

3D-Grundlagen

*Was, abgesehen von einem 3D-Grafiker,
brauche ich, um eine 3D-Szene auf
meinen Bildschirm zu zaubern?*

Das 3D-Koordinatensystem

2D Koordinatensysteme in Director

Auch in früheren Versionen von Director hatten wir bereits mit verschiedenen Koordinatensystemen zu tun:

Jeder einzelne Darsteller besitzt ein eigenes Koordinatensystem. Des- sen Bezugspunkt – oder auch Ursprung – befindet sich meist in der linken oberen Ecke der »Bounding Box« des betreffenden Darstellers. Eine Aus- nahme hiervon bilden beispielsweise Bitmap-Darsteller, deren Registrie- rungspunkt frei gewählt werden kann. Die Position des Registrie- rungs- punkts für diesen Darsteller wird immer in Relation zu diesem Ursprung angegeben.

Wird jedoch ein Darsteller als Sprite auf der Bühne platziert, so kann eine Positionsangabe relativ zur eigenen linken oberen Ecke nicht funk- tionieren! Deshalb braucht man Eigenschaften wie `locH` und `locV`, um die Position eines Sprites auf der Bühne anzugeben. Ursprung dieses Koor- dinatensystems ist die linke obere Ecke der Bühne.

Quasi-3D: Die Eigenschaft `locZ`

Mit der Einführung der Eigenschaft `locZ` konnten wir dann nicht nur die Position eines Sprites auf der Bühne bestimmen, sondern gewissermaßen auch dessen »räumliche« Anordnung: Ist der Sprite ganz im Vordergrund? Oder wird er von anderen Sprites überlagert?

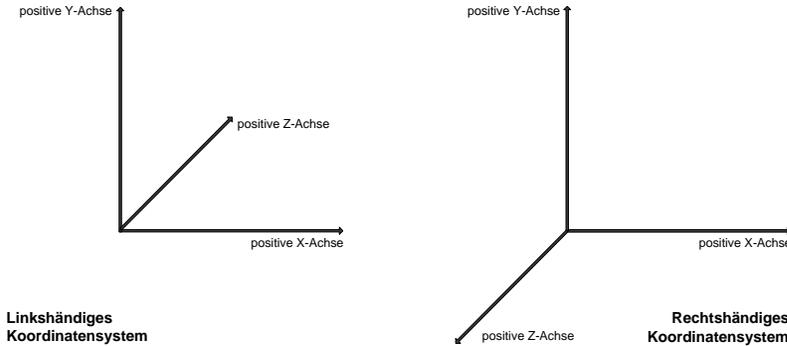
3D-Koordinatensysteme

Ähnlich ist es auch im 3D-Raum: Die Position jedes einzelnen Punkts lässt sich mit Hilfe von drei Koordinaten (X, Y, Z) eindeutig bestimmen. Wie auch beim zweidimensionalen Koordinatensystem gibt die X-Achse die horizontale und die Y-Achse die vertikale Entfernung eines Punkts zum Ursprung an. An der Z-Achse lässt sich die räumliche »Tiefe« messen.

Dabei gibt es zwei Möglichkeiten: Zeigt die positive Z-Achse nach vorn, so spricht man von einem rechtshändigen Koordinatensystem. Zeigt

die positive Z-Achse nach hinten, so hat man es mit einem linkshändigen Koordinatensystem zu tun.

In Director wird beispielsweise im Normalfall ein rechtshändiges Koordinatensystem verwendet, bei dem die positive Y-Achse nach oben zeigt.



▲ **Abbildung 1**
Linkshändiges und rechtshändiges Koordinatensystem

Koordinatensysteme in 3ds max

In 3ds max laufen die Achsen ein bisschen anders: die positive X-Achse zeigt zwar vom Benutzer aus nach rechts, aber die positive Y-Achse zeigt nicht etwa nach oben, sondern nach hinten. Dafür zeigt die positive Z-Achse nach oben. Somit haben wir zwar auch ein rechtshändiges Koordinatensystem, doch im Gegensatz zu dem in Director ist es um 90° gedreht.

Werden nun Objekte bzw. 3D-Szenen aus 3D Max nach Shockwave 3D exportiert, so werden die Koordinatensysteme aus 3ds max beibehalten. Das bedeutet, dass diese Objekte nach dem Import aussehen können, als würden sie »auf der Seite liegen«. Auch sämtliche Transformationen beziehen sich dann auf das Max-Koordinatensystem und können daher unerwartete Ergebnisse liefern.

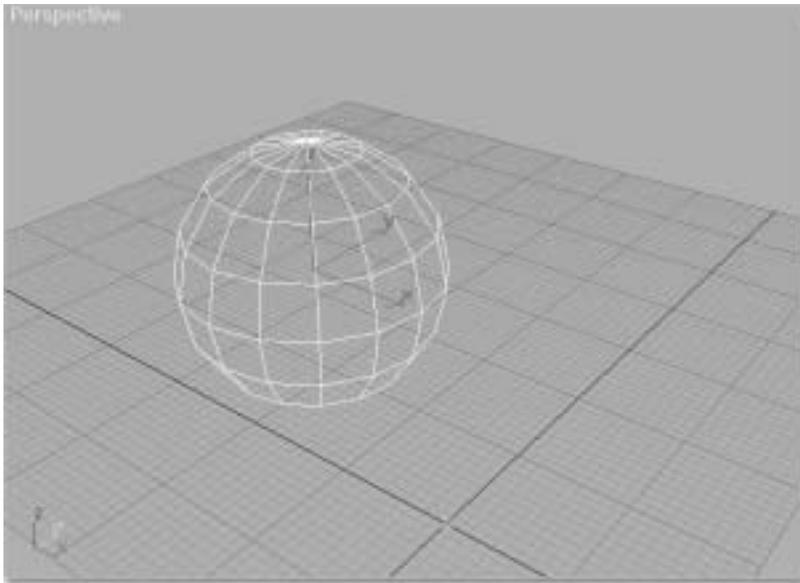
Um solche Objekte an das Koordinatensystem in Director anzupassen, gibt es zwei Möglichkeiten:

- ▶ Sie legen das Objekt im 3D-Programm bereits so an, dass es quasi »auf dem Rücken liegt«, dann ist es in Director korrekt ausgerichtet.
- ▶ Oder Sie drehen das Objekt in Director um 90°, um die korrekte Ausrichtung zu erhalten.

Werden dagegen komplette 3D Darsteller komplett per Lingo erzeugt, so werden die üblichen Director-Koordinatensysteme verwendet: X-Achse horizontal nach rechts, Y-Achse vertikal nach oben, Z-Achse nach hinten.

Lokale Koordinaten (»Objektkoordinaten«)

Wie auch jeder Director-Darsteller sein eigenes Koordinatensystem mit einem eigenen Ursprung hat, so hat auch jedes 3D-Objekt einen lokalen Ursprung. Genauer gesagt: Jedes Modell wird zunächst im Ursprung der 3D-Welt erzeugt. Damit hat jeder Punkt des Modells eine Koordinate bezogen auf diesen Mittelpunkt. Wird das Modell an einen anderen Punkt der 3D-Welt verschoben, so wird im Normalfall nur der Mittelpunkt verschoben, und die anderen Punkte behalten ihre Relation zum Modell-Mittelpunkt bei.



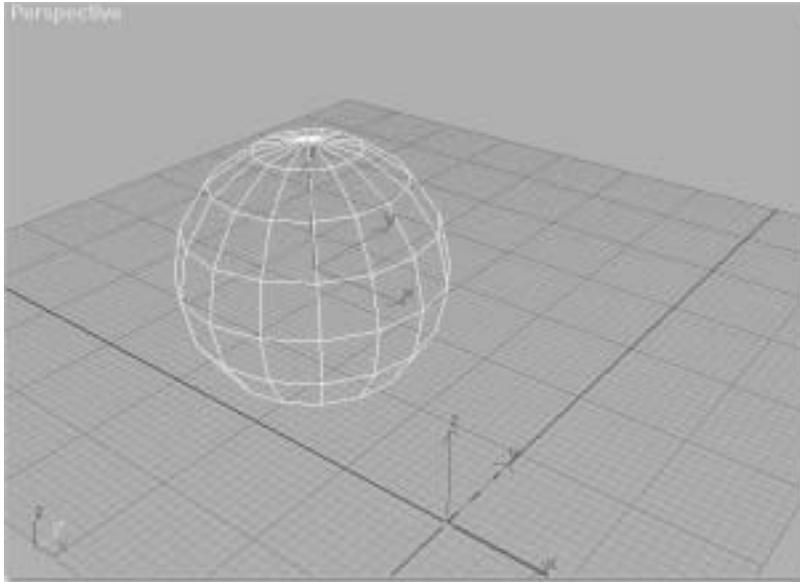
▲ Abbildung 2

Modell einer Kugel mit den lokalen Achsen (X, Y, Z)

Globale Koordinaten (»Weltkoordinaten«)

Will man nun jedoch verschiedene 3D-Modelle in einer Welt anordnen, so braucht man ein Koordinatensystem, das unabhängig von dem einzelnen Modell ist. Dieses globale Koordinatensystem wird auch »Weltkoordinatensystem« genannt.

Ähnlich wie die Sprites auf der Bühne können die Modelle innerhalb dieses Systems angeordnet und bewegt werden.



▲ **Abbildung 3**
Modell einer Kugel mit lokalen und globalen Achsen

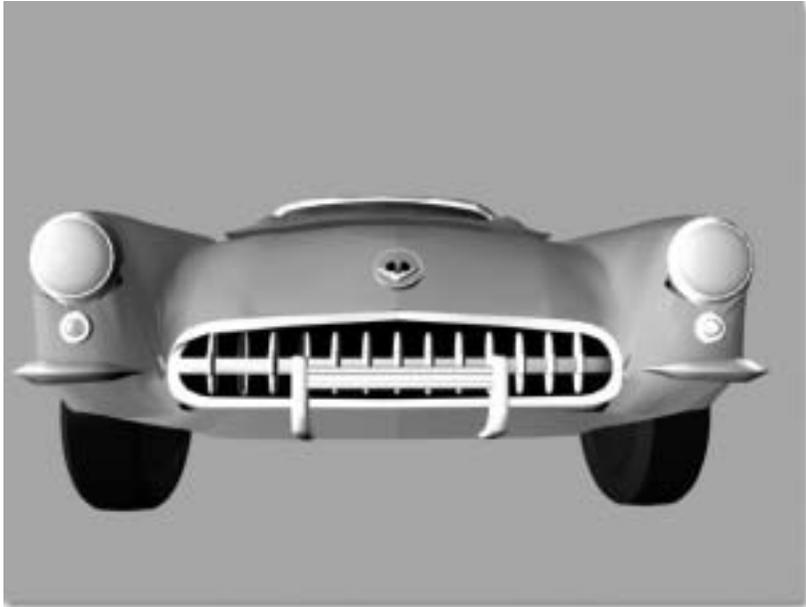
Alles Ansichtssache: Die Ausrichtung der Kamera

Eine 3D-Welt kann riesig sein – meist sieht man dann immer nur einen kleinen Ausschnitt davon. Hier bestimmen die Kameras, was der Betrachter zu sehen bekommt und was nicht. Ähnlich wie bei einer richtigen Filmkamera kann man bestimmen, welches Objekt gezeigt werden soll, und vor allem wie das Objekt gezeigt wird.

Ein Auto ist ein Auto ist ein Auto

Wir haben hier das Modell eines Autos. An diesem Modell selbst wird nichts verändert. Die unterschiedlichen Eindrücke der Bilder werden hier alle durch Veränderungen an der Kamera erreicht.

Dabei kommt es nicht nur darauf an, wie weit die Kamera von dem Objekt entfernt ist. Ausschlaggebend ist auch, von wo aus sie auf das Objekt gerichtet wird und wie groß der Betrachtungswinkel sein soll.



▲ **Abbildung 4**
Frontansicht eines Autos



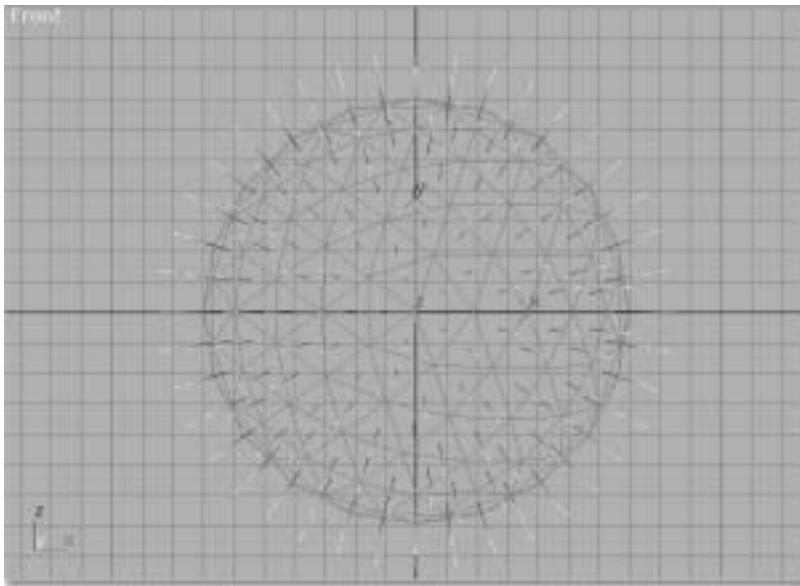
▲ **Abbildung 5**
Das gleiche Automodell, Ansicht von oben

Welche Elemente werden in einer 3D-Szene benötigt?

Modelle

Woraus besteht ein 3D-Modell? Im Normalfall setzt sich ein 3D-Objekt aus Flächen, so genannten Polygonen, zusammen. Diese Flächen werden durch ihre Eckpunkte (Vertices) definiert. Außerdem gehört zu jeder Fläche ein Normalenvektor, der bestimmt, in welche Richtung diese Fläche »zeigt«.

Ein Normalenvektor ist ein Vektor, der senkrecht, also im Winkel von 90° , zu einer Fläche steht. Dabei zeigt der Normalenvektor immer nach »außen«. Macht man beispielsweise bei dem Modell einer Kugel die Normalenvektoren sichtbar, so sieht die Kugel beinahe aus wie ein Igel, der sich zusammenrollt. Dabei sind nur Flächen eines Modells sichtbar, deren Normalenvektoren in Richtung der Kamera zeigen. Zeigen die Normalenvektoren von der Kamera weg, so ist die Fläche nicht sichtbar.



▲ **Abbildung 6**
Modell einer Kugel mit sichtbaren Normalenvektoren

Die Normalenvektoren werden vom 3D-Programm gebraucht, um die Reflexion bzw. Brechung von auftreffenden Lichtstrahlen berechnen zu können. Bei glatten, undurchsichtigen Materialien beispielsweise wird das Licht getreu der Formel »Einfallswinkel gleich Ausfallswinkel« reflektiert.

Licht

In 2D-Welten haben die Objekte in der Regel statische Farben, die sich nicht verändern. Bewegt man das Bild eines roten Autos innerhalb von Director vom linken Rand der Bühne zum rechten, so wird das Auto immer gleich aussehen.

In einer 3D-Welt dagegen bestimmen die vorhandenen Lichtquellen, wie die einzelnen Flächen der Modelle für den Betrachter aussehen werden. So kann ein rotes Auto, das sich durch eine 3D-Szene bewegt, völlig unterschiedlich aussehen – je nachdem, wie es gerade beleuchtet wird.

Fährt ein Auto beispielsweise unter einer Straßenlaterne hindurch, so wird es zunächst gut von vorne-oben ausgeleuchtet. Die Farbe erscheint relativ leuchtend und kräftig, es sind viele Details erkennbar. Sobald das Auto jedoch die Laterne passiert hat, so sieht der Betrachter es im Gegenlicht. Das Auto erscheint jetzt eher als Umriss, eine Farbe ist kaum noch zu erkennen.

Die Hauptaufgabe der Beleuchtung ist also, einer Szene durch Kontraste und Schatten »Leben einzuhauchen«.

Generell unterscheiden wir vier Arten von Lichtquellen:

- ▶ Umgebungslicht (Ambient Light)
- ▶ Punktförmige Lichtquellen (Point oder Omni Lights)
- ▶ Gerichtetes Licht (Directional Light)
- ▶ Spotlichter (Spotlights), die eigentlich auch zu den Punktlichtern gehören, weil die Lichtquelle punktförmig ist

Das Licht, das in einer 3D-Szene eigentlich immer vorhanden ist, selbst wenn keine extra Lichtquellen definiert wurden, ist das **Umgebungslicht**. Es beschreibt die durchschnittliche Helligkeit der Szene. Ein Beispiel für eine solche Beleuchtung wäre das Tageslicht an einem bewölkten Tag. Die Lichtstrahlen werden in der Wolkendecke in alle möglichen Richtungen abgelenkt, so entsteht ein diffuses, gleichmäßiges Licht ohne großartige Kontraste oder Schatten.

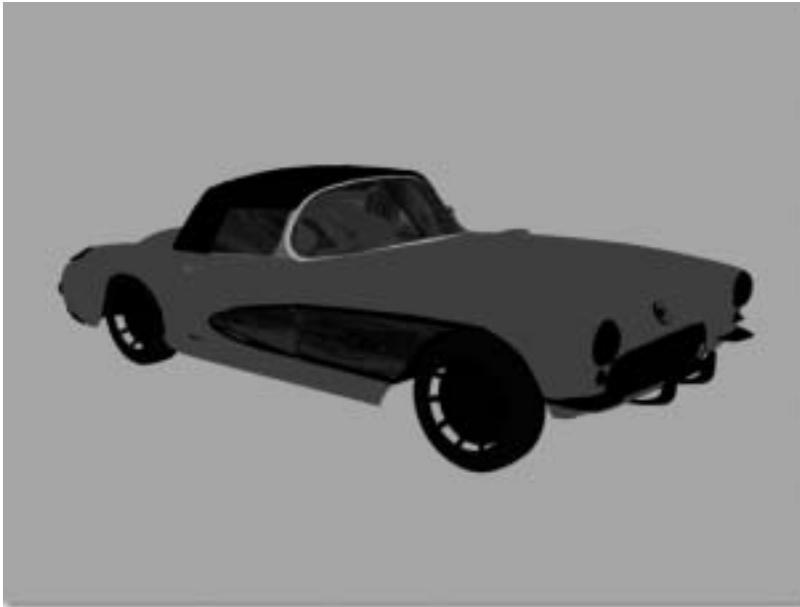
Ein nur mit dem Umgebungslicht beleuchtetes Modell können wir zwar sehen, es wirkt aber relativ flach und leblos. Je höher die Intensität des Umgebungslichts, desto heller wird eine Szene dargestellt. Allerdings gibt es mit zunehmender Helligkeit immer weniger Kontraste und die Szene wirkt immer flacher.



Farbiges Umgebungslicht

Dadurch, dass das Umgebungslicht immer und überall da ist, nehmen von anderen Lichtquellen erzeugte Schatten immer die Farbe des Umgebungslichts an. Um die Atmosphäre einer Szene

zu verstärken, kann man dem Umgebungslicht die Komplementärfarbe der Hauptlichtquelle zuweisen. Ist das Hauptlicht beispielsweise das gelbe Licht des Vollmonds, so kann dieser Eindruck verstärkt werden, indem man dem Umgebungslicht einen dunkelvioletten Farbton zuweist.



▲ **Abbildung 7**

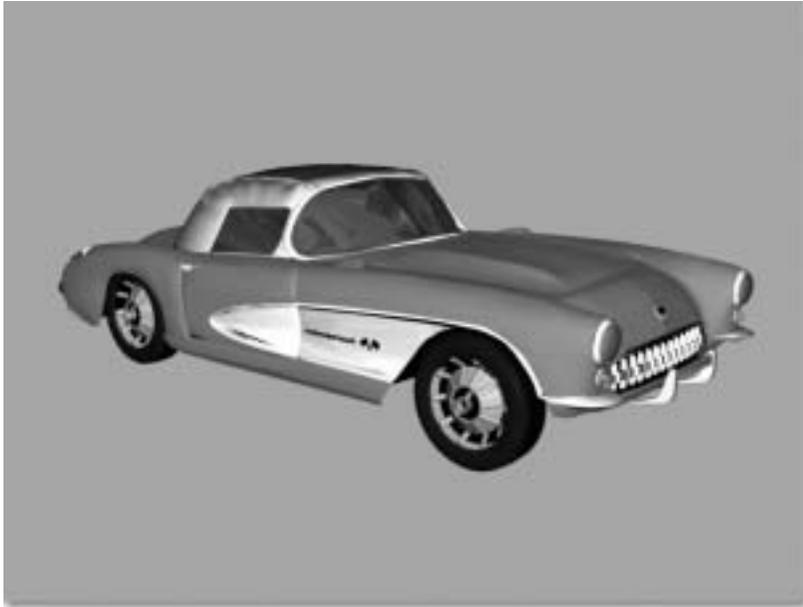
Modell eines Autos, nur mit dem Umgebungslicht beleuchtet

Punktförmige Lichtquellen senden Lichtstrahlen in alle Richtungen, wie z. B. eine einfache Glühbirne, die von der Decke hängt – oder auch wie die Sonne.

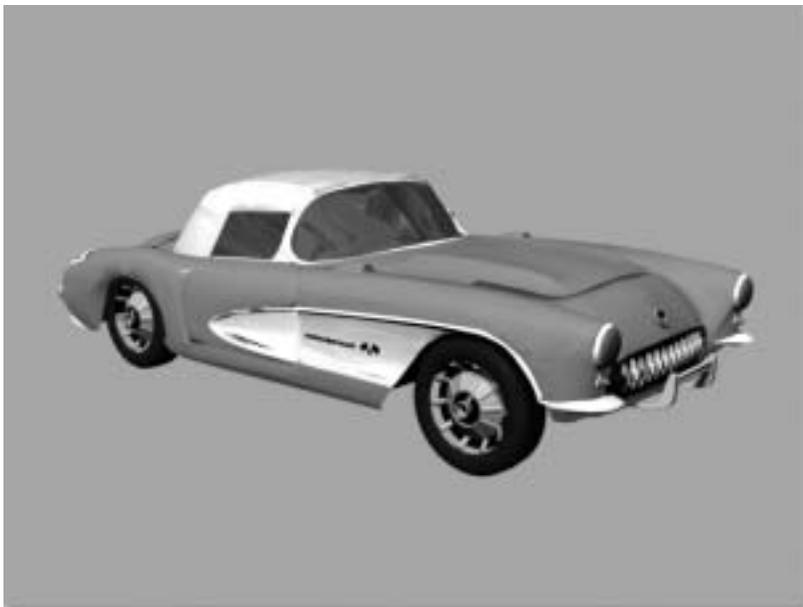
Punktlichter werden in der Regel dazu verwendet, die Grundstimmung einer Szene zu bestimmen. So kann beispielsweise ein orangefarbenes Punktlicht mit geringer Intensität dazu verwendet werden, die Grundstimmung eines Kaminfeuers zu erzeugen.

Je weiter ein Punktlicht von einem Objekt entfernt ist, desto paralleler sind die Lichtstrahlen, die auf dem Objekt auftreffen, und desto gleichmäßiger wird das Objekt ausgeleuchtet.

Gerichtete Lichtquellen senden parallele Strahlen auf die Szene. Anders als bei Spots gibt es keinen Lichtkegel, sondern die Lichtstrahlen sind quasi überall, ähnlich wie bei Punktlichtern. Die Strahlenintensität ist in der Regel gleichmäßig.



▲ **Abbildung 8**
Das gleiche Modell, von einem Punktlicht beleuchtet



▲ **Abbildung 9**
Das gleiche Modell mit gerichtetem Licht beleuchtet

Spotlichter sind etwa eine Taschenlampe oder ein Scheinwerfer. Im Extremfall (z.B. bei einem Laserstrahl) senden sie Lichtstrahlen zu einem bestimmten Punkt. Meist wird jedoch ein Lichtkegel definiert, an dessen Rand die Intensität der Strahlen schwächer ist als im Zentrum. Spots werden im Normalfall dazu verwendet, Akzente in einer Szene zu setzen, da sie starke Kontraste und Schatten erzeugen können.



▲ **Abbildung 10**
Das gleiche Modell mit einem Spot beleuchtet

Innerhalb von Director werden Lichter behandelt wie andere 3D-Objekte auch: Sie können zu Modellhierarchien hinzugefügt werden, dabei können sie sowohl untergeordnete Objekte (auch: Child- oder Kindobjekte) als auch übergeordnete Objekte (auch: Parent- oder Elternobjekte) haben.

Wo viel Licht ist, ist auch viel Schatten?

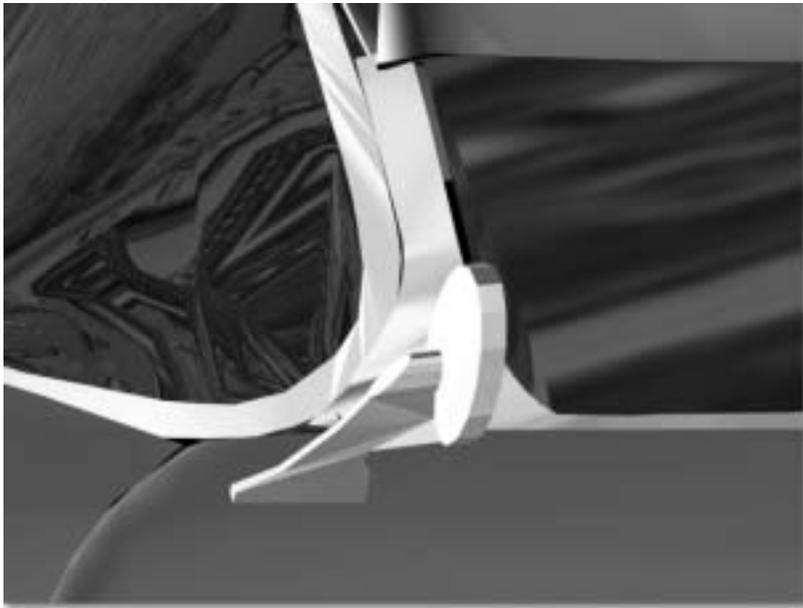
In 3ds max kann der Benutzer bei den meisten Lichtarten einstellen, ob ein Lichtobjekt Schatten werfen soll oder nicht. Allerdings werden diese Einstellungen in Shockwave 3D nicht übernommen. Das bedeutet, daß in Shockwave 3D-Objekte generell keine Schatten werfen können – egal ob statisch oder animiert!



Kamera

Wir haben ein Objekt (das Auto), wir haben die Beleuchtung (die Straßenlaterne). Was uns jetzt noch fehlt, ist ein »Einblick« in die 3D-Welt. Wir müssen noch definieren, wie diese Szene betrachtet werden soll. Dazu brauchen wir eine oder mehrere Kameras.

Jede Kamera hat einen Punkt, auf den sie gerichtet ist (target). Außerdem kann mit Hilfe des Betrachtungswinkels (field of view) der Bildausschnitt gewählt werden. Ähnlich wie auch bei »echten« Kameras sind Einstellungen vom extremen Weitwinkel bis hin zur extremen Nahaufnahme möglich.



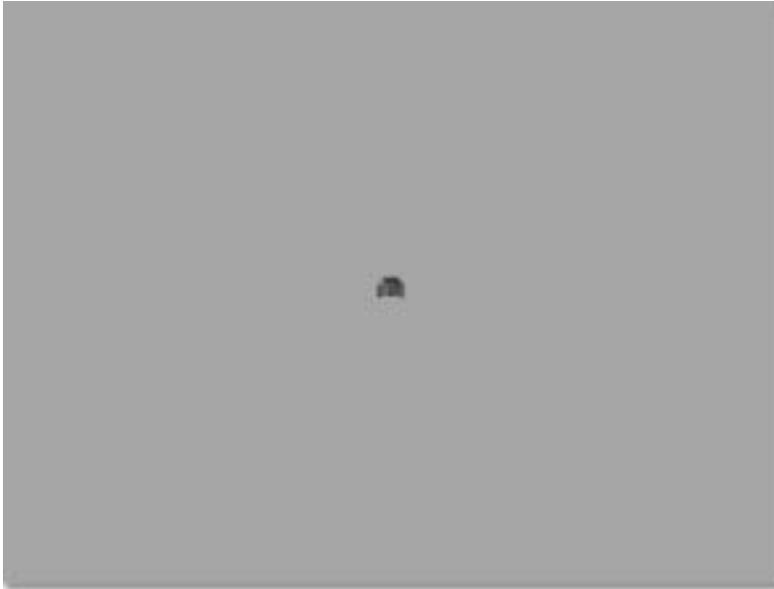
▲ **Abbildung 11**
Detailaufnahme des Automodells

Innerhalb von Director bestimmen Kameras, wie Sprites eine 3D-Szene darstellen.



Scharfe Sache!

Anders als bei echten Kameras ist in einer 3D-Welt in der Regel der komplette Bildausschnitt scharf! Ein Effekt wie Tiefenschärfe lässt sich also nur mit zusätzlichem Aufwand erreichen.



▲ **Abbildung 12**
Das selbe Automodell, mit einem Weitwinkel aufgenommen

Bewegung im 3D-Raum

Translation

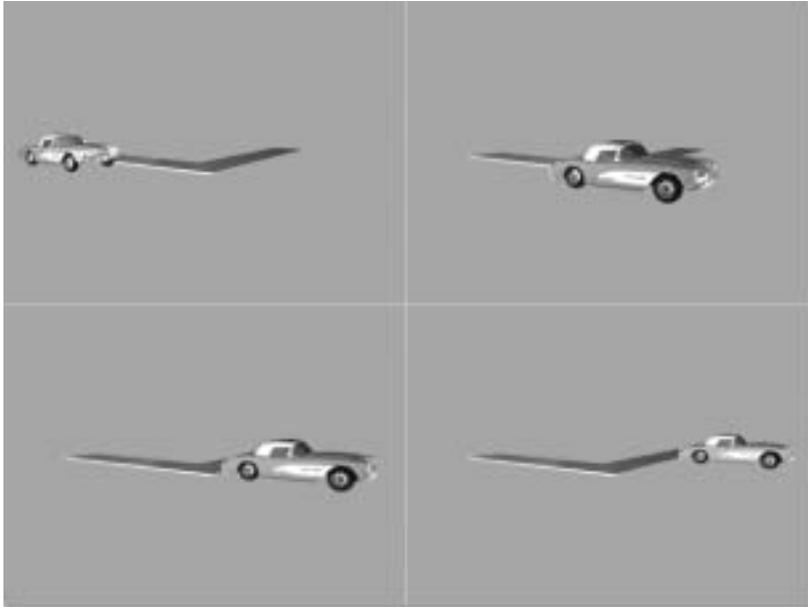
Eine Translation beschreibt eine Verschiebung im Raum. Dabei wird der »Mittelpunkt« des Objekts entlang einer Geraden oder einer Kurve verschoben. Das Objekt behält dabei seine Orientierung bei.

Wird ein Objekt in einer Animation lediglich verschoben, kann das recht merkwürdig aussehen: Man stelle sich nur einmal ein Auto vor, das um eine Kurve fährt, aber dabei seine Fahrtrichtung nicht ändert!

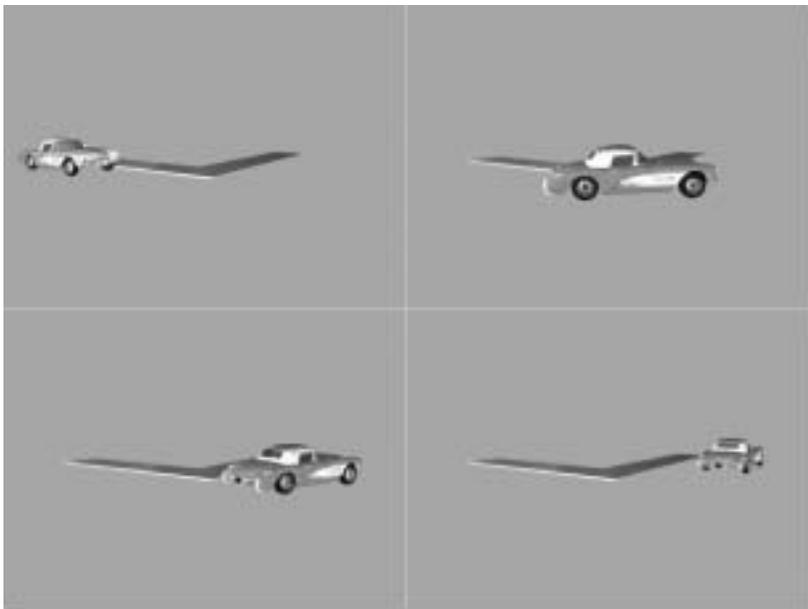
Rotation

Um realistische Bewegungen zu simulieren, brauchen wir also auf jeden Fall auch immer Rotationen. Eine Rotation beschreibt die Drehung eines Objekts. Dabei kann sich das Modell um eine oder auch um mehrere Achsen drehen.

Das Drehzentrum kann dabei innerhalb oder außerhalb des Modells liegen. Liegt das Drehzentrum außerhalb, so kann man sich vorstellen, dass jeder Punkt des Modells auf einer Kugel um einen bestimmten Punkt bewegt wird. Jeder Punkt des Modells behält also immer den gleichen Abstand vom Drehzentrum bei.



▲ **Abbildung 13**
»Kurvenfahrt« nur mit Verschiebung



▲ **Abbildung 14**
Kurvenfahrt mit Verschiebung und Drehung des Modells

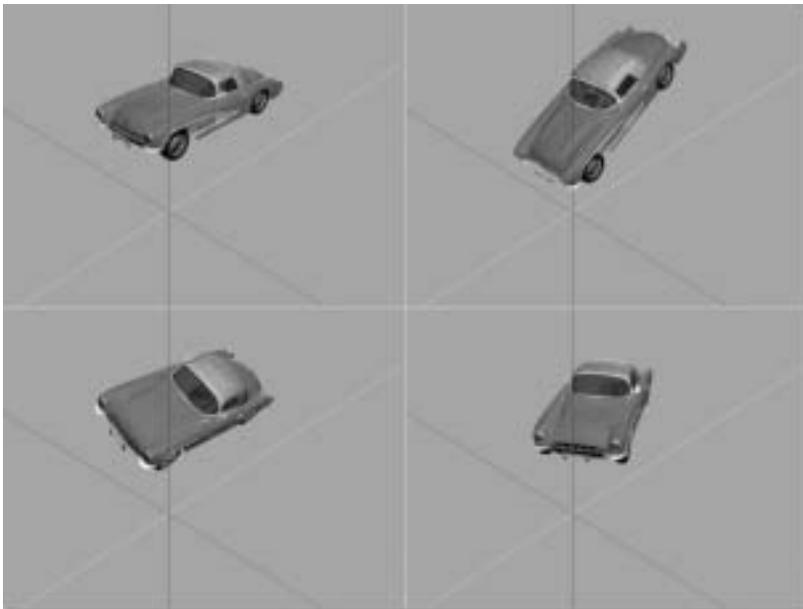
Hierarchien (Parent-Child linking)

Objekte in 3D-Szenen befinden sich meist in hierarchischen Strukturen. Das Gesamtmodell »Auto« beispielsweise besteht aus einer Karosserie und vier Rädern. Die Räder haben dabei als »Parent« (»Elternobjekt« oder auch übergeordnetes Objekt) die Karosserie. Das bedeutet folgendes: Wird das Auto – die Karosserie – bewegt, so bewegen sich die Räder relativ zur Karosserie mit.

Anders ist es jedoch, wenn Child-Objekte (bzw. untergeordnete Objekte) bewegt werden: Das Elternobjekt bewegt sich dann nicht automatisch mit: Werden z.B. bei einer Lenkbewegung die Vorderräder des Autos gedreht, so dreht sich die Karosserie nicht mit.

Lokales vs. globales Koordinatensystem

Wie bereits beschrieben, ist jedes Objekt in zwei Koordinatensystemen »zu Hause«. So können beispielsweise Rotationen relativ zu dem Objektmittelpunkt bzw. den Objektachsen stattfinden. Dabei bleibt das Objekt im Normalfall an der gleichen Stelle in der 3D-Welt.



▲ Abbildung 15

Jeweils Drehung um die lokale X-, Y- und Z-Achse

Es ist aber auch möglich, das Objekt um eine der Weltachsen oder auch um den Weltmittelpunkt zu drehen. An den folgenden Bildern können wir erkennen, dass sich bei einer Rotation um eine der Weltachsen im Normalfall auch die Position des Objektes innerhalb der 3D-Welt ändert.



▲ **Abbildung 16**
Jeweils Drehung um die globale X-, Y- und Z-Achse

Neben dem eigenen Objektmittelpunkt und dem Weltursprung können als Bezugspunkte ebenfalls andere Objekte dienen: Ein Objekt wird dann relativ zu einem anderen Objekt bewegt. So kann beispielsweise eine Kamera auf einer Kreisbahn um ein Objekt rotieren.