

Für die Lichtquellen gilt folgendes:

Zwei Lichtquellen sollen die Szene beleuchten und zwar: von oben schräg links und von unten schräg rechts.

1.Aufgabe: a) Erläutere das Verfahren, wie man in PovRay eine Animation erstellt. Nimm als Beispiel die Drehung eines Würfels um die eigene Achse.

Welche Bedeutung hat die clock-Variable?

Erläutere auch die Bedeutung der sog. INI-Datei. Erkläre die mit einem Fragezeichen versehenen Begriffe.

Antialias=On

Antialias_Threshold=0.5

Antialias_Depth=5

Input_File_Name=?

Initial_Frame=?

Final_Frame=?

Initial_Clock=?

Final_Clock=?

Cyclic_Animation=on

Pause_when_Done=off

2.Aufgabe:

a) Welche Werte nimmt die Variable clock in der folgenden INI-Datei an? Gib die ersten 10 Werte an.

Antialias=On

Antialias_Threshold=0.5

Antialias_Depth=5

Input_File_Name=?

Initial_Frame=1

Final_Frame=10

Initial_Clock=0

Final_Clock=1

Cyclic_Animation=on

Pause_when_Done=off

b) Welche Werte nimmt die Variable clock in der folgenden INI-Datei an? Gib die ersten 10 Werte an.

Antialias=On

Antialias_Threshold=0.5

Antialias_Depth=5

Input_File_Name=?

Initial_Frame=1

Final_Frame=50

Initial_Clock=0

Final_Clock=2

Cyclic_Animation=on

Pause_when_Done=off

3.Aufgabe: Erläutere die Struktur der WHILE-Schleife an folgendem Beispiel: Auf der x-Achse sollen 5 Würfel in einer Reihe erzeugt werden. Der mittlere Würfel soll im Ursprung des Koordinatensystems platziert sein.

4. Aufgabe: a) Schreibe ein POV-Programm, in dem 4 Würfel auf der z-Achse angeordnet werden. Der erste Würfel befindet sich dabei im Koordinatenursprung.
 b) Schreibe ein POV-Programm, in dem 5 Kugeln auf der y-Achse angeordnet sind. Dabei soll die erste Kugel ihren Mittelpunkt in $\langle 0,0,-4 \rangle$ haben.

5. Aufgabe:

a) Was passiert mit folgender WHILE-Schleife? Erläutere!

```
#declare Ball=sphere{<0,0,0>, 0.5 pigment{Red}}
#declare Nr = 0;
#declare EndNr = 4;
#while (Nr<EndNr+1)
  object{Ball translate<Nr,Nr,0> pigment{Red}}
  #declare Nr = Nr+1
#end
```

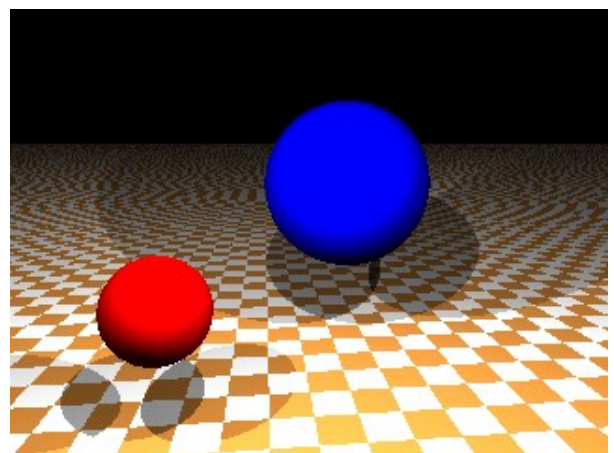
b) Was passiert mit folgender WHILE-Schleife? Erläutere!

```
#declare Ball=sphere{<0,0,0>, 0.5 pigment{Red}}
#declare Nr = 0;
#declare EndNr = 4;
#while (Nr<EndNr+1)
  object{Ball translate<Nr,Nr,Nr> pigment{Red}}
  #declare Nr = Nr+1
#end
```

6. Aufgabe: Erläutere die Begriffe an selbstständig gewählten Beispielen: **translate**, **rotate** und **scale**.

Beispiel:

```
#include "colors.inc"
#include "textures.inc"
#include "shapes.inc"
//Unser Standpunkt, das Auge des Betrachters
camera{ location<0,3,-9> // hier stehe ich
  look_at<1,1,1> } // da schaue ich hin
// die Lichtquelle
light_source {<0,5,-5> color White }
// die rote Kugel
sphere {<-2,0,-1>,1 pigment {Red} }
// die blaue Kugel
sphere{<2,2,2>,2 pigment {Blue} }
// die unendliche Fläche
plane {y,-3.0 pigment{ checker color White color Gold} }
```



Lösung

1.Aufgabe: Eine Animation erfordert zwei Dateien. Zum einen muss in einer POV-Ray-Datei die gewünschte Szene erzeugt werden. Um die benötigten Einzelbilder zu erzeugen, muss eine sog. INI-Datei angelegt werden, die mit Hilfe der clock-Variablen die Einzelbilder erzeugt. Diese Einzelbilder werden dann mit einem speziellen Programm z.B. BMP-to-AVI zu einer Video-Datei zusammengefügt werden.

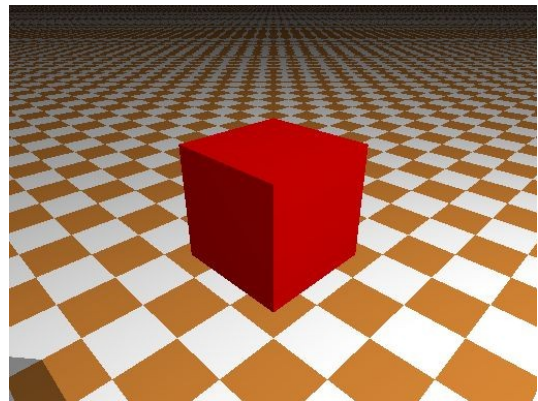
Rotierender Würfel

1.Die POV-Ray-Datei

```
#include "colors.inc"
#include "shapes.inc"
camera{ location<-2,1.5,-2> look_at<0,0,0>}
```

```
light_source{<5,5,-5>color White}
light_source{<-5,5,-5>color White}
```

```
box{ <-1,-1,-1>,<1,1,1> pigment {Red}
scale<0.5,0.5,0.5>
rotate<0,-360*clock,0>}
```



```
plane {y,-3.0 pigment{ checker color White color Gold} }
```

2.Die INI-Datei

```
Antialias=Off
Antialias_Threshold=0.1
Antialias_Depth=2
Input_File_Name=aufgabe1.pov
Initial_Frame=1
Final_Frame=20
Initial_Clock=0
Final_Clock=1
Cyclic_Animation=-K
Pause_when_Done=+P
```

Die clock-Variable wird benötigt, um unterschiedliche Bilder zu erzeugen. Die Werte werden automatisch verändert. Dieses hängt mit den Variablen Initial_Frame, Final_Frame, Initial_Clock und Final_Clock zusammen. In der obigen INI-Datei ergeben sich folgende Werte: Es werden 20 Bilder erzeugt (Final_Frame - Initial_Frame+1 = 20), die Clock-Variable überstreicht einen Wertebereich von 0 bis 1. Teilt man diesen Bereich in 20 Teile für die 20 Bilder auf, so erhält man für die Clock-Variable folgende Werte: 0, $1/20 = 0,05$, $2 * 1/20 = 0,1$, $3 * 1/20 = 0,15$ $20 * 1/20 = 1$.

2.Aufgabe: a)Wie in Aufgabe erläutert, ergibt sich für die Werte der Clock-Variablen:
0- 0,1 - 0,2 - 0,3 - 0,4 - 0,5 - 0,6 - 0,7 - 0,8 - 0,9 - 1

a) Wie in Aufgabe erläutert, ergibt sich für die Werte der Clock-Variablen:
0- 0,04 - 0,08 - 0,12 - 0,16 - 0,20 - 0,24 - 0,28 - 0,32 - 0,36 2

3. Aufgabe:

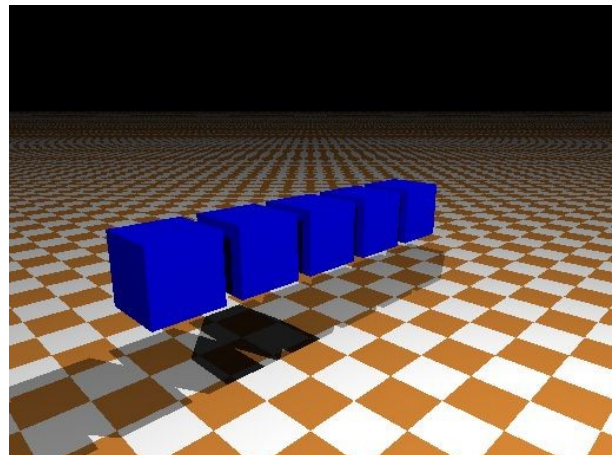
```
#while {Bedingung}  
    // Solange die Bedingung wahr ist, werden die  
    // folgenden Anweisungen ausgeführt
```

```
{Anweisungen}
```

```
#end // Hier ist die Schleife zu Ende
```

Hier nun das Programm für 5 Würfel auf der x-Achse, beginnend im Ursprung

```
#include "colors.inc"  
#include "shapes.inc"  
camera{ location<-2,1.5,-4> look_at<2,0,0>}  
  
light_source{<5,5,-5>color White}  
light_source{<-5,5,-5>color White}  
  
#declare Wuerfel = box{<-1,-1,-1><1,1,1> pigment {Blue} scale<0.4,0.4,0.4>};  
#declare XWert = 0;  
#declare XEnde = 5;  
  
#while (XWert<XEnde)  
    object{Wuerfel translate<XWert,0,0>}  
    #declare XWert = XWert + 1; // next Nr  
#end  
  
plane {y,-3.0 pigment{ checker color White  
color Gold} }
```



4. Aufgabe: a)

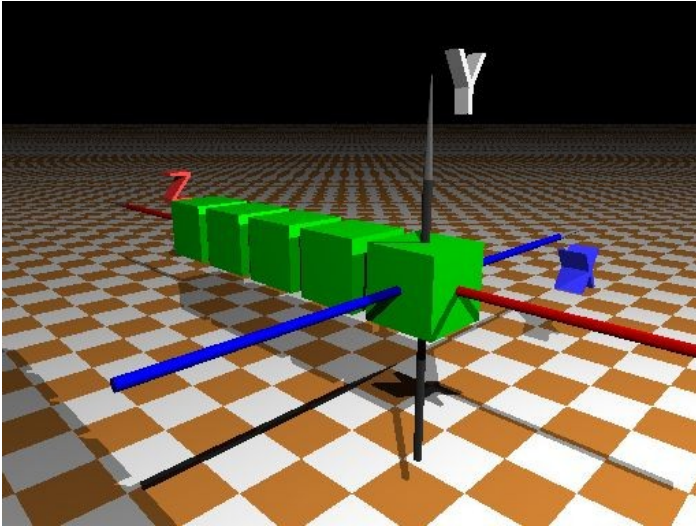
```
#include "colors.inc"  
#include "shapes.inc"  
camera{ location<-4,1.5,-3> look_at<0,0,1>}  
  
light_source{<5,5,-5>color White}  
light_source{<-5,5,-5>color White}
```

```

#declare Wuerfel = box{<-1,-1,-1><1,1,1> pigment {Green} scale<0.4,0.4,0.4>};
#declare ZWert = 0;
#declare ZEnde = 5;

#while (ZWert<ZEnde)
  object{Wuerfel translate<0,0,ZWert>}
#declare ZWert = ZWert + 1; // next Nr
#end

```



Zur Übersichtlichkeit ist das Koordinatenkreuz mitgezeichnet worden.

b) 5 Kugel auf der y-Achse, beginnend bei <0,0,-4>

```

#include "colors.inc"
#include "shapes.inc"
camera{ location<-4,1.5,-10> look_at<0,1,0>}

light_source{<5,5,-5>color White}
light_source{<-5,5,-5>color White}

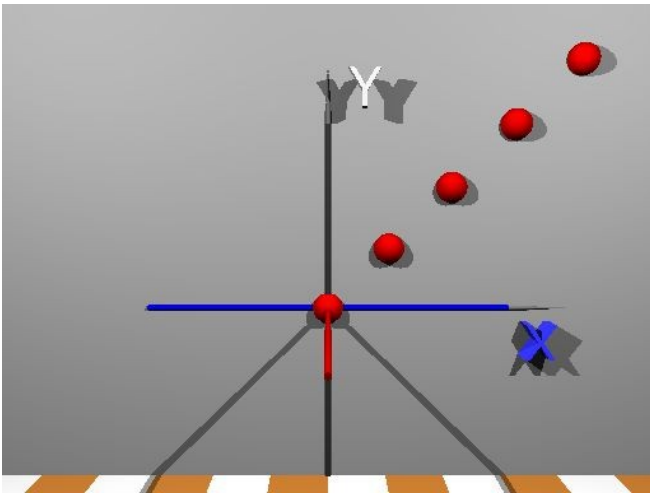
#declare Kugel = sphere{<0,0,0>,1 pigment {Green} scale<0.4,0.4,0.4>};
#declare YWert = 0;
#declare YEnde = 5;

#while (YWert<YEnde)
  object{Kugel translate<0,YWert,-4>}
  #declare YWert = YWert + 1; // next Nr
#end

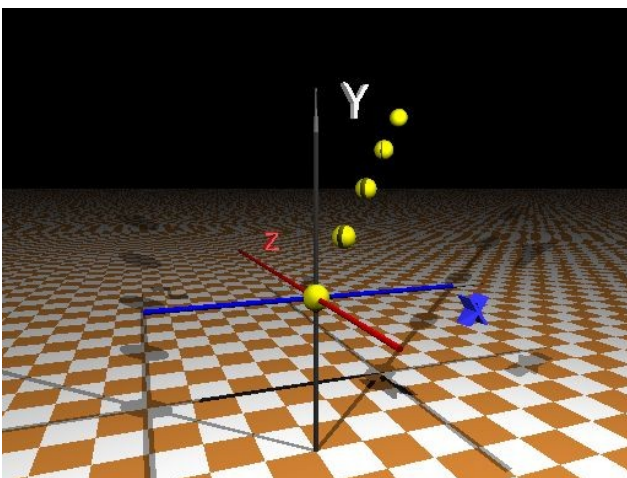
```

5. Aufgabe: a)

Hier werden 5 Kugeln in der xy-Ebene in einer Diagonalen angeordnet. Beim Befehl translate wird sowohl in x- als auch gleichzeitig in y-Richtung um denselben Wert verschoben.



b) Hier werden die Kugeln in einer Raumdiagonalen eines Würfels angeordnet. Der Befehl `translate` verschiebt in alle drei Richtungen um denselben Wert.



6. Aufgabe:

Mit dem Befehl **translate** können Objekte verschoben werden und zwar in alle drei Richtungen: `translate<xRichtung,yRichtung,zRichtung>`

Mit dem Befehl **rotate** können Objekte um die drei Koordinatenachsen gedreht werden. Angegeben wird der Winkel in Grad.

Mit dem Befehl **scale** können Objekte gestaucht oder gestreckt werden. Achtung: Soll in eine Richtung keine Veränderung stattfinden, so muss der Wert 1 eingegeben werden. Beispiel: `scale<1.5,0.8,1>`:

In x-Richtung wird vergrößert (mal 1.5), in y-Richtung wird verkleinert (mal 0.8) und in z-Richtung wird nichts verändert an der Skalierung (mal 1).

as