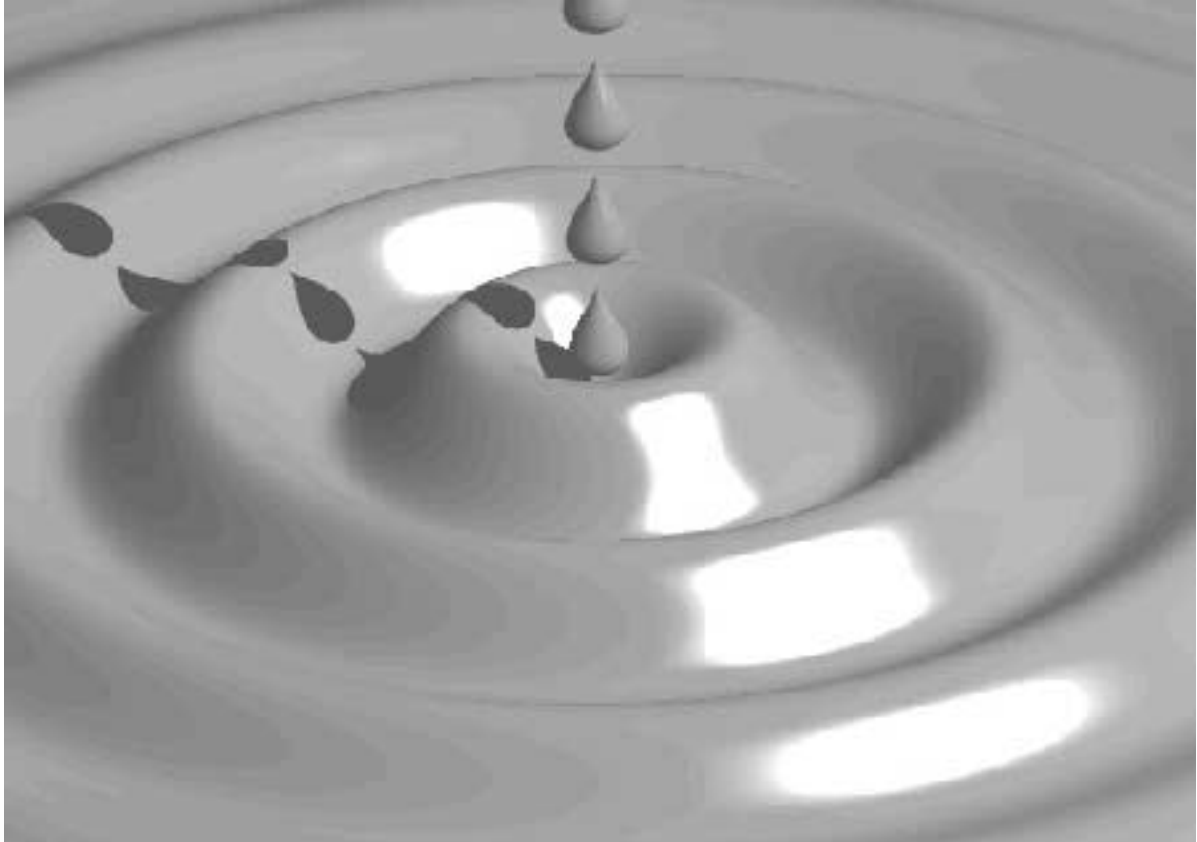


POV-Ray for Windows

Version 3.5



Inhaltsverzeichnis

Einführung	2
Das erste Bild	3
Geometrische Grundformen	4
Die Kugel	4
Der Quader	4
Der Kegel.....	5
Die Ebene	5
Der Torus	6
Vereinigungen, Schnitte und Differenzen	7
Vereinigungen	7
Schnittmenge.....	8
Herausschneiden oder Differenz bilden	8
Rotationskörper	9
Prismen.....	10
Zeichen und Texte	10
Animationen	11

Einführung

Raytracing (Strahlverfolgung) ist ein Verfahren, mit dem fotorealistische 3D-Bilder erzeugt werden. Um eine Vorstellung von der Funktionsweise zu bekommen, schaue man sich das Bild auf der vorherigen Seite oben an. Der Computer berechnet die Bildschirmpunkte indem er die Lichtstrahlen ausgehend vom Beobachter verfolgt. Dabei beachtet er Winkel, Farbe und Reflexionseigenschaften der Oberflächen und die Helligkeit der Lichtquelle. Raytracing wird z.B. bei der Herstellung von computer-generierten Trickfilmen oder bei der Darstellung von Automodellen angewendet. POV-Ray 3.5 für Windows© hat mehrerer Vorteile:

- Es ist Freeware (Download unter www.povray.org).
- Es ist relativ einfach zu erlernen
- Die Ergebnisse können sich sehen lassen.

Ein Nachteil für den Unterricht liegt vielleicht darin, dass man doch ca. 4 Unterrichtsstunden benötigt, um die Programmiersprache von POV-Ray zu erlernen. Danach können die Schüler anspruchsvolle Aufgaben lösen. Da die Online-Anleitung zwar sehr gut aber auf Englisch ist, hier eine kleine Kurzanleitung auf Deutsch. Beachten Sie folgende Schreibweisen:

Zur Kurzanleitung

Buttons des Programms werde so dargestellt:

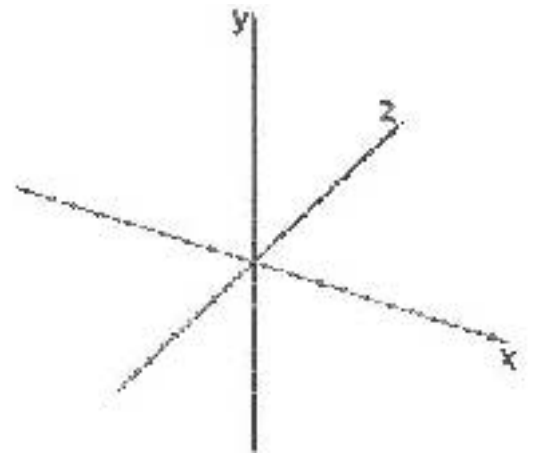
RUN

(Mit RUN wird das 3D-Bild berechnet)

Befehle und Programmteile werden so dargestellt:

include

Ein Nachteil von POV-Ray ist das linkshändige und daher gewöhnungsbedürftige Koordinatensystem. Hier liegt das bekannte x-y-KKS in der Zeichenebene und z in die Blatt- oder Tischebene.



Die Kurzanleitung folgt ist stets in drei Teile gegliedert:

1. Beispielprogramm
2. Erläuterungen
3. Übungen

Das Beispielprogramm kann entweder abgeschrieben oder - falls auf CD/Diskette/Festplatte vorhanden - in POV-Ray geladen werden. Die Erläuterungen beschränken sich auf die im Programm neuen Elemente. Die Übungen sollen Anregungen sein für die Erkundung der vorgestellten Befehle. Übungen in ***Fettkursiv*** sind komplexer oder anspruchsvoller. Im Unterricht speichern die Schüler Ihre Dateien zur Kontrolle ab. Der Dateiname für diese Übungsdateien ist in Klammern angegeben. (Kugel1.pov)

Das erste Bild

Beispielprogramm 11Kugel.pov:

```
#include "colors.inc"
background {color White}
camera {
    location <1,2,-3>
    look_at <0,0,0>
    angle 42}

light_source { <3,3,-3> color White}
sphere { <0,0,0> 1
    texture{pigment {color Blue}} }
```



Erläuterungen zum Programm:

Bestimmte Befehle sind für jedes POV-Ray-File obligatorisch. Dazu gehören:

#include:	Damit werden vordefinierte Farben, Oberflächen, Materialien, etc geladen.
background:	legt die Hintergrundfarbe des Bildes fest (White, Black, Gray,...)
{...}	Angabe von Befehlen gehören in geschweifte Klammern
<...>	Koordinatenangaben gehören in spitze Klammern
camera {location	legt die Position der Kamera im KKS fest, (hier x=1, y=2, z=-3)
look_at	legt den Punkt fest, den die Kamera anvisiert (hier den Koordinatenursprung)
angle	legt den Blickwinkel fest
light_source	legt die Position der Lichtquelle(n) fest
sphere	definiert eine Kugel, mit (0,0,0) als Mittelpunkt und r=1 als Radius
texture	legt die Oberflächenstruktur fest
pigment	einfachste Oberflächenstruktur: nur Farbpigmente
color Blue	legt die Farbe der Pigmente fest

Übungen

Bitte laden Sie dazu das Programm 11Kugel.pov. Bei den folgenden Übungen werden Grundeinstellungen geändert, um ein Gespür für das Programm zu bekommen. Dabei kann es passieren, dass kein Bild zu sehen ist: Dann schaut die Kamera vielleicht an die falsche Stelle oder die Lichtquelle ist hinter dem Objekt.

1. Gehen Sie mit der Kamera weiter weg von dem Objekt. (z.B.: z=-20)
2. Verändern Sie den Bildausschnitt mit Hilfe von camera und angle
3. Positionieren Sie die Lichtquelle so, dass ein Halbmondbild entsteht
4. Geben Sie dem Mond die Farbe Weiß.
5. Fügen Sie eine weitere Lichtquelle hinzu. Speichern unter (1Kugel.pov).

Geometrische Grundformen

POV-Ray besitzt eine Reihe von geometrischen Grundformen. Diese können direkt benutzt werden. Im folgenden werden die dazugehörigen Befehle nur ganz kurz vorgestellt.

Die Kugel

siehe "Das erste Bild"

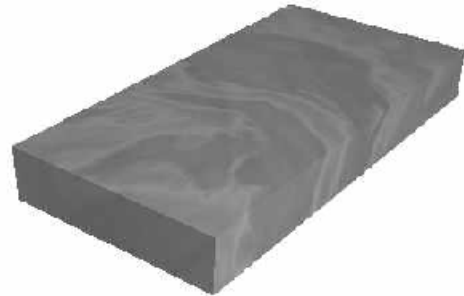
Der Quader

Beispielprogramm 12Quader.pov:

```
#include "colors.inc"
#include "stones.inc"
background {color White}
camera {
    location <2,4,-6>
    look_at <0,0,0>
    angle 42}

light_source { <3,3,-3> color White}

box {
    <-1, 0, -1>, // linke untere Ecke
    < 1, 0.5, 3> // rechte obere Ecke
    texture {
        T_Stone1 // Pre-defined from stones.inc
        scale 4 // Scale by the same amount in all
                // directions
    }
    rotate <0,20,0> }
```



Erläuterungen

#include "stones.inc"	Es werden Oberflächenstrukturen von Steinen geladen
box	definiert einen Quader mit den Eckpunkten (-1,0,-1) und (1,0.5,3)
// linke...	// leitet einen Kommentar ein, der ignoriert wird
texture T_Stone1	legt die Oberflächenstruktur fest
scale 4	vergrößert die Oberflächenstruktur
rotate	dreht das Objekt box um die y-Achse mit 20°

Hinweis: Beachten Sie, dass texture und rotate zum Box-Objekt gehören und deshalb in {...} eingeklammert sind.

Übungen

1. Stellen Sie einen Würfel dar, dessen Mittelpunkt im Koordinatenursprung liegt.
2. Legen Sie eine Kugel aus Stein auf den Würfel.
3. Legen Sie eine Kugel mit dem Würfelvolumen in den Ursprung. (1Würfel.pov).

Der Kegel

Beispielprogramm 13Kegel.pov:

```
#include "colors.inc"
#include "stones.inc"
background {color White}
camera {
    location <2,4,-6>
    look_at <0,0,0>
    angle 42}

light_source { <3,3,-3> color White}

cone {
    <0, 1, 0>, 1.0    // Center and radius of one end
    <1, 2, 3>, 0.3    // Center and radius of other end
    texture { T_Stone35 scale 4 }
    translate <-1,-2,0>
}
```



Erläuterungen

Cone	definiert einen Kegel
<0, 1, 0>, 1.0	definiert den Mittelpunkt des 1. Deckkreises und dessen Radius
<1, 2, 3>, 0.3	definiert den Mittelpunkt des 2. Deckkreises und dessen Radius
translate <-1,-2,0>	verschiebt das Objekt mit dem Vektor (-1,-2,0)

Hinweis:
Wenn beide Radien gleich sind, entsteht ein Zylinder.

Übungen

1. Zeichnen Sie einen Kegel mit einem passenden Zylinder darauf. (1Zylinder.pov)

Die Ebene

Ebenen in POV-Ray haben Ähnlichkeit mit der Normalform der Ebenengleichung. Es wird der Normalenvektor angegeben und der Abstand zum Koordinatenursprung.

Beispielprogramm: 14Ebene.pov

```
#include "colors.inc"
background {color White}
camera {
    location <2,4,-6>
    look_at <0,0,0>
    angle 42}

light_source { <3,3,-3> color White}

plane { <0, 1, 0>, -1
    pigment {
        checker color Red, color Blue
    }
}
```



Erläuterungen

`plane { <0, 1, 0>, -1`

Definiert eine Ebene mit dem Normalenvektor (senkrecht auf der Ebene) (0,1,0) und einem Abstand von -1 vom Ursprung

`checker color Red, color Blue`

Definiert ein Schachbrettmuster als Farbe der Ebene

Übungen

1. Stellen Sie die drei Koordinatenebenen eines räumlichen KKS' dar. (1KKS.pov)
2. Stellen Sie eine Ebene so dar, dass man den Horizont sieht.
3. Zeichnen Sie eine Kugel und legen Sie eine Tangentialebene daran. (1Tangentialebene.pov)

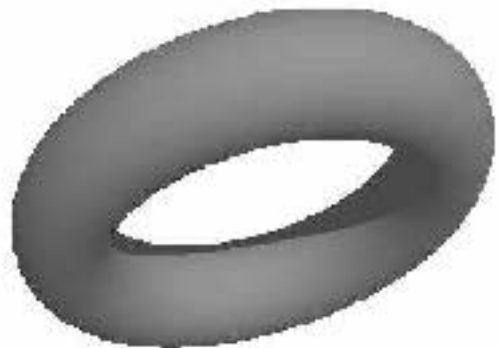
Der Torus

Unter Torus versteht man einen Ring deren Querschnitte Kreise konstanter Größe sind.

Beispielprogramm 15Torus.pov

```
#include "colors.inc"
background {color White}
camera {
    location <0, .1, -25>
    look_at 0
    angle 30
}

light_source { <300, 300, -1000> White }
torus {
    4, 1 // major and minor radius
    rotate <30,30,30>
    pigment { Green }
}
```



Erläuterungen

`torus {4 ,1}` definiert einen Torus. Der Durchmesser des „Fahrradschlauchs“ ist 4. Der Durchmesser des Querschnittes (Reifenbreite) ist 1.

`rotate <30,30,30>` dreht den Torus aus seiner Normallage um je 30° um die x-,y- und z-Achse.

Übungen

1. Zeichnen Sie einen Torus in seiner Normallage.
2. Zeichnen Sie einen Torus, der auf einer Ebene stehen wie ein Rad. (1Torus.pov)

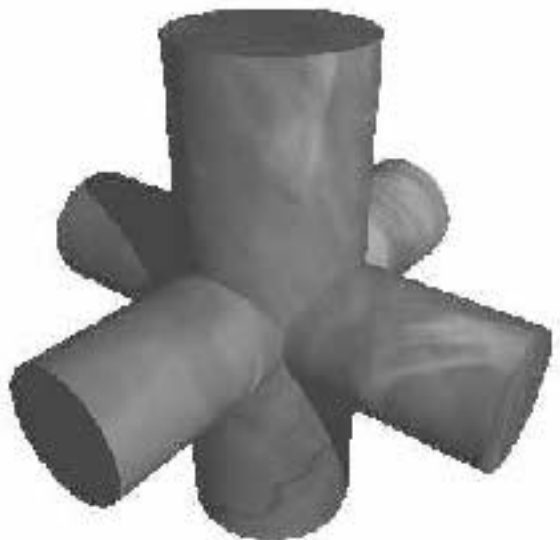
Vereinigungen, Schnitte und Differenzen

Vereinigungen

Möchte man ein komplexes Gebilde bauen, braucht man Vereinigungen. Drehungen und Farbgebung brauchen dann nicht für jedes Teilobjekt definiert zu werden, sondern nur einmal für das gesamte Objekt.

Beispielprogramm 21Vereinigung.pov

```
#include "colors.inc"
#include "stones.inc"
background {color White}
camera {
    location <1,2,-3>*1.5
    look_at <0,0,0>}
union{
    cylinder {
        <-2,0,0>,
        < 2,0,0>,
        0.5}
    cylinder {
        <0,-2,0>,
        < 0,2,0>,
        0.7}
    cylinder {
        <0,0,-2>,
        < 0,0,2>,
        0.5}
    rotate <0,20,0>
    texture {T_Stone20 scale 6}
}
light_source { <3,3,-3> color White}
```



Erläuterungen

`union{` enthält beliebige Objekte, die dann als ein ganzes behandelt werden. Die Eigenschaften des Gesamtobjekts werden am Ende - nach der Definition der Teilobjekte - angegeben.

`cylinder` definiert einen Zylinder mit den Mittelpunkten der Deck-Kreise und deren Radius.

`rotate <0,20,0>` bezieht sich hier auf alle Objekte die zur Vereinigung gehören: Drehung um die y-Achse mit 20°.

`texture {T_Stone20 scale 6}` bezieht sich ebenfalls auf das gesamte geometrische Objekt: Steinstruktur.

Übungen

1. Verdünnen Sie die Zylinder und drehen Sie ihn wie in ein 3D-Koordinatensystem.
2. Zeichnen Sie ein 3D-Koordinatensystem deren Achsen aus dünnen Zylindern und Spitzen aus kleinen Kegel bestehen. (3D-Koordinaten.pov)



Schnittmenge

Dabei entsteht ein Körper, der in allen Teilobjekten enthalten ist.

Beispielprogramm 22Schnittmenge.pov

```
#include "colors.inc"
#include "stones.inc"

background {color White}
camera {
    location <0,0,-2>
    look_at <0,0,0>
    angle 90
}
intersection{
    sphere {<-1,0,0>,1.5}
    sphere {<1,0,0>,1.5}

    rotate <0,10,0>
    texture {T_Stone14 scale 1}
}

light_source { <3,3,-3> color White}
light_source { <-3,-3,-3> color White}
```



Erläuterungen

`intersection{ }` es entsteht ein geometrisches Objekt, dass der Durchschnitt aller Teilobjekte ist. Der Durchschnitt zweier versetzter Kugeln ist eine Linse.

`light_source` es wurde eine zweite Lichtquelle eingeführt, um die Rückseite zu beleuchten.

Übungen

1. Bilden Sie den Durchschnitt aus einem Würfel und einer Kugel (2Schnitt.pov)

Herausschneiden oder Differenz bilden

Beispielprogramm 23Differenz.pov

```
#include "colors.inc"
#include "stones.inc"
background {color White}
camera {
    location <0,0,-2>
    look_at <0,0,0>
    angle 90
}
difference {

    intersection {
        sphere {<-1,0,0>,1.5}
        sphere {<1,0,0>,1.5}
    }
    cone {<-3,0,0>,.5
        <3,0,0>,.5}
    rotate <0,60,30>
    texture {T_Stone14 scale 1}
}

light_source { <3,3,-3> color White}
light_source { <-3,-3,-3> color White}
```



Erläuterungen

`difference {...}` schneidet aus der Linse (wieder als Durchschnittsmenge zweier Kugeln definiert) einen Zylinder heraus.

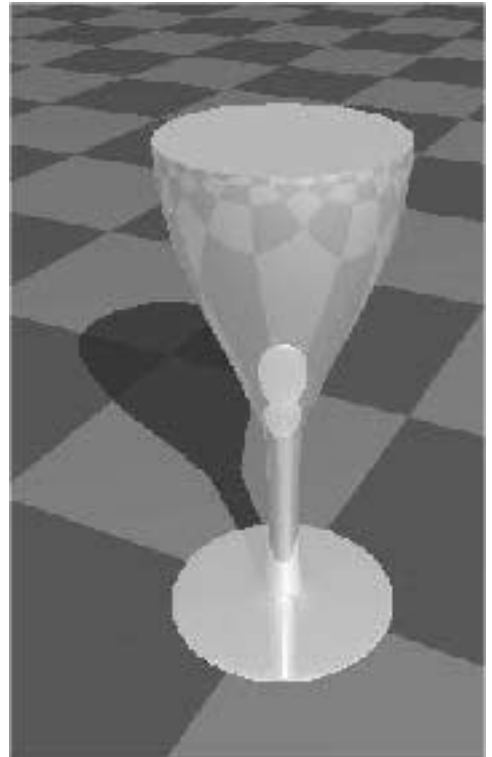
Übungen

Schneiden Sie aus einer Kugel einen Quader heraus.

Rotationskörper

Beispielprogramm 31Rotationskörper.pov

```
#include "colors.inc"
#include "golds.inc"
    camera {
        location <10, 15, -20>
        look_at <0, 5, 0>
        angle 45
    }
    background { color White}
    light_source { <100, 100, -100> color White}
    plane {<0,1,0>, 0
        pigment {checker color Red, color Green scale
10 }
    }
    sor {
        8,
        <0.0, -0.5>,
        <3.0, 0.0>,
        <1.0, 0.2>,
        <0.5, 0.4>,
        <0.5, 4.0>,
        <1.0, 5.0>,
        <3.0, 10.0>,
        <1.0, 11.0>
        texture { T_Gold_1B }
    }
```



Erläuterungen

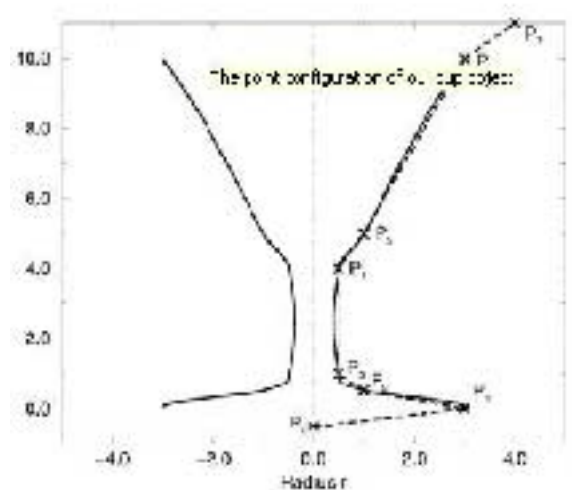
#include "golds.inc" ermöglicht die Benutzung von Gold-Oberflächen.

sor { 8, ...} definiert einen Rotationskörper mit 8 Stützpunkten. Die Punkte P_0 bis P_7 werden als 2D-Koordinaten angegeben und geben den Querschnitt an, der dann um die y-Achse rotiert wird. Die Punkt P_0 und P_7 selbst gehören nicht zum Objekt – sie geben die Tangentenrichtung in den Punkten P_1 bzw. P_6 an. Die Punkte werden mit Kurven verbunden.

texture { T_Gold_1B } definiert eine Goldoberfläche für das Objekt.

Übungen

1. Variieren Sie den Punkt P_7 und beobachten Sie den Einfluss auf die Form des Glases.
2. Stellen Sie ein reales Glas (von zu Hause) mit Hilfe von PovRay 3dimensional dar. (3Glas.pov)



Prismen

Beispielprogramm 41Prisma.pov

```
#include "colors.inc"
background{White}
camera {
  angle 20
  location <2, 10, -30>
  look_at <0, 1, 0>
}
light_source { <20, 20, -20> color White }
prism {
  linear_sweep
  linear_spline
  0, // Höhe der Grundfläche
  1, // Höhe der Deckfläche
  7, // Zahl der Eckpunkte plus 1
  <3,5>, <-3,5>, <-5,0>, <-3,-5>, <3, -5>, <5,0>, <3,5>
  pigment { Green }
}
```



Erläuterungen

Angaben nach // sind Erklärungen zum Programm

`prism { linear_sweep...` definiert ein Prisma

`linear_spline` definiert die Grundfläche durch die Angabe von 2D-Koordinaten der Eckpunkte.

Beachte: Der letzte Punkte ist gleich dem Startpunkt.

Übung

Zeichnen Sie eine gleichseitige 5-seitige Pyramide. Ersetzen Sie dabei `linear_sweep` durch `conic_sweep`.

Zeichen und Texte

Beispielprogramm 51Text.pov

```
#include "colors.inc"
#include "stones.inc"
background {White}
camera {
  location <0, 1, -10>
  look_at 0
  angle 35
}
light_source { <500,500,-1000> White }
plane {
  y,0
  pigment { checker Green White }
}
text {
  ttf "arial.ttf" "POV-RAY 3.5" .3, 0 scale 2
  texture { T_Stone15 scale 4}
  translate <-1.5,.4,-2>
  rotate <5,25,5>
}
```



Erläuterungen

`text { ttf "arial.ttf"...` schreibt einen Text in der true-type-font Arial

`.3` gibt die räumliche Tiefe der Buchstaben an

`0` gibt die Steigung der Schrift an (Offset)

`scale 2` vergrößert die Schrift um den Faktor 2

Übungen

Variieren Sie das Programm so, dass der Schriftzug nicht mehr abgeschnitten wird. (5Schrift.pov)

Beispiel 52Text.pov

```
#include "colors.inc"
camera {
    location <-5,.15,-2>
    look_at <.3,.2,1>
    angle 35
}
light_source { <500,500,-1000> White }
plane {
    y,0
    texture {
        pigment { SeaGreen }
        finish { reflection .35 specular 1 }
        normal { ripples .35 turbulence .5 scale
.25 }
    }
}
text {
    ttf "Arial.ttf" "POV-RAY 3.5" .08, 0.1*y
    pigment { BrightGold }
    finish { reflection .25 specular 1 }
    translate -3*x
}
#include "skies.inc"
sky_sphere { S_Cloud5 }
```



Animationen

Um eine Animation herzustellen sind folgende Schritte zu durchlaufen:

1. POV-Ray-Bild programmieren mit einer Variable namens „clock“
2. Programmierung einer INI-Datei. Siehe unten.
3. INI-Datei rendern. Dabei werden BMP-Bilder erzeugt.
4. BMP-Bilder mit dem Programm „bmp2avi“ in ein avi-Film umsetzen

Beispielprogramm mit clock:

```
#include "colors.inc"
#include "stones.inc"
background {White}
camera {
    location <0, 1, -10>
    look_at 0
    angle 35
}
light_source { <500,500,-1000> White }
plane {
    y,0
    pigment { checker Green White }
}
text {
    ttf "arial.ttf" "POV-RAY 3.5" .3, 0
scale .6
    texture { T_Stone15 scale 4}
    translate <-1.7,.4,-2>
    rotate <5,clock,5>
}
```



INI-Datei zum rendern: 61Animation.ini

```
Antialias=On
Antialias_Threshold=0.2
Antialias_Depth=3
Input_File_Name=61Animation.pov

Initial_Frame=1
Final_Frame=30
Initial_Clock=0
Final_Clock=30

Pause_when_Done=off
Cyclic_Animation=off
```

Schlussbemerkungen:

Weitere Hilfen finden Sie in der Hilfe zu POV-Ray 3.5 im Kapitel „beginning tutorial“.

Eine gute deutsche Anleitung findet man unter:

http://www.f-lohmueeller.de/pov_tut/pov__ger.htm

Komplexe Aufgabe

Bearbeiten Sie eine komplexe Aufgabe. Programmieren Sie möglichst effektiv mit Hilfe von Vereinigungen (unions) damit sie Gruppen von Elementen manipulieren können (und nicht jedes Objekt einzeln). Notieren Sie Schwierigkeiten und Herangehensweisen für eine *Präsentation*.

Vorschläge

1. **Stilleben:** Erstellen Sie ein Stilleben bestehend aus einem Tisch mit realen Gefäßen und geometrischen Formen (Rotationskörpern).
2. **Kegelschnitte:** Schneiden Sie einen Kegel mit einer Ebene und stellen Sie die verschiedenen Schnittmengen übersichtlich dar.
3. **Technik:** Konstruieren Sie ein Fahrzeug oder anderes technisches Gerät (Zahnräder) und erzeugen Sie dazu eine Animation.