

01 Einführung

Das nachfolgende Script will die Originaldokumentation von OpenSCAD nicht ersetzen. Du solltest zu jedem Sprachdetail die entspr. Teile auf www.openscad.org konsultieren, um das, was ich hier nicht erwähne, nachzulesen. Da der Sprachumfang am Anfang verwirrend sein kann, kannst du den Ausflug auch auf später verschieben. Ebenso hilfreich wie die Dokumentation selbst ist der Spickzettel. Er enthält zu allen Schlüsselwörtern entspr. Links zur Erklärung.

Die Aufgaben in diesem Script sollen das bisher Gelernte anwenden – greife bitte nicht zu Dingen, die noch nicht erklärt wurden – es ist Absicht, dass zunächst eine umständliche Lösung herauskommt, die später auf elegante Weise verbessert wird. Fehlende Angaben bekommst du aus dem Internet oder dürfen geschätzt werden.

Das Script wird in mehreren Teilen veröffentlicht. Das gibt dir Gelegenheit, die Aufgaben in Ruhe zu lösen. Geplantes Intervall ist ca. 1 Woche. Du kannst deine Lösungen an mich senden und erhältst im Gegenzug meine Musterlösungen. Diese werden dann z.T. für Folgeaufgaben gebraucht.

1. Allgemeines

- Das Programmfenster ist in drei Teile aufgeteilt: Editor, Vorschau und Konsole. Statt dem eingebauten Editor kann man einen beliebigen anderen Editor verwenden und dann das Editorfenster ausblenden. Auch das Konsolenfenster kann ausgeblendet werden.
- Im Editor(-Fenster) werden die Konstruktionsanweisungen in Form von Text unter Verwendung der Schlüsselwörter formuliert. Die Gesamtheit des Textes beschreibt das 3D-Objekt.
- Das Konsolenfenster zeigt Programmausgaben, Fehlermeldungen und numerische Werte zur Komplexität des Objekts an.
- OpenSCAD berechnet zunächst mittels der F5-Taste eine Vorschau des Objektes. Bei Verwendung eines externen Editors kann OpenSCAD automatisch eine Vorschau generieren, sobald der Text im externen Editor gespeichert wird. Will man das Objekt als .stl-Datei exportieren, muss es vorher mittels der F6-Taste gerendert werden.
- Mit gedrückter linken Maustaste wird das Objekt im Vorschaufenster gedreht, eine gedrückte rechte Maustaste verschiebt das Objekt. Das Scrollrad ist für das Zoomen zuständig. Die Bedienung unter Linux / MacOS kann hier abweichen.
- OpenSCAD kennt keine Variablen. Allerdings sind Konstanten verwendbar. Es gibt deswegen auch keine Konstrukte wie x=x+1;
- OpenSCAD ist nicht C, C++, C# oder ähnliches. Die Syntax lehnt sich aber an C an. Insbesondere die Tatsache wo etwas (in einer Zeile) steht ist unwichtig. Einrückungen machen den Text lesbar!
- Einzeilige Kommentare werden mit // eingeleitet. Mehrzeilige mit /* begonnen und mit */ beendet.
- OpenSCAD kennt keine Integer. Alle numerischen Werte sind Fließkommawerte. Obwohl nirgends eine Dimensionsangabe möglich oder nötig ist, sind per Konvention alle Abmessungen in Millimeter. Neben den Zahlen gibt es noch die boolschen Ausdrüche (wie true und false), Zeichenketten (wie "Ich bin eine Zeichenkette.") und Vektoren (wie [10, 15.5, 20]).
- OpenSCAD kennt geometrische Grundobjekte sowohl für den 2D- (circle, square, polygon, text) als auch für den 3D-Raum (sphere, cube, cylinder, polyhedron). 2D-Konstrukte werden, obwohl sie in der dritten Dimension keine Ausdehnung haben, immer mit einem Millimeter in der Vorschau gezeichnet. Beim Rendern bekommt man daraufhin aber u.U. Fehlermeldungen.
- Ein Objekt wird durch Transformation (translate, rotate, scale, resize, mirror, multimatrix, color, offset, hull, minkowski) verändert. Meist sind mehrere Transformationen am gleichen Objekt nötig. Das Resultat ist ebenfalls als Objekt für weitere Manipulationen verfügbar. Es gibt Manipulationen, die auf mehrere Objekte wirken oder mehrere Objekte benötigen.
- Mit boolschen Operationen (intersection, union, difference) bildet man Schnitt-, Vereinigungs- oder Teilmengen. OpenSCAD bildet automatisch immer eine Vereinigungsmenge.



 Mitels Modifizierern kann man in der Entwurfsphase eines Objektes Teile hervorheben, auslassen oder bevorzugen.

Debug Modifizierer: # das Objekt wird rot/transparent eingefärbt

Root Modifizierer:
 ! restlichen Text ignorieren

Disable Modifizierer: * diesen Teil ignorieren

2. Syntax

- Am Ende einer Anweisung steht ein Semikolon: ;
- Eine Anweisung ist z.B. eine Folge von Manipulationen an einem Objekt:

Mehrere Anweisungen werden in geschweifte Klammern gesetzt: {}

3. 3D-Objekte

- Quader: cube([x-Ausdehnung, y-Ausdehnung, z-Ausdehnung], center=true / false); Wenn center=true angegeben wird, ist der Mittelpunkt des Quaders mit dem Koordinatenursprung identisch.
- Kugel: sphere (d=Durchmesser);
- Zylinder: cylinder(d=Durchmesser, h=Höhe, center=true / false);
- Kegel: cylinder(d1=Durchmesser_unten, d2=Durchmesser_oben, h=Höhe, center = true / false);

3.1 Aufgabe 1

Baue eine Treppe mit 5 Stufen, Breite 100mm, Höhe 50mm. Variiere den Parameter center= und erkläre das Verhalten.

3.2 Aufgabe 2

Baue einen Verkehrs-Leitkegel. "Grundsätzlich besteht der Leitkegel aus einer stabilen Fußplatte und einem aufgehenden konisch geformten Körper. Die Fußplatte ist deutlich breiter und schwerer als der aufgehende Körper." (Wikipedia)





4. Verschiebung

• translate([x-Verschiebung, y-Verschiebung, z-Verschiebung]) Eine Verschiebung positioniert das Objekt an einer beliebigen Position im Raum.

4.1 Aufgabe 3

Baue eine Stufenpyramide aus 5 Stufen.



4.2 Aufgabe 4

Deinem Brettspiel ist ein Spielkegel verloren gegangen. Du hast einem anderen Kegel folgende Maße abgenommen:

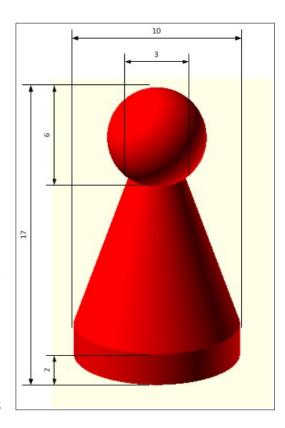
- Basis Ø 10mm,
- Basis-Höhe 2mm,
- ø unter der Kugel 3mm,
- Kugel ø 6mm,
- · Höhe des Spielkegels 17mm.

Konstruiere eine Ersatzspielfigur ähnlich der Zeichnung rechts.

<u>Tipp:</u> OpenSCAD geht sparsam mit den Ressourcen des Rechners um. Das zeigt sich vor allem bei Rundungen. Je mehr Facetten zu berechnen sind, desto länger dauert es, bis die Vorschau oder das Rendering erscheint. Verantwortlich ist die Konstante **\$fn**.

Du kannst die Anzahl der Facetten bei jedem runden Objekt einzeln z.B. mittels sphere(d=10, \$fn=200); oder global für das Gesamtobjekt beeinflussen.

Setze an den Anfang des Textes die Anweisung **\$fn=50**; und variiere den Wert, bis Du mit dem Ergebnis zufrieden bist. Erzeuge dann mit der F6-Taste das Rendering und <u>warte solange</u> bis das Ergebnis erscheint.



Beachte, dass bei 50 Facetten der Slicer auch 50 Fahrbefehle erzeugt, wenn das Objekt gedruckt werden soll. 50 Fahrbefehle enthalten je 50 Beschleunigungs- und Abbremsvorgänge. Verwende ein vernünftiges Verhältnis zwischen der Anzahl der Facetten und dem Durchmesser der Rundung.

5. Rotation

```
rotate([x-Winkel, y-Winkel, z-Winkel]);
rotiert um die Achse(n)
rotate([a=Winkel, v=[x, y, z]);
rotiert um den Vektor
```

5.1 Aufgabe 5

Stelle die Stufenpyramide aus Aufgabe 3 auf eine Ecke der untersten Stufe.

