

# DREHBUCH RENDERING

Computergrafik.Online

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

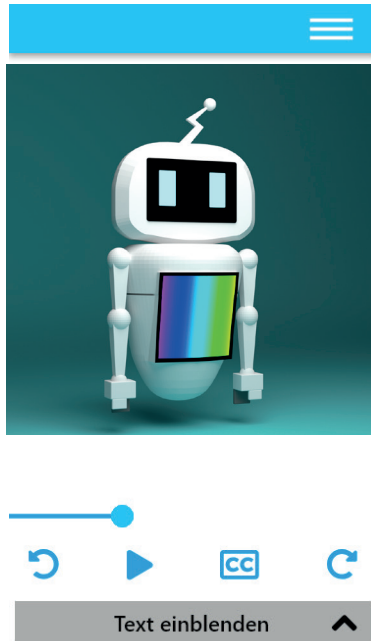
**Hochschule Furtwangen University**  
**Fakultät Digitale Medien**

**Version: 1.5**  
**Letzte Änderung: 06.12.2018**  
**Autor: Berdan Der**



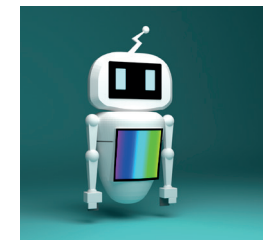
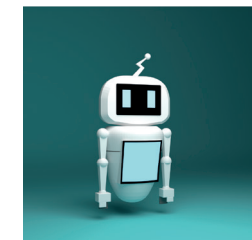
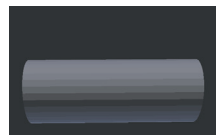
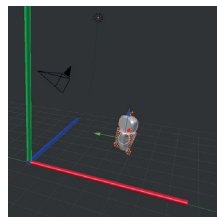
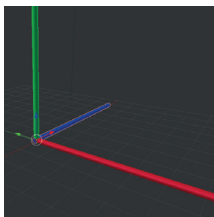
9.1 Einleitung/Anwendung	1
9.2 mathematische Grundlagen	2
9.3 Modell-Transformation	3
9.4 Augepunkt-Transformation	4
9.5 Projektions-Transformation	5
9.6 Clipping	6
9.7 Culling	7
9.8 Rasterisierung	8
9.9 Verdeckungsrechnung – z-Buffer	9
9.10 Raytracing	10
9.11 Raytracing - Interaktion	11
9.12 Volumengrafik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

## 9.1 Einleitung/Anwendung

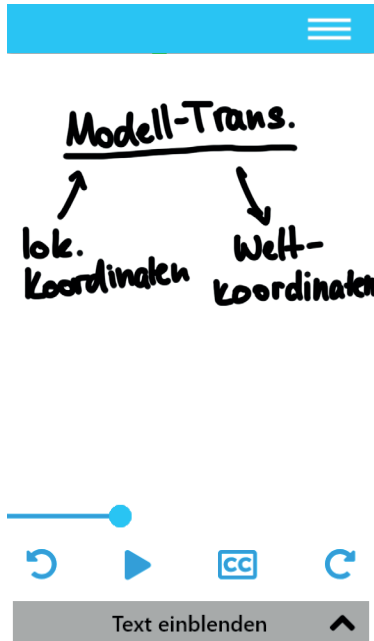


Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090101 Rendern stammt vom englischen Wort „to render“ und heißt zu deutsch „etwas ausgeben“. Das Rendering bezeichnet den Vorgang, ein Bild zu generieren. Beim Rendern aus einer 3D-Szene werden insbesondere Meshes, Kameras und Lichtquellen berücksichtigt.</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 Umwandlung der Koordinaten des Meshes und die Rasterisierung.</p> <p>090103 Des Weiteren werden aber auch Sichtbarkeits- und Beleuchtungsberechnungen durchgeführt, Texturen gemappt und spezielle Effekte dargestellt.</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>

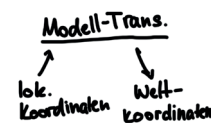
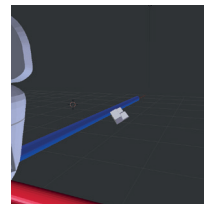
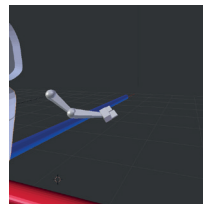
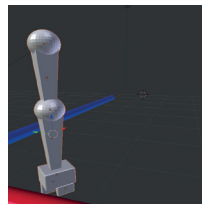
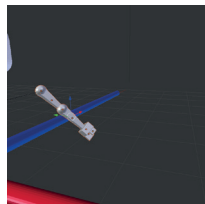
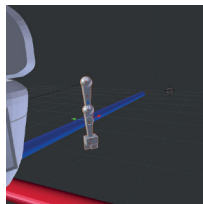
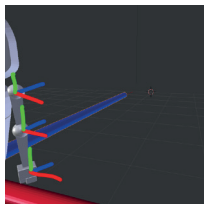
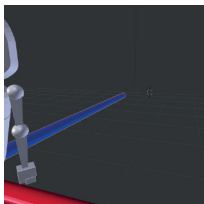
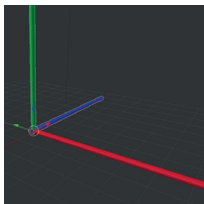
rendern  
=  
Bildsynthese



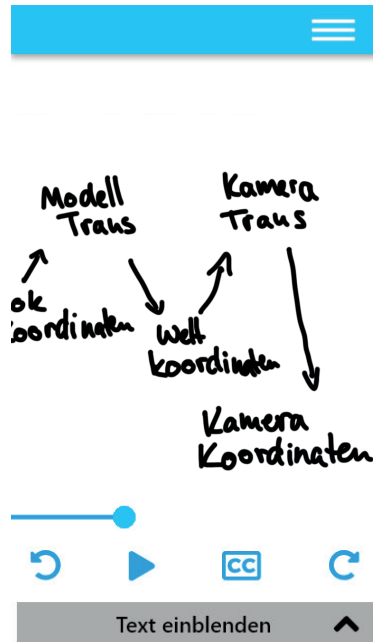
### 9.3 Modell Transformation



Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090301 Ein Objekt, dass sich in einem dreidimensionalen Raum aufhält, wird normalerweise durch sein lokales Koordinatensystem definiert. Ohne Transformationen würden alle Objekte im Weltursprung liegen.</p> <p>090302 Durch Translationen, Rotationen und Skalierungen wird ein Mesh an die gewünschte Stelle im Raum gebracht.</p> <p>090303 Die endgültige Position eines Objektes hängt von der Reihenfolge der Transformationen in der Szenenhirarchie ab.</p> <p>090304 Um diese Transformationsverkettung aufzulösen werden die Objekte in ein Weltkoordinatensystem übertragen.</p> <p>090305 Von nun an liegen die Koordinaten nicht mehr in lokalen Koordinaten, sondern in Weltkoordinaten vor.</p>	<p>090301 lokales Koordinatensystem = Objektkoordinaten</p> <p>090302 Verschiebung (Translation) Drehung (Rotation) Vergrößerung bzw. Verkleinerung (Skalierung)</p> <p>090304 Modelltransformation: lokales Koordinatensystem --&gt; globales Koordinatensystem</p>	<p>090301 -Es erscheint ein Koordinatensystem mit einem Objekt (roboterarm) -Auf dem Objekt erscheint ein Objektkoordinatensystem</p> <p>090302 -Das Objekt wird verschiedenen Transformationen unterzogen</p> <p>090303 -auf den verschiedenen hierarchien erscheinen Objektkoordinatensysteme und es wird eine verkettete Transformation ausgeführt</p> <p>090304 Der Großteil der Arms bist auf eine Hierarchiestufe verschwindet, da man diese nun alleine betrachten kann</p> <p>090305 Es erscheint ein Schema, in welchem klar wird, welche Koordinaten zu diesem Zeitpunkt vorliegen.</p>

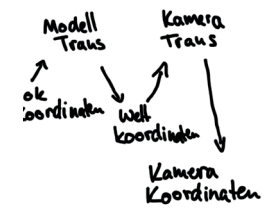
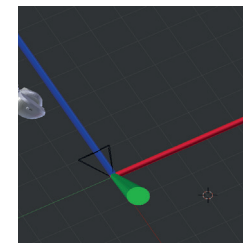
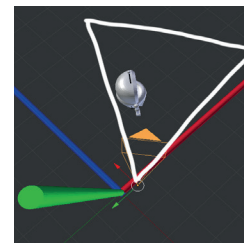
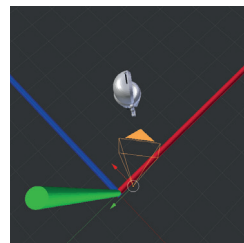
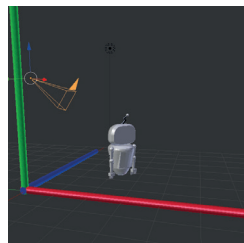
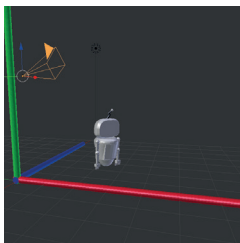


## 9.4 Kamera-Transformation

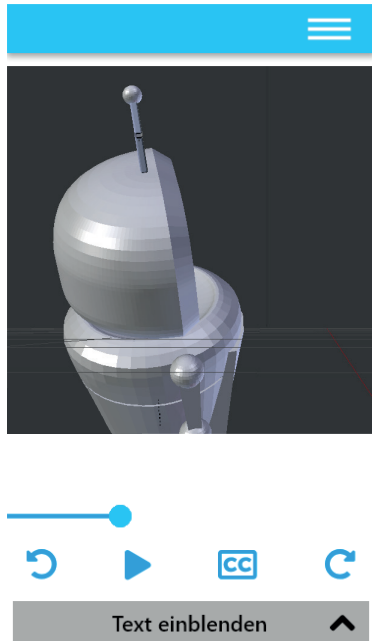


Sprechertext	Screen text / Notizen	Regieanweisungen
<p>090401 Bei der Kamera-Transformation, die auf Englisch viewing transformation genannt wird, werden zunächst die Betrachtungsparameter bestimmt.</p> <p>090402 Das heißt es wird die Betrachterposition, die Blickrichtung und das Sichtvolumen der Kamera bestimmt.</p> <p>090403 Beim Sichtvolumen oder auch View Frustrum handelt es sich um den Bereich, der von der Kamera eingefangen wird.</p> <p>090404 Nach dem Setzen der Kamera wird die Szene derartig transformiert, dass die Betrachterposition im Ursprung liegt. Dabei ist die Ausrichtung der Kamera entlang der z-Achse.</p> <p>090406 Nach diesem Schritt liegen die Weltkoordinaten des Meshes nun als Kamerakoordinaten vor</p>	<p>090401 Augenpunkt-Transformation = (engl.) Viewing Transformation</p> <p>090404 Veränderung der Position und Blickrichtung der Kamera</p>	<p>090401 -Es erscheint eine Einblendung der Begrifflichkeit</p> <p>090402 -Es erscheint ein Koordinatensystem mit Kamera und Objekt und einem Sichtvolumen</p> <p>090403 Das Sichtvolumen wird hervorgehoben</p> <p>090404 Die Kamera mitsamt Meshes wird in den Ursprung verschoben</p> <p>090406 Es erscheint ein Schema, in welchem klar wird, welche Koordinaten zu diesem Zeitpunkt vorliegen.</p>

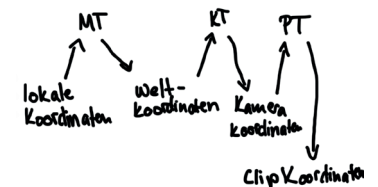
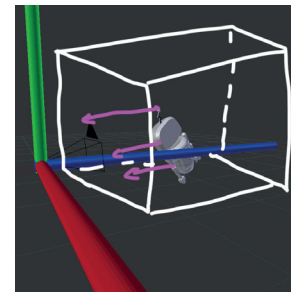
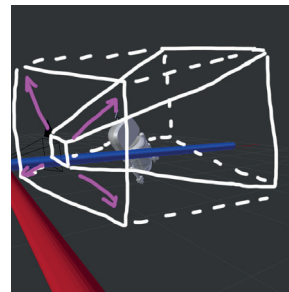
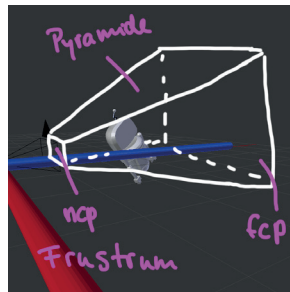
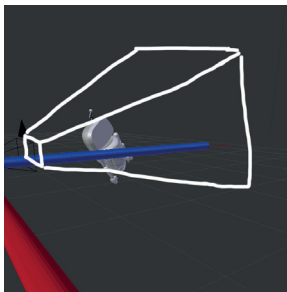
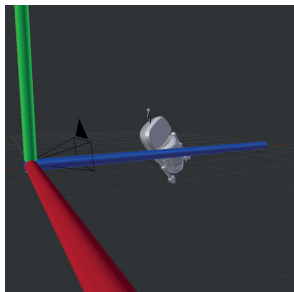
Kamera-Transformation  
=  
Viewing Transformation



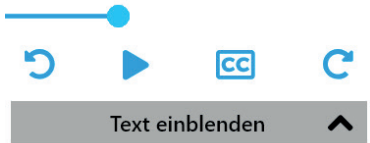
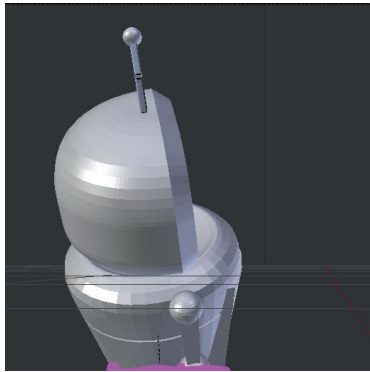
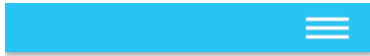
## 9.5 Projektions Transformation



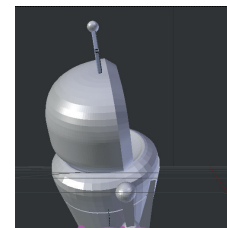
Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090501 Nach der Modell- und der Kamera Transformation befinden sich alle Eckpunkte, welche auch Vertices genannt werden, an den gewünschten Positionen.</p> <p>090502 Bei der Projektions-Transformation wird das View Frustrum näher betrachtet. Das Frustrum hat die Form eines Pyramidenstumpfes und besteht aus einer Far Clipping Plane und einer Near Clipping Plane, die auch Projektionsebene genannt wird.</p> <p>090503 Die Meshes die dargestellt werden sollen liegen innerhalb des Frustrums.</p> <p>090504 Um eine orthografische Projektion zu ermöglichen, muss die Near Clipping Plane derartig transformiert werden, dass ein Einheitswürfel entsteht.</p> <p>090505 Daraufhin werden die Vertices orthogonal auf die Projektionsebene projiziert.</p> <p>090406 Nach diesem Schritt liegen die Koordinaten als Clip-Koordinaten vor.</p>	<p>090501 Eckpunkt = Vertice</p>	<p>090501 Es erscheint eine Kurze Animation zu den vorherigen Transformationen</p> <p>090502 Es erscheint ein Frustrum, und die Bestandteile werden aufgezeigt</p> <p>090503 Die Meshes erscheinen innerhalb des Frustrums</p> <p>090504 Die Near Clipping Plane wird auf die selbe Größe der Far Clipping Plane transformiert</p> <p>090505 -Es erscheinen parallele Strahlen von den Vertices aus auf die Projektionsebene -Das Bild darauf wird erkennbar und das Frustrum staucht sich zusammen zu einer Ebene</p> <p>090506 Die Clip-Koordinaten erscheinen im Schema</p>

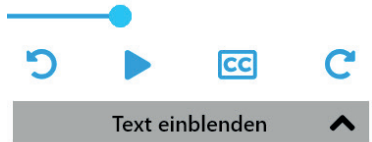
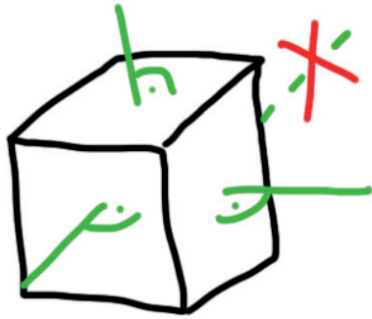


## 9.6 Clipping

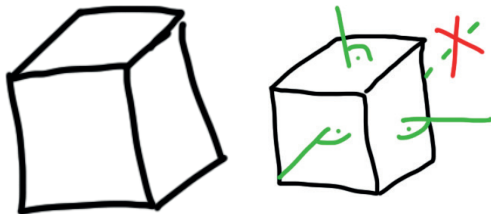
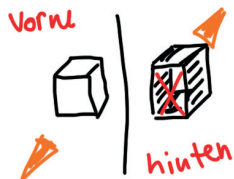


Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090601 Beim Clipping geht es darum Flächen, die vom sichtbaren Volumen nicht mehr eingefangen werden können aus der Szene zu entfernen.</p> <p>090602 Nach der Projektionstransformation wird überprüft, welche Meshes vollständig im sichtbarem Bereich 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.</p> <p>090605 Punkte die innerhalb der Grenze liegen werden in ihrer Geometrie belassen, Punkte außerhalb entfernt.</p> <p>090606 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>



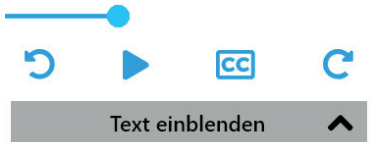
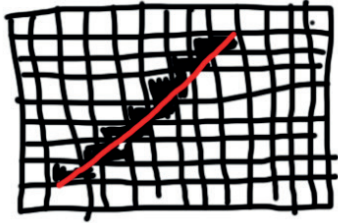


Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090701 Beim Culling geht es darum Flächen, die vom Betrachter nicht mehr wahrgenommen werden können aus der Szene zu entfernen.</p> <p>090702 Durch das Backface-Culling werden die Polygone aus der Szene entfernt, die vom Betrachter abgewandt sind.</p> <p>090703 Ob eine Fläche sichtbar oder nicht sichtbar ist wird mit Hilfe des Normalenvektors entschieden.</p> <p>090704 Ein Normalenvektor ist ein Vektor einer Fläche, der zu diesem orthogonal ist.</p> <p>090705 Zeigt der Normalenvektor zum Beispiel in Richtung der Kamera, hat es zur Folge, dass der Betrachter die Vorderseite sieht. Ist der Normalenvektor <math>n</math> von der Kamera abgewandt heißt das, dass es sich bei der Fläche um eine Rückseite eines-Meshes handelt.</p> <p>090706 Damit kann diese Fläche entfernt werden.</p>	<p>090701 Culling dient dazu Meshes die komplett außerhalb des sichtbaren Volumens wegzuschneiden</p> <p>090702 -spezielle Form: Backfaceculling</p> <p>090705 Normalenvektor steht senkrecht zur Fläche</p>	<p>090702 -Es erscheint ein Objekt mit Vorder- und Hinteransicht. Auf das Objekt ist eine Kamera gerichtet und der hintere wird entfernt</p> <p>090803-090804 -Es erscheint ein Objekt auf dem Normalen erscheinen. die Normalen die von der Kamera abgewandt sind werden entfernt.</p>

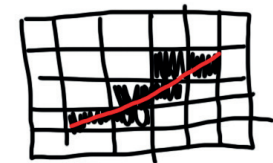
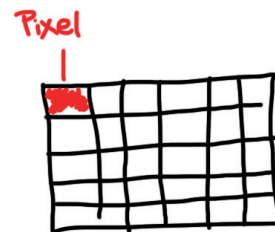
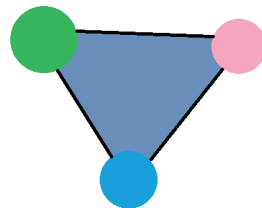
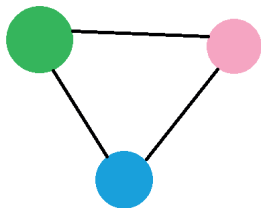




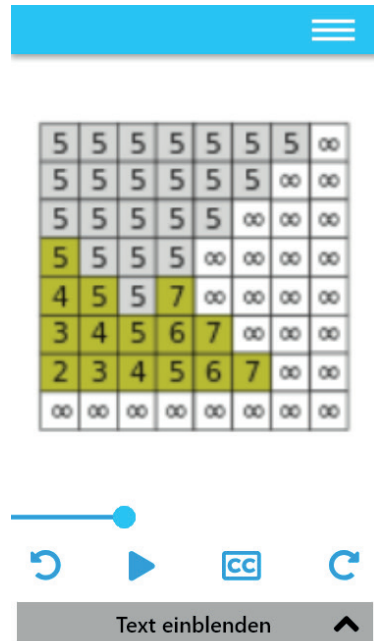
## 9.8 Rasterisierung



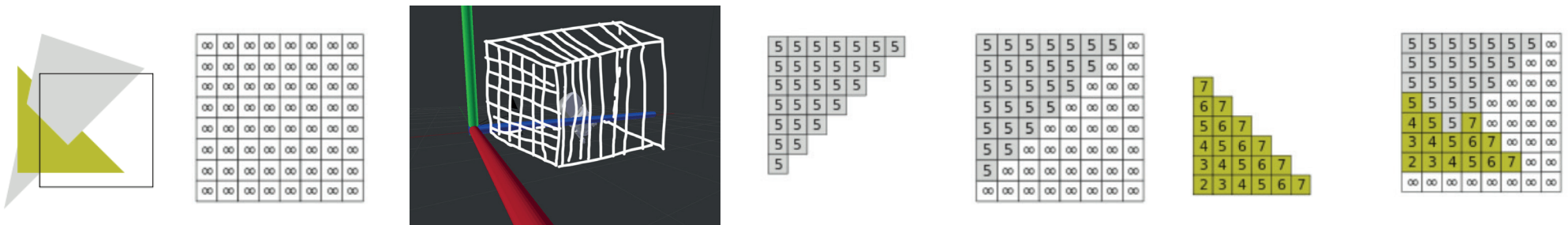
Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090801 Bis zu diesem Zeitpunkt liegt für jeden Vertex eines Polygons ein Farbwert vor. Um Das Bild jedoch später auf einem Monitor darstellen zu können, muss für jedes Pixel der Farbwert, durch Beleuchtungsberechnungen, berechnet werden.</p> <p>090802 Zur besseren Unterscheidbarkeit werden die Pixel in diesem Schritt Fragmente genannt.</p> <p>090803 Ein weiterer Grund für die Unterschiedliche Bezeichnung ist, dass ein Fragment mehrere Daten speichern kann. Dazu gehört zum Beispiel der Alpha-Wert, der Transparenzen beschreibt, und der z-Wert, der für die Verdeckungsrechnung wichtig ist.</p> <p>090804 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>



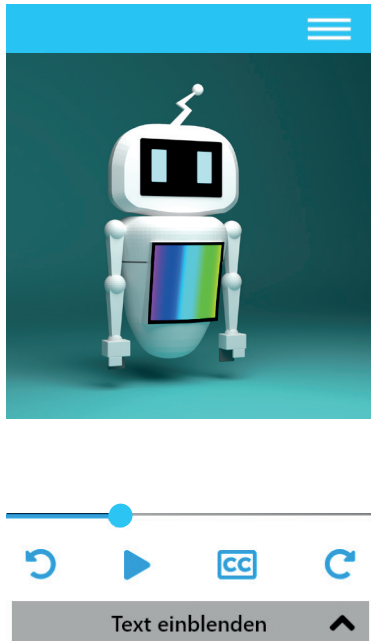
## 9.9 Verdeckungsberechnung/z-Buffer



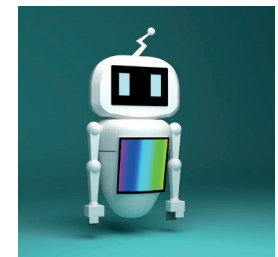
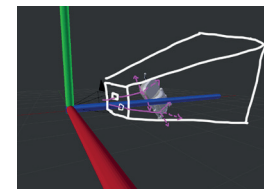
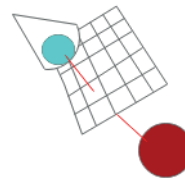
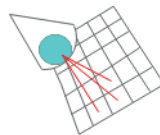
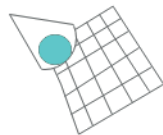
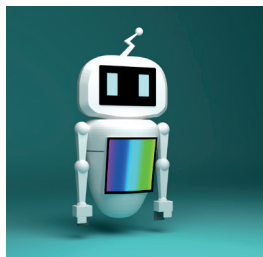
Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090901 Bei einer Szene in der mehrere Objekte zu sehen sind, kann es dazu kommen, dass ein Objekt A vor einem Objekt B ist, oder dieses auch schneidet.</p> <p>090902 Um dies korrekt darzustellen, werden in der Computergrafik Verdeckungsberechnungen, 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 gleicheBild nur mit falscher Verdeckungsberechnung</p> <p>090903-090905 Auf dem Frustrum wird ein Raster dargestellt. Alle Objekte werden auf dem Raster abgebildet. Gleichzeitig dazu wird ein 2-D-Schema nebenan gerastert</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>



## 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>091002 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.</p> <p>091006 Anschließend werden die gebrochenen bzw. die reflektierten Strahlen berechnet. Diese werden als Sekundärstrahlen bezeichnet.</p> <p>091007 Der Strahl endet, wenn er die maximale Anzahl von Schritten erreicht, auf kein weiteres Objekt oder auf eine Lichtquelle trifft.</p> <p>091008 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>



## 9.11 Raytracing – Interaktion



- ☐ Shading
- ☐ Raytracing

### Anweisungen

091201

Wähle mittels der Radio Button zwischen reinem Shading und Raytracing aus und betrachte die Änderungen.



Sprechertext	Screen text / Notizen	Regieanweisungen
<p>091201 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>091202 Die Volumengrafik basiert auf dem Strahlentransport, der beschreibt, wie sich Licht auf dem Weg durch ein Volumen verhält.</p> <p>091203 Beim Rendern einer Volumengrafik unterscheidet man vier Schritte:</p> <p>091204 <u>1. der Klassifikation:</u> 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.</p> <p>091205 <u>2. der Interpolation:</u> 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.</p> <p>091206 <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>091207 <u>4. der Composition:</u> 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.</p>	<p>091201 Volumengrafik = transparente Objekte</p> <p>Voxel = Gitterpunkt in einem dreidimensionalen Gitter.</p> <p>091203-091207 vier Render Schritte: 1. Klassifikation 2. Interpolation 3. Shading 4. Composition</p>	<p>091201 Es wird ein Voxelgitter eingeblendet und anhanddessen ein Voxel gezeigt</p> <p>091203 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>

