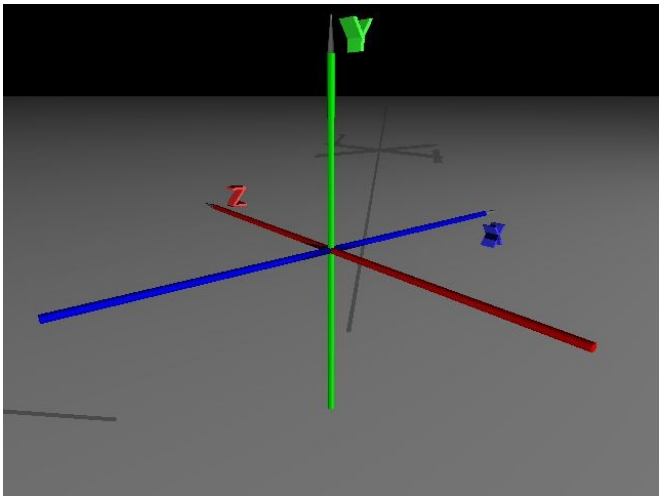
2.Arbeit Informatik Jahrgangsstufe 9 25. Januar 2012

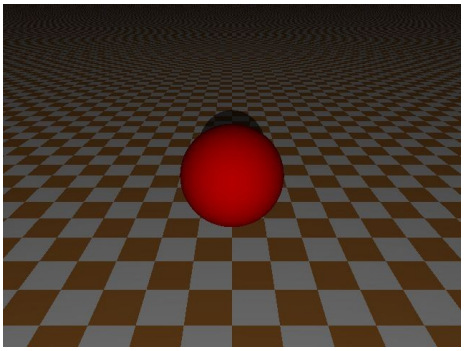
Für die Lichtquellen gilt folgendes:
Zwei Lichtquellen sollen die Szene beleuchten und zwar: von oben schräg links und von oben schräg rechts.

1.Aufgabe: Erläutere den Begriff RayTracing. Stelle das Koordinatensystem dar und gib an, welche Befehle erforderlich sind.
Beschreibe den camera-Befehl und gib ein einfaches Beispiel (Kugel im Koordinatenursprung). Was versteht man unter den include-Dateien, welche gibt es und wozu dienen diese speziell?

Lösung:



Ray-Tracing: Mit PovRay werden fotorealistische Bilder erzeugt. Insbesondere wird der Strahlenverlauf von Lichtquellen berechnet und im Bild realisiert.



```
#include "colors.inc"
#include "textures.inc"
#include "shapes.inc"
//Unser Standpunkt, das Auge des Betrachters
camera{ location<0,3,-6> // hier stehe ich
 look_at<0,0,0> }        // da schaue ich hin
// die Lichtquelle
light_source {<0,5,-15> color White  }
sphere {<0,0,0>,1    pigment {Red}    }
plane {y,-3.0   pigment{ checker color White color Gold}  }
```

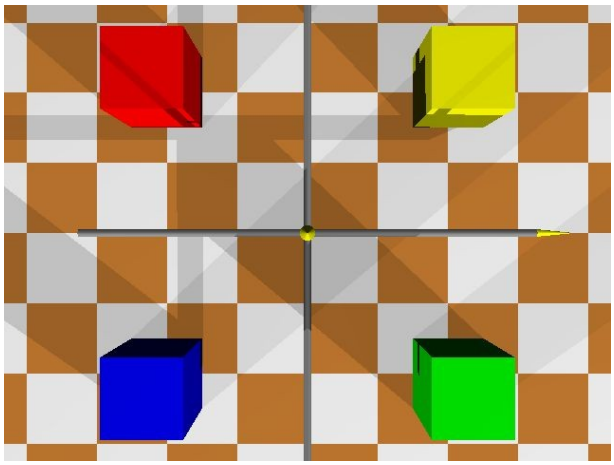
Die Include-Dateien stellen Texturen, Körpern und Farben mit Hilfe von Variablen zur Verfügung (z.B. für die Farbe rot: Red).
Die wichtigsten Include-Dateien sind: colors.inc, textures.inc und shapes.inc.

2.Aufgabe: Schreibe ein Programm, das die Darstellung von 4 unterschiedlich gefärbten Würfeln, angeordnet in den Ecken eines Quadrats, ermöglicht.

Lösung:

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

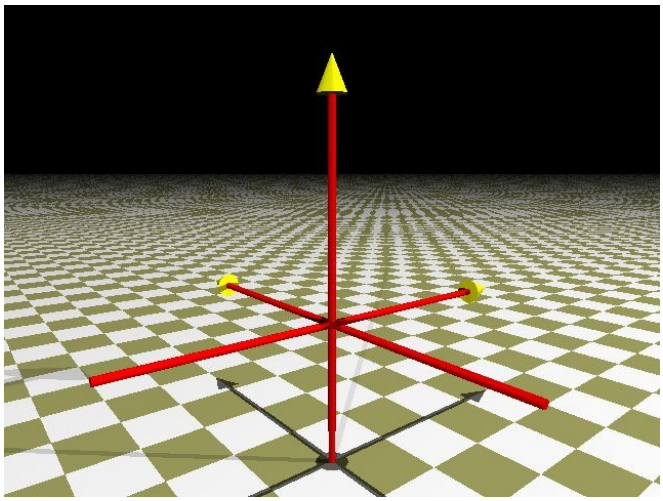
```
#include "textures.inc"
#include "shapes.inc"

camera{
 location<0,6,0>   // hier stehe ich
 look_at<0,0,>      // da schaue ich hin
 }

// die Lichtquelle
 light_source {  <20,6,-18> color White  }

light_source { <-20,2,-18> color White    }

light_source {  <0,40,0> color White  }

#declare Wuerfel=box{<-0.5,-0.5,-0.5><0.5,0.5,0.5> }

object{Wuerfel pigment{Blue} translate<-2,0,-2>}
object{Wuerfel pigment{Red} translate<-2,0,2>}
object{Wuerfel pigment{Green} translate<2,0,-2>}
object{Wuerfel pigment{Yellow} translate<2,0,2>}
```



3. Aufgabe: Stelle das xyz-Koordinatensystem (jeweils von -4 bis 4) mit Hilfe von dünnen Zylindern dar. Dabei soll in die positive Richtung jeweils ein Kegel auf den Zylinder aufgesetzt sein.

Lösung:

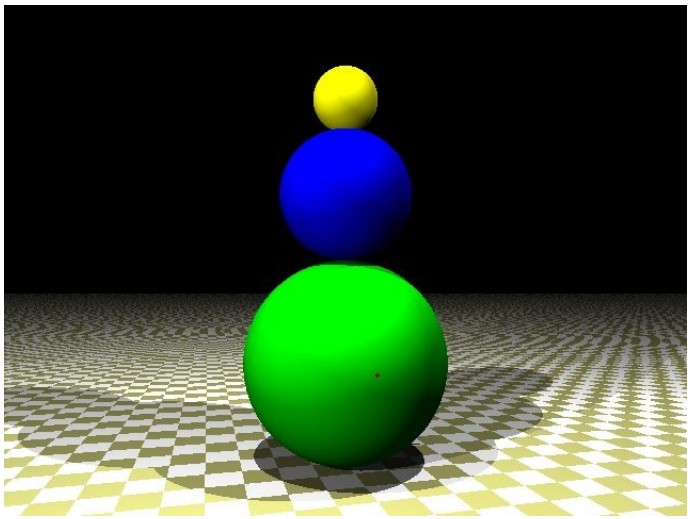
```
camera{ location<-4,2,-5>    look_at<0,1,0>       }

// die Lichtquelle
light_source {  <20,6,-18> color White    }

light_source {  <-20,2,-18> color White   }

light_source {   <0,40,0> color White  }

#declare xAchse=union{cylinder{<-3,0,0><3,0,0>0.05 pigment{Red}}
cone{<3,0,0>0.2<3.5,0,0>0 pigment{Yellow}}}

#declare yAchse=union{cylinder{<0,-3,0><0,3,0>0.05
pigment{Red}}
cone{<0,3,0>0.2<0,3.5,0>0 pigment{Yellow}}}

#declare zAchse=union{cylinder{<0,0,-3><0,0,3>0.05
pigment{Red}}
cone{<0,0,3>0.2<0,0,3.5>0 pigment{Yellow}}}

object{xAchse}
object{yAchse}
object{zAchse}

plane {y,-2  pigment{     checker color White color
Khaki}     }
```



4.Aufgabe: Schreibe ein Programm, das die Szene von 3 übereinander gestellter unterschiedlich gefärbter Kugeln darstellt. Die erste Kugel soll im Mittelpunkt des Koordinatensystems stehen und den Radius 3 haben, die zweite den Radius 2 und die dritte den Radius 1.

Lösung:

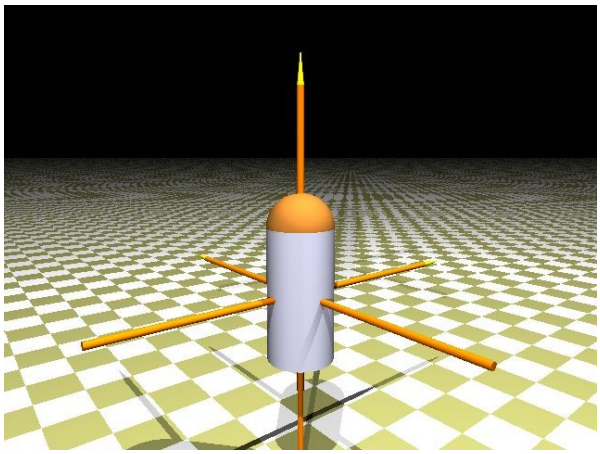
```
camera{
 location<-4,2,-15>   // hier stehe ich
 look_at<0,3,0>       // da schaue ich hin
 }

// die Lichtquelle
light_source {  <5,13,-5> color White   }

light_source {  <-5,13,-5> color White  }

light_source {  <-10,3,-10> color White  }

sphere{<0,0,0>,3 pigment{Green}}
sphere{<0,5,0>,2 pigment{Blue}}
sphere{<0,8,0>,1 pigment{Yellow}}
```



5. Aufgabe: Erzeuge mit Hilfe von PovRay einen Zylinder, auf dem eine Halbkugel unterschiedlicher Farbe aufgesetzt ist.

Lösung:

```
camera{
 location<-4,2,-5>   // hier stehe ich
 look_at<0,1,0>      // da schaue ich hin
 }

// die Lichtquelle
light_source {   <5,13,-5> color White   }

light_source {   <-5,13,-5> color White   }

light_source {    <-10,3,-10> color White  }

cylinder{<0,-1,0><0,1,0> 0.5 pigment{Quartz}}
sphere{<0,1,0>,0.5 pigment{Gold}}
```

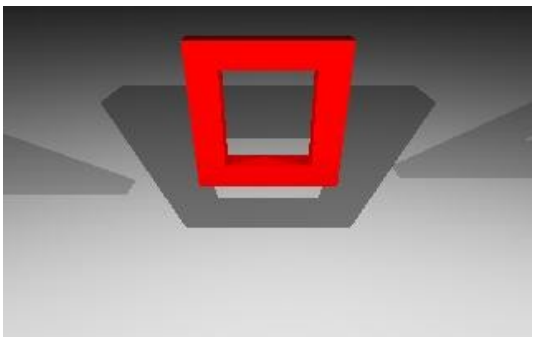


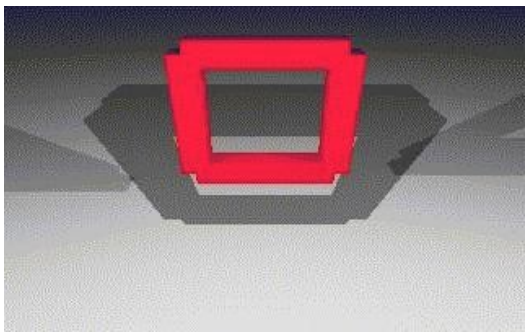
7. Aufgabe: Mit Hilfe des Befehls text soll der Text ASGSG-Marl auf einen Quader ähnlich dem Bild geschrieben werden. Der Schriftzug und der Quader sollen dabei ein Objekt darstellen. Erläutere deine Überlegungen und schreibe ein Programm dazu.

Lösung:

```
camera{
 location<0,4,-12>   // hier stehe ich
 look_at<0,0,0>      // da schaue ich hin
 }

// die Lichtquelle
light_source {  <-20,6,-50> color red 1 green 1 blue 1  }
light_source {  <20,4,-50> color red 1 green 1 blue 1  }

//Der Text
#declare text1=text{ttf "verdana.ttf","ASGSG - Marl",2,0
   texture{  pigment{
     color rgb<0.4,0.4,0.4>}
         finish{ambient 0.1
                diffuse 0.85}
                       }
                }

object{text1 scale<1.2,1.7,0.4> translate<-4,0.5,-3>}

#declare Kasten=box{<-1,-1,-1><1,1,1> texture{
                  pigment{color rgb<1,0.65,0>}
                  finish{ambient 0.15
                         diffuse 0.85} }
}


object{Kasten translate<0.1,0.5,0> scale<2.4,0.6,0.6>*2.4}
```

8.Aufgabe: Definiert wird ein Balken wird mit Hilfe des box-Befehls ***(#declare block = box{<-1,-0.2,-0.2><1,0.2,0.2> pigment{Red}}.*** Mit Hilfe dieser Variablen block sollen die beiden Vierecke erzeugt werden. Wie
lauten diese Befehle? Diskutiere, woran die unterschiedliche Darstellung der Rahmen liegt.



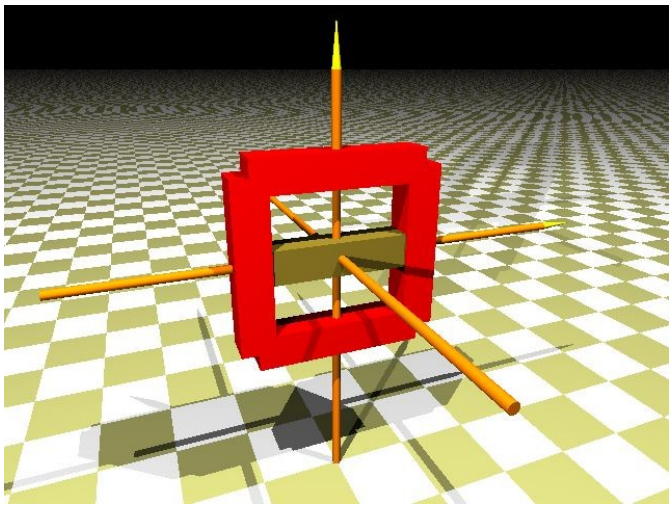


Anhang:
Der Text-Befehl:
#declare text1 = text{ttf font, zeile, text_dicke, text_abstandx*x+text_abstandy*y
texture{text_textur}}

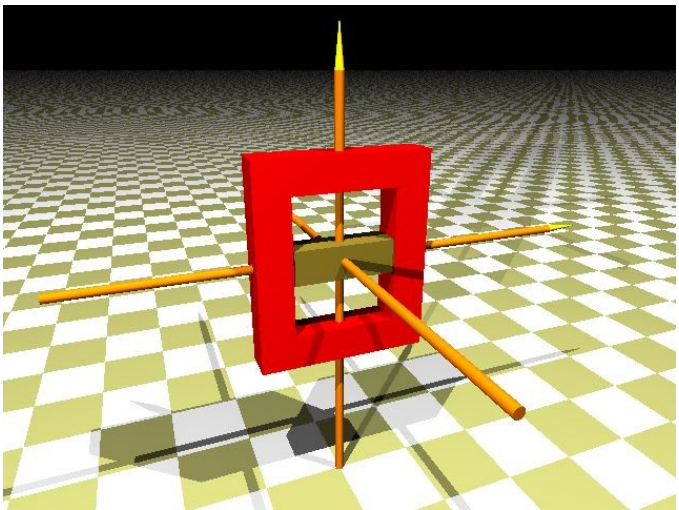
```
#declare text1=text{ttf "verdana.ttf","M A R L",3,0
   texture{  pigment{ color rgb<1,0.65,0>} finish{ambient 0.1  diffuse 0.85} } }
```

Lösung:

```
camera{
 location<-2,2,-5>   // hier stehe ich
 look_at<0,0,0>      // da schaue ich hin
 }
// die Lichtquelle
light_source {   <5,13,-5> color White   }

light_source {  <-5,13,-5> color White   }

light_source {   <-10,3,-10> color White  }

#declare block = box{<-1,-0.2,-0.2><1,0.2,0.2>
pigment{Red}}

object{block pigment{Bronze} scale<1,1,0.9>}

object{block rotate<0,0,90> translate<1,0,0>}
object{block rotate<0,0,90> translate<-1,0,0>}
object{block  translate<0,1,0>}
object{block  translate<0,-1,0>}
```



```
camera{
 location<-2,2,-5>   // hier stehe ich
 look_at<0,0,0>      // da schaue ich hin
 }

// die Lichtquelle
light_source {  <5,13,-5> color White   }

light_source {  <-5,13,-5> color White   }
```



```
light_source {   <-10,3,-10> color White  }

#declare block = box{<-1,-0.2,-0.2><1,0.2,0.2> pigment{Red}}

object{block pigment{Bronze} scale<1,1,0.9>}

object{block rotate<0,0,90> translate<0.8,0,0>}
object{block rotate<0,0,90> translate<-0.8,0,0>}
object{block  translate<0,1,0>}
object{block  translate<0,-1,0>}
```

Wenn die Verschiebung in x-Richtung verkleinert wird, erhält man das Viereck ohne Zacken an den Ecken.