DREHBUCH RENDERING

Computergrafik.Online

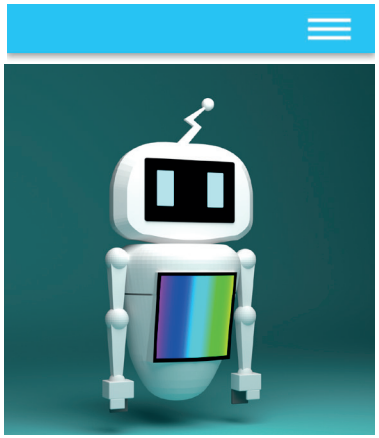
Betreuer: Prof. Jirka Dell'Oro-Friedl
Wintersemester 2018/2019

Hochschule Furtwangen University
Fakultät Digitale Medien

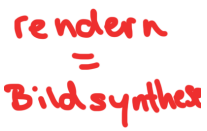
Version: 1.5
Letzte Änderung: 06.12.2018
Autor: Berdan Der

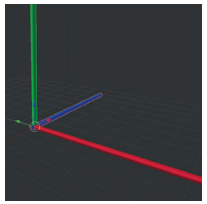
HOCHSCHULE FURTWANGEN UNIVERSITY | HFU

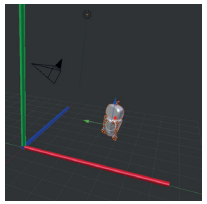
9.1 Einleitung/Anwendung

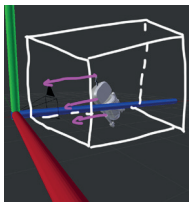


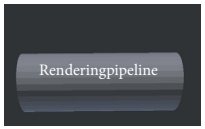
Sprechertext	Screentext / Notizen	Regieanweisungen
090101 Rendern stammt vom englischen Wort „to render“ und heißt zu deutsch „etwas ausgeben“. Das Rendering bezeichnet den Vorgang, ein Bild zu generieren. Beim Rendern einer 3D-Szene werden im Wesentlichen Meshes, Kameras und Lichtquellen berücksichtigt. 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. 090103 Des Weiteren werden aber auch Sichtbarkeits- und Beleuchtungsberechnungen durchgeführt, Texturen gemappt und spezielle Effekte dargestellt.	090101 - Rendern (dt. Bildsynthese) - Aus einer Szene wird ein Bild erzeugt 090102 Prozess des Renderings in der Rendering Pipeline	090101 -Es erscheint eine Einblendung der Begrifflichkeit -Danach erscheint eine Szene mit Objekt, Kamera und Licht -Daraufhin erscheint das Kamera-Frustrum mit dem Mesh darin. Dieses wird auf die Ner-Plane abgebildet und das Frustrum zusammengestaucht 090102 -Es erscheint eine Rendering Pipeline, die in Transformation und Rasterisieung aufgeteilt wird. - Daraufhin läuft ein Mesh durch

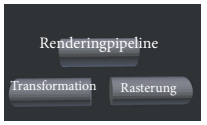


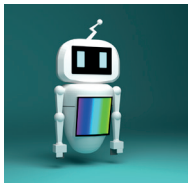




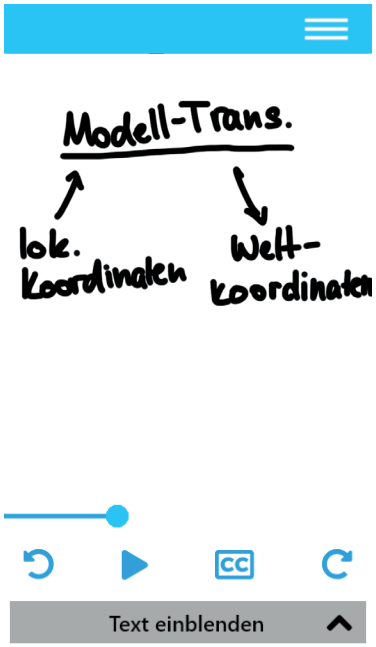




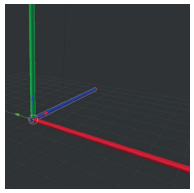
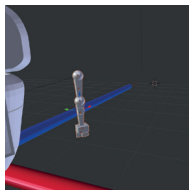
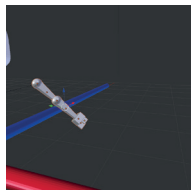
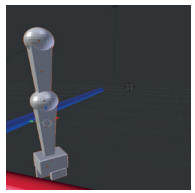
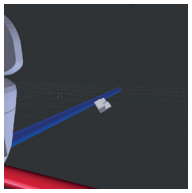
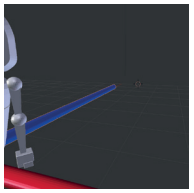
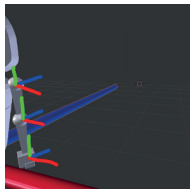
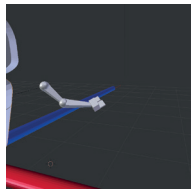
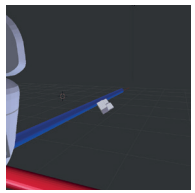
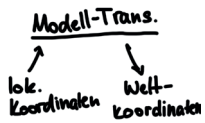




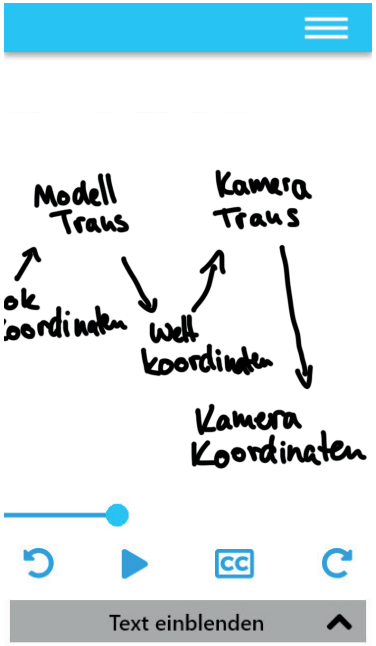
9.2 Modell Transformation



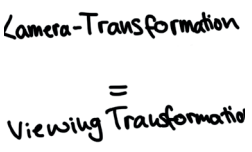
Sprechertext	Screentext / Notizen	Regieanweisungen
090201 Ein Mesh, wird normalerweise durch sein lokales Koordinatensytem definiert. Ohne Transformationen würden alle Objekte im Welturspung liegen. 090202 Durch Translationen, Rotationen und Skalierungen wird ein Mesh an die gewünschte Stelle im Raum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

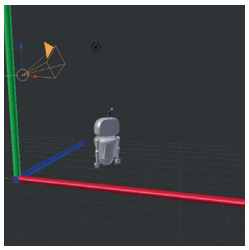











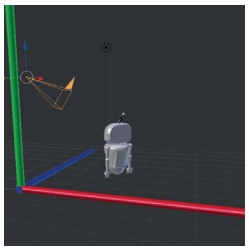
9.3 Kamera-Transformation

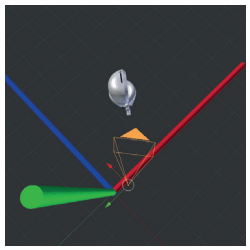


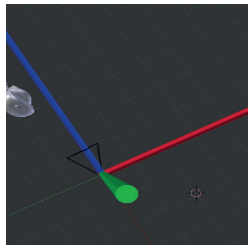
Sprechertext	Screentext / Notizen	Regieanweisungen
090301 Bei der Kamera-Transformation, die auf Englisch viewing transformation genannt wird, werden die Kamera und alle Meshes so transformiert, dass die Kamera im Ursprung liegt. 090302 Die Ausrichtung der Kamera veräuft nach der Transformation entlang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

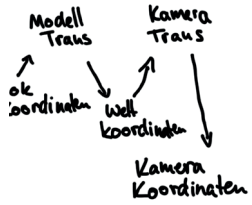








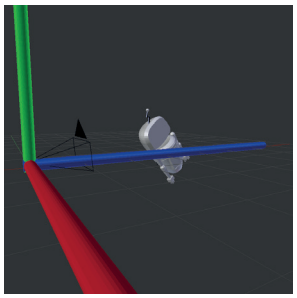


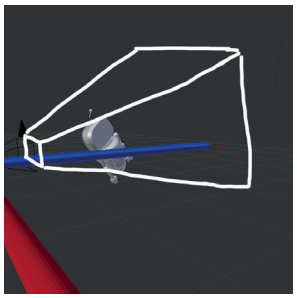


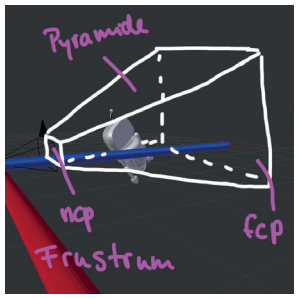
9.4 Projektions Transformation

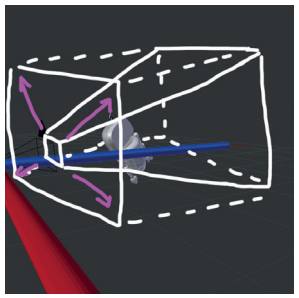
Text einblenden

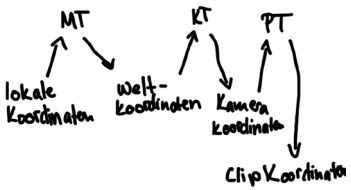
Sprechertext	Screentext / Notizen	Regieanweisungen
090401 Nach der Modell- und der Kamera Transformation befinden sich alle Eckpunkte, wlelche auch Vertices geannt werden, an ihren endgültigen Positionen im Raum. 090402 Bei der Projektions-Transformation wird das View Frustrum näher betrachtet. Bei der perspektivischeProjektion hat das Frustrum hat die Form eines Pyramidenstumpfes und besitzt eineFar Clipping Plane und einer Near Clipping Plane, die auch Projektionsebene genannt wird. 090403 Meshes innerhalb des Frustrums werden dargestellt. 090404 Als nächstes wird das das Frustrum mitsamt der Inhalte derartig transformiert , dass ein Quader entsteht. 090405 Nach diesem Schritt liegen die Koordinaten als Clip-Koordinaten im sogenannten Clip-Space vor.	090401 Eckpunkt = Vertex	090401 Es erscheint eine Kurze Animation zu den vorherigen Transformationen 090402 Es erscheint ein Frustrum, und die Bestandteile werden aufgezeigt 090403 Die Meshes erscheinen innerhalb des Frustrums 090404 Die Near Clipping Plane wird auf die selbe Größe der Far Clipping Plane transformiert 090405 Die Clip-Koordinaten erscheinen im Schema





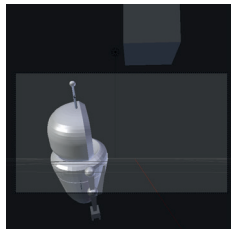




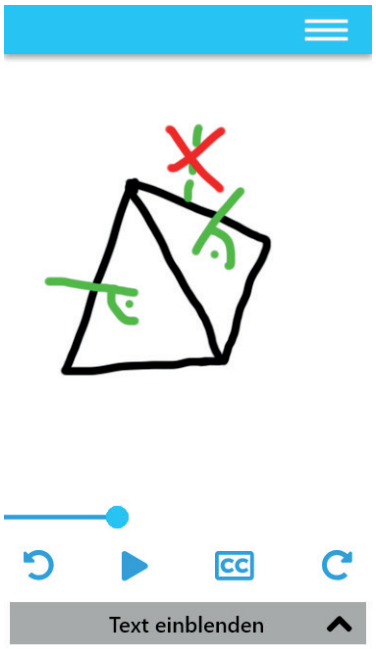


Text einblenden

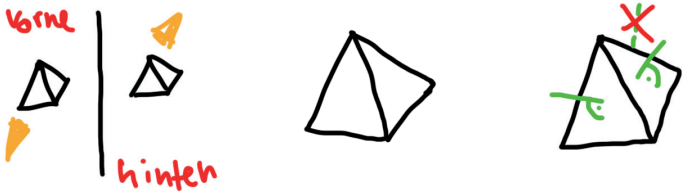
Sprechertext	Screentext / Notizen	Regieanweisungen
090501 Beim Clipping geht es darum Flächen, die vom sichtbaren Volumen nicht mehr eingefangen werden können aus der Szene zu entfernen. 090502 Nach der Projektionstransformation wird überprüft, welche Meshes vollständig im sichtbarem Bereich liegen. 090503 Elemente die gänzlich außerhalb des Sichtfensters liegen werden komplett entfernt. 090504 Für jede Kante des Sichtfensters wird geprüft, ob sich der Vertex eines Objekts inner- oder außerhalb der Kante befindet. 090505 Punkte die innerhalb der Grenze liegen werden in ihrer Geometrie belassen, Punkte außerhalb entfernt. 090506 An der Grenze des Sichtfensters werden neue Vertices kreiert. Dieses Verfahren wird auch Sutherland Hodgeman Clipping genannt.	090501 Clipping dient dazu Geometrien außerhalb des sichtbaren Volumens wegzuschneiden	090501-090503 -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 090504 -Es erscheint ein Objekt mit Vertices und eine kante des Sichtfensters -Daraufhin werden neue Vertices berechnet und der überstehende Teil wird abgeschnitten

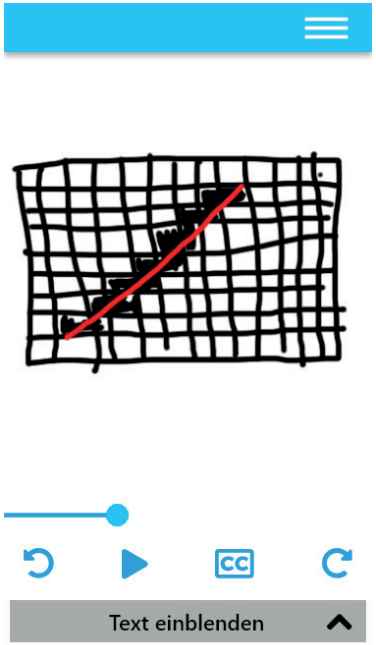




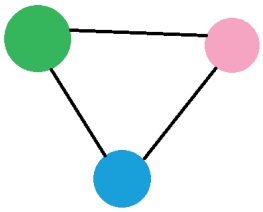


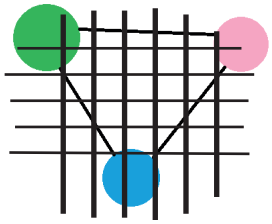
Sprechertext	Screentext / Notizen	Regieanweisungen
090601 Beim Culling geht es darum Flächen, die vom Betrachter nicht wahrgenommen werden können aus der Szene zu entfernen. 090602 Durch das Backface-Culling werden die Polygone aus der Szene entfernt, die vom Betrachter abgewandt sind. 090603 Ob eine Fläche sichtbar oder nicht sichtbar ist wird mit Hilfe des Normalenvektors entschieden. 090604 Ein Normalenvektor ist ein Vektor, der zu senkrecht auf der zugehörigen Fläche steht 090605 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. 090606 Damit kann diese Fläche entfernt werden.	090601 Culling dient dazu Meshes die komplett außerhalb des sichtbaren Volumens wegzuschneiden 090602 -spezielle Form: Backfaceculling 090605 Normalenvektor steht senkrecht zur Fläche	090602 -Es erscheint ein Objekt mit Vorder- und Hinteransicht. Auf das Objekt ist eine Kamera gerichtet und der hintere wird entfernt 090603-090604 -Es erscheint ein Objekt auf dem Normalen erscheinen. die Normalen die von der Kamera abgewandt sind werden entfernt.

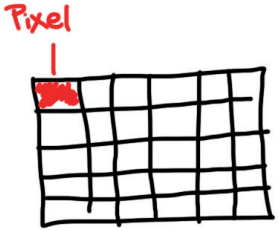


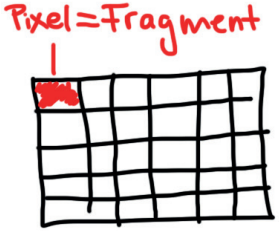


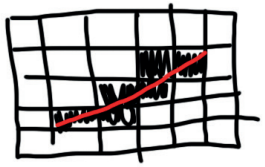
Sprechertext	Screentext / Notizen	Regieanweisungen
090701 Bis zu diesem Zeitpunkt liegt für jeden Vertex eines Polygons ein Farbwert vor. ZUnächst werden das Polygon gerastert. 090702 Während des Rasterisierungsschrittes werden die Rasterelemente nicht Pixel, sondern Fragmente genannt. 090703 Ein Grund für die unterschiedliche Bezeichnung ist, dass ein Fragment mehr und auch andere Daten speichern kann als ein Pixel. Dazu gehört zum Beispiel der Alpha-Wert, der Transparenzen beschreibt, und einen Tiefen-Wert, der für die Verdeckungsberechnung wichtig ist. 090704 Bei diesem Schritt werden die Flächen in Fragmente aufgeteilt und der Farbwert für jedes einzelne Fragment wird berechnet.	090702 Fragment = stellvertretend für Pixel	090701 Es erscheint Polygon, dessen Vertice Farbwerte enthalten Es erscheint ein Raster 090702 Im Raster leuchtet eine Fläche auf, die ein Pixel darstellt. Anhand dessen wird der Begriff Fragment eingeführt 090703 Es wird eine Linie bzw ein anderes beliebiges Objekt eingeblendet, welches den Flächen angenähert wird.

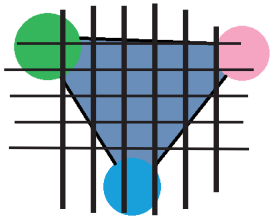


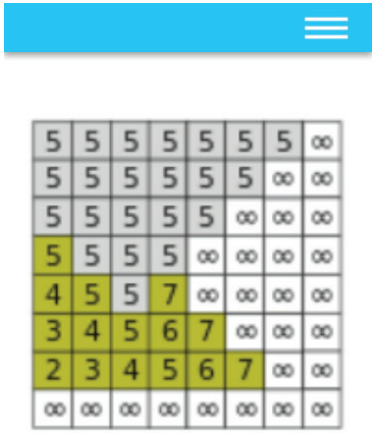




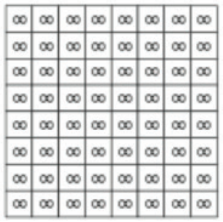


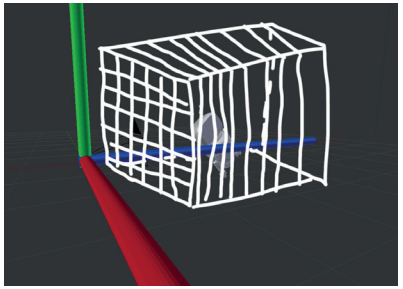


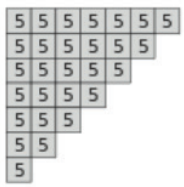


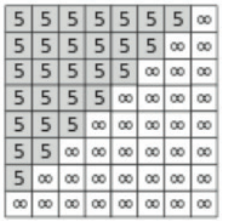


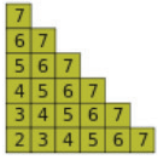
Sprechertext	Screentext / Notizen	Regieanweisungen
090801 Bei einer Szene in der mehrere Meshes zu sehen sind, kann es dazu kommen, dass ein Mesh A vor einem anderen verdeckt wird, oder dieses auch schneidet.		090801 -Es erscheint ein Bild, bei welchem sich Objekte überschneiden -Daraufhin erscheint das gleicheBild nur mit falscher Verdeckungsberechnung
090802 Um dies korrekt darzustellen, werden in der Computergrafik Verdeckungsberechnungen, wie z. B. der z-Buffer-Algorithmus angewandt.	090802 Verdeckungsberechnung durch z-Buffer-Algorithmus	
090803 Die Grundidee des z-Buffer-Algorithmuses ist es für jeden Pixel die Tiefeninformation bzw. den z-Wert zu speichern.	090803 z-Buffer-Algorithmus speichert für jeden Pixel z-Wert	090803-090805 Auf dem Frustrum wird ein Raster dargestellt. Alle Objekte werden auf dem Raster abgebildet. Gleichzeitig dazu wird ein 2-D-Schema nebenan gerastert
090804 Es muss geprüft werden ob ein Pixel näher an der Kamera liegt als ein vorher berechneter. Dazu muss der z-Wert kleiner sein.		Falls der aktuell gerasterte Punkt näher am Betrachter liegt als der davor gerasterte Punkt, wird dieser durch das aktuelle ersetzt.
090805 Falls ja, werden Farbwerte und z-Buffer für den Pixel überschrieben , andernfalls werden die alten Werte beibehalten.	090805 Je kleiner der z-Wert eines Pixels, desto näher ist er am Betrachter	Dabei wird die Distanz zum Betrachter eingetragen. Anhand dieser weiß an, welche Objekte wie überschnitten und überlagert sind und wie die Objekte dartgestellt werden müssen.

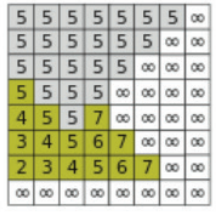


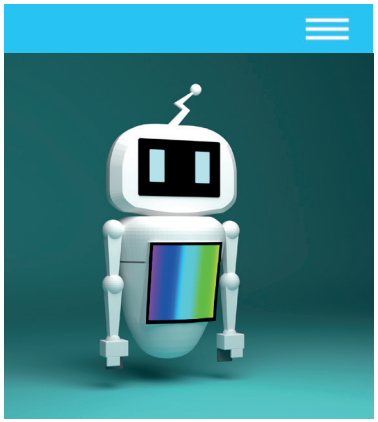


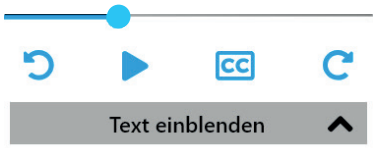




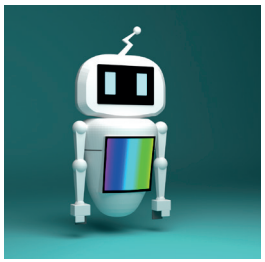


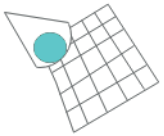


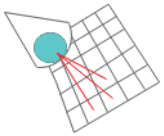


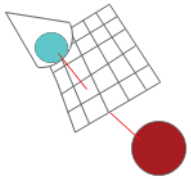


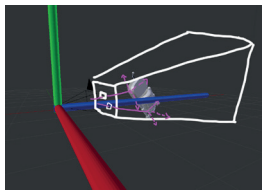
Sprechertext	Screentext / Notizen	Regieanweisungen
090901 Beim realistischen Rendern liegt das Hauptaugenmerk auf der physikalischen Korrektheit der Darstellung des gerenderten Bildes. Dafür sollte jedoch eine höhere Rechenzeiten in Kauf genommen werden. 090902 Raytracing – zu Deutsch „Strahlen verfolgen“ – ist in erster Linie ein Algorithmus zu Verdeckungsberechnung. 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. 090904 Für jeden Pixel wird ein Strahl ausgesandt. Diese werden Primärstrahl genannt. 090905 Die Primärstrahlen ermitteln Schnittpunkte mit Meshes aus der Szene. 090906 Anschließend werden die gebrochenen bzw. die reflektierten Strahlen berechnet. Diese werden als Sekundärstrahlen bezeichnet. 090907 Der Strahl endet, wenn er die maximale Anzahl von Schritten erreicht, auf kein weiteres Objekt oder auf eine Lichtquelle trifft. 090908 Das Raytracing bringt den Vorteil, dass korrekte Objektspiegelungen und Schatten dargestellt werden können.	090902 Raytracing (dt. Strahlen verfolgen“) 090903 -Aussendung von Strahlen vom Betrachter aus -Für jedes Rasterelement ein Strahl 090904 Primärstrahl: Strahl von Betrachter auf Pixel 09097 Sekundärstrahl: reflektierte/ gebrochene Strahlen	090901 Es wird ein Bild von eien Szene gezeigt, welches den Raytracing-Algorithmus verwendet. 090903 Es wird ein Auge eingeblendet Es wird eine Bildeben eingeblendet 090904 Es erscheint ein Raster. Es schießen Strahlen aus dem Auge durch jedes Rasterelement. Daraufhin wird geprüft, ob der Strahl ein Objekt trifft. 090906 Es erscheinen Normalen Oberflächenstücke bekommen eine Farbe 090907 Sekundärstrahlen enstehen 090908 Ein fertig gerendertes Bild entsteht

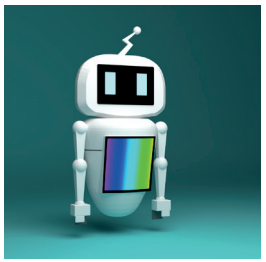




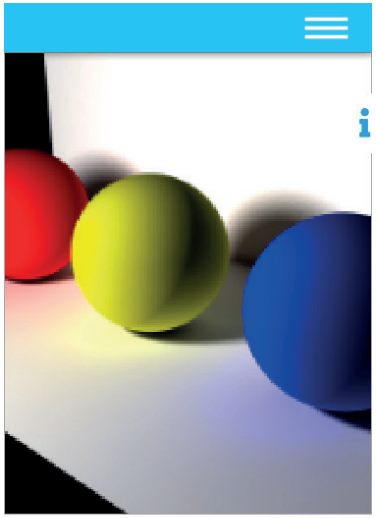








9.10 Raytracing – Interaktion



Anweisungen
091001 Wähle zwischen reinem Shading und Raytracing aus und betrachte die Änderungen.

Sprechertext	Screentext / Notizen	Regieanweisungen
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 Analogei zum Pixel leitet sich aus den Begriffen „volume“ und „element“ ab. 091102 Die Volumengrafik basiert auf dem Strahlentransport, der beschreibt, wie sich Licht auf dem Weg durch ein Volumen verhält. 091103 Beim Rendern einer Volumengrafik unterscheidet man vier Schritte: 091104 1. der Klassifikation: Hier werden den Voxeln Materialeigenschaften gegeben. Bei der Erzeugung des Voxels besitzt dieser zunäckhst nur eine Eigenschaft. Weitere müssen bei der Klassifikation vom Benutzer vorgegeben werden. Eine Eigenschaft könnte zum Beispiel sein, wie sehr das Voxel spiegeln soll. 091105 2. der Interpolation: 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. 091106 3. dem Shading: Beim Shading wird bestimmt, wie viel Licht von einem Voxel aus in Richtung des Betrachters reflektiert wird und welche Farbe es hat. 091107 4. der Composition: 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.	091101 Volumengrafik = transparente Objekte Voxel = Gitterpunkt in einem dreidimensionalen Gitter. 091103-091107 vier Render Schritte: 1. Klassifikation 2. Interpolation 3. Shading 4. Composition	091101 Es wird ein Voxelgitter eingebledet und anhanddessen ein Voxel gezeigt 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 unterchsiedlichen Lichtstufen einer Linie werden miteinander verrechnet Zum Schluss wird eine Volumengrafik eingeblendet, die sich dreht.