

DREHBUCH RENDERING

Computergrafik.Online

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

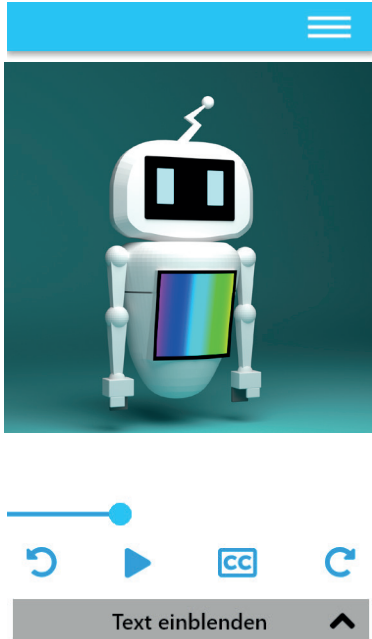
Hochschule Furtwangen University
Fakultät Digitale Medien

Version: 1.6
Letzte Änderung: 11.12.2018
Autor: Berdan Der



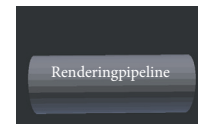
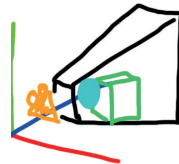
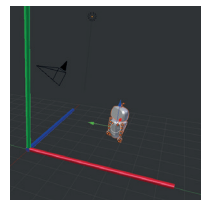
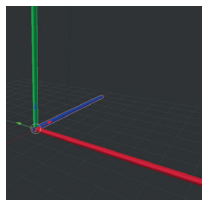
9.1 Einleitung	1
9.2 Modell-Transformation	2
9.3 Kamera-Transformation	3
9.4 Projektions-Transformation	4
9.5 Projektions-Transformation - Interaktion	5
9.6 Clipping	6
9.7 Culling	7
9.8 Rasterisierung	8
9.9 Verdeckungsrechnung – z-Buffer	9
9.10 Raytracing	10
9.11 Raytracing - Interaktion	11
9.12 Volumengrafik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

9.1 Einleitung/Anwendung

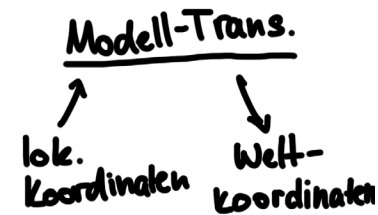


Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090101 Rendern stammt vom englischen Wort „to render“ und heißt zu deutsch „etwas ausgeben“. Das Rendering bezeichnet den Vorgang, ein Bild zu generieren. Beim Rendern einer 3D-Szene werden im Wesentlichen Meshes, Lichtquellen und Kame- ras berücksichtigt. Daraufhin erfolgt eine Projektion der Szene auf die Projektionsebene.</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 Beleuch- tungsberechnungen 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 Pipe- line</p>	<p>090101 -Es erscheint eine Einblendung der Begrifflichkeit -Danach erscheint eine Szene mit Objekt, Kamera und Licht -Daraufhin erscheint das Kame- ra-Frustum mit dem Mesh darin. Dieses wird auf die Near-Plane (Pro- jektionsebene)abgebildet</p> <p>090102 -Es erscheint eine Rendering Pipe- line, die in Transformation und Rast- erisierung aufgeteilt wird. -Daraufhin läuft ein Mesh durch</p> <p>090103 - Das fertige Bild mit Texturen wird angezeigt</p>

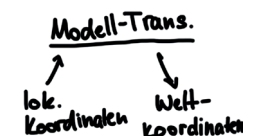
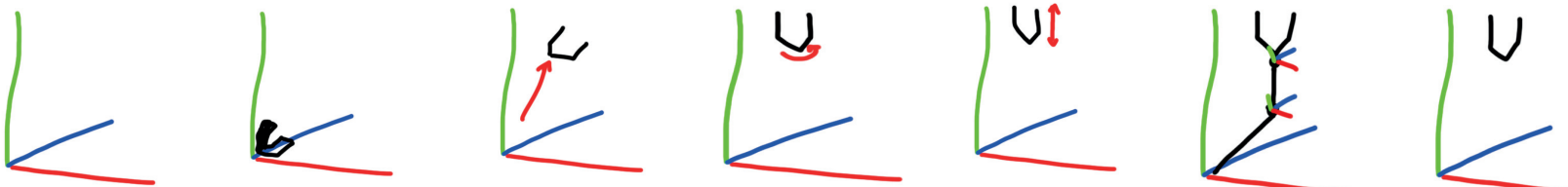
rendern
=
Bildsynthese



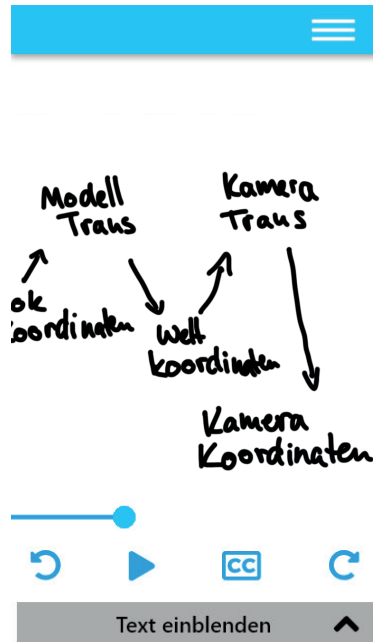
9.2 Modell Transformation



Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090201 Ein Mesh, wird normalerweise durch sein lokales Koordinatensystem definiert. Ohne Transformationen würden alle Objekte im Weltursprung liegen.</p> <p>090202 Durch Translationen, Rotationen und Skalierungen wird ein Mesh an die gewünschte Stelle im Raum gebracht.</p> <p>090203 Die endgültige Position eines Meshes hängt von der Reihenfolge der Transformationen in der Szenenhierarchie ab.</p> <p>090204 Um diese Transformationsverkettung aufzulösen werden die Vertexkoordinaten des Meshes in das Weltkoordinatensystem übertragen.</p> <p>090205 Von nun an liegen die Koordinaten nicht mehr als transformierte lokale Koordinaten, sondern in Weltkoordinaten vor.</p>	<p>090201 lokales Koordinatensystem = Objektkoordinaten</p> <p>090202 Verschiebung (Translation) Drehung (Rotation) Vergrößerung bzw. Verkleinerung (Skalierung)</p> <p>090204 Modelltransformation: lokales Koordinatensystem --> globales Koordinatensystem</p>	<p>090201 -Es erscheint ein Koordinatensystem mit einem Objekt (Roboterhand) im Weltursprung -Auf dem Objekt erscheint ein Objektkoordinatensystem</p> <p>090202 -Das Objekt wird verschiedenen Transformationen unterzogen</p> <p>090203 -Es erscheint zusätzlich der Oberarm gefolgt vom Unterarm. Darufhin wird der Arm verketteten Transformationen unterzogen</p> <p>090204 Der Großteil der Arms bist auf die Hand verschwindet, da man diese nun alleine betrachten kann</p> <p>090205 Es erscheint ein Schema, in welchem klar wird, welche Koordinaten zu diesem Zeitpunkt vorliegen.</p>



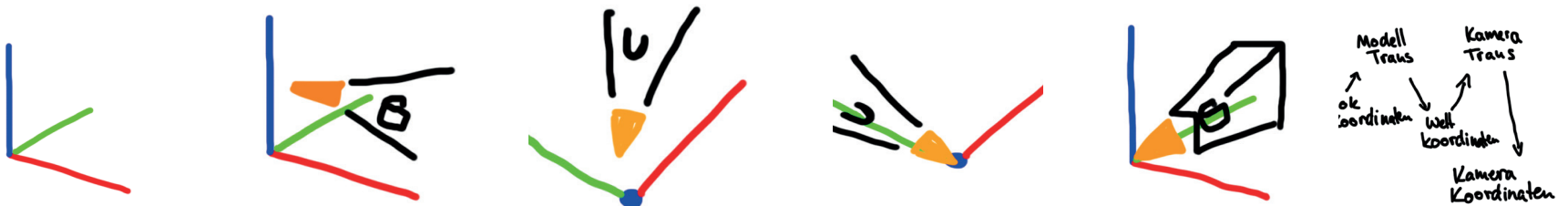
9.3 Kamera-Transformation



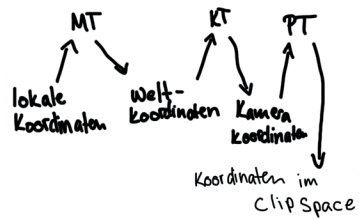
Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090301 Bei der Kamera-Transformation, die auf Englisch viewing transformation genannt wird, werden die Kamera und alle Meshes so transformiert, dass die Kamera im Ursprung liegt.</p> <p>090302 Die Ausrichtung der Kamera verläuft nach der Transformation entlang der z-Achse.</p> <p>090303 Nach diesem Schritt liegen die Weltkoordinaten des Meshes nun als Kamerakoordinaten vor.</p>	<p>090301 Kamera-Transformation = (engl.) Viewing Transformation</p> <p>090301-090302 Veränderung der Position und Blickrichtung der Kamera</p>	<p>090301-090302 -Es erscheint eine Einblendung der Begrifflichkeit -Es erscheint ein Koordinatensystem mit Kamera und Objekt und einem Sichtvolumen -Die Kamera mitsamt Meshes wird in den Ursprung verschoben und ausgerichtet</p> <p>090303 Es erscheint ein Schema, in welchem klar wird, welche Koordinaten zu diesem Zeitpunkt vorliegen.</p>

Kamera-Transformation

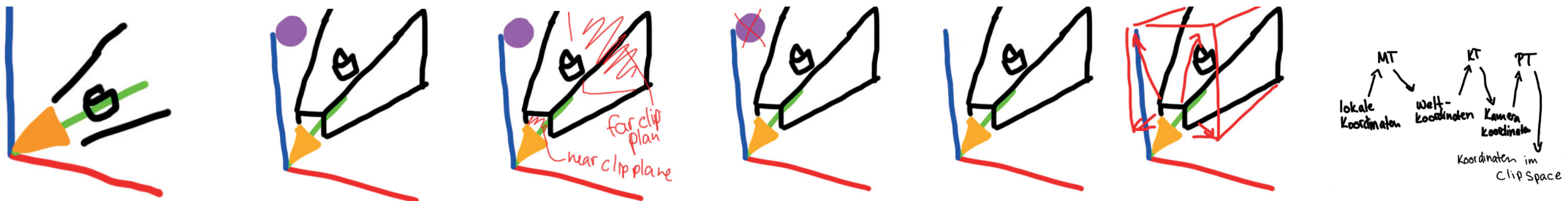
=
Viewing Transformation



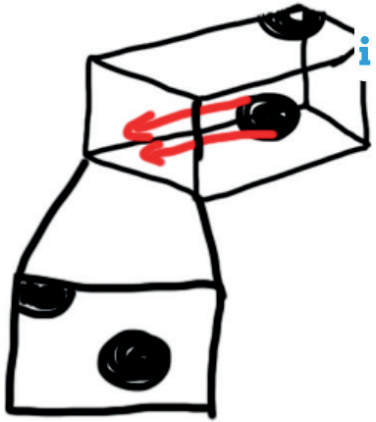
9.4 Projektions Transformation



Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090401 Nach der Modell- und der Kamera-Transformation liegen alle Koordinaten der Meshes so vor, dass sie gut weiterverarbeitet werden können.</p> <p>090402 Bei der Projektions-Transformation wird das View Frustum näher betrachtet. Bei der perspektivische Projektion hat das Frustum die Form eines Pyramidenstumpfes und besitzt eine Far Clipping Plane und eine Near Clipping Plane, die auch Projektionsebene genannt wird.</p> <p>090403 Meshes außerhalb des Frustums werden nicht dargestellt.</p> <p>090404 Als nächstes wird das das Frustrum mitsamt der Inhalte derart transformiert, dass ein Quader entsteht.</p> <p>090405 Nach diesem Schritt liegen die Koordinaten im sogenannten Clip-Space vor.</p> <p>090406 Die Meshes sind nun perspektivisch verzerrt. Bei der orthografischen Projektion sind die Meshes nicht perspektivisch verzerrt, da das Frustrum schon ein Quader ist.</p>	<p>090401</p> <p>090402 -Frustum bei perspektivischer Projektion = Pyramidenstumpf -Near Clipping Plane = Projektionsebene</p>	<p>090401 Es erscheint die Szene, wie sie nach der Kamera-Transformation vorliegt</p> <p>090402 Es erscheint ein Frustrum, und die Bestandteile werden aufgezeigt</p> <p>090403 Die Meshes die außerhalb des Frustums liegen werden entfernt</p> <p>090404 Die Near Clipping Plane wird auf die selbe Größe der Far Clipping Plane transformiert</p>



9.5 Projektions-Transformation– Interaktion



☐ perspektivische Projektion

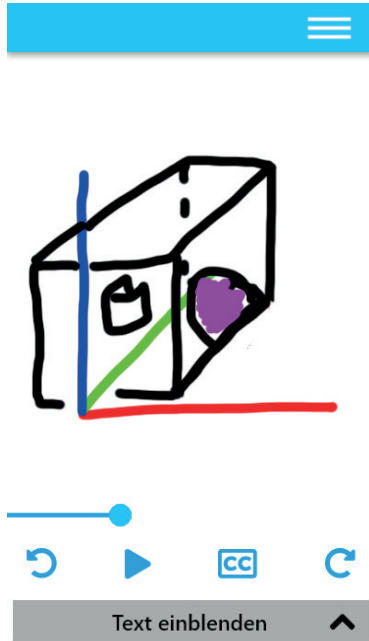
☒ orthografische Projektion

Anweisungen

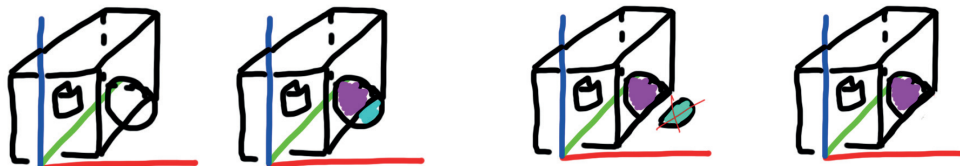
090501

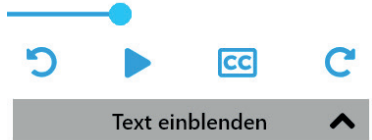
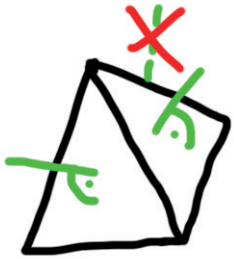
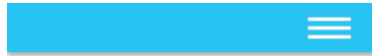
Wähle zwischen perspektivischer und orthografischer Projektion aus und betrachte die Änderungen.

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 Meshes, die teilweise im Frustum liegen, werden geschnitten und in außen- und innenliegende Meshes aufgeteilt.</p> <p>090604 Meshes die gänzlich außerhalb des Frustums liegen werden komplett entfernt.</p>	<p>090601 Clipping dient dazu Geometrien außerhalb des sichtbaren Volumens wegzuschneiden</p>	<p>090601 Es erscheint eine Szene mit Meshes. Des Weiteren wird das Frustum abgebildet</p> <p>090602 Eine Fläche wird überprüft, ob sie Meshes schneidet</p> <p>090603 Das Mesh wird in zwei Teile zerlegt(inner- und außerhalb). Das Stück Mesh das außerhalb liegt stößt sich leicht ab und verschwindet dabei</p> <p>090604 Meshes, die sich gänzlich außerhalb des Frustums befinden, verschwinden auch</p>

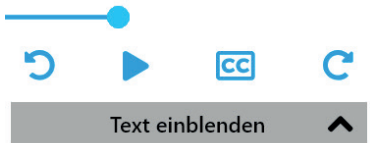
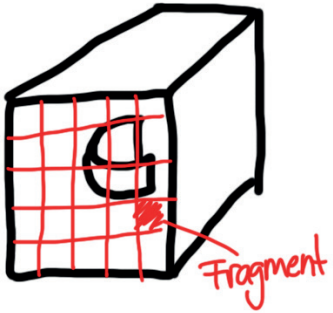




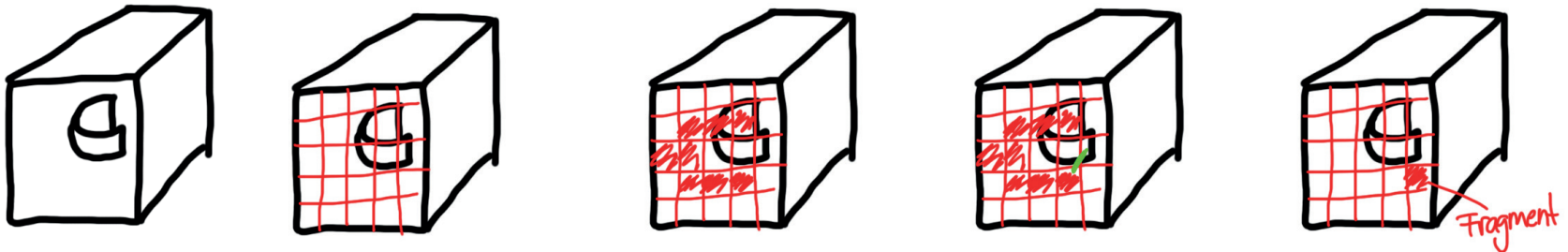
Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090701 Beim Culling geht es darum Flächen, die vom Betrachter nicht wahrgenommen werden können aus der Szene zu entfernen.</p> <p>090702 Durch das Backface-Culling werden die Polygone aus der Szene entfernt, die vom Betrachter abgewandt sind.</p> <p>090703 Ob eine Fläche sichtbar oder nicht sichtbar ist wird mit Hilfe des Normalenvektors entschieden.</p> <p>090704 Ein Normalenvektor ist ein Vektor, der zu senkrecht auf der zugehörigen Fläche steht.</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 n 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>090601 Culling dient dazu Meshes die komplett außerhalb des sichtbaren Volumens liegen wegzuschneiden</p> <p>090602 -spezielle Form: Backfaceculling</p> <p>090605 Normalenvektor steht senkrecht zur Fläche</p>	<p>090602 -Es erscheint ein Trigon. Auf das Objekt wird eine Kamera gerichtet</p> <p>090603-090604 -Es erscheint auf dem Objekt eine Normale. Daraufhin fängt sich das Trigon an zu drehen</p> <p>090705-090706 Erreicht das Trigon eine Stellung, bei der der Normalenvektor von der Kamera abgewandt ist, verschwindet das Trigon. Der Normalenvektor rotiert weiterhin</p>



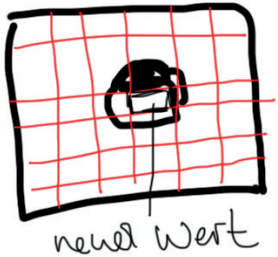
9.8 Rasterisierung



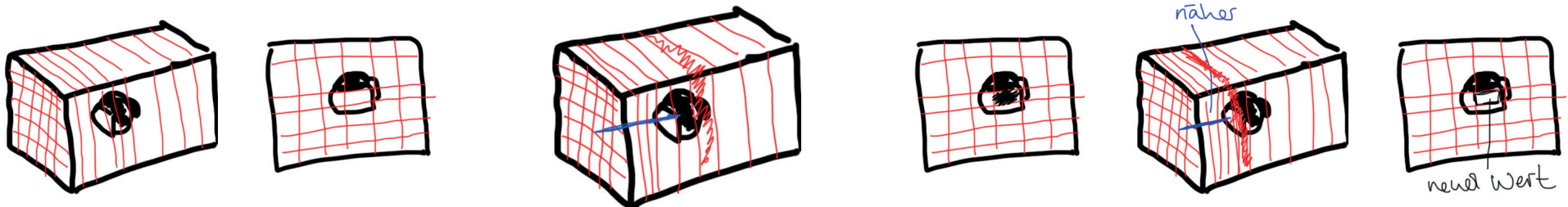
Sprechertext	Screen text / Notizen	Regieanweisungen
<p>090801 Zunächst wird im Rasterisierungsschritt die Projektionsebene gerastert.</p> <p>090802 Die Meshes im Frustum werden nun auf diesem Raster dargestellt.</p> <p>090803 Die Rasterisierung der Projektionsebene liefert nicht unbedingt die endgültigen Pixel des fertigen Bildes.</p> <p>090804 Das liegt daran, dass noch Transparenzen und mögliche Verdeckungen berechnet werden müssen.</p> <p>090805 Deswegen werden die Rasterelemente während dem Rasterisierungsschritt auch Fragmente genannt.</p>	<p>090805 Rasterelement heißt bei der Rasterisierung Fragment</p>	<p>090801 Es erscheint das Frustum mit einem Mesh (Hand). Daraufhin wird die Projektionsebene gerastert</p> <p>090802 Die Vertices des Meshes werden nun auf dem Raster abgebildet</p> <p>090803 Ein Rasterelement leuchtet auf</p> <p>090804 Es wird eine Streife zu einem Vertex aufgezeichnet</p> <p>090805 Der Begriff Fragment erscheint neben einem Rasterelement</p>



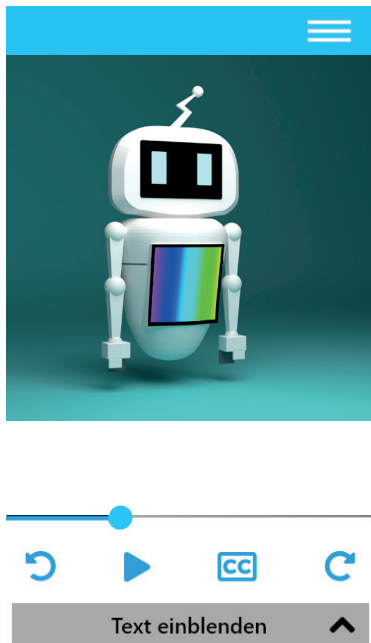
9.9 Verdeckungsberechnung/z-Buffer



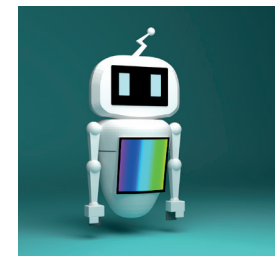
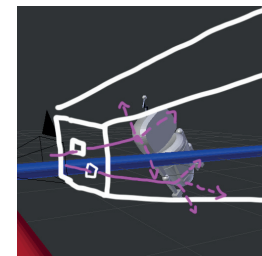
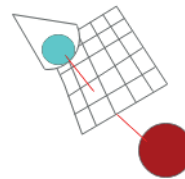
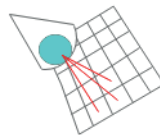
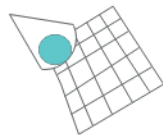
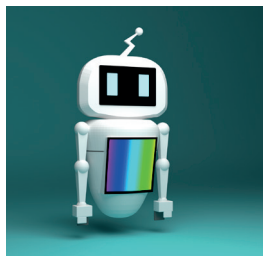
Sprechertext	Screentext / Notizen	Regieanweisungen
<p>090901 Bei einer Szene in der mehrere Meshes zu sehen sind, kann es dazu kommen, dass ein Mesh A vor einem anderen verdeckt wird, oder dieses auch schneidet.</p> <p>090902 Um dies korrekt darzustellen, werden in der Computergrafik Verdeckungsberechnungen, 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 das Frustum mit mehreren Meshes (Hand, die einen Gegenstand hält) und einer gerasterten Projektionsebene</p> <p>090902-090903 Nebenan erscheint das Abbild der Projektionseben</p> <p>090904 Ein Strahl läuft von einem Raster-element zu einem Vertice. dabei trifft er Den Vertice des Meshes, das weiter hinten im Frustum liegt. Dieser wird auf der Projektionsebene dargestellt. Darufhin erscheint der gleiche Strahl nochmal. Diesmal trifft er aber auf einen näheren Punkt. Dieser wird auch auf der Projektionsebene dargestellt. Dabei überschreibt die Abbildung die vorherige.</p>



9.9 Raytracing



Sprechertext	Screen text / Notizen	Regieanweisungen
<p>090901 Beim realistischen Rendern liegt das Hauptaugenmerk auf der physikalischen Korrektheit der Darstellung des gerenderten Bildes. Dafür sollte jedoch eine höhere Rechenzeit in Kauf genommen werden.</p> <p>090902 Raytracing – zu Deutsch „Strahlen verfolgen“ – ist in erster Linie ein Algorithmus zur Verdeckungsrechnung.</p> <p>090903 Dieser basiert auf dem Aussenden von Strahlen vom Betrachterblickpunkt. Des Weiteren ist eine Bildebene vorhanden, die in Pixel unterteilt ist und dem später gerenderten Rasterbild entspricht.</p> <p>090904 Für jeden Pixel wird ein Strahl ausgesandt. Diese werden Primärstrahl genannt.</p> <p>090905 Die Primärstrahlen ermitteln Schnittpunkte mit Flächen.</p> <p>090906 Anschließend werden die gebrochenen bzw. die reflektierten Strahlen berechnet. Diese werden als Sekundärstrahlen bezeichnet.</p> <p>090907 Der Strahl endet, wenn er die maximale Anzahl von Schritten erreicht, auf kein weiteres Objekt oder auf eine Lichtquelle trifft.</p> <p>090908 Das Raytracing bringt den Vorteil, dass korrekte Objektspiegelungen und Schatten dargestellt werden können.</p>	<p>090902 Raytracing (dt. Strahlen verfolgen“)</p> <p>090903 - Aussendung von Strahlen vom Betrachter aus - Für jedes Rasterelement ein Strahl</p> <p>090904 Primärstrahl: Strahl von Betrachter auf Pixel</p> <p>090907 Sekundärstrahl: reflektierte/ gebrochene Strahlen</p>	<p>090901 Es wird ein Bild von einer Szene gezeigt, welches den Raytracing-Algorithmus verwendet.</p> <p>090903 Es wird ein Auge eingeblendet Es wird eine Bildebene eingeblendet</p> <p>090904 Es erscheint ein Raster. Es schießen Strahlen aus dem Auge durch jedes Rasterelement. Daraufhin wird geprüft, ob der Strahl ein Objekt trifft.</p> <p>090906 Es erscheinen Normalen Oberflächenstücke bekommen eine Farbe</p> <p>090907 Sekundärstrahlen entstehen</p> <p>090908 Ein fertig gerendertes Bild entsteht</p>



9.10 Raytracing – Interaktion



- ☒ Shading
☐ Raytracing

Anweisungen

091001

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



Sprechertext	Screentext / Notizen	Regieanweisungen
<p>091101 Volumengrafiken sind in der Lage transparente Objekte und Objekte ohne scharfe Abgrenzungen, wie z. B. Wolken, zu modellieren. Diese bestehen aus Voxeln. Voxel bezeichnet einen Gitterpunkt in einem dreidimensionalen Gitter. Dies entspricht einem Pixel in einem 2D-Bild, einer Rastergrafik. Der Begriff Voxel ist eine Analogie zum Pixel leitet sich aus den Begriffen „volume“ und „element“ ab.</p> <p>091102 Die Volumengrafik basiert auf dem Strahlentransport, der beschreibt, wie sich Licht auf dem Weg durch ein Volumen verhält.</p> <p>091103 Beim Rendern einer Volumengrafik unterscheidet man vier Schritte:</p> <p>091104 <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>091105 <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>091106 <u>3. dem Shading:</u> Beim Shading wird bestimmt, wie viel Licht von einem Voxel aus in Richtung des Betrachters ausgeht und welche Farbe es hat.</p> <p>091107 <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>091101 Volumengrafik = transparente Objekte</p> <p>Voxel = Gitterpunkt in einem dreidimensionalen Gitter.</p> <p>091103-091107 vier Render Schritte: 1. Klassifikation 2. Interpolation 3. Shading 4. Composition</p>	<p>091101 Es wird ein Voxelgitter eingeblendet und anhanddessen ein Voxel gezeigt</p> <p>091103-091107 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>

