

# DREHBUCH RENDERING

Computergrafik.Online

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

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

**Version: 1.3**  
**Letzte Änderung: 21.10.2018**  
**Autor: Berdan Der**

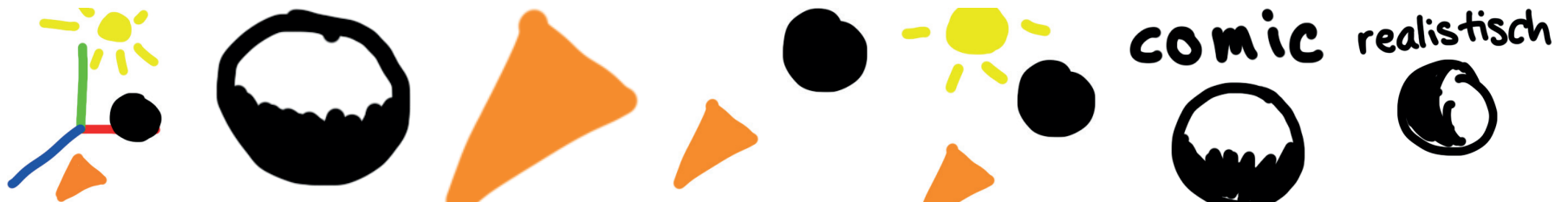


9.1 Einleitung/Anwendung	1
9.2 mathematische Grundlagen	2
9.3 Modell-Transformation	3
9.4 Augepunkt-Transformation	4
9.5 Projektions-Transformation	5
9.6 Window-Viewport-Transformation	6
9.7 Clipping und Culling	7
9.8 Rasterisierung	8
9.9 Verdeckungsrechnung – z-Buffer	9
9.10 Raytracing	10
9.11 Radiosity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



Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090101 Rendern heißt zu Deutsch Bildsynthese und bezeichnet den Vorgang, ein Bild aus einer Szene zu generieren. Die Szene besteht dabei aus Objekten, Kameras und Lichtquellen.</p>	<p>090101 - Rendern (dt. Bildsynthese) - Aus einer Szene wird ein Bild erzeugt</p>	<p>090101/090102 -Es erscheint eine Einblendung der Begrifflichkeit -Danach erscheint eine Szene mit Objekt, Kamera und Licht</p>
<p>090102 Dies geschieht in mehreren Stufen, die zusammengefasst als Rendering Pipeline bezeichnet werden.</p>	<p>Hier die Stufen vorstellen z.B. 1. Alle Flächen werden auf das Weltkoordinatensystem bezogen 2. Alle Flächen werden auf das Kamerakordinatensystem bezogen 3. Nicht sichtbare Flächen werden eliminiert (Clipping/Culling) 4. Beleuchtung auf Vertices wird berechnet 5. Vertices werden auf die Bildebene projiziert 6. Es wird gerastert und Fragmente berechnet 7. Verdeckungen auf Fragment-Ebene wird berechnet (Z-Buffer) (Quelle: Wikipedia, bitte prüfen...!)</p>	<p>090102 -Es werden Szenen gezeigt, die auf verschiedene Arten gerendert wurden (comic/realistisch)</p>
<p>090103 Beim Durchlaufen eines Objektes durch eine Rendering Pipeline, wird dieses mehreren unterschiedlichen Transformationen unterzogen. Des Weiteren werden zum Beispiel Verdeckungsberechnungen und Beleuchtungsberechnungen aufgestellt.</p>	<p>Der Absatz raus, denn er bezieht sich nur auf den letzten Schritt</p>	<p>090103 -Es werden Szenen gezeigt, die auf verschiedene Arten gerendert wurden (comic/realistisch)</p>
<p>090104 Das Rendering ist in der Computergrafik ein unumgängliches Thema, das zum Beispiel dafür sorgt, dass Animationsfilme und Computerspiele einen eigenen Look bekommen. So sieht manches z. B. eher comichaft und anderes realistisch aus.</p>		<p>090104 -Es werden Szenen gezeigt, die auf verschiedene Arten gerendert wurden (comic/realistisch)</p>

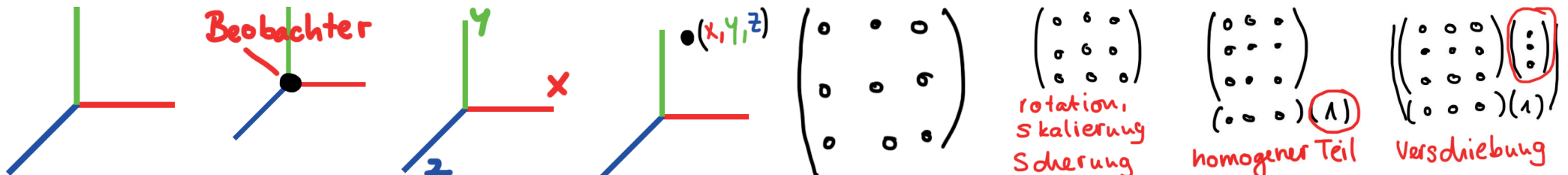
rendern  
= Bildsynthese



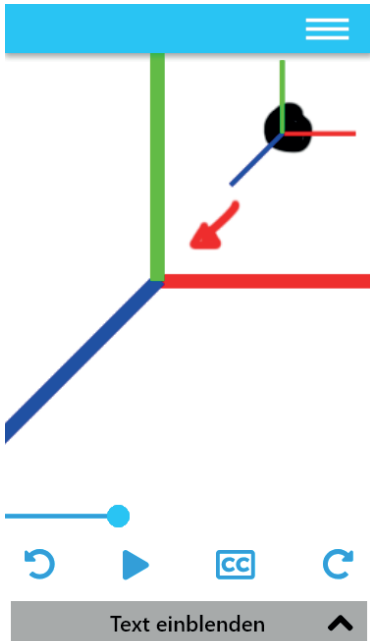
Ist diese Seite  
erforderlich um die  
anderen Inhalte zu  
begreifen? Wenn nicht,  
raus damit!!



Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090201 In der Computergrafik wird ein dreidimensionales Koordinatensystem mit drei aufeinander senkrecht stehenden Achsen x, y und z verwendet.</p>	<p>090201 dreidimensionales Koordinatensystem: x,y,z</p>	<p>090201 -Es erscheint ein Koordinatensystem</p>
<p>090202 Der Beobachter wird standardmäßig in den Ursprung gesetzt. Die positive x-Achse zeigt bezüglich des Bildschirms nach rechts. Die positive y-Achse nach oben und die positive z-Achse steht senkrecht zum Bildschirm.</p>	<p>090202 Beobachter ist im Ursprung</p>	<p>090202 -Der Beobachter wird in den Ursprung gesetzt und die Koordinatenachsen werden benannt</p>
<p>090203 Ein Punkt in einer Szene bzw. in einem dreidimensionalen Raum wird durch drei Koordinaten beschrieben - x, y und z. Eine Transformation eines Objekts kann durch eine 3x3-Matrix beschrieben werden. Dieser Teil ist für die Rotation, Skalierung und Scherung zuständig.</p>	<p>090203 Transformation durch eine 3x3-Matrix</p>	<p>090203 -Es erscheint ein Punkt in der Szene, dessen Koordinatenanteile definiert werden -Es erscheint eine 3x3-Matrix und es wird gezeigt wofür dieser Anteil zuständig ist</p>
<p>090204 Um im späteren Verlauf der Rendering Pipeline einige Dinge zu vereinfachen führt man noch einen homogenen Teil ein, der unabhängig ist. Dieser Faktor w liegt standardgemäß bei 1.</p>	<p>090204 homogener Teil w = 1</p>	<p>090204 -Es erscheint ein homogener Teil</p>
<p>090205 Des Weiteren enthält die nun 4x4 große Matrix noch einen Anteil der zur Verschiebung dient.</p>		<p>090205 -Es erscheint ein weiterer Anteil und wird auf eine 4x4-Matrix erweitert</p>



### 9.3 Modell Transformation

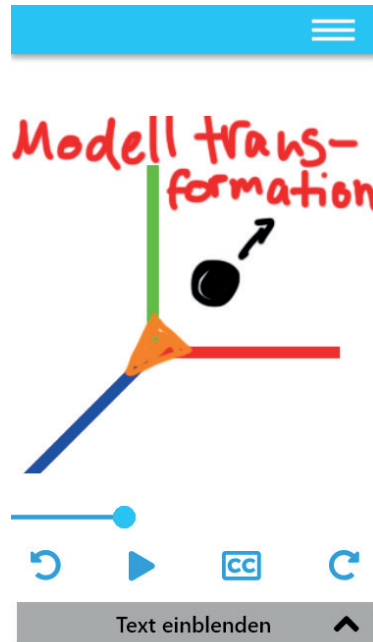


Sprechertext		Regieanweisungen
<p>090301 Ein Objekt, dass sich in einem dreidimensionalen Raum aufhält, wird normalerweise durch ihr lokales Koordinatensystem definiert. Dabei handelt es sich um die Objektkoordinaten, jedoch müssen sie in ein globales Koordinatensystem überführt werden.</p> <p>090302 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>090301 lokales Koordinatensystem = Objektkoordinaten globales Koordinatensystem = Weltkoordinaten</p> <p>090302 lok. KS -&gt; 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>

Der Ansatz ist schlecht. Besser das Mesh im Raum darstellen, mit den verschiedenen verketteten Transformationen. Dann die resultierende Transformation darstellen. Das ist die benötigte.



## 9.4 Augenpunkt-Transformation



Sprechertext	Screen- / Notizen	Screen- / Notizen
<p>090401 Bei der Augenpunkt-Transformation, die auf Englisch Viewing Transformation genannt wird, ändert man die Position und die Blickrichtung des Augenpunktes bzw. 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 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>090404 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>090404 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 Ein Koordinatensystem mit Objekt und Kamer wird gezeigt, wobei Kamera und Objekt aufeinander liegen</p> <p>090403/090404 -Zuletzt wird das Objekt verstezt.</p>

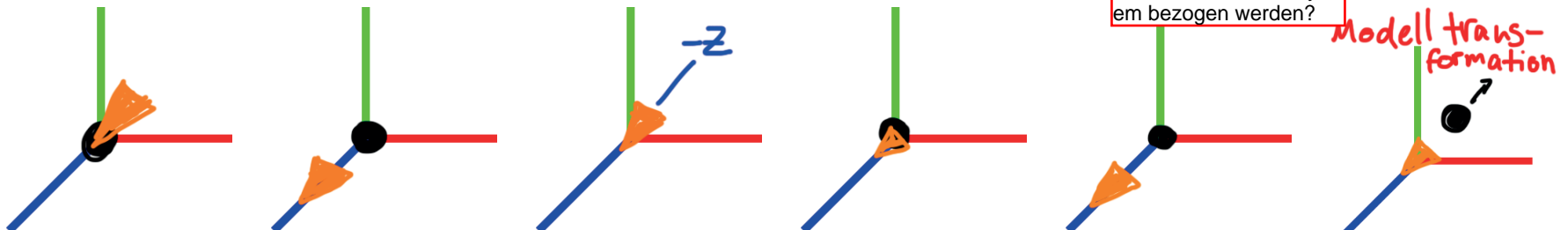
Bei Wikipedia heißt es Kameratransformation. Das erscheint mir eingängiger

Ich bezweifle, dass beim Rendern die Kamera verschoben wird

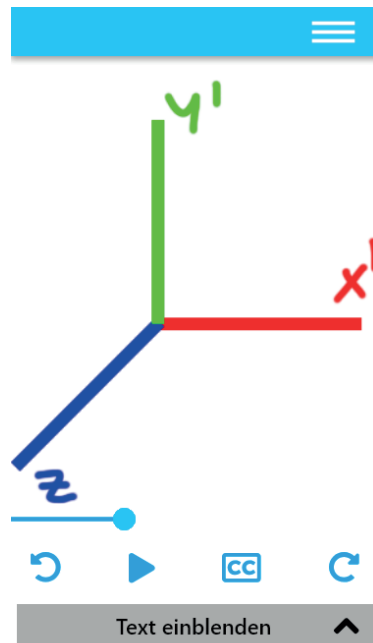
Die ganze Seite ist völlig unverständlich und die Relevanz wird nicht klar.

Geht es nicht einfach darum, dass alle Meshes nun auf das Kamerakoordinatensystem bezogen werden?

Augenpunkt-Transformation = Viewing Transformation



## 9.5 Projektions Transformation



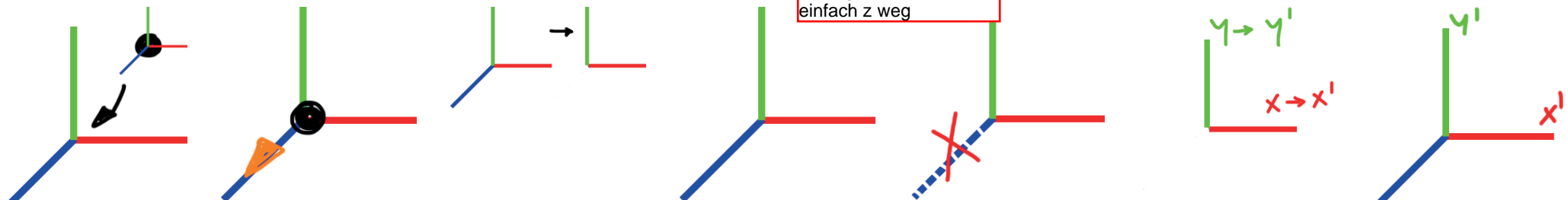
Ist es nicht so, dass hier in einen Cubus mit den Koordinaten -1,-1,-1 bis 1,1,1 transformiert wird?

Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090501 Nach der Modell- und der Viewing Transformation befinden sich alle Eckpunkte, welche auch Vertices genannt werden, an der gewünschten Position.</p> <p>090502 Um aber ein zweidimensionales Abbild der dreidimensionalen Szene zu erhalten, muss eine Projektions-Transformation vollzogen werden.</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>090504 Da man jedoch später die z-Werte noch braucht, bleiben diese erhalten und nur die x-y-Ebene wird transformiert.</p> <p>090505 In der Praxis sind besonders zwei Projektions-Transformationen von großer Bedeutung: die orthografische und die perspektivische.</p>	<p>090501 Eckpunkt = Vertice</p> <p>090502 Projektion = KS transformieren</p> <p>090504 Vorgehensweise: z-Werte erhalten und x-y-Ebene transformieren</p> <p>090505 orthografische und die perspektivische Projektions Transformation</p>	<p>090501 Es erscheint eine Kurze Animation zu den vorherigen Transformationen</p> <p>090503 Es erscheint wieder ein dreidimensionales KS bei welchem die z-Achse verschwindet und die übrigen umgewandelt werden</p> <p>090504 die z-Achse erscheint wieder</p>

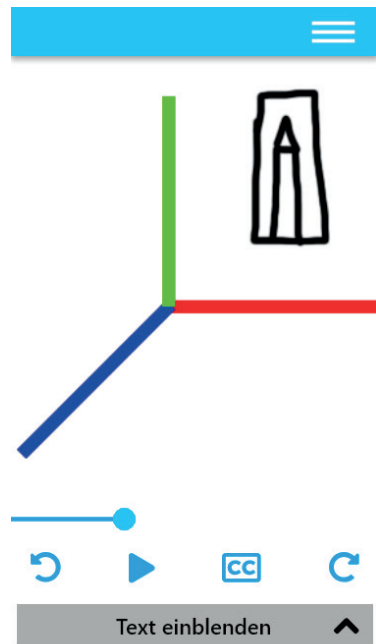
Die Herumschieberei ist Mist. Das wird doch beim Modellieren der Szene gemacht. Hier ist doch entscheidend, dass nun die Vertices in einem für die Projektion günstigen Koordinatensystem definiert sind.

völlig unverständlich

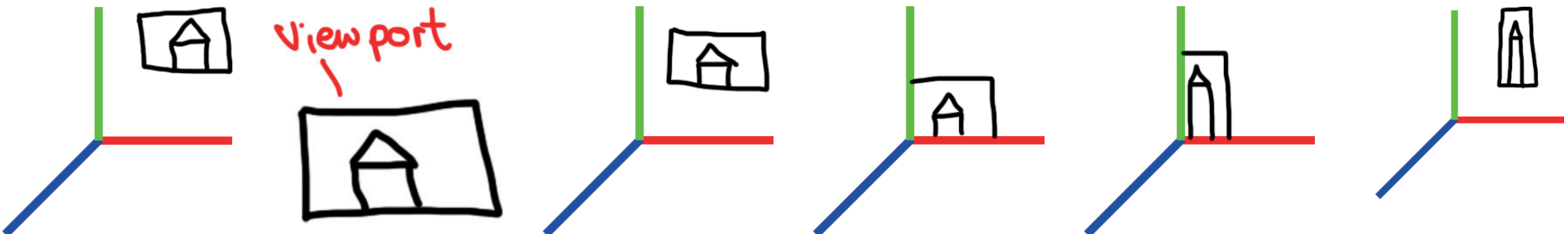
Das Schöne bei der Orthografischen: es fällt einfach z weg



## 9.6 Window-Viewport-Transformation



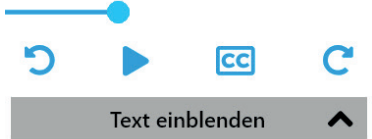
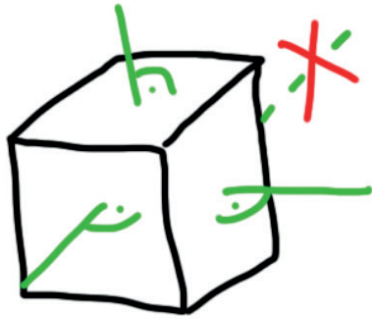
Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090601 Ein Viewport ist ein Ausschnitt einer Szene, der später als Bild dargestellt werden soll. Er entspricht dem sichtbaren Bereich einer Szene.</p>	<p>090601 Viewport = Ausschnitt der Szene, der als Bild generiert werden soll</p>	<p>090601 -Es erscheint ein dreidimensionales KS mit einem Objekt, das von der Projektionsfläche eingefangen wird -Daraufhin erscheint eine Bild des Viewports</p>
<p>090602 Da Objekte durch Weltkoordinaten definiert werden und diese in einem Viewport dargestellt werden sollen müssen die Weltkoordinaten auf Bildschirmkoordinaten umgerechnet werden. Diese werden auch Window-Koordinaten genannt.</p>	<p>jetzt sollten wir im View-Space-Koordinatensystem sein!</p> <p>Warum brauchen wir Screenkoordinaten um im Viewport darzustellen?</p>	<p>090602 -Das erste Bild wird wieder gezeigt</p>
<p>090603 Die Umwandlung der Koordinaten wird durch eine Window-Viewport-Transformation erreicht.</p>	<p>Window-Viewport-Transformation = Umwandlung Weltkoordinaten in Window-Koordinaten</p>	
<p>090604 Als erstes wird der Viewport durch eine Translation in den Koordinatenursprung verschoben.</p>		<p>090604 Der Viewport wird in den Ursprung verlegt</p>
<p>090605 Daraufhin wird der Viewport im Ursprung auf die Größe des Bildschirmfensters angepasst.</p>		<p>090605 Die Größe des Viewports wird nun angepasst</p>
<p>090606 Als letztes wird der Viewport an die richtige Stelle auf dem Bildschirm positioniert.</p>	<p>Alles nicht nachvollziehbar</p>	<p>090606 Der Viewport wird wieder an der richtigen Stelle positioniert</p>





## 9.7 Clipping und Culling

An der Stelle erklären, an der es in der Pipeline kommt.



### Sprechertext

090701

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

090702

Das Clipping wird immer eingesetzt, ohne dass ein Culling Algorithmus aktiv war.

090703

Nach der Projektionstransformation wird überprüft, ob alle Primitive vollständig im sichtbarem Bereich liegen.

090704

Elemente die gänzlich außerhalb des Sichtfensters liegen werden komplett entfernt.

090705

Für jede Kante des Sichtfensters wird geprüft, ob sich der Vertex eines Objekts inner- oder außerhalb der Kante befindet. Punkte die innerhalb der Granze 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.

090706

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

090707

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

090708

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.

Das heißt, wenn Culling dann kein Clipping?

### Screentext / Notizen

090701

Clipping bzw Culling dient dazu Geometrien außerhalb des sichtbaren Volumens wegzuschneiden

### Regieanweisungen

090701-090704

-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

090705

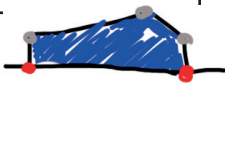
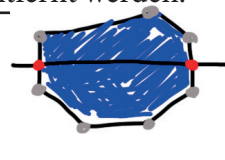
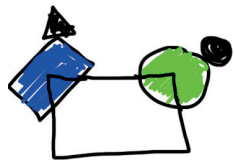
-Es erscheint ein Objekt mit Vertices und eine kante des Sichtfensters  
-Daraufhin werden neue Vertices berechnet und der überstehende Teil wird abgeschnitten

090706

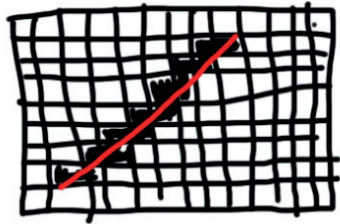
-Es erscheint ein Objekt mit Vorder- und Hinteransicht. Auf das Objekt ist eine Kamera gerichtet und der hintere wird entfernt

090707-090708

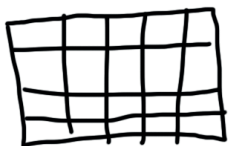
-Es erscheint ein Objekt auf dem Normalen erscheinen. die Normalen die von der Kamera abgewandt sind werden entfernt.



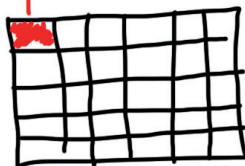
## 9.8 Rasterisierung



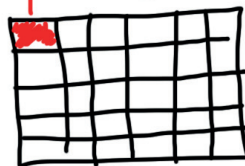
Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090801 Alle Ausgabegeräte haben eine feste Auflösung mit einem festen Raster.</p> <p>090802 Dieses basiert auf Pixeln. Da Pixel keine Punkte sondern Flächen sind, müssen alle Objekte innerhalb des Renderings einer Rasterisierung unterzogen werden. Hierbei werden die endgültigen Farbwerte durch Interpolation der Farbwerte zwischen den Vertices berechnet. Zur besseren Unterscheidbarkeit werden die Flächen in diesem Schritt auch Fragmente genannt.</p> <p>090803 Bei einer Rasterisierung werden die Objekte auf dem Raster dargestellt, indem man sie den Flächen annähert.</p> <p>090804 Falls ein Raster eine niedrige Auflösung hat, jedoch ein komplexeres Objekt darstellen soll, kann es dazu kommen, dass das Endergebnis Stufen aufweist. Hier spricht man von einer Unterabtastung bzw. dem Aliasing. Dies ist häufig bei schrägen Kanten ein Problem.</p>	<p>Vorsicht, es gibt immer noch Vectorscreens</p> <p>Das klingt irreführend. Den Begriff Flächen nutzen wir in anderem Kontext. Was soll die Aussage bringen?</p> <p>090802 Fragemnt = Rasterfläche Rasterisierung = Interpolation der Farbwerte zwischen den Vertices eines Polygons</p> <p>090803 Rasterisierung = Annäherung der Punkte an Flächen</p> <p>090804 Aliasing = Unterabtastung</p> <p>Völlig unverständlich</p> <p>Das kennen wir schon aus Kapitel Vektorgrafik: Prüfen und entsprechend raus damit.</p>	<p>090801 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> <p>090804 Da das zuletzt gerasterte Objekt unterabgetastet wurde, wird hier der Begriff Aliasing eingeführt und das Objekt wird auf einem feinerem Raster erneut gerastert.</p>



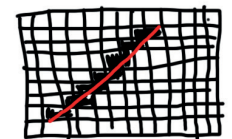
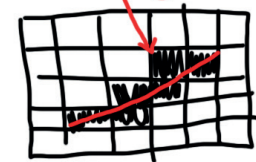
Pixel



Pixel=Fragment



Aliasing



## 9.9 Verdeckungsrechnung/z-Buffer



5	5	5	5	5	5	5	00
5	5	5	5	5	5	00	00
5	5	5	5	5	00	00	00
5	5	5	5	00	00	00	00
4	5	5	7	00	00	00	00
3	4	5	6	7	00	00	00
2	3	4	5	6	7	00	00
00	00	00	00	00	00	00	00



### Sprechertext

090901

Bei der Verdeckung geht die menschliche Wahrnehmung davon aus, dass ein Objekt A, das ein Objekt B verdeckt näher am Betrachter liegen muss. Wird dies jedoch nicht korrekt dargestellt ist der Beobachter irritiert und das Bild wirkt unrealistisch.

090902

Um dies korrekt darzustellen wird der z-Buffer-Algorithmus gebraucht.

090903

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

090904

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

090905

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

Der Satz irritiert eher, als dass er hilft.  
Es ist intuitiv, dass Verdeckungen dargestellt werden müssen

Im Text also darauf eingehen: Flächen druchdringen sich.icc

090902

Verdeckungsrechnung durch z-Buffer-Algorithmus

090903

z-Buffer-Algorithmus speichert für jeden Pixel z-Wert

090905

Je kleiner der z-Wert eines Pixels, desto näher ist er am Betrachter

Eher Fragmente, oder? Der Pixel wird am Ende definiert...

Fragmente-Begriff klären!

### Regieanweisungen

090901

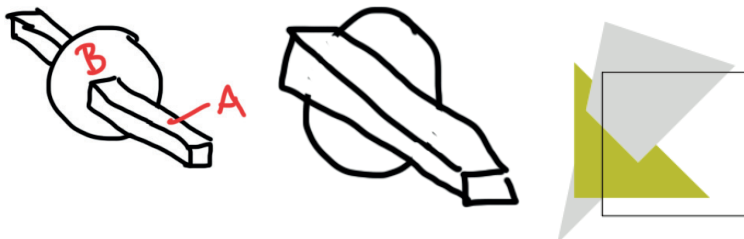
-Es erscheint ein Bild, bei welchem sich Objekte überschneiden  
-Daraufhin erscheint das gleiche Bild nur mit falscher Verdeckungsrechnung

090903-090905

Es wird ein Raster dargestellt. Alle Objekte werden auf dem Raster abgebildet.

Falls der aktuell gerasterte Punkt näher am Betrachter liegt als der davor gerasterte Punkt, wird dieser durch das aktuelle ersetzt.

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.



00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00

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	00
5	5	5	5	5	5	00	00
5	5	5	5	5	00	00	00
5	5	5	5	00	00	00	00
5	5	5	00	00	00	00	00
5	5	00	00	00	00	00	00
5	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00

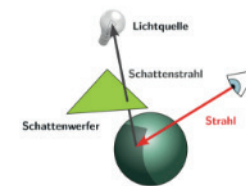
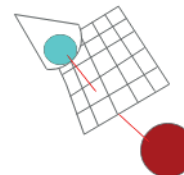
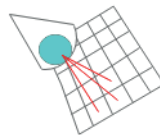
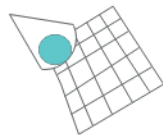
7							
6	7						
5	6	7					
4	5	6	7				
3	4	5	6	7			
2	3	4	5	6	7		

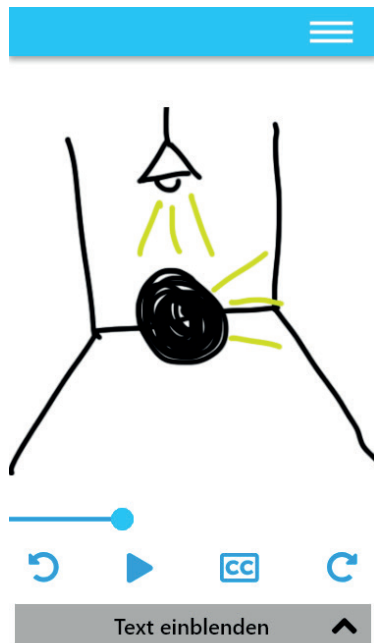
5	5	5	5	5	5	5	00
5	5	5	5	5	5	00	00
5	5	5	5	5	00	00	00
5	5	5	5	00	00	00	00
4	5	5	7	00	00	00	00
3	4	5	6	7	00	00	00
2	3	4	5	6	7	00	00
00	00	00	00	00	00	00	00

## 9.10 Raytracing



Sprechertext	Was bedeutet Korrektheit in diesem Zusammenhang?	Screentext / Notizen	Regieanweisungen
091001 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.			091001 Es wird ein Bild von einer Szene gezeigt, welches den Raytracing-Algorithmus verwendet.
091002 Raytracing – zu Deutsch „Strahlen verfolgen“ – ist in erster Linie ein Algorithmus zur Verdeckungsrechnung.		091002 Raytracing (dt. Strahlen verfolgen“)	
091003 Dieses basiert auf dem Aussenden von Strahlen vom Betrachterblickpunkt aus.		091003 Aussendung von Strahlen vom Betrachter aus	091003 Es wird ein Auge eingeblendet
091004 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.	Pixel?	091004 Für jedes Rasterelement ein Strahl	091004 Es erscheint ein Raster. Es schießen Strahlen aus dem Auge durch jedes Rasterelement. Daraufhin wird geprüft, ob der Strahl ein Objekt trifft.
091005 Für jeden Schnittpunkt werden daraufhin Farbbeiträge berechnet.	Das erklärt überhaupt nicht den eingangs so wichtig erklärten Realismus!	091005 Farben für jeden Schnittpunkt berechnen	





Sprechertext	Screen text / 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 Strahlenaustausch zwischen Oberflächenstücken, den sogenannten Patches, statt.</p> <p>091104 Von jeder Fläche geht ein konstanter Lichtstrom aus, der sich aus emittiertem Lichtstrom, falls es sich um eine Lichtquelle handelt, und reflektiertem 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>Was ist ein Strahlenaustausch</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 Raytracing und Radiosity aus. Betrachte die Änderungen.

Kann man Shading nicht wählen???





Sprechertext	Screentext / Notizen	Regieanweisungen
<p>091301</p> <p>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</p> <p>Die Volumengrafik basiert auf dem Strahlentransport, der beschreibt, wie sich Licht auf dem Weg durch ein Volumen verhält.</p> <p>091303</p> <p>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</p> <p>Volumengrafik = transparente Objekte</p> <p>Voxel = Gitterpunkt in einem dreidimensionalen Gitter.</p> <p>091303</p> <p>vier Render Schritte:</p> <ol style="list-style-type: none"><li>1. Klassifikation</li><li>2. Interpolation</li><li>3. Shading</li><li>4. Composition</li></ol>	<p>091301</p> <p>Es wird ein Voxelgitter eingeblendet und anhanddessen ein Voxel gezeigt</p> <p>091303</p> <p>Die vier Schritte werden erklärt:</p> <ol style="list-style-type: none"><li>1) Es werden Eigenschaften verschiedener Transparenzstufen gezeigt</li><li>2) Voxel werden am Lichtstrahl interpoliert</li><li>3) Die Voxelflächen erhalten Normalen und eine Beleuchtung</li><li>4) Die unterschiedlichen Lichtstufen einer Linie werden miteinander verrechnet</li></ol> <p>Zum Schluss wird eine Volumengrafik eingeblendet, die sich dreht.</p>

