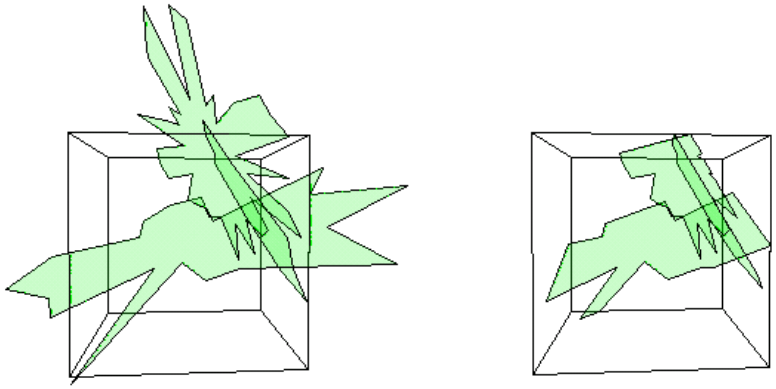
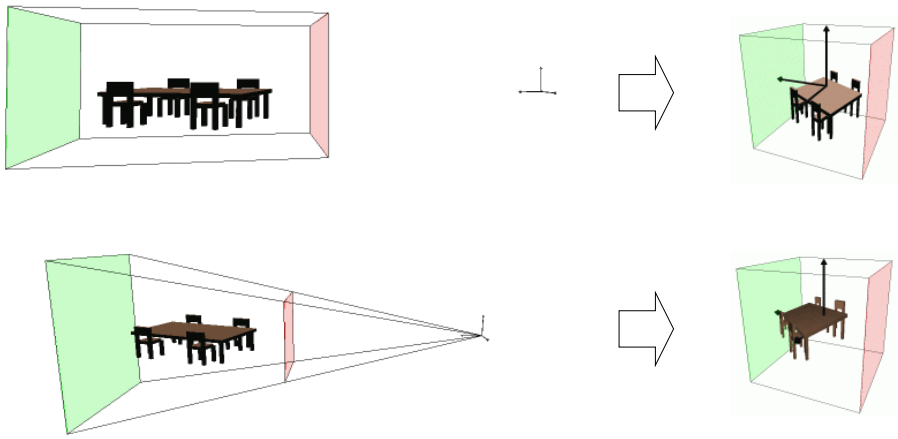
Aufgabe 2.1



Implementieren Sie den Clipping-Algorithmus von Sutherland-Hodgman. Den Algorithmus finden Sie auf der Rückseite des Blattes. Erzeugen Sie mindestens drei unterschiedliche Polygone und stellen Sie diese dar. Beschneiden Sie anschließend die Polygone durch einen Würfel. Stellen Sie die beschnittenen Polygone ebenfalls dar.

Aufgabe 2.2



Schreiben Sie ein Programm, das eine beliebige Szene je nach gewählter Projektionsart in einen Einheitswürfel transformiert. Visualisieren Sie zunächst das sichtbare Volumen. Anschließend plazieren Sie mehrere Objekte in dem sichtbaren Volumen. Transformieren Sie zum Schluß die Szene, so daß aus dem Volumen ein Einheitswürfel wird.

Clipping-Algorithmus von Sutherland-Hodgman.

```
Vector
intersection(Point p0,Point p1,Plane &clipPlane)
{
  Vector v=p1-p0;
  float t=-(clipPlane*p0)/(clipPlane*v);
  return p0+v*t;
}

bool
inside(Point &p,Plane &clipPlane)
{
  return clipPlane.distance(p)>0;
}

Polygon
clipPolygon(Polygon polygon,vector<Plane> &clipPlanes)
{
  Polygon clippedPolygon;
  Point p0,p1,pI;
  bool p0flag, p1flag;
  uint i,j;

  for (j=0;j<clipPlanes.size();j++) {
    p0=polygon[0];
    p0flag=inside(p0,clipPlanes[j]);
    for (i=1;i<=polygon.size();i++) {
      p1=polygon[i];
      p1flag=inside(p1,clipPlanes[j]);
      if (p0flag) {
        if (p1flag)
          clippedPolygon.add(p1);
        else {
          pI=intersection(p0,p1,clipPlanes[j]);
          clippedPolygon.add(pI);
        }
      } else if (p1flag) {
        pI=intersection(p0,p1,clipPlanes[j]);
        clippedPolygon.add(pI);
        clippedPolygon.add(p1);
      }
      p0=p1;
      p0flag=p1flag;
    }
    polygon=clippedPolygon;
    clippedPolygon.clear();
  }
  return polygon;
}
```