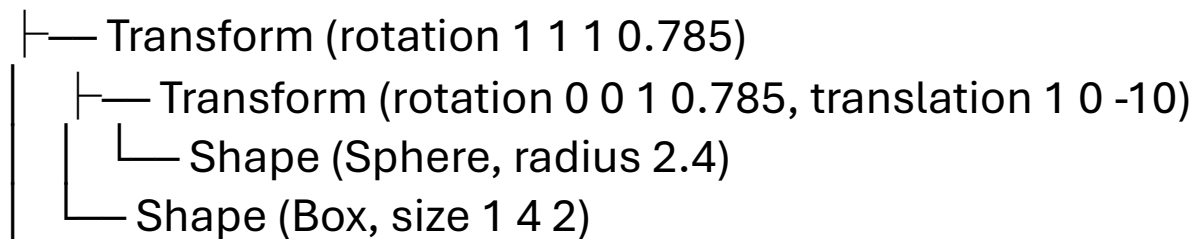


VRML-Miniprojekt:

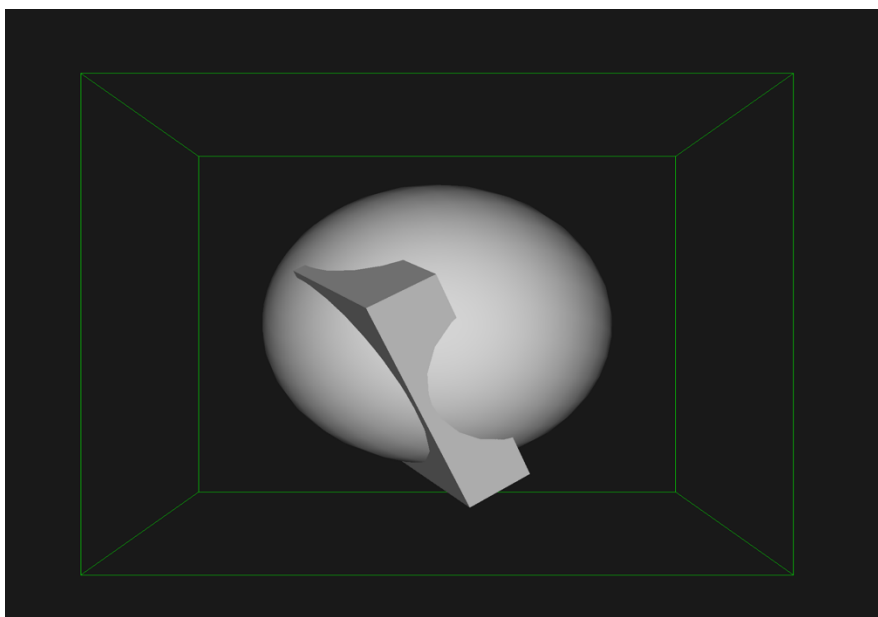
Aufgabe 1:

b. Der Szenengraph, der nur Transform- und Shape-Nodes enthält, sieht folgendermaßen aus:

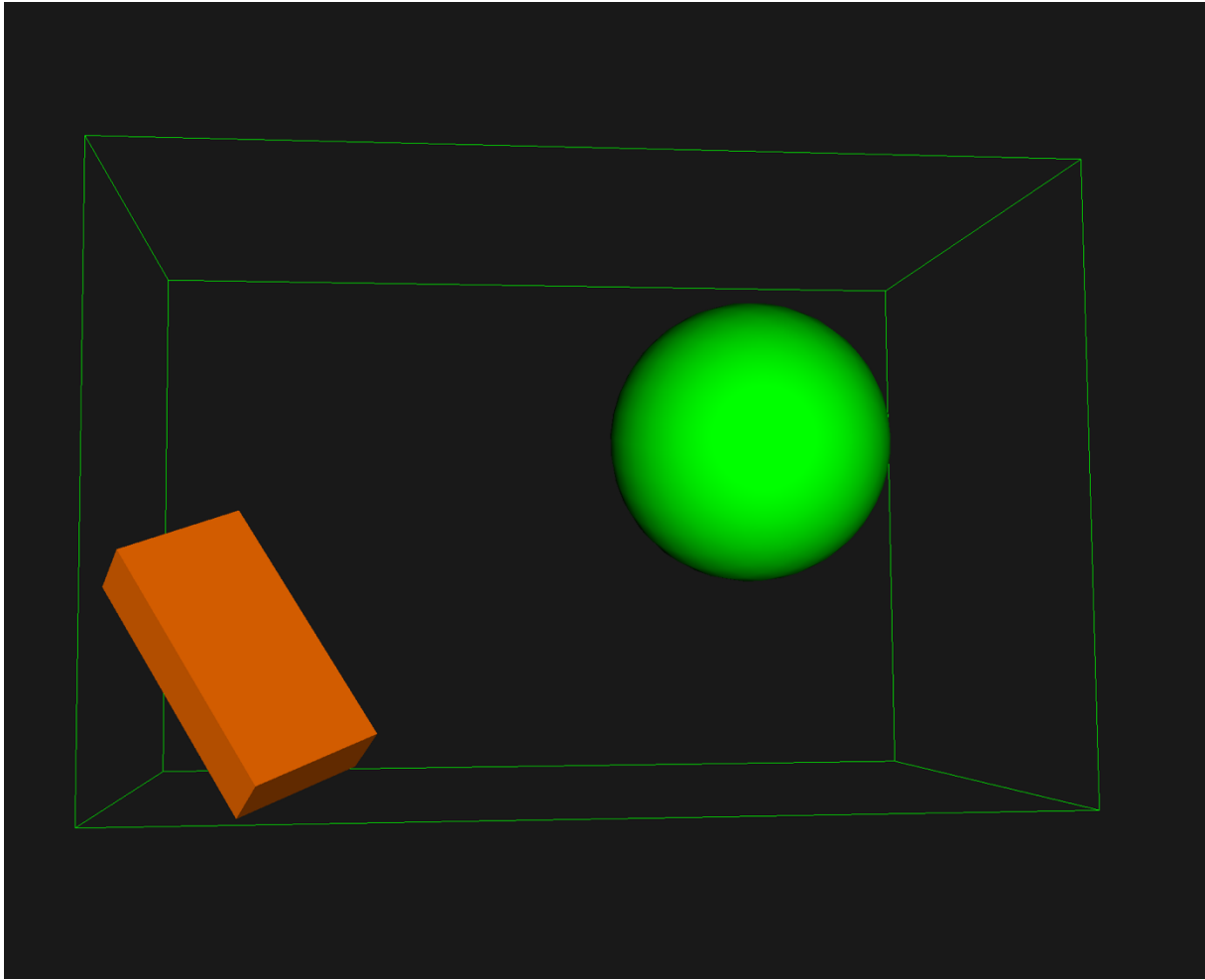
Root (outer Transform)



```
c. Transform {  
  rotation 0 0 1 0.785  
  translation 1 0 -10  
  children [...]  
}
```



d.



e.

Root (outer Transform)

- ├─ Transform (rotation 1 1 1 0.785)
 - ├─ Transform (rotation 0 0 1 0.785, translation 1 0 -10)
 - ├─ Shape (Sphere)
 - ├─ Appearance
 - ├─ Material (diffuseColor 0 1 0) # Grün für die Kugel
 - ├─ Shape (Box)
 - ├─ Appearance
 - ├─ Material (diffuseColor 1 0.5 0) # Orange für den Quader

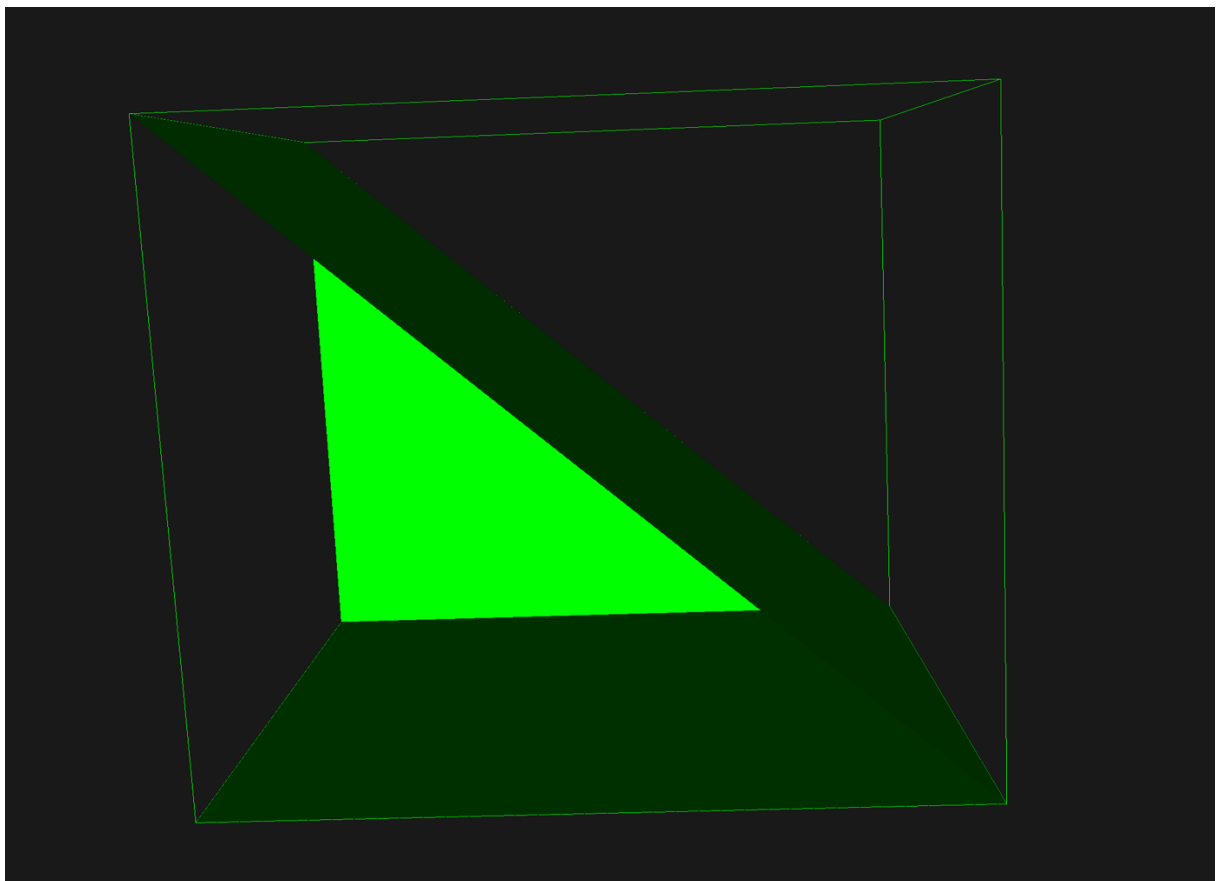
Aufgabe 2:

a. Das Prisma hat die Form eines dreieckigen Quaders, wobei eine Seite offen oder unvollständig ist. Es besteht aus drei vollständigen Flächen:

- Eine dreieckige Grundfläche (A-C-B).
- Zwei rechteckige Seitenflächen (A-B-E-D und B-C-F-E).

b. Aus der Standardblickrichtung des Browsers das Prisma unsichtbar, weil der VRML-Code nur teilweise korrekt ist und bestimmte Koordinaten für das Prisma fehlen. Der Code definiert nur einige der Kanten und Flächen, wodurch kein geschlossenes Volumen entsteht.

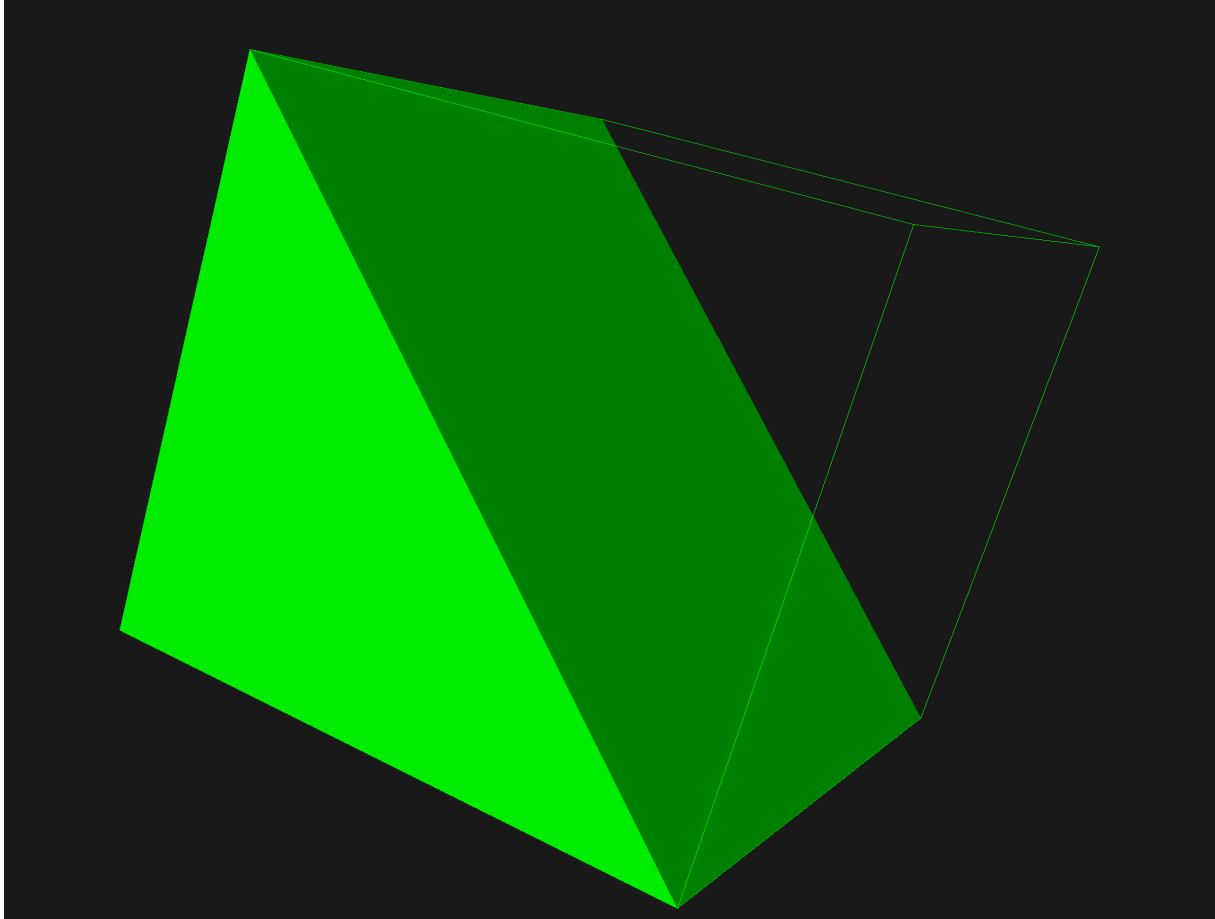
c.



Das Setzen von solid FALSE ermöglicht es, die Rückseiten der Flächen zu sehen. Dadurch kann ein Modell vollständig sichtbar gemacht werden, was nützlich ist, wenn man das Objekt von innen oder aus verschiedenen Perspektiven betrachten möchte, ohne dass Flächen ausgeblendet werden.

d. Mit diesen Ergänzungen ist das Prisma vollständig definiert, und alle sechs Flächen (vier Seitenflächen und zwei Grundflächen) sind in coordIndex angegeben.

coordIndex [0 2 1 -1 0 1 4 3 -1 1 2 5 4 -1 **2 0 3 5 -1** CADF (ergänzt) **3 4 5 -1** DEF (ergänzt)]



e.

1. Berechnung des Mittelpunkts des Vierecks $B C F E$ BCFE:

Das Viereck $B C F E$ BCFE wird durch die Punkte B , C , F , und E gebildet, die im Code die Koordinaten haben:

$B = (1, 0, 0)$ $B = (1, 0, 0)$ $C = (0, 1, 0)$ $C = (0, 1, 0)$ $F = (0, 1, 1)$ $F = (0, 1, 1)$ $E = (1, 0, 1)$ $E = (1, 0, 1)$

Um den Mittelpunkt dieses Vierecks zu berechnen, nehmen wir die Durchschnittskordinaten dieser vier Punkte.

Die Koordinaten des Mittelpunkts sind:

Mittelpunkt = $(\frac{1+0+0+1}{4}, \frac{0+1+1+0}{4}, \frac{0+0+1+1}{4}) = (0.5, 0.5, 0.5)$

Mittelpunkt = $(\frac{4 \cdot 1 + 0 + 0 + 1}{4}, \frac{4 \cdot 0 + 1 + 1 + 0}{4}, \frac{4 \cdot 0 + 0 + 1 + 1}{4}) = (0.5, 0.5, 0.5)$

Dies ist der Punkt, an dem die Spitze des Kegels den Prisma berühren soll.

2. Platzierung und Drehung des Kegels Translation (translation 0.5 0.5 0.5):

Der Kegel wird mit einem Transform-Knoten verschoben (übersetzt), damit seine Spitze genau auf den berechneten Mittelpunkt des Vierecks $B C F E$ BCFE zeigt.

Die translation 0.5 0.5 0.5 verschiebt den Kegel so, dass seine Spitze auf den Punkt $(0.5, 0.5, 0.5)$ zeigt – den Mittelpunkt des Vierecks.

Rotation (rotation 1 0 0 1.5708):

Ein Kegel im VRML-Format ist standardmäßig entlang der positiven y -Achse orientiert, mit der Spitze unten und der Basis oben. Um die Basis des Kegels parallel zum Viereck $B C F E$ zu machen, muss der Kegel um die x -Achse gedreht werden.

Die rotation 1 0 0 1.5708 rotiert den Kegel um 90 Grad (1.5708 Bogenmaß) entlang der x -Achse. Dadurch liegt die Basis des Kegels jetzt parallel zur xz -Ebene, genau wie das Viereck $B C F E$.

```
#VRML V2.0 utf8

DEF Prisma Shape {
  geometry IndexedFaceSet {
    coord Coordinate {
      point [
        0 0 0, # Punkt A
        1 0 0, # Punkt B
        0 1 0, # Punkt C
        0 0 1, # Punkt D
        1 0 1, # Punkt E
        0 1 1 # Punkt F
      ]
    }
    coordIndex [
      0 2 1 -1 # Dreieck ACB
      0 1 4 3 -1 # Viereck ABED
      1 2 5 4 -1 # Viereck BCFD
      2 0 3 5 -1 # Viereck CADF (ergänzt)
      3 4 5 -1 # Dreieck DEF (ergänzt)
    ]
  }
  solid FALSE
  appearance Appearance {
    material Material {
      diffuseColor 0 1 0
    }
  }
}

Transform {
  translation 0.5 0.5 0.5
  rotation 1 0 0 1.5708 # Rotiert den Kegel entlang der x-Achse um 90 Grad
  children [
    Shape {
      geometry Cone {
        bottomRadius 0.5
        height 1
      }
      appearance Appearance {
        material Material {
          diffuseColor 1 0 0
        }
      }
    }
  ]
}
```

