

```
GLfloat KL=100.0f;
```

```
...
```

```
glBegin(GL_POLYGON);           //Vorderseite
glColor4f(1.0f, 0.0f, 0.0f, 1.0f); //ROT
glVertex3f(-KL/2.0f, -KL/2.0f, +KL/2.0f);
glColor4f(1.0f, 1.0f, 0.0f, 1.0f); //GELB
glVertex3f(+KL/2.0f,-KL/2.0f,+KL/2.0f);
glColor4f(1.0f, 1.0f, 1.0f, 1.0f); //WEISS
glVertex3f(+KL/2.0f, +KL/2.0f, +KL/2.0f);
glColor4f(1.0f, 0.0f, 1.0f, 1.0f); //MAGENTA
glVertex3f(-KL/2.0f, +KL/2.0f, +KL/2.0f);
glEnd();
```



Abb. 9.2: Definition eines Primitivs

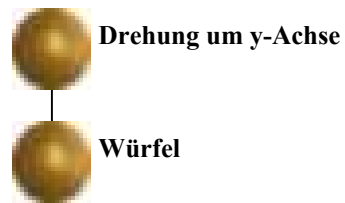
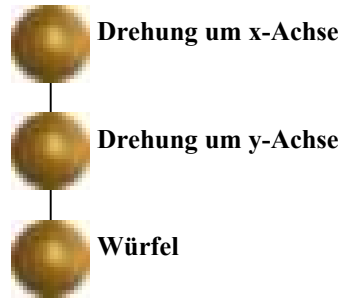


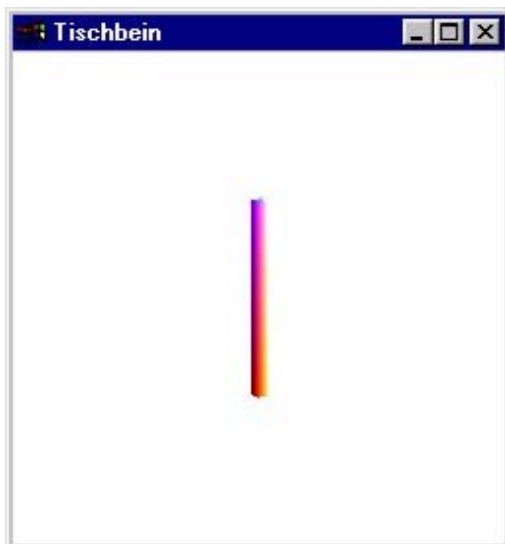
Abb. 9.3: Drehung um die y-Achse

```
glRotatef (30.0f, 0.0f, 1.0f, 0.0f);
Wuerfel ();
```



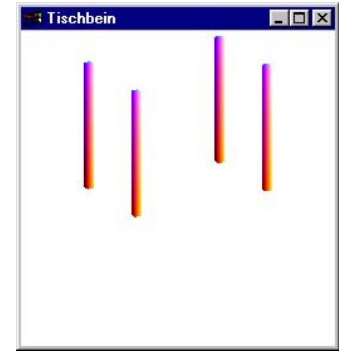
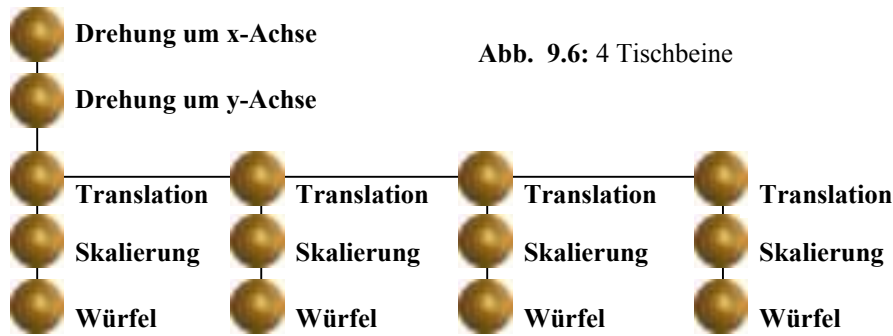
**Abb. 9.4:** zusätzliche Drehung um die x-Achse um 20 Grad

```
glRotatef (20.0f, 1.0f, 0.0f, 0.0f);
glRotatef (30.0f, 0.0f, 1.0f, 0.0f);
Wuerfel ();
```



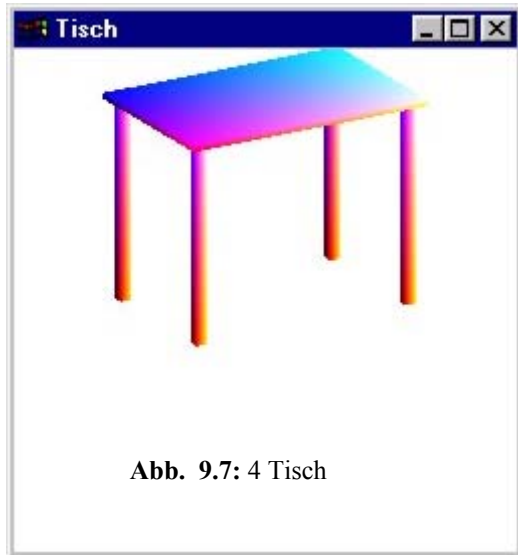
**Abb. 9.5:** Skalierter und gedrehter Quader

```
glRotatef (20.0f, 1.0f, 0.0f, 0.0f);
glRotatef (30.0f, 0.0f, 1.0f, 0.0f);
glScalef (0.04f, 0.7f, 0.04f);
Wuerfel ();
```



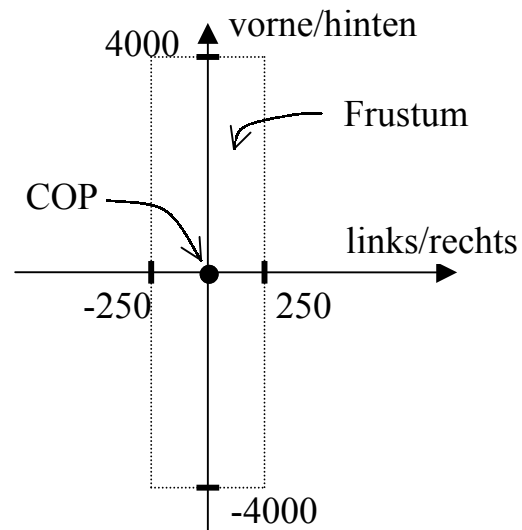
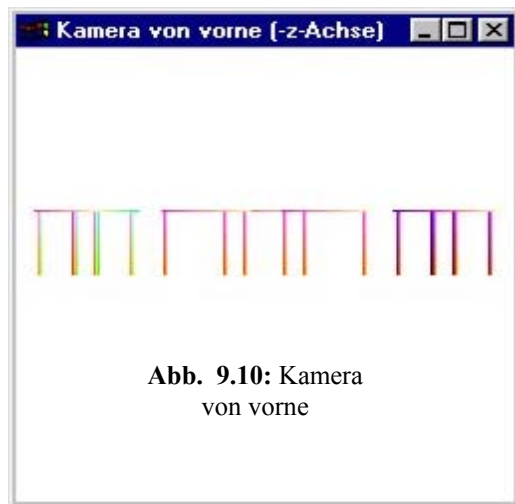
```
glRotatef(20.0f, 1.0f, 0.0f, 0.0f);
glRotatef(30.0f, 0.0f, 1.0f, 0.0f);
```

```
glPushMatrix();           // Tischbein links, unten
  glTranslatef(-40.0f, 35.0f, 25.0f);
  glScalef(0.04f, 0.7f, 0.04f);
  Wuerfel();
glPopMatrix();
glPushMatrix();           // Tischbein rechts, unten
  glTranslatef(+40.0f, 35.0f, 25.0f);
  glScalef(0.04f, 0.7f, 0.04f);
  Wuerfel();
glPopMatrix();
glPushMatrix();           // Tischbein rechts, oben
  glTranslatef(+40.0f, 35.0f, -25.0f);
  glScalef(0.04f, 0.7f, 0.04f);
  Wuerfel();
glPopMatrix();
glPushMatrix();           // Tischbein rechts, oben
  glTranslatef(-40.0f, 35.0f, -25.0f);
  glScalef(0.04f, 0.7f, 0.04f);
  Wuerfel();
glPopMatrix();
```



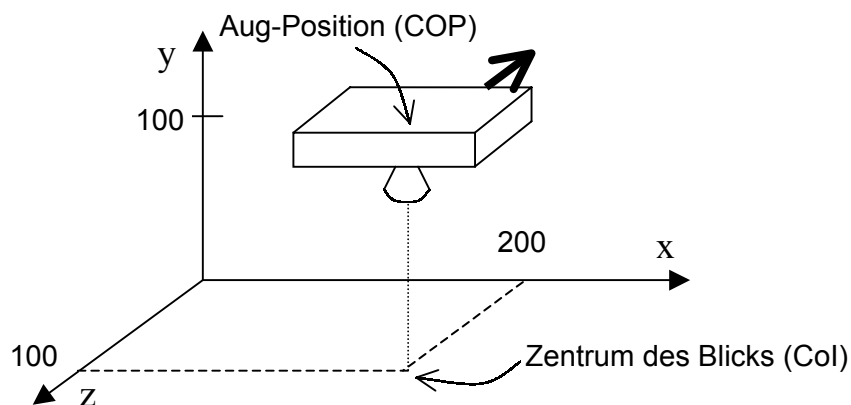
```
// Tischplatte
glPushMatrix(); // Tischplatte
glTranslatef(0.0f, 71.0f, 0.0f);
glScalef(0.9f, 0.02f, 0.6f);
Wuerfel();
glPopMatrix();

// Gesamter Tisch
glRotatef(20.0f, 1.0f, 0.0f, 0.0f);
glRotatef(30.0f, 0.0f, 1.0f, 0.0f);
Tisch();
```



```
// orthographische Projektion
glOrtho(-KL*2.5, +KL*2.5, -KL*2.5, +KL*2.5, -40*KL, +40*KL);
// Abstand der linken, rechten, unteren, oberen, vorderen und hinteren
// Clipping-Ebene vom COP; hier: KL=100
```

```
//Blick von oben senkrecht nach unten
gluLookAt(200.0, 100.0, 100.0, 200.0, 0.0, 100.0, 0.0, 0.0, -1.0);
// Aug-Position (x,y,z), Zentrum des Blicks (x,y,z) und Aufwaertsvektor (x,y,z)
```



```
// Gesamte Szene drehen und verkleinern
glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glScalef(0.25f, 0.25f, 0.25f);
```

```
glPushMatrix(); // Tisch links unten
glTranslatef(-60.0f, 0.0f, 80.0f);
Tisch();
glPopMatrix();
```

```
glPushMatrix(); // Tisch rechts unten
glTranslatef(+60.0f, 0.0f, 80.0f);
Tisch();
glPopMatrix();
```

```
glPushMatrix(); // Tisch Mitte links
glTranslatef(-180.0f, 0.0f, 0.0f);
glRotatef(-45.0f, 0.0f, 1.0f, 0.0f);
Tisch();
glPopMatrix();
```

```
glPushMatrix(); // Tisch Mitte rechts
glTranslatef(+180.0f, 0.0f, 0.0f);
glRotatef(+45.0f, 0.0f, 1.0f, 0.0f);
Tisch();
glPopMatrix();
```

```
glPushMatrix();// Tisch oben
glTranslatef(0.0f, 0.0f, -80.0f);
Tisch();
glPopMatrix();
```

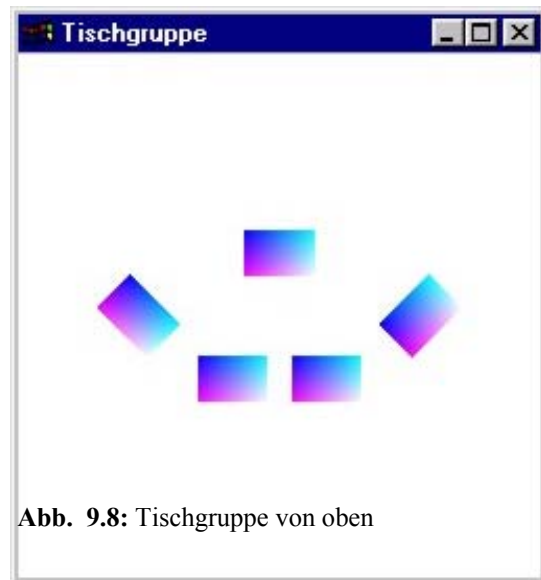


Abb. 9.8: Tischgruppe von oben

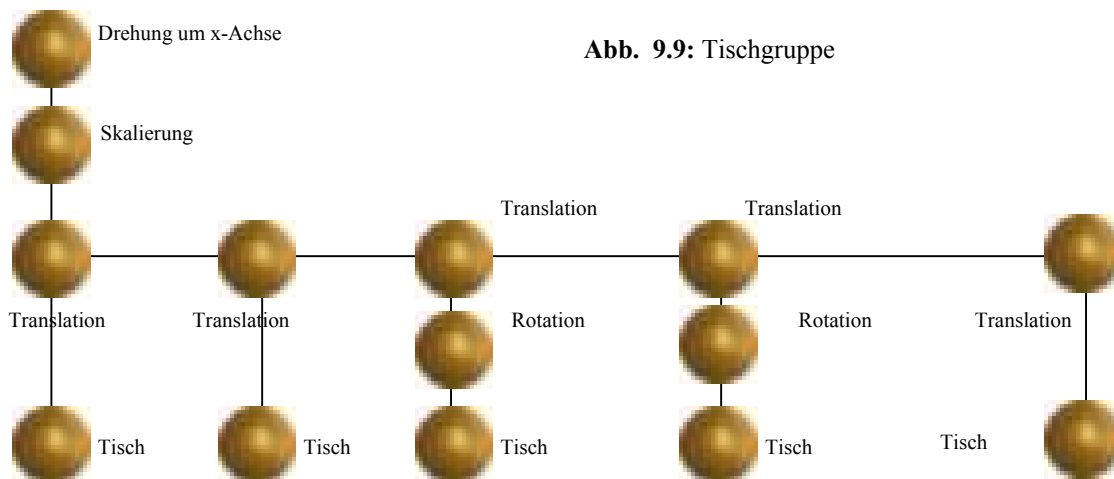
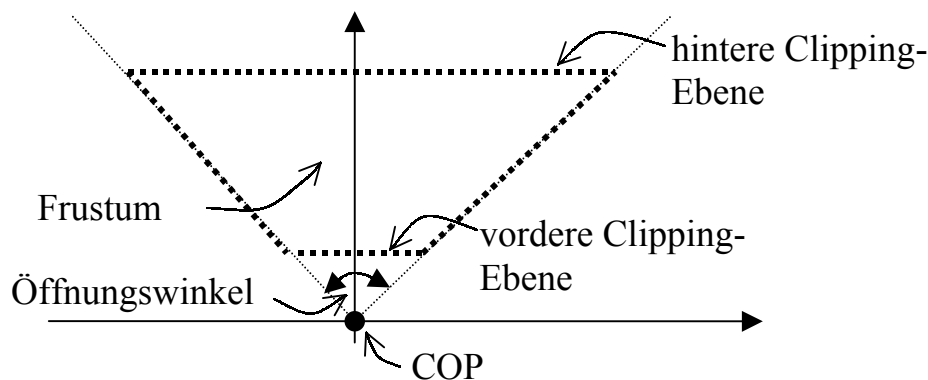


Abb. 9.9: Tischgruppe

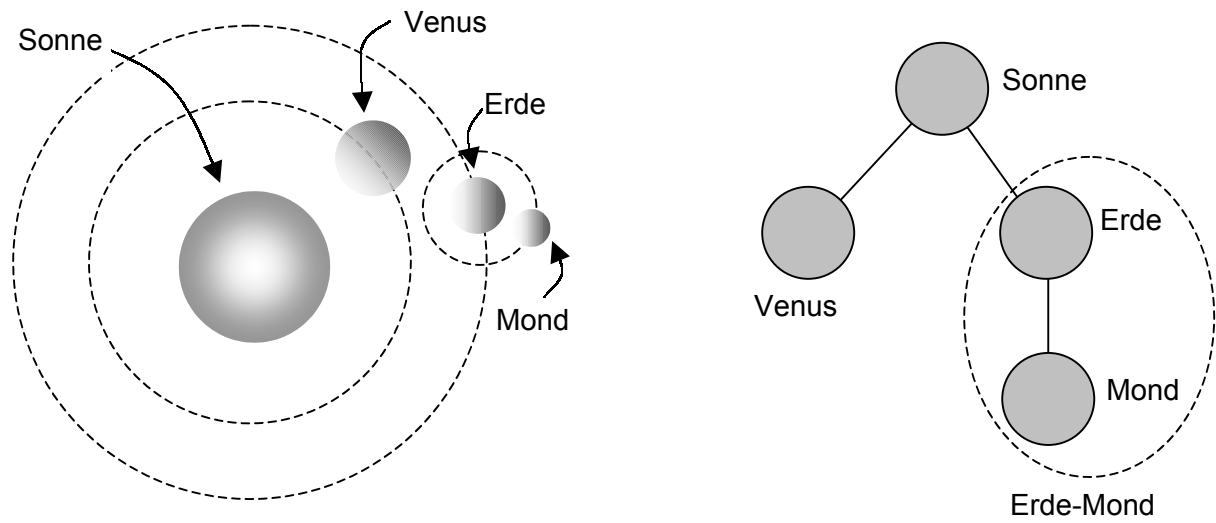


// Perspektivische Projektion  
 gluPerspective(90.0, 1.0, 1.5\*KL, 40\*KL); // Öffnungswinkel, Seiten-  
 // verhältnis, Abstand der vorderen und hinteren Clipping-Ebene vom COP

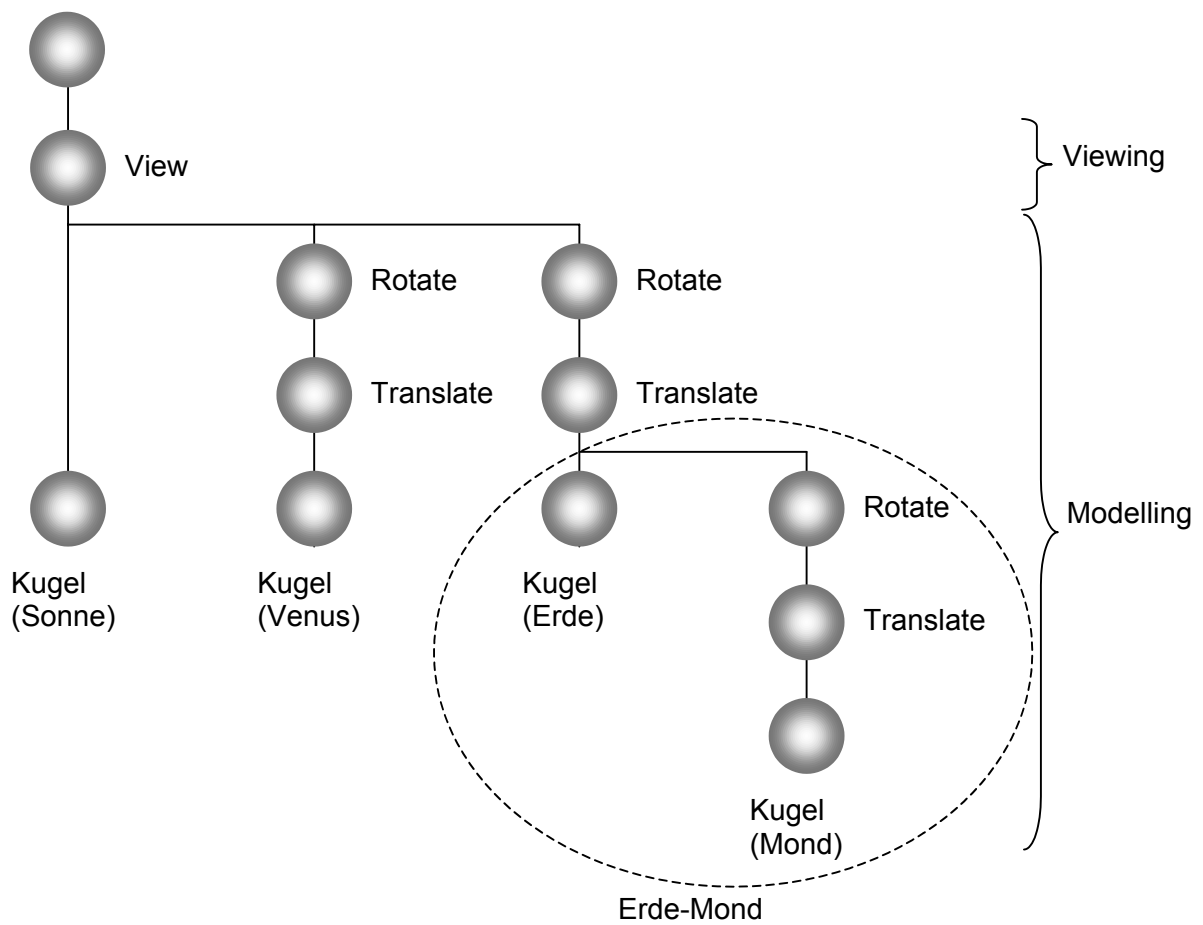


**Hierarchisches Modellieren mit OpenGL (Beispiel: Planetensystem)**

1.) Szene entwerfen(links) mit Abhängigkeiten (rechts)  
(Vorgehen: Top-Down)



2.) Szenegraph erstellen  
(Vorgehen: Bottom-Up)



3.) Umsetzen in OpenGL-Programm (hier nur entsprechende Teile; nicht oo)

```

:
GLfloat winkel=0.0; //steuert die Rotation der Planeten
:
void Sonne(void) {
    glPushMatrix();
        glColor3f(1.0f, 0.8f, 0.0f);
        glutSolidSphere(0.25, 40, 40);           // vorgegebene Kugel
    glPopMatrix();
}

void Venus(void) {
    glPushMatrix();
        glColor3f(0.5f, 0.5f, 0.8f);
        glRotatef(2.0*winkel, 0.0f, 1.0f, 0.0f);
        glTranslatef(-0.9f, 0.0f, 0.0f);
        glutSolidSphere(0.09, 20, 20);         // vorgegebene Kugel
    glPopMatrix();
}

void Erde(void) {
    glPushMatrix();
        glColor3f(0.0f, 0.5f, 1.0f);
        glutSolidSphere(0.1, 20, 20);          // vorgegebene Kugel
    glPopMatrix();
}

void Mond(void) {
    glPushMatrix();
        glColor3f(1.0f, 1.0f, 0.0f);
        glRotatef(4.0*winkel, 0.0f, 1.0f, 0.0f);
        glTranslatef(0.2f, 0.0f, 0.0f);
        glutSolidSphere(0.02, 20, 20);         // vorgegebene Kugel
    glPopMatrix();
}

void ErdeMond(void) {
    glPushMatrix();
        glRotatef(winkel, 0.0f, 1.0f, 0.0f);
        glTranslatef(1.3f, 0.0f, 0.0f);
        Erde();
        Mond();
    glPopMatrix();
}
:
void RenderScene(void) {
:
    glLoadIdentity (); // Modellierungs-/Viewing-Matrix = Einheits-Matrix
:
    Sonne();
    Venus();
    ErdeMond();
:
}
:
void Animate () {
    winkel = winkel + 1.0; // bei schnellen Rechnern bzw. Graphikkarten
    if (winkel>=360.0)     // Schrittweite für "winkel" kleiner waehlen
        winkel = 0.0;
}
:

```