8. Graphische Objekte und ihre Erzeugung

Um ein graphisches Objekt auf einem Ausgabegerät darzustellen sind eine Reihe von Schritten erforderlich, die in diesem Kapitel besprochen werden sollen. (Die Mathematik hierzu folgt im Kapitel 10). Dies sei an einem Beispiel erläutert:
Die Aufgabe soll darin bestehen eine Szene (z.B. einen Raum mit Mobiliar, wie ein Hörsaal) zu erstellen und auszugeben (zu rendern).

- In einem ersten Schritt muss das Mobiliar definiert werden. Ein Tisch beispielsweise besitzt eine bestimmte Form und Größe (geometrische Daten) und Attribute wie die Oberflächenbeschaffenheit (Abb. 8.1a).
- Der Tisch muss, so wie die übrigen Möbelstücke, im Raum platziert werden (Abb. 8.1b).
- Es muss festgelegt werden, welche Ansicht bzw. welcher Ausschnitt des Raums auf dem Ausgabegerät erscheinen soll (Definition einer Sicht, hier im gewählten Beispiel die komplette Szene von oben).
- Die gewählte Sicht muss in geeigneter Größe und an der richtigen Stelle der Ausgabefläche dargestellt werden (Abb. 8.1c).

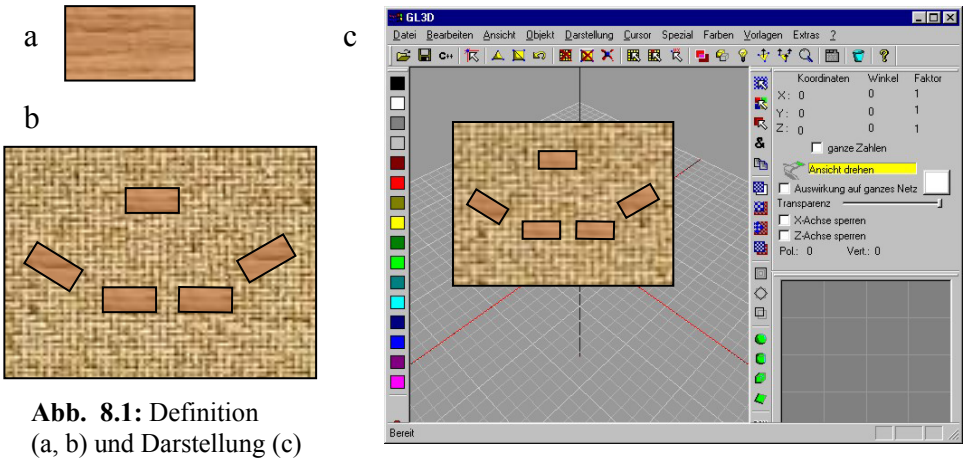


Abb. 8.1: Definition (a, b) und Darstellung (c)

8.1. Koordinaten und Attribute

Um die Form und Größe der Grundobjekte festzulegen, benötigen wir zunächst ein Koordinatensystem in dem das Objekt definiert werden kann.
Am bekanntesten sind ***kartesische Koordinatensysteme***. Zu Modellierungszwecken werden dabei i.a. die rechtshändigen Varianten verwendet (s. Abb. 8.2). Objekte wie der Tisch im obigen Beispiel können hierin sehr einfach beschrieben werden (s. Abb. 8.3).

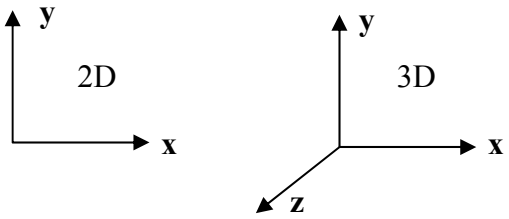


Abb. 8.2: Kartesische Koordinatensysteme

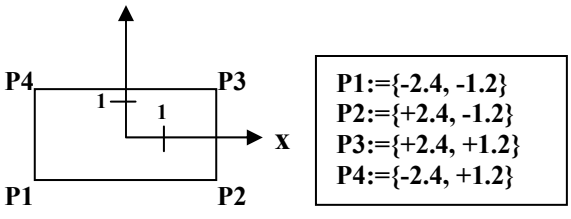


Abb. 8.3 Definition eines Rechtecks in kartesischen Koordinaten

Beachten Sie, dass die Koordinaten hier nicht ganzzahlig (also in Pixel) angegeben werden. Koordinatensysteme, in denen die Grundobjekte modelliert werden, bezeichnet man als ***Modellierungskoordinatensysteme*** oder auch "***körpereigene Koordinatensysteme***". Jedes Objekt das modelliert wird besitzt sein eigenes (daher "körpereigenes") Koordinatensystem. Die Koordinaten können i.a. beliebig als Integer- oder Gleitkommagrößen angegeben werden.
Sind die geometrischen Größen festgelegt, können dem Objekt nun ***graphische Attribute*** zugeordnet werden. Beispiele hierfür sind in Abb. 8.4 zu sehen.

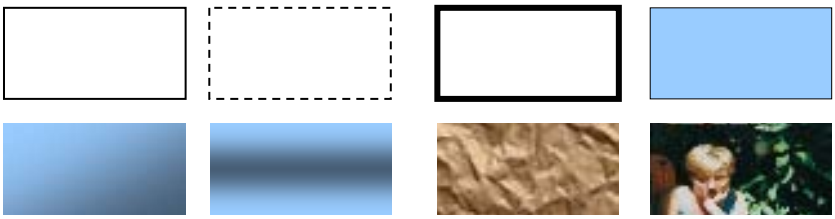


Abb. 8.4 Beispiele für graphische Attribute

Von links oben nach rechts unten sind dies:

- Fläche mit einfachem Rand (z.B. als Drahtgitter, sodass darunter liegende Objekte zu sehen sind.
- Rand gestrichelt
- Randdicke erhöht
- Fläche gefüllt
- Eckpunkte mit unterschiedlicher Farbe, die dann über die Fläche interpoliert wird.
- Fläche mit Reflexionseigenschaften
- Synthetische Textur auf Fläche gemapt (="geklebt")
- JPEG-Bild auf Fläche gemapt (wie man sieht, hier verzerrt).

Wie hätte man vorzugehen, wenn der Tisch nicht rechteckig, sondern rund (kreisförmig) wäre? Wir erinnern uns, dass es für einen Kreis eine Abhängigkeit der y-Koordinate von der x-Koordinate gibt. Diese spiegelt sich in den beiden Funktionen $y = \pm\sqrt{R^2 - x^2}$ wider.

Ein Kreis kann sehr einfach durch 2 Angaben beschrieben werden: Radius und Mittelpunkt. Legen wir das körpereigene Koordinatensystem für den Kreis so fest, dass der Mittelpunkt des Kreises im Ursprung liegt, kann der Kreis allein durch seinen Radius beschrieben werden.

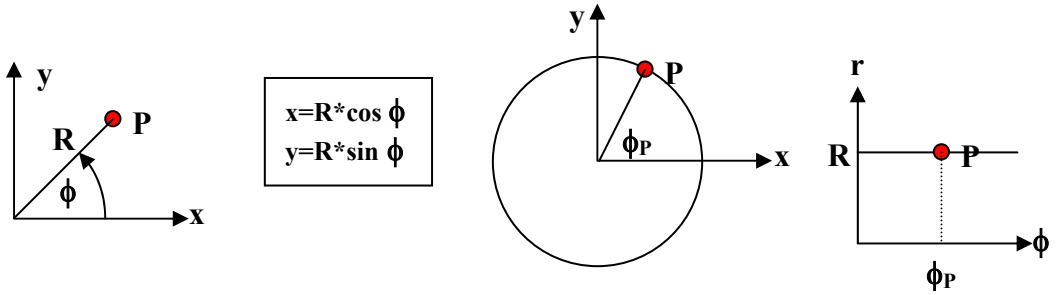
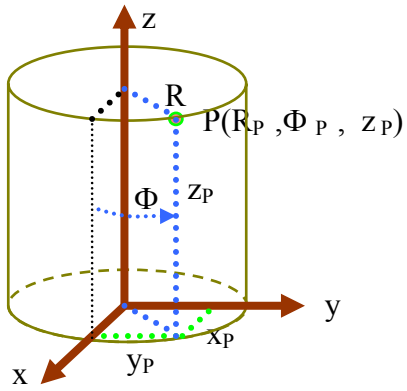


Abb. 8.5 Kreisdefinition in Polarkoordinaten

Der Kreis kann in einem Modell zwar mit dieser Angabe gespeichert werden, zur Ausgabe muss er jedoch in ein regelmäßiges Polygon umgewandelt werden. Dies gilt für alle gekrümmtem Kurven. Eine hohe Dichte der aufeinanderfolgenden Eckpunkte des Polygons gaukeln dem Betrachter eine glatte runde Form vor. Die Form $y = \pm\sqrt{R^2 - x^2}$ ist zur Berechnung der Stützpunkte ungeeignet. Wir verwenden stattdessen ***Polarkoordinaten*** (Abb. 8.5). Hierbei können die x- und y-Koordinaten der Eckpunkte durch gleichmäßige Veränderung des Winkels ϕ gewonnen werden. Polarkoordinaten werden immer dann verwendet, wenn die

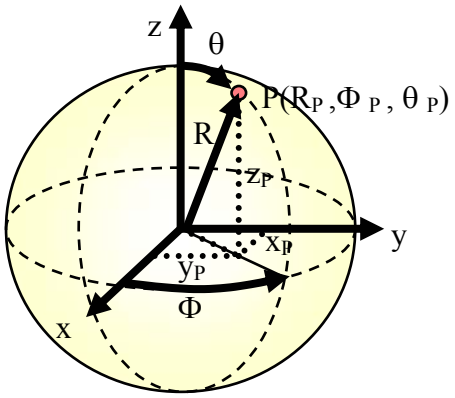
einzelnen Punkte des Objekts durch Drehung um eine Achse gewonnen werden können. Dies ist z.B. auch bei der Definition eines spiralförmigen Objekts der Fall.

In 3D kommen noch 2 weitere Koordinatensysteme hinzu, die für Objektmodellierungen verwendet werden. Dieses sind die ***Zylinderkoordinaten*** und die ***Kugelkoordinaten***. Will man ein Objekt wie eine Dose, eine Tonne, einen Kegel etc., ganz allgemein einen Körper mit Rotationssymmetrie um eine Achse, modellieren, dann verwendet man Zylinderkoordinaten. Diese entstehen aus Polarkoordinaten durch Hinzufügen eines z- Wertes.



x = R * cos Φ y = R * sin Φ z = z

Abb. 8. 6: Zylinderkoordinaten



x = R * sin θ * cos Φ
y = R * sin θ * sin Φ
z = R * cos θ

Abb. 8. 7: Kugelkoordinaten

Will man dagegen Körper mit Rotationssymmetrie bezügliche eines Punktes beschreiben (kugelförmige Gebilde), so verwendet man Kugelkoordinaten. Der Zusammenhang zwischen diesen beiden Koordinatensystemen und kartesischen Koordinaten ist in den Abbildungen 8.6 und 8.7 dargestellt.

8.2. Objekthierarchien

Nachdem nun die einzelnen Objekte in ihren körpereigenen Modellierungskoordinaten definiert worden sind, ist die nächste Aufgabe, sie in größeren Gruppen zusammenzufassen. Dies ist am Beispiel der Abb. 8.8 zu sehen. Bezüglich des übergeordneten Modellierungskoordinatensystems der Gruppe werden die elementaren Objekte positioniert und orientiert. Wir wollen uns diesen Vorgang veranschaulichen. Auf die mathematischen Hintergründe wird in Kapitel 10, auf programmierungstechnische Gesichtspunkte im Kapitel 9 eingegangen.

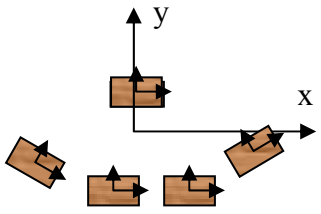


Abb. 8.8: Modellierungskoordinatensystem einer Gruppe

Abbildung 8.9 zeigt das Ergebnis einer Objektverschiebung (***Translation***). Zu den Koordinaten des Objekts werden entsprechend der Größe der Verschiebung Werte hinzuaddiert. Hier in y-Richtung ein positiver Wert, in x-Richtung der Wert 0.

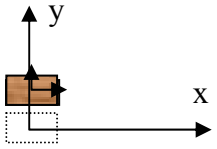


Abb. 8.9: Translation

Ein interessanteres Beispiel ist die Drehung (***Rotation***) eines Objekts um einen beliebigen Punkt. In der Regel sind Objekte in einem ***körpereigenen Koordinatensystem*** definiert (s.o.). Um beim Beispiel zu bleiben: Die Form des Mobiliars (Tisch, Stuhl) soll natürlich von seiner späteren Position im Raum unabhängig sein. Abbildung 8.10 zeigt diesen Sachverhalt (gestricheltes Rechteck mit Mittelpunkt im Ursprung). Das Objekt soll nun an anderer Position um den Winkel α gedreht werden.

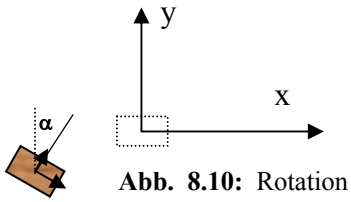


Abb. 8.10: Rotation

Im Prinzip erfolgt diese ***Transformation*** stets in 2 Schritten. Diese sind eine Drehung des Objekts um den Ursprung und eine Verschiebung zur gewünschten Position. Die Abbildungen 8.11 und 8.12 zeigen, dass es bei der Durchführung entscheidend auf die ***Reihenfolge*** ankommt, in der Transformationen durchgeführt werden.

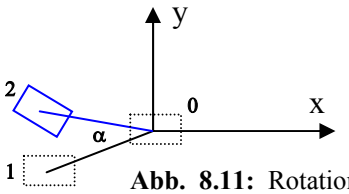


Abb. 8.11: Rotation (Translation+Rotation)

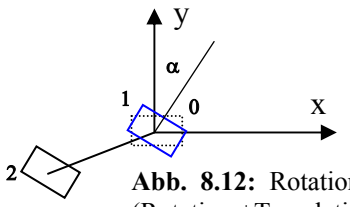


Abb. 8.12: Rotation (Rotation +Translation)

Wird das Objekt zunächst in die Position 1 verschoben und anschließend eine Drehung (um den Ursprung!) um den Winkel α durchgeführt, so ergibt sich das (fehlerhafte) Ergebnis 2 Abbildung 8.11). Wie aus Abbildung 8.12 ersichtlich ist, muss zur Erzielung des korrekten Ergebnisses zuerst die Drehung (1) und erst danach die Verschiebung durchgeführt werden.

Schlussfolgerung: Bei Transformationen, die in mehreren Schritten durchgeführt werden, ist das Ergebnis i.a. von der Reihenfolge der einzelnen Transformationen abhängig. Wegen einer mathematischen Begründung müssen Sie sich bis zum Kapitel 10 gedulden.

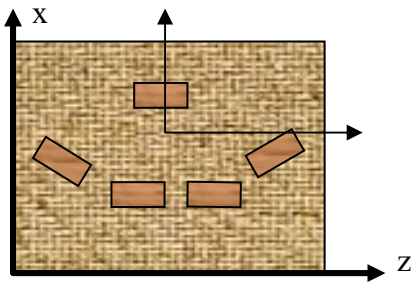


Abb. 8.13: Einbettung in WKS

Durch die Zusammenfassung einzelner Objekte in einem neuen Koordinatensystem ist ein ***Teilmodell*** entstanden. Derartige Teilmodelle können erneut transformiert und zu komplexeren Einheiten zusammengefasst werden. Hierbei kann eine vielstufige ***Objekthierarchie*** entstehen. Am Ende, also in der obersten Ebene der Hierarchie, werden die Teilmodelle in die ***Szene*** (Miniwelt) eingebettet. Das Koordinatensystem der obersten Ebene der Hierarchie wird üblicherweise als ***Weltkoordinatensystem*** (WKS) bezeichnet. Im Beispiel der Abb. 8.13 ist zur Einbettung des Teilmodells in das WKS eine Rotation und eine Translation erforderlich (Beachten Sie, dass das gewählte Beispiel 3D sein soll und die WKS- y-Koordinaten die Höheninformation der Szene beinhalten).

8.3. Die virtuelle Kamera

Die Modellierung einer Szene kann nicht Selbstzweck sein. Ein Modell kann dazu dienen um damit Berechnungen durchzuführen. Im Falle der Modellierung graphischer Objekte ist das Ziel in der Regel eine Visualisierung der Szene. Den Vorgang der Erzeugung einer (möglichst realistischen) Darstellung des Modells auf einem graphischen Ausgabegeräts bezeichnet man als ***Rendering***. Um eine Szene erfolgreich zu rendern sind Angaben darüber erforderlich

- von wo aus (Position des "Betrachters=Kamera") und unter welchem Winkel (Orientierung des "Betrachters"), (= ***äußere Kameraparameter***)
- welcher Ausschnitt der Szene ("Gesichtsfeld" des "Betrachters", Art des Kameraobjektivs) (= ***innere Kameraparameter***).

Dies wird als Definition einer ***View***, einer ***Sicht,*** bezeichnet. Zur Definition einer View wird eine ***virtuelle Kamera*** mit sehr einfachen Eigenschaften (Lochkamera) verwendet (s. Kapitel 4). Diese Kamera besitzt ihr eigenes ***Kamerakoordinatensystem***. Im Gegensatz zu den Modellierungskoordinatensystemen ist dies üblicherweise ein Linkssystem. Dies bedeutet, dass die z-Achsen von Modellierungskoordinaten- und Kamerakoordinatensystem entgegengesetzt gerichtet sind.

Abbildung 8.14 zeigt das Einbringen einer Kamera in die Szene. Im linken Bild ist die Lage und Orientierung des Kamerakoordinatensystems (***äußere Kameraparameter***) bezüglich des WKS dargestellt. Rechts ist der Ausschnitt der Szene zu sehen, der von der Kamera gesehen wird. Der schraffierte Bereich ist (wie in der realen Welt) 3-dimensional und hat (bei perspektivischen Projektionen) die Form einer Pyramide, deren Spitze im Brennpunkt der Kamera liegt. Je spitzer die Pyramide ist, desto mehr entspricht dies einem Teleobjektiv. Eine flache Pyramide entspricht einem Weitwinkel(!)-Objektiv und führt zu starken perspektivischen Verzerrungen der "Aufnahme". Eine Besonderheit der virtuellen Kamera besteht darin, dass der Sichtbereich nicht nur seitlich, sondern auch vorne und hinten begrenzt wird. Hierdurch

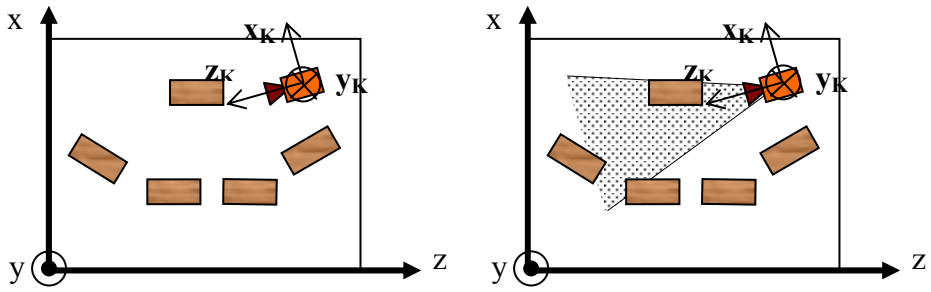


Abb. 8.14: Kamera: Ort, Orientierung und Ausschnitt

wird der Pyramide zum Einen die Spitze abgeschnitten und zum Anderen wird die Pyramide in ihrer Gesamthöhe begrenzt. Der für die Kamera sichtbare Bereich entspricht geometrisch damit einem Pyramidenstumpf, auch ***Frustum*** (Abb. 8.15) genannt.

- **1** ***Brennpunkt*** (COP)
- **2** vordere ***Clipping-Ebene***
- **3** hintere Clipping-Ebene
- **4** seitliche Clipping-Ebenen

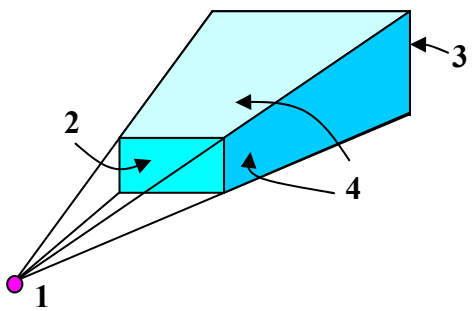


Abb. 8.15: Frustum

Der Vorgang des Ausblendens (=Abschneidens) nicht sichtbarer Teile der Szene wird als ***Clipping*** bezeichnet. Die Festlegung der geometrischen Abmessungen des Frustums entspricht dem Vorgang der Einstellung einer (einfachen) Kamera und damit der Festlegung der sog. ***inneren Kameraparameter***.

Um eine Szene zu rendern, müssen also die Weltkoordinaten noch

1. in das Koordinatensystem der Kamera überführt (transformiert) werden (= ***Definition einer View***)
2. entsprechend den Einstellungen der Kamera verzerrt und anschließend nach 2D projiziert werden (= ***Definition einer Projektion***).
3. Hierauf folgt noch der Schritt der Umwandlung in Gerätekoordinaten (= ***Vektor-Rasterkonvertierung***)

8.4. Szenengraphen

Hierarchien, wie sie im vorausgehenden Abschnitt beschrieben wurden, können in Datenstrukturen gespeichert werden, die als ***gerichtete, zyklenfreie (schleifenfreie) Graphen*** bezeichnet werden.

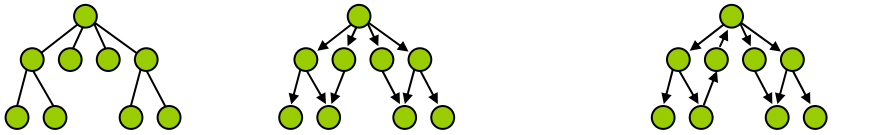


Abb. 8.15: Unterschied Baum und Graph

Im Unterschied zu einem Baum können bei einem gerichteten schleifenfreien Graphen auf einen untergeordneten Knoten mehrere Verweise existieren (Abb. 8.15). Wie bei einem Baum wird der oberste Knoten der Hierarchie als ***Wurzel*** (= ***root***) bezeichnet. Die Knoten der untersten Ebene, jene die keine Kinder haben, heißen ***Blätter*** (= ***Endknoten***, ***terminale Knoten***). Die Wurzel repräsentiert die gesamte Szene. Die Blätter enthalten die Grundelemente und damit die Geometrie, die letztendlich gerendert wird. Die übrigen Knoten enthalten neben den Verweisen auf ihre Kinder möglicherweise die Transformationen, die auf alle ihre Kinder angewendet werden. In der graphischen Datenverarbeitung wird eine derartige Hierarchie häufig als ***Szenengraph*** bezeichnet.

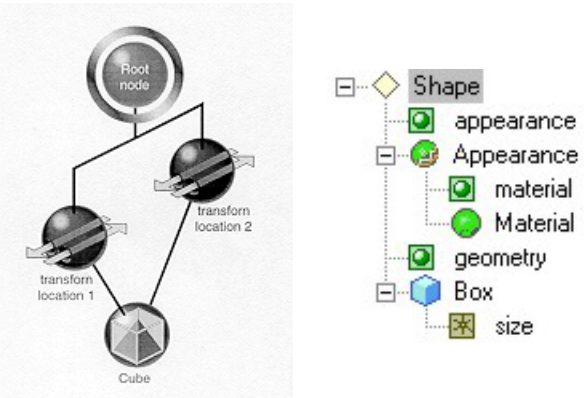


Abb. 8.16: Beispiele 2 einfache Szenengraphen

Szenengraphen werden sehr unterschiedlich dokumentiert. 2 einfache Beispiele hierfür sind in Abb. 8.16 zu sehen.

Das linke Beispiel zeigt eine einfache Szene, die einen Würfel an 2 verschiedenen Positionen enthält. Die rechte Szene enthält einen Würfel, dessen Materialeigenschaften abweichend von Defaultwerten definiert wurden.

Zur Erzeugung von Szenengraphen s. Kapitel 9.