p> <p>12030504 Ist der Winkel zwischen der Normale und dem gebildeten Vektor größer als 90° dann wird das Polygon aus der Szene entfernt.</p> <p>12030505 Beim Clipping werden Teile aus der Szene ermittelt, die im späteren Bild nicht zu sehen sind und nicht mehr im View Space liegen.</p> <p>12030506 Dabei wird für jede Kante des Sichtfensters einzeln getestet, ob sich der Eckpunkt inner- oder außerhalb der Kante des Sichtfensters befinden. Punkte innerhalb der Grenze werden in ihrer Geometrie belassen, Punkte außerhalb entfernt. Dieses Verfahren wird auch Sutherland Hodgeman Clipping genannt.</p>	<p>12030502 Culling: Polygone entfernen, die der Betrachter nicht sehen kann</p> <p>Winkel zwischen Normale und Betrachter-Vektor $>90^\circ$</p> <p>12030505 Clipping: Polygone entfernen, die außerhalb des Sichtfensters liegen</p>	<p>12030501 Zu Anfang ist ein Sichtfeld und ein Objekt zu sehen. Daraufhin wird alles außerhalb des Sichtfelds weggeschnitten.</p> <p>12030502/12030503/ 12030504 Es erscheint ein Würfel, der von einem Betrachter gefolgt wird. Daraufhin werden die Normalen der Flächen dargestellt und die Winkel zum Betrachter gemessen. Flächen, bei denen der Winkel $>90^\circ$ sind werden verworfen.</p> <p>12030505/ 12030506 Ein Sichtfenster wird eingeblendet. Daraufhin sieht man ein Objekt. Jetzt wird bei jedem Punkt getestet, ob diese inner- oder außerhalb des Fensters liegt. An Schnittpunkten werden neue Punkte gebildet. Alle Flächen außerhalb werden verworfen und an den neuen Punkten entstehen neue Kanten.</p>



1.3.6 Rasterisierung



Sprechertext	Screentext / Notizen	Regieanweisungen
<p>12030601 Mit der Rasterung wir die Umwandlung der Geometrien in Fragmente bezeichnet.</p> <p>12030602 Die Primitiven die gerastert werden, bilden Flächen. Fragmente selbst sind noch keine fertigen Pixel, sondern zeigen nur deren spätere Position.</p> <p>12030603 Des Weiteren wird bei übereinanderliegenden Polygonen ermittelt, welche näher am Betrachter liegen. Hierfür wird ein Z-Buffer benötigt.</p> <p>12030604 Des weiteren könne Fragmente auch eingefärbt/beleuchtet werden. Dies hängt immer von der Beleuchtung, Textur und anderen Materialeigenschaften zusammen.</p> <p>12030605 Stark vereinfacht kann man sagen, dass bei der Rasterisierung Farbwerte für jeden Pixel eines Primitivs, welcher innerhalb des Sichtfensters liegt berechnet wird.</p>	<p>12030601 Rasterung = Umwandlung Gemetrie in Fragmente</p> <p>12030605 berechnet Farbwerte für Pixel die im Sichtfenster liegen</p>	<p>Es werden verschiedene Geometrien aufgezeigt , die dan gerastert werden und Fragmente bilden.</p>



1.3.7 Texturierung

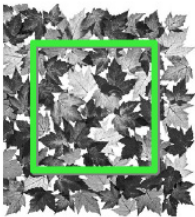
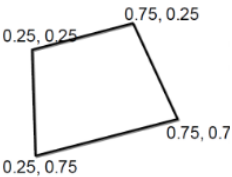




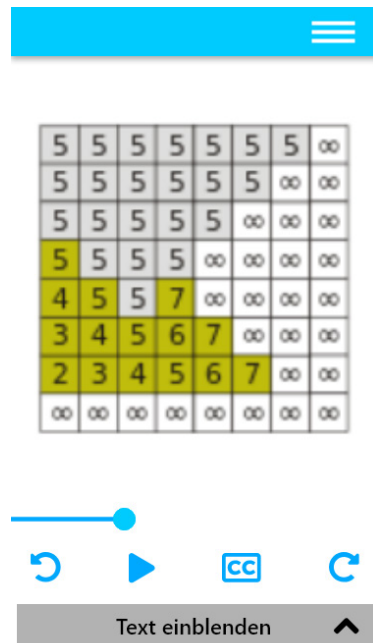


Text einblenden

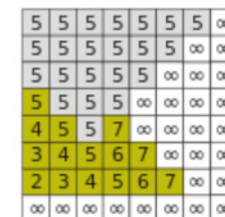
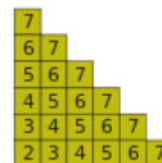
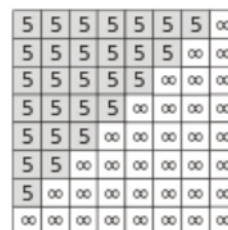
Sprechertext	Screentext / Notizen	Regieanweisungen
<p>12030701 Jeder Eckpunkt des Objekts kann neben seiner 3D- Koordinaten im Raum zusätzlich auch noch mit einer 2D-Texturkoordinate versehen werden.</p> <p>12030702 Durch die Texturkoordinaten wird definiert, wie eine Textur auf dem Objekt abgebildet werden soll.</p>	<p>12030701 3D-Koordinaten werden 2D-Texturkoordinaten zugewiesen</p> <p>12030702 Textur wird je nach Texturkoordinaten auf dem Objekt abgebildet</p>	<p>12030701 Dem Objekt werden Texturkoordinaten zugewiesen.</p> <p>12030702 Das Objekt wird anhand der Koordinaten mit der Textur gefüllt.</p>



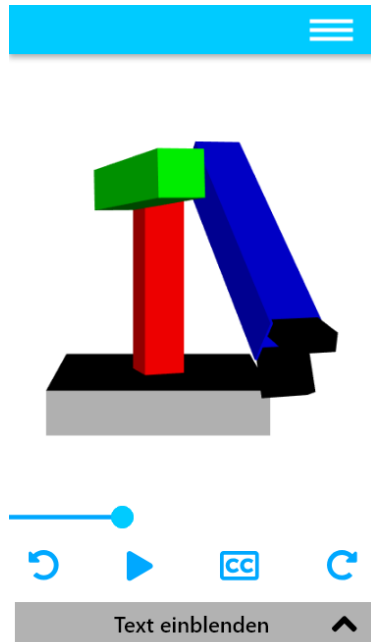
1.3.8 Blending



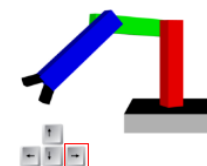
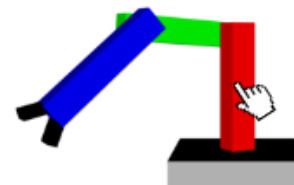
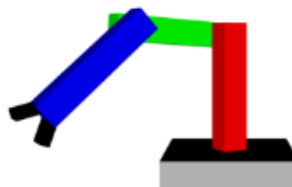
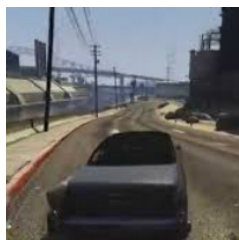
Sprechertext	Screentext / Notizen	Regieanweisungen
<p>12030801 Beim Blending wird bei übereinanderliegenden Polygonen ermittelt, welche sichtbar sind und näher am Betrachter liegen.</p> <p>12030802 Dafür wird ein Z-Buffer verwendet. Dieser enthält Informationen über die Entfernung der Objektflächen von der Kamera und ermittelt, welche Polygone verdeckt sind und welche dargestellt werden müssen.</p> <p>12030803 Zuerst werden alle Polygone der Objekte gerastert.</p> <p>12030804 Danach wird ermittelt, welche Fragmente näher am Betrachter liegen und ob sie Fragmente anderer Objekte verdecken.</p> <p>12030805 Falls das aktuell betrachtete Fragment näher am Betrachter liegt als das des davor gerasterten Punktes, wird dieser durch das aktuelle Polygon ersetzt.</p>	<p>12030801 Ermittlung durch Z-Buffer, welche Polygone für Betrachter sichtbar sind und welche näher an ihm sind</p>	<p>12030801 Es erscheint ein Bild, bei welchem sich Objekte überschneiden</p> <p>12030803-12030805 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 an, welche Objekte wie überschritten und überlagert sind und wie die Objekte dargestellt werden müssen.</p>



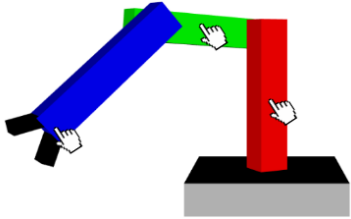
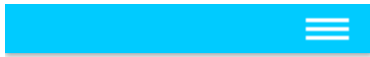
1.4 Echtzeit Rendering



Sprechertext	Screentext / Notizen	Regieanweisungen
<p>120401 Bei den Arten des Renderns wird zwischen realistischem und einem in Echtzeit unterschieden.</p> <p>120402 Hierbei wird eine Reihe von Bildern mit hoher Bildfrequenz berechnet und die Szene durch den Anwender interaktiv verändert.</p> <p>120403 Die Anwendung liegt z. B. bei Spielen oder VR-Anwendungen.</p> <p>120404 Klicke auf ein Arm-Segment des Greifarms und bewege ihn mittels der Pfeiltasten.</p>	<p>120402 Reihe von Bildern mit hoher Bildfrequenz berechnet</p> <p>kann von Anwender interaktiv verändert werden</p>	<p>120401 Es wird ein Bild gezeigt, welches realistisch gerendert wurde und eins welches auf Echtzeit basiert(Video)</p> <p>120402 Daraufhin wird ein Greifarm gezeigt, der mittels Pfeiltasten bewegt wird</p>



1.4.1 Echtzeit Rendering – Interaktion

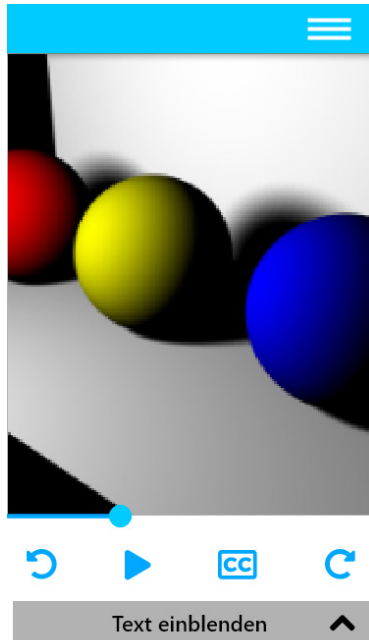


Anweisungen

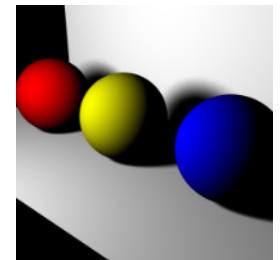
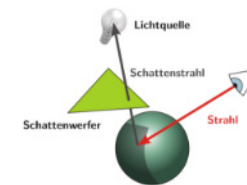
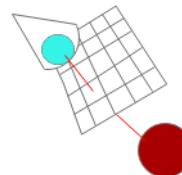
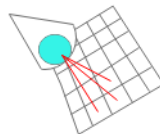
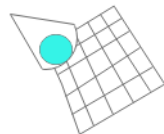
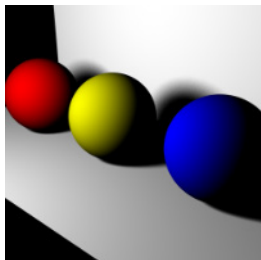
12040101

Klicke auf ein Arm-Segment des Greifarms und bewege ihn mittels der Pfeiltasten.

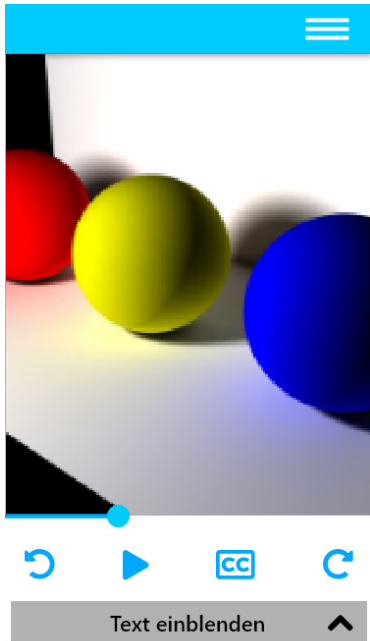
1.5.1 Raytracing



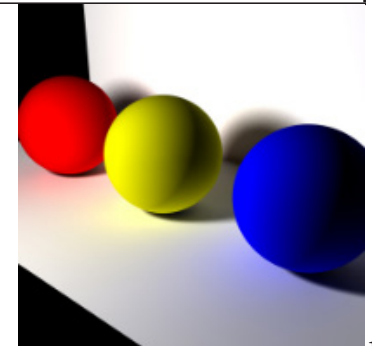
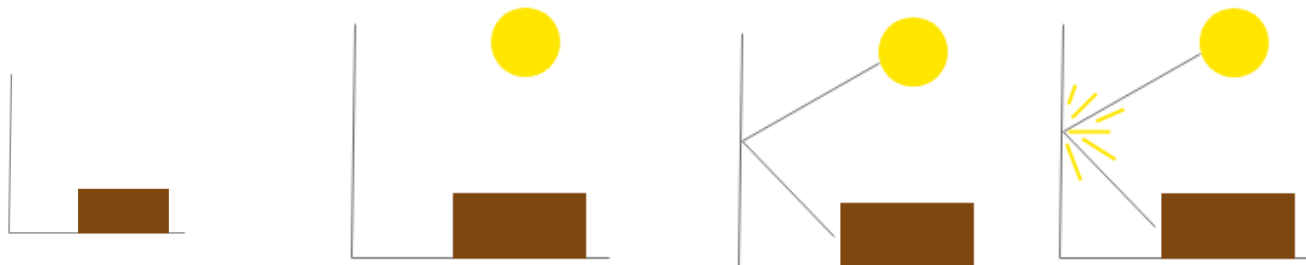
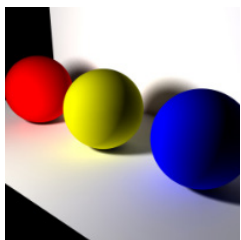
Sprechertext	Screentext / Notizen	Regieanweisungen
<p>12050101 Beim realistischen Rendern liegt das Hauptaugenmerk auf der Korrektheit der Darstellung des gerenderten Bildes. Die Bildqualität und die physikalische Korrektheit spielen eine besondere Rolle, wobei man höhere Rechenzeiten in Kauf nehmen muss.</p> <p>12050102 Raytracing – zu Deutsch „Strahlen verfolgen“ – ist in erster Linie ein Algorithmus zur Verdeckungsrechnung.</p> <p>12050103 Dieses basiert auf dem Aussenden von Strahlen vom Betrachterblickpunkt aus.</p> <p>12050104 Da das Bild an einem Monitor ausgegeben wird, der über ein Raster verfügt, betrachtet man für jedes Rasterelement nur einen Strahl. Dabei prüft man, ob sich ein Objekt mit dem Strahl schneidet.</p> <p>12050105 Stark vereinfacht kann man sagen, dass für jedes Element des Sichtfensters ein Strahl gesendet wird und daraus Schnittpunkte mit der Geometrie und deren Farbbeiträge berechnet werden.</p> <p>12050106 Der Vorteil bei dieser Vorgehensweise ist die realistische Simulation des Lichtes. Jedoch muss man mit Abzügen in der Performance rechnen.</p>	<p>12050102 Raytracing (dt. Strahlen verfolgen“)</p> <p>12050103 Aussendung von Strahlen vom Betrachter aus</p> <p>12050104 Für jedes Rasterelement ein Strahl</p> <p>schneidet Strahl ein Objekt?</p>	<p>12050101 Es wird ein Bild von einer Szene gezeigt, welches den Raytracing-Algorithmus verwendet.</p> <p>12050103 Es wird ein Auge eingeblendet</p> <p>12050104 Es erscheint ein Raster. Es schießen Strahlen aus dem Auge durch jedes Rasterelement. Daraufhin wird geprüft, ob der Strahl ein Objekt trifft.</p>



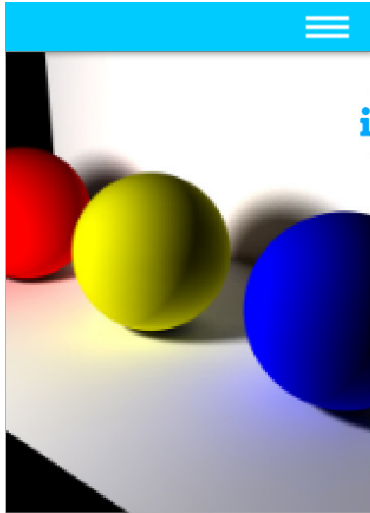
1.5.2 Radiosity



Sprechertext	Screentext / Notizen	Regieanweisungen
<p>12050201 Radiosity beschreibt den Vorgang der Ausbreitung von Licht in einer diffus reflektierenden Umgebung.</p> <p>12050202 Hierbei beeinflussen sich die Flächen gegenseitig, da sie ein Teil des Lichtes reflektieren. Somit werden Objekte zu weiteren Lichtquellen. Dadurch erhalten auch Flächen, die nicht direkt beleuchtet werden, eine bestimmte Helligkeit.</p> <p>12050203 Ein Vorteil gegenüber Raytracing ist, dass die Lichtverteilung blickpunktunabhängig berechnet wird.</p> <p>12050204 Radiosity eignet sich besonders zum Rendern statischer oder weniger animierter Szenen in Echtzeit, sofern eine zeitaufwändige Berechnung vertretbar ist.</p> <p>12050204 Wähle mittels der Radio Button zwischen Raytracing und Radiosity aus. Betrachte die Änderungen.</p>	<p>12050201 Radiosity = Vorgang der Lichtausbreitung</p> <p>12050202 Objekte reflektieren Licht und werden zu auch zu Lichtquellen</p> <p>12050203 blickpunktunabhängig</p>	<p>12050201 Es wird ein Bild gezeigt, welches mit dem Radiosity-Algorithmus gerendert wurde.</p> <p>12050202 Es erscheint ein Szene mit Lichtquelle. Diese strahlt Strahlen aus. Objekte die von den Strahlen getroffen werden reflektieren und fangen an selbst zu einer Lichtquelle zweiter Ordnung zu werden.</p>



1.5.2.1 Radiosity – Interaktion



☐ Raytracing

☒ Radiosity

Anweisungen

1205020101

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



Sprechertext	Screentext / Notizen	Regieanweisungen
<p>120601 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>120602 Die Volumengrafik basiert auf dem Strahlentransport, der beschreibt, wie sich Licht auf dem Weg durch ein Volumen verhält.</p> <p>120603 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>Volumengrafik = transparente Objekte</p> <p>Voxel = Gitterpunkt in einem dreidimensionalen Gitter.</p> <p>vier Render Schritte: 1. Klassifikation 2. Interpolation 3. Shading 4. Composition</p>	<p>120601 Es wird ein Voxelgitter eingeblendet und anhanddessen ein Voxel gezeigt</p> <p>120603 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>Zu schluss wird eine Volumengrafik eingeblendet, die sich dreht.</p>

