3DPrinting - Digitale Konstruktion

OpenSCAD Workshop

Mag. Cornelia Haslinger (Bsc.)

Universität Salzburg, Fachbereich Mathematik, AG Technische Mathematik

UNIVERSITÄT SALZBURG

Übersicht

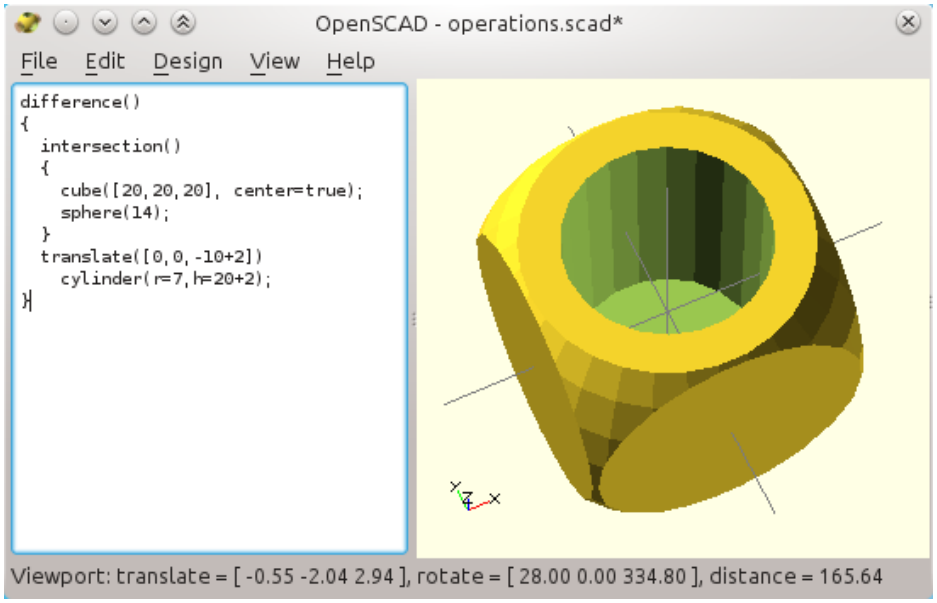
UNIVERSITÄT SALZBURG

OpenSCAD

OpenScad

- nicht-visuelles, programmierorientiertes **solid-modeling tool**
- entwickelt von Marius Kintel (2015)
- erzeugt 3D Volumens-Modelle **solid 3D CAD models**
- Export von STL und OFF Dateien
- Fokus: CAD-Aspekte

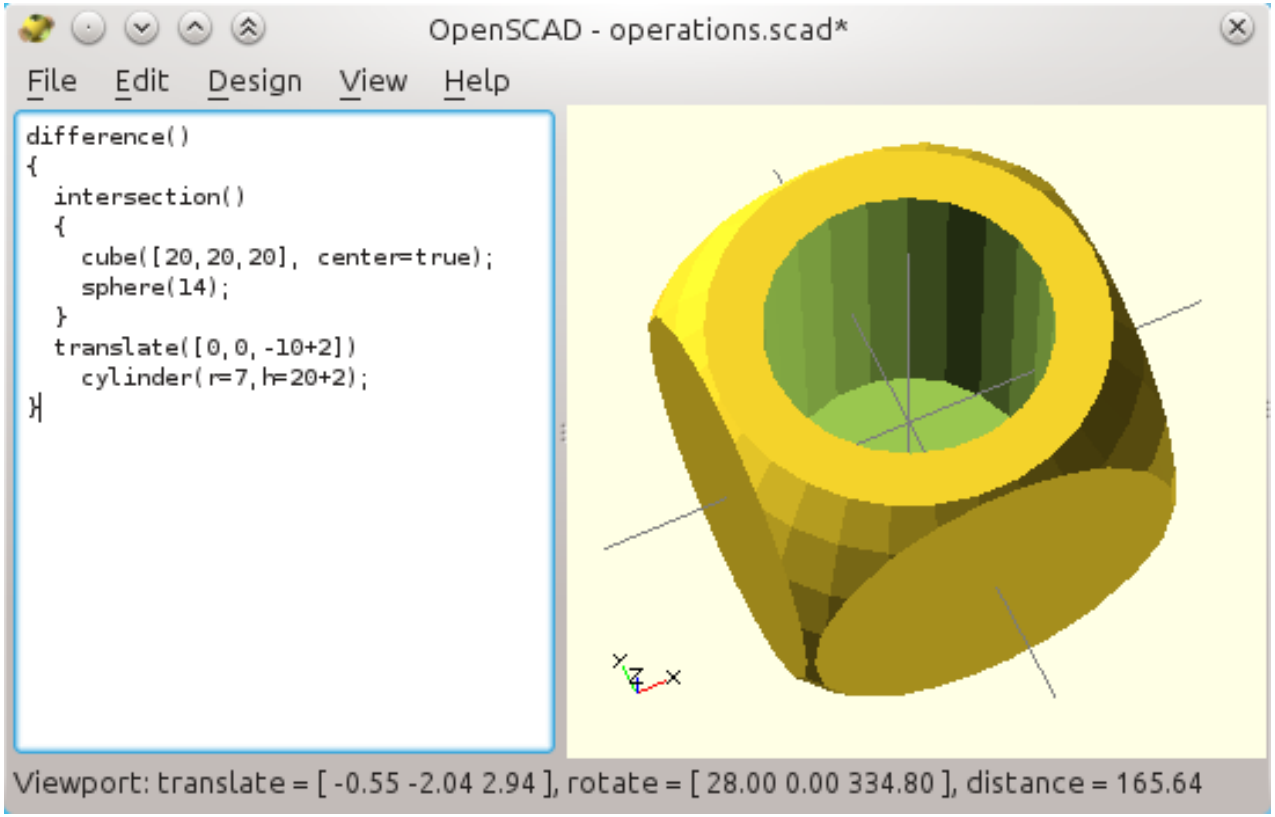
OpenSCAD



- 3D-Compiler:
  - liest ein `script file` das das Objekt beschreibt
  - erzeugt daraus ein 3D-Objekt
- Vorteil:
  - volle Kontrolle über den Konstruktionsprozess
  - jederzeit Änderung der Parameter möglich

UNIVERSITÄT SALZBURG

Main modelling techniques



- CSG
- Extrusion von 2D-Outlines

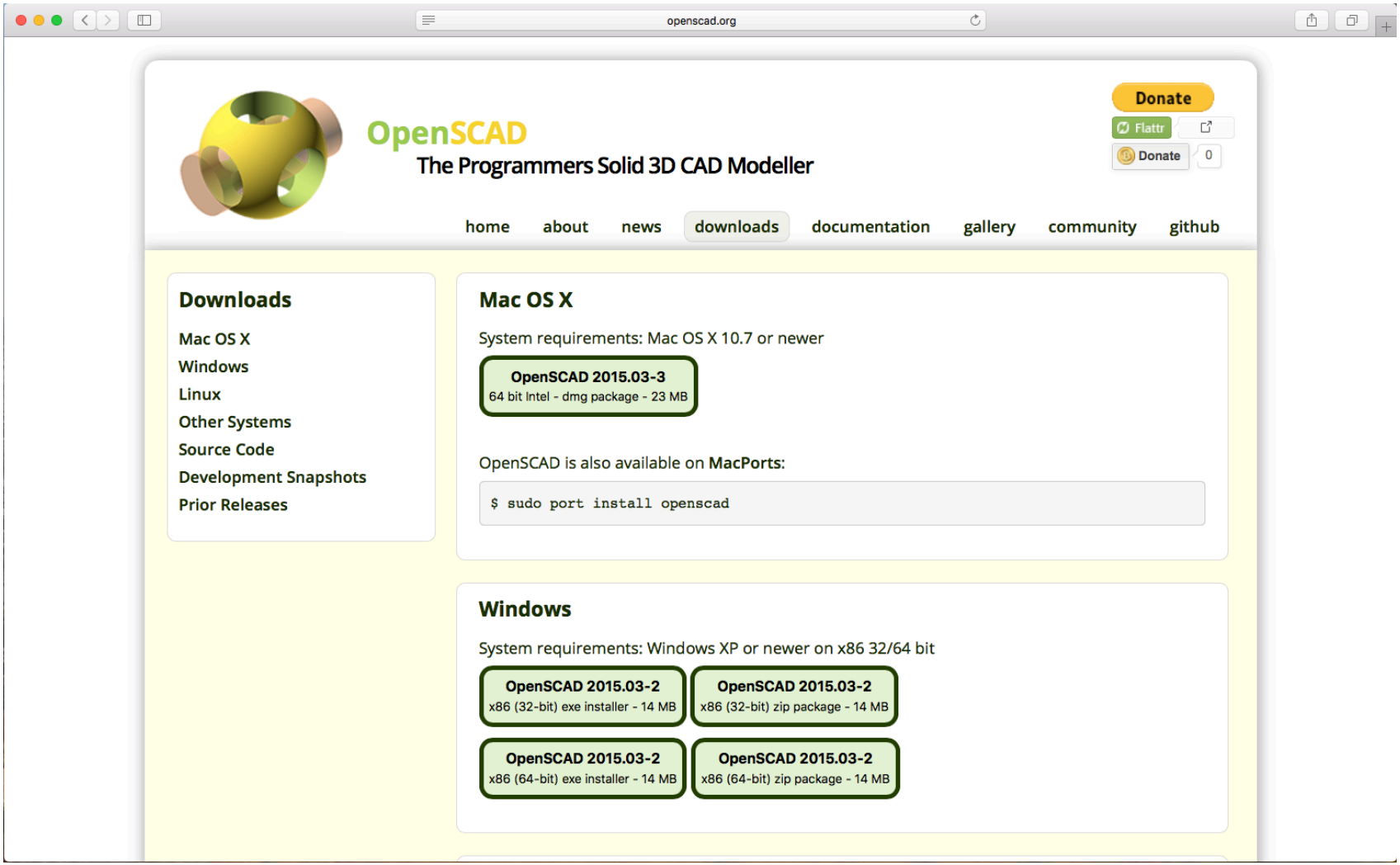
UNIVERSITÄT SALZBURG

OpenSCAD: Installation

www.openscad.org

UNIVERSITÄT SALZBURG

OpenSCAD: Installation



www.openscad.org → Downloads

UNIVERSITÄT SALZBURG

OpenSCAD: Workspace

OpenSCAD – logo.scad

File Edit Design View Help

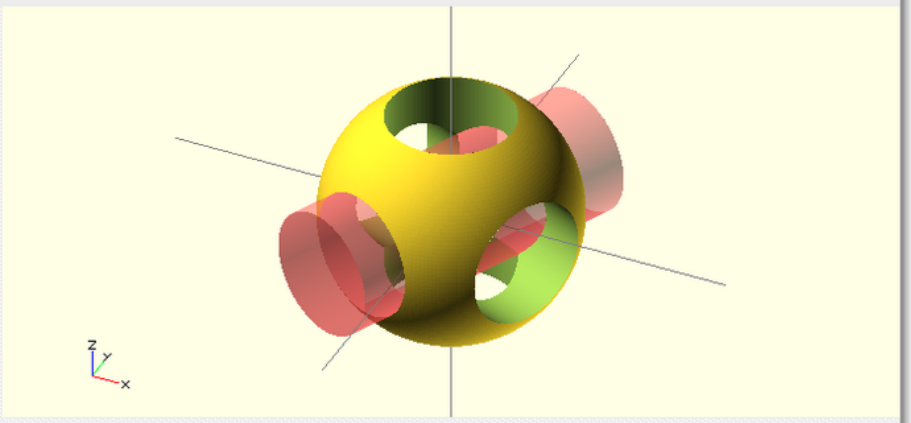
```
$fn = 200;

r = 20;
hole = 10;

module cut() {
   cylinder(r = hole, h = 2.5 * r, center = true);
}

module logo() {
   difference() {
      sphere(r = r);
      cut();
      rotate([0, 90, 0]) cut();
      #rotate([90, 0, 0]) cut();
   }
}

rotate([0, 0, -25]) logo();
```



```
Compiling design (CSG Tree generation)...
Compiling design (CSG Products generation)...
PolySets in cache: 4
PolySet cache size in bytes: 4434272
CGAL Polyhedrons in cache: 0
CGAL cache size in bytes: 0
Compiling design (CSG Products normalization)...
Compiling highlights (1 CSG Trees)...
Normalized CSG tree has 4 elements
CSG generation finished.
Total rendering time: 0 hours, 0 minutes, 0 seconds
```

Viewport: translate = [0.00 0.00 0.00], rotate = [55.00 0.00 25.00], distance = 450.00

text editor console window viewing area

UNIVERSITÄT SALZBURG

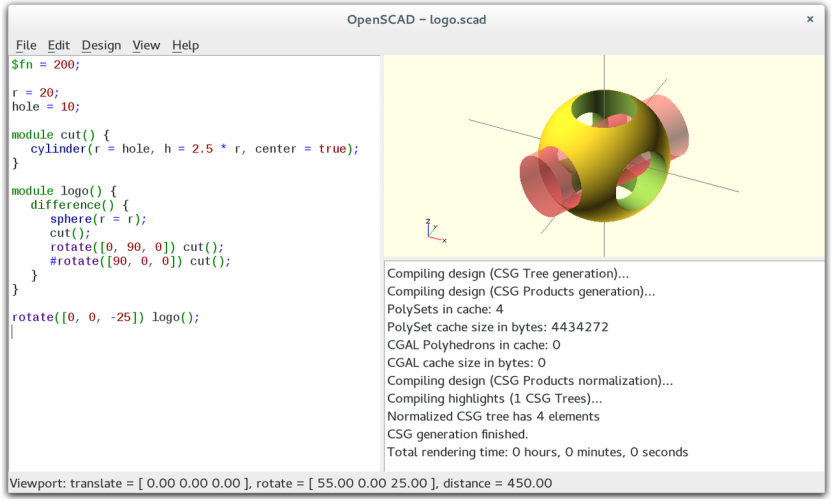
OpenSCAD: Workspace

`console window`

- Status information, warnings and errors

`text editor`

- provides basic editing features
- needs to be **compiled** to be viewed



`viewing area`

- Preview and rendering output
- Axes menu and standard perspektives

UNIVERSITÄT SALZBURG

OpenSCAD: Programmiersprache

```
object();
variable = value;
operator()   action();
operator() { action();    action(); }
operator()   operator() { action(); action(); }
operator() { operator()   action();
             operator() { action(); action(); } }
```

```
Examples

  cube(5);
  x = 4+y;
  rotate(40) square(5,10);
  translate([10,5]) { circle(5); square(4); }
  rotate(60) color("red") { circle(5); square(4); }
  color("blue") { translate([5,3,0]) sphere(5); rotate([45,0,45]) { cylinder(10); cube([5,6,7]);
} }
```

UNIVERSITÄT SALZBURG

OpenSCAD: Cheat Sheet

OpenSCAD CheatSheet v2015.03

Syntax
var = value;
module name(…) { … }
name();
function name(…) = …
name();
include <….scad>
use <….scad>

2D
circle(radius | d=diameter)
square(size,center)
square([width,height],center)
polygon([points])
polygon([points],[paths])
text(text, size, font,
halign, valign, spacing,
direction, language, script)

3D
sphere(radius | d=diameter)
cube(size, center)
cube([width,depth,height], center)
cylinder(h,r|d,center)
cylinder(h,r1|d1,r2|d2,center)
polyhedron(points, triangles, convexity)

Transformations
translate([x,y,z])
rotate([x,y,z])
scale([x,y,z])
resize([x,y,z],auto)
mirror([x,y,z])
multmatrix(m)
color("colorname",alpha)
color([r,g,b,a])
offset(r|delta,chamfer)
hull()
minkowski()

Boolean operations
union()
difference()
intersection()

Modifier Characters
* disable
! show only
highlight / debug
% transparent / background

Mathematical
abs
sign
sin
cos
tan
acos
asin
atan
atan2
floor
round
ceil
ln
len
let
log
pow
sqrt
exp
rands
min
max

Functions
concat
lookup
str
chr
search
version
version_num
norm
cross
parent_module(idx)

Other
echo(…)
for (i = [start:end]) { … }
for (i = [start:step:end]) { … }
for (i = […,…,…]) { … }
intersection_for(i = [start:end]) { … }
intersection_for(i = [start:step:end]) { … }
intersection_for(i = […,…,…]) { … }
if (…) { … }
~~assign (…) { … }~~
import("….stl")
linear_extrude(height,center,convexity,twist,slices,scale)
rotate_extrude(angle,convexity)
surface(file = "….dat",center,convexity)
projection(cut)
render(convexity)
children([idx])

List Comprehensions
Generate [for (i = *range|list*) i]
Conditions [for (i = …) if (condition(i)) i]
Assignments [for (i = …) let (assignments) a]

Special variables
$fa minimum angle
$fs minimum size
$fn number of fragments
$t animation step
$vpr viewport rotation angles in degrees
$vpt viewport translation
$vpd viewport camera distance
$children number of module children

Links: Official website | Code | Issues | Manual | MCAD library | Forum | Other links

Edit me on GitHub
By Peter Uithoven @ Fablab Amersfoort (CC-BY

UNIVERSITÄT SALZBURG

Übersicht

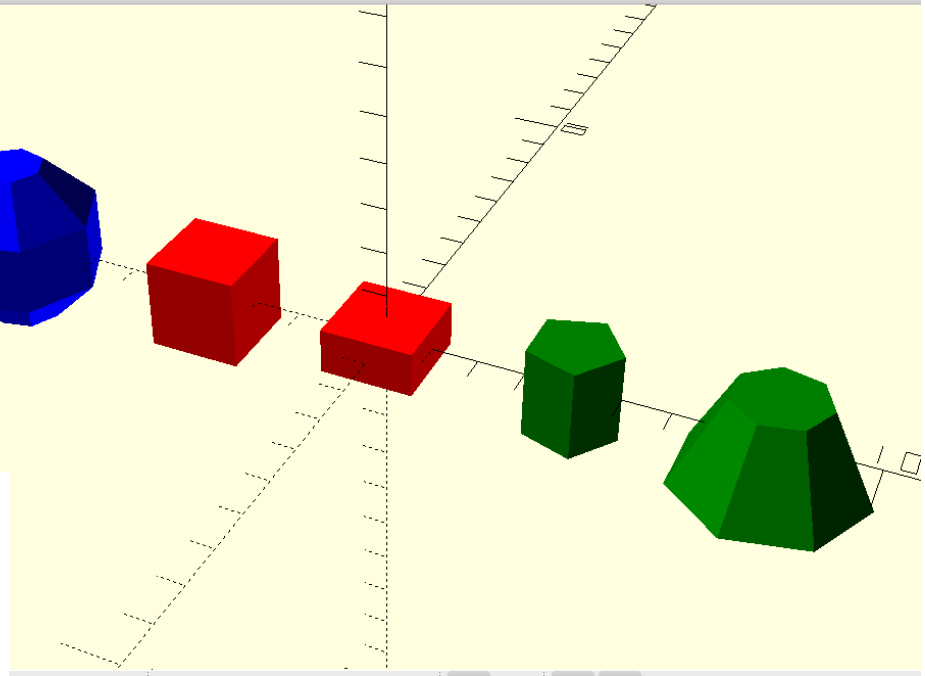
UNIVERSITÄT SALZBURG

OpenSCAD: 3D-Primitive

Unbenannt.scad

Editor

```
translate([-9,0,0]) color("Blue")
    sphere(2);
translate([-4,0,0]) color("Red")
    cube(2, center=true);
translate([0,0,0]) color("Red")
    cube([2,2,1], center=true);
translate([4,0,0]) color("Green")
    cylinder(2,1,center=true);
translate([8,0,0]) color("Green")
    cylinder(2,2,1,center=true);
```



Konsole

```
s/chaslinger/Documents/OpenSCAD/backups/unsaved-backup-JjKM5812.scad
Tree generation)...
Products generation)...

bytes: 14448
che: 0
s: 0
Products normalization)...
s 5 elements
shed.
ours, 0 minutes, 0 seconds
```

OpenSCAD 2015.03

3D

sphere(radius | d=diameter)

cube(size, center)

cube([width,depth,height], center)

cylinder(h,r|d,center)

cylinder(h,r1|d1,r2|d2,center)

polyhedron(points, triangles, convexity)

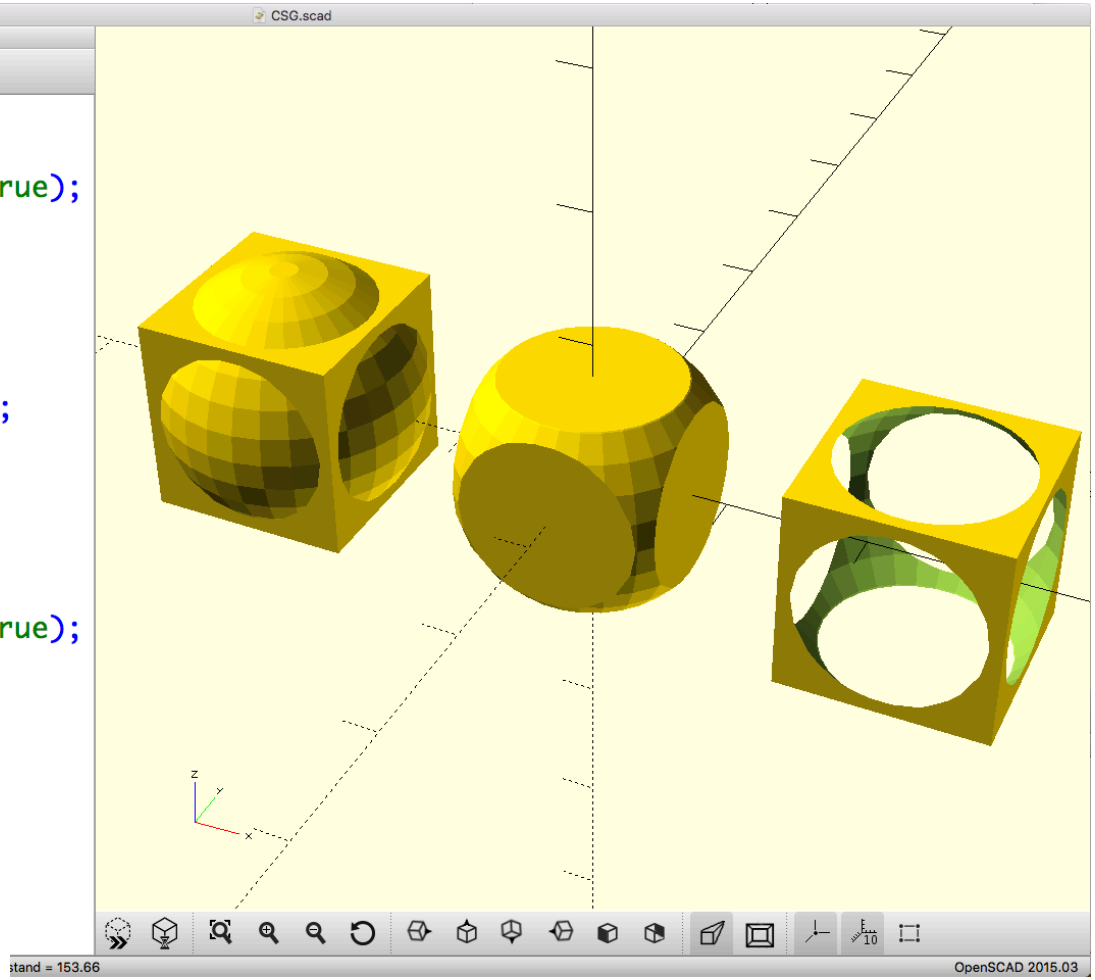
UNIVERSITÄT SALZBURG

OpenSCAD: Boolesche Operatoren

```
translate([-24,0,0]) {
    union() {
        cube(15, center=true);
        sphere(10);
    }
}

intersection() {
    cube(15, center=true);
    sphere(10);
}

translate([24,0,0]) {
    difference() {
        cube(15, center=true);
        sphere(10);
    }
```



Boolean operations

union()

difference()

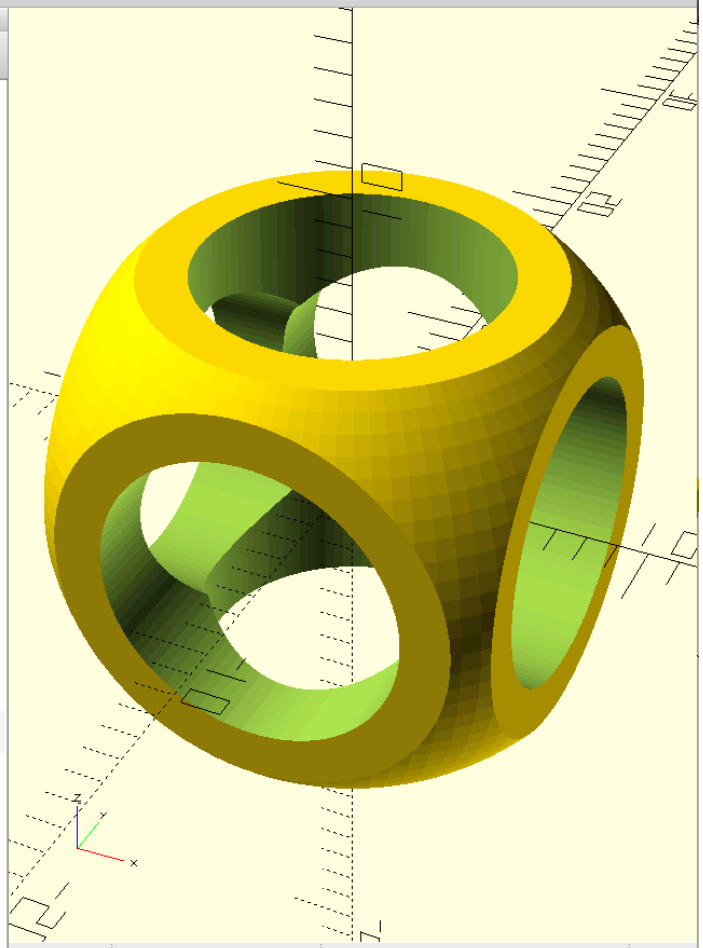
intersection()

OpenSCAD: Konstruktion **Dice**

Dice.scad

Editor

```
difference(){
    intersection(){
        cube(15, center=true);
        sphere(10);
    }
    union(){
        union(){
            cylinder(h=20, r=5, center=true);
            rotate([0,90,0])
                cylinder(h=20, r=5, center=true);
        }
        rotate([90,0,0])
            cylinder(h=20, r=5, center=true);
    }
}
$fn=100;
```



Ansicht: Verschiebung = [-0.00 -0.00 -0.00], Rotation = [55.00 0.00 25.00], Abstand = 75.45

OpenSCAD 2015.03

Video: OpenSCAD_CSG

UNIVERSITÄT SALZBURG

OpenSCAD: Konstruktion **Dice** mit Modulen

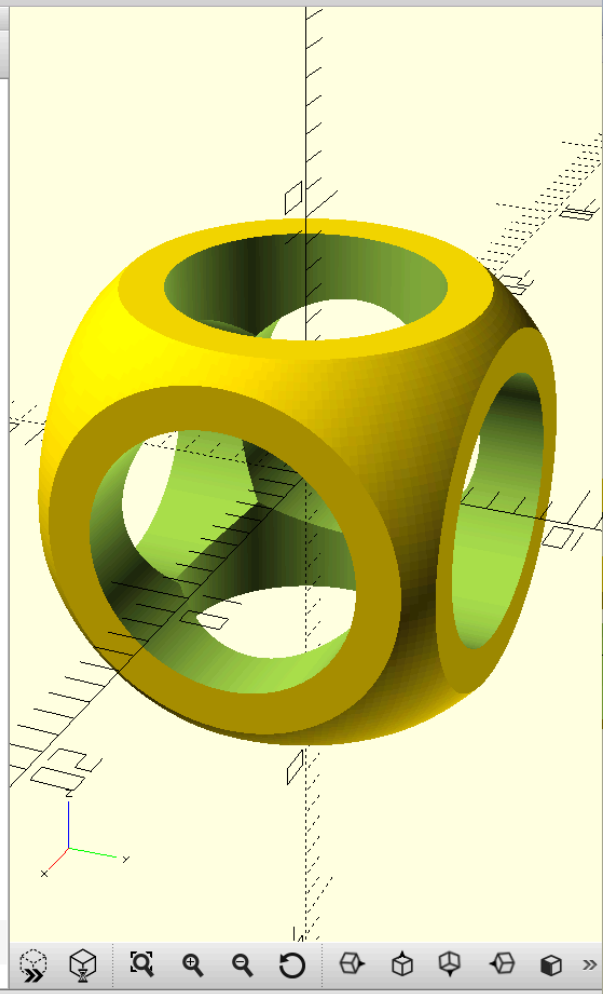
Dice.scad

Editor

```
module cylKreuz() {
    cylinder(h=20, r=5, center=true);
    rotate([0,90,0])
        cylinder(h=20, r=5, center=true);
    rotate([90,0,0])
        cylinder(h=20, r=5, center=true);
}
module dice() {
    intersection(){
        cube(15, center=true);
        sphere(10);
    }
}

difference() {
    dice();
    cylKreuz();
}

$fn=200;
```



Ansicht: Verschiebung = [-0.00 -0.00 -0.00], Rotation = [63.40 0.00 113.00], Abstand = 86.23

OpenSCAD 2015.03

Video: OpenSCAD_CSG_modules

UNIVERSITÄT SALZBURG

Übersicht

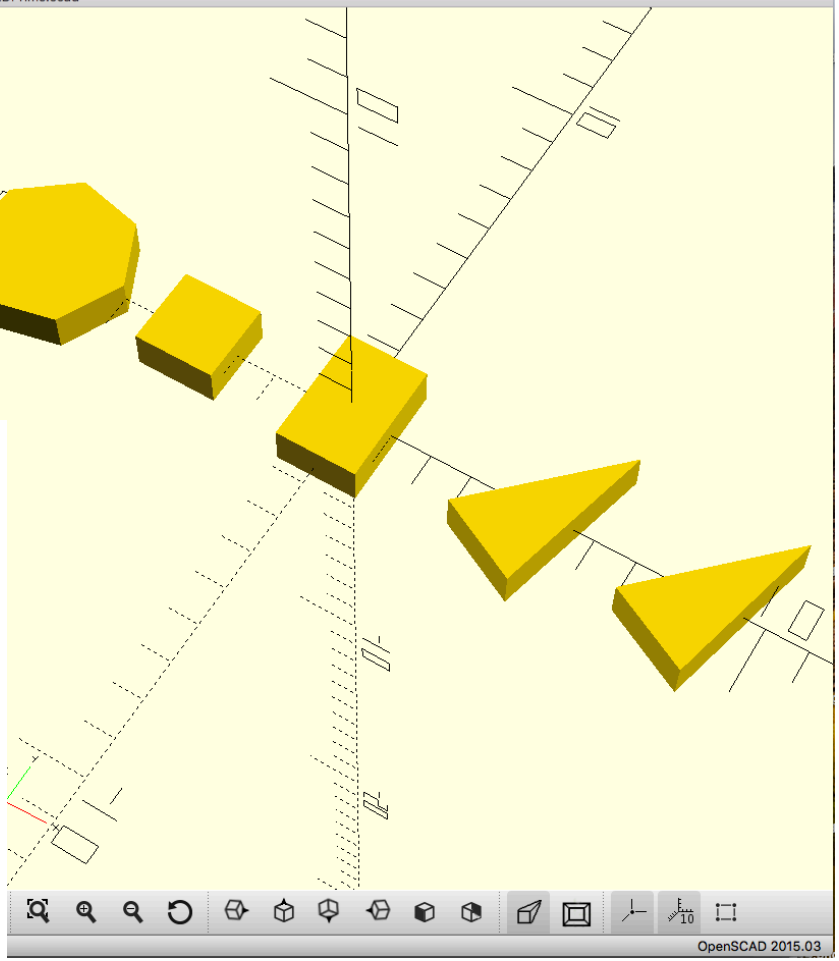
UNIVERSITÄT SALZBURG

OpenSCAD: 2D-Primitive

2DPrims.scad

Editor

```
translate([-8,0,0])
    circle(2);
translate([-4,0,0])
    square(2,center=true);
translate([-0,0,0])
    square([2,3],center=true);
translate([4,0,0])
    polygon([
```

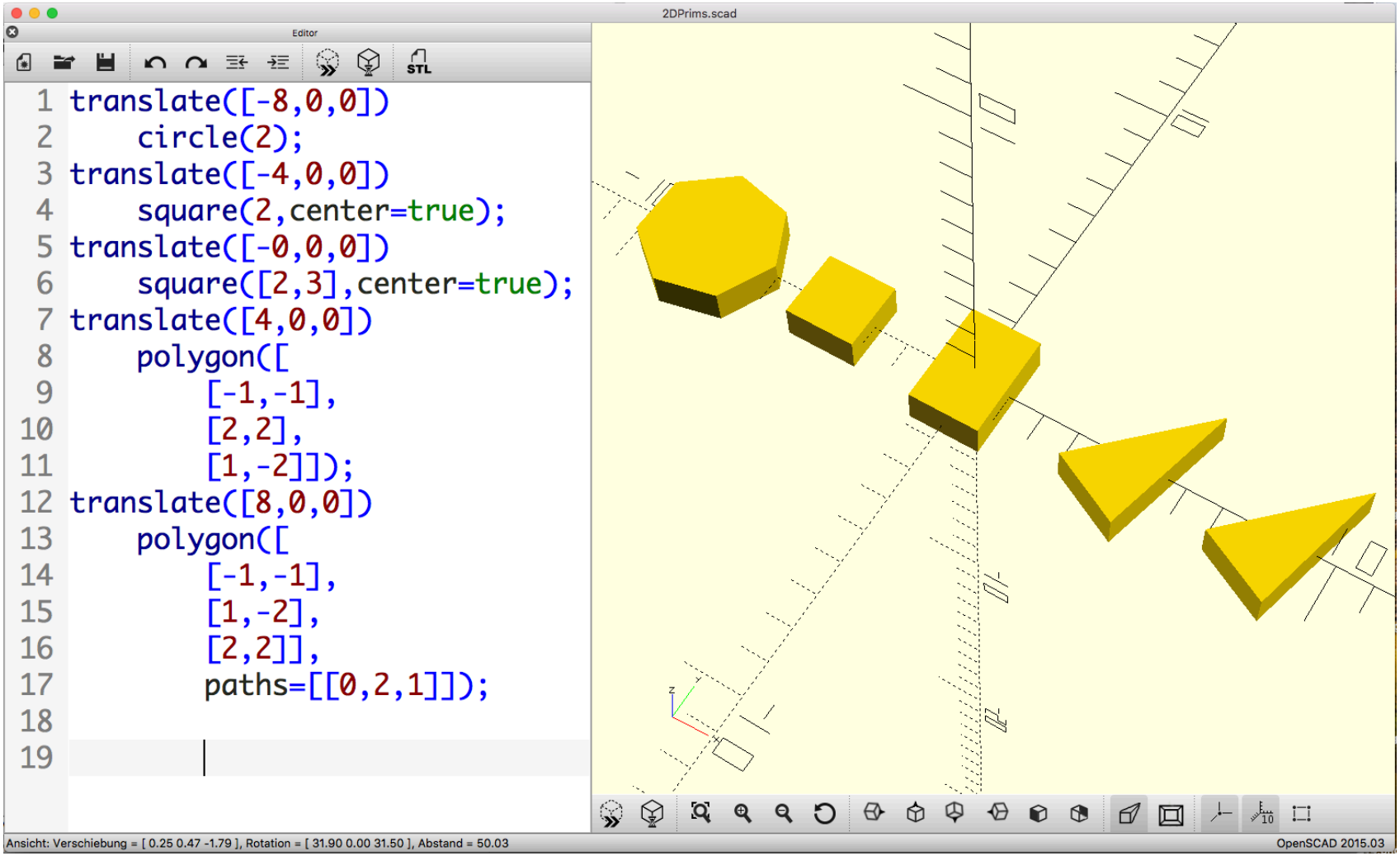


2D

```
circle(radius | d=diameter)
square(size,center)
square([width,height],center)
polygon([points])
polygon([points],[paths])
text(text, size, font,
     halign, valign, spacing,
     direction, language, script)
```

UNIVERSITÄT SALZBURG

OpenSCAD: 2D-Primitive

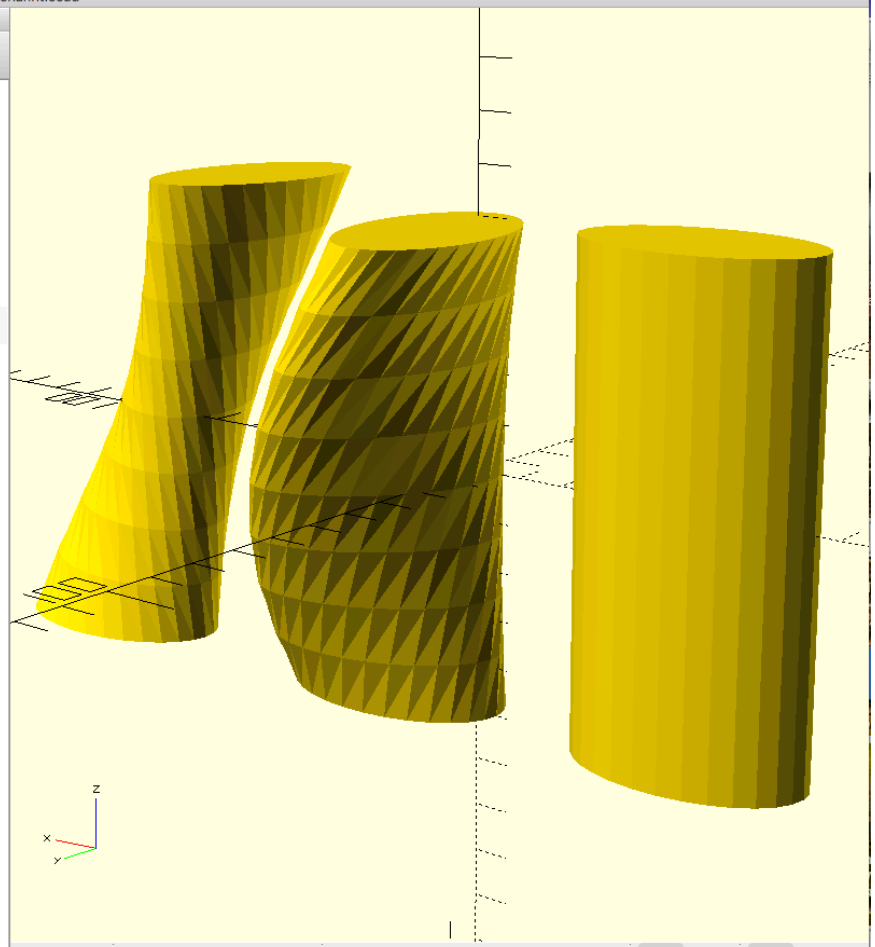


OpenSCAD: Lineare Extrusion

Unbenannt.scad

Editor

```
translate([-50,0,0])
linear_extrude(
    height = 100,
    center = true,
    convexity = 10,
    twist = 0)
        resize([50,30]) circle(d=20);
linear_extrude(
    height = 100,
    center = true,
    convexity = 10,
    twist = -100)
        translate([20, 0, 0])
            resize([50,30]) circle(d=20);
translate([80,0,0])
linear_extrude(
    height = 100,
    center = true,
    convexity = 10,
    twist = 100)
        translate([20, 0, 0])
            resize([50,30]) circle(d=20);
```

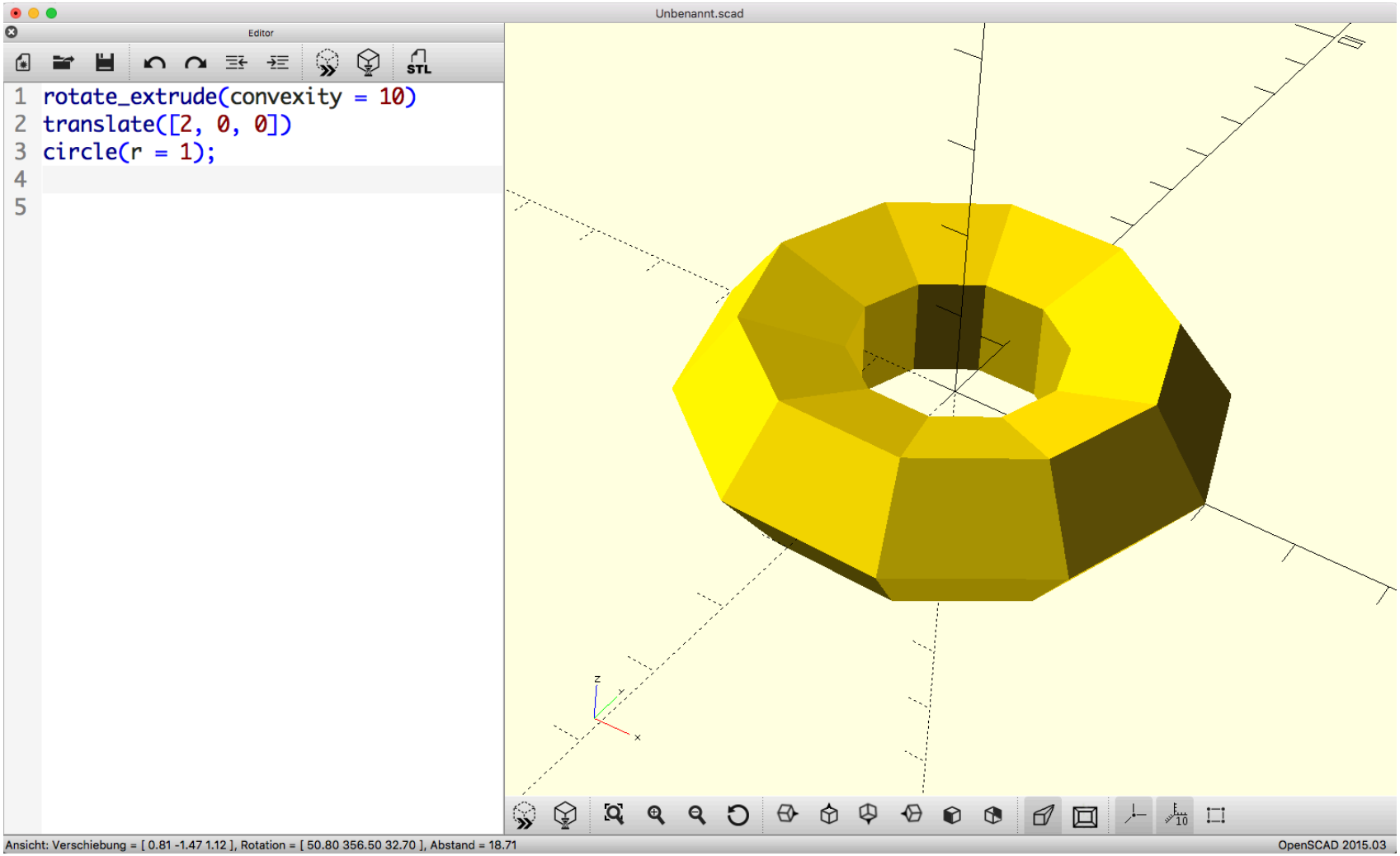


Ansicht: Verschiebung = [5.28 5.31 -0.70], Rotation = [75.30 0.00 218.20], Abstand = 458.84

OpenSCAD 2015.03

UNIVERSITÄT SALZBURG

OpenSCAD: Rotation um eine Achse



UNIVERSITÄT SALZBURG

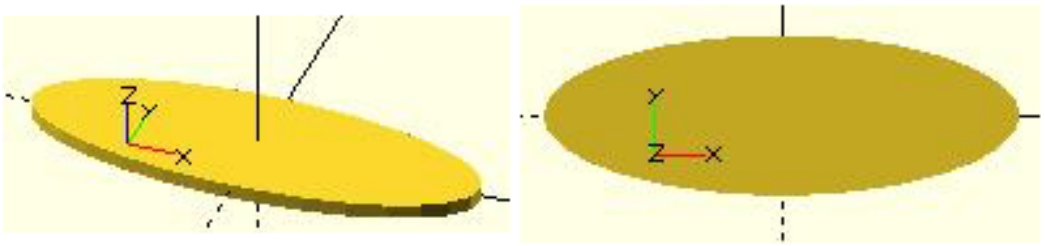
Übersicht

UNIVERSITÄT SALZBURG

OpenSCAD: Ellipse mit `resize` und `scale`

Transformations

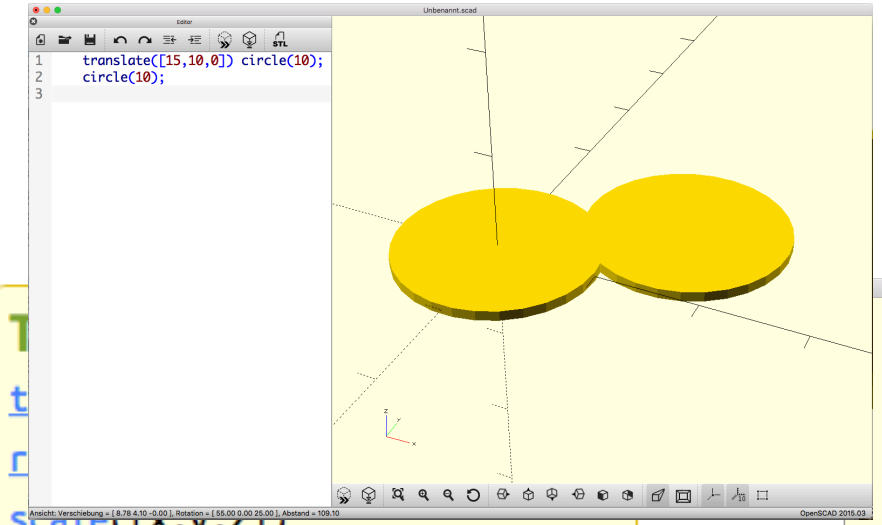
```
translate([x,y,z])
rotate([x,y,z])
scale([x,y,z])
resize([x,y,z],auto)
mirror([x,y,z])
multmatrix(m)
color("colorname",alpha)
color([r,g,b,a])
offset(r|delta,chamfer)
hull()
minkowski()
```



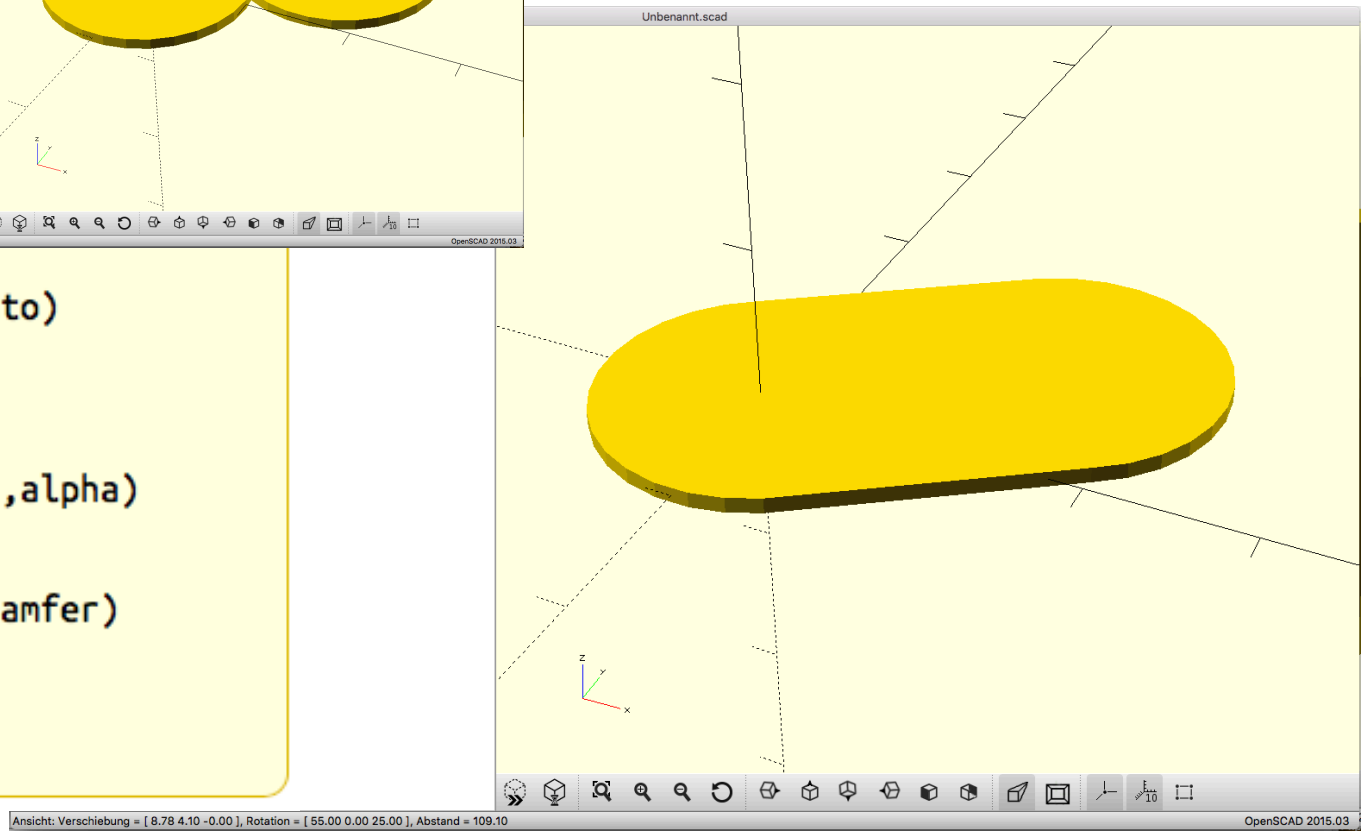
```
equivalent scripts for this example
 resize([30,10])circle(d=20);
 scale([1.5,.5])circle(d=20);
```

UNIVERSITÄT SALZBURG

OpenSCAD: hull

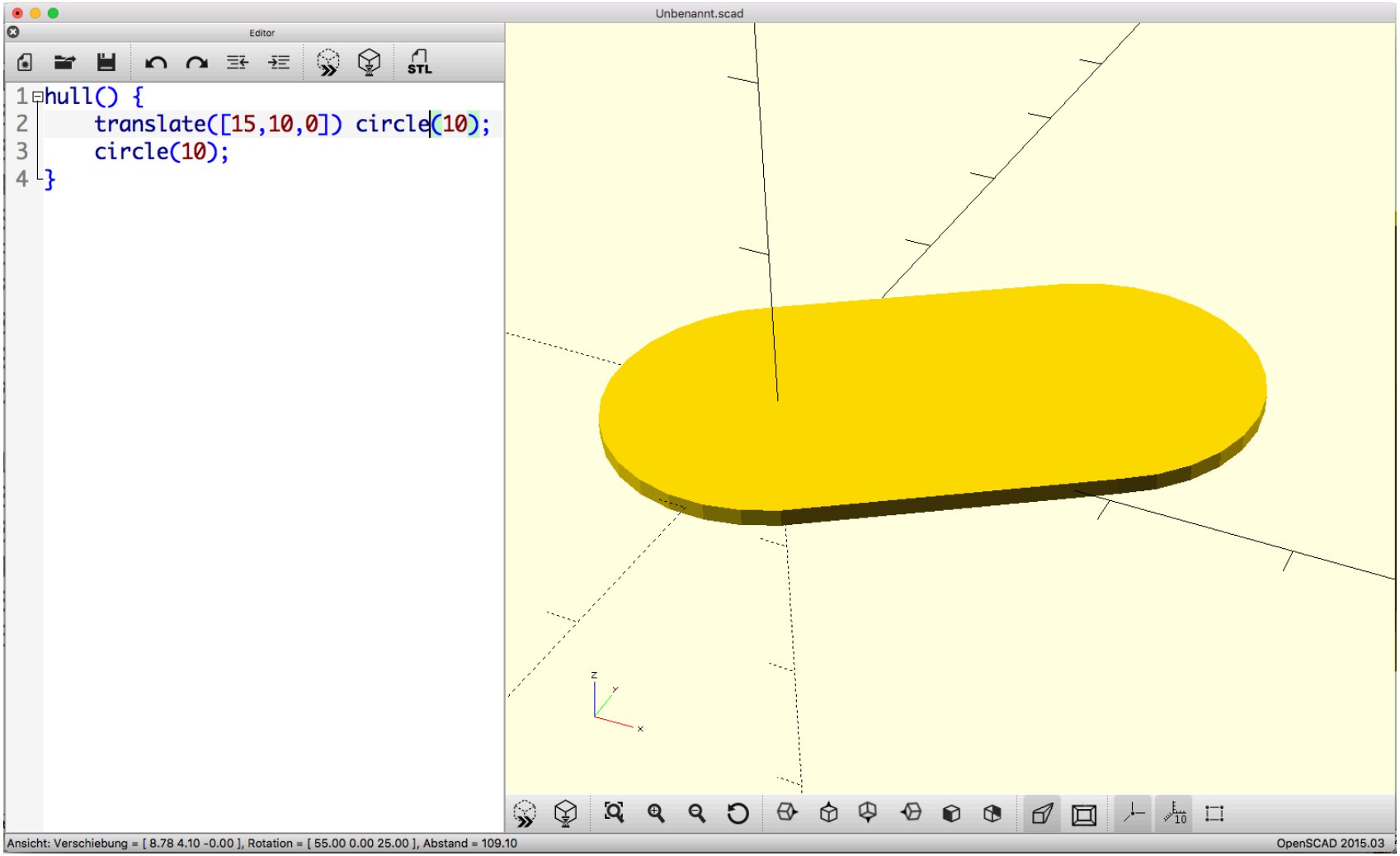



```
resize([x,y,z],auto)
mirror([x,y,z])
multmatrix(m)
color("colorname",alpha)
color([r,g,b,a])
offset(r|delta,chamfer)
hull()
minkowski()
```



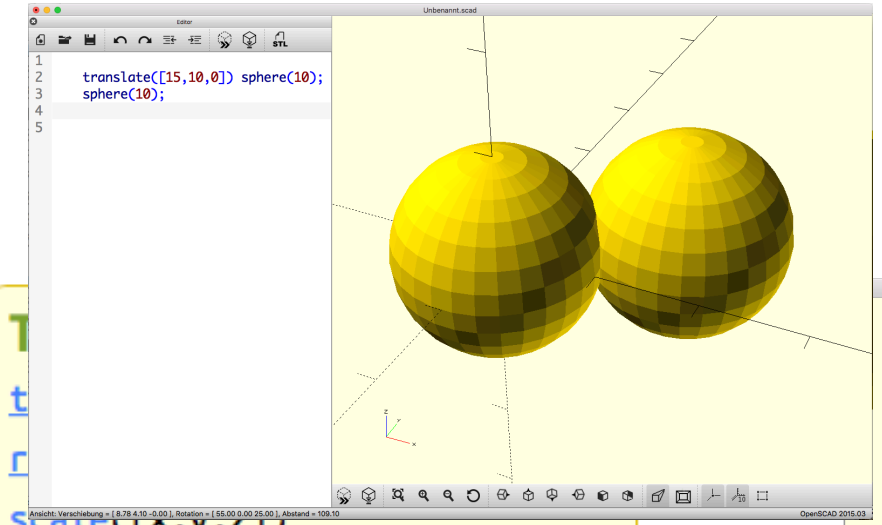
UNIVERSITÄT SALZBURG

OpenSCAD: `hull`

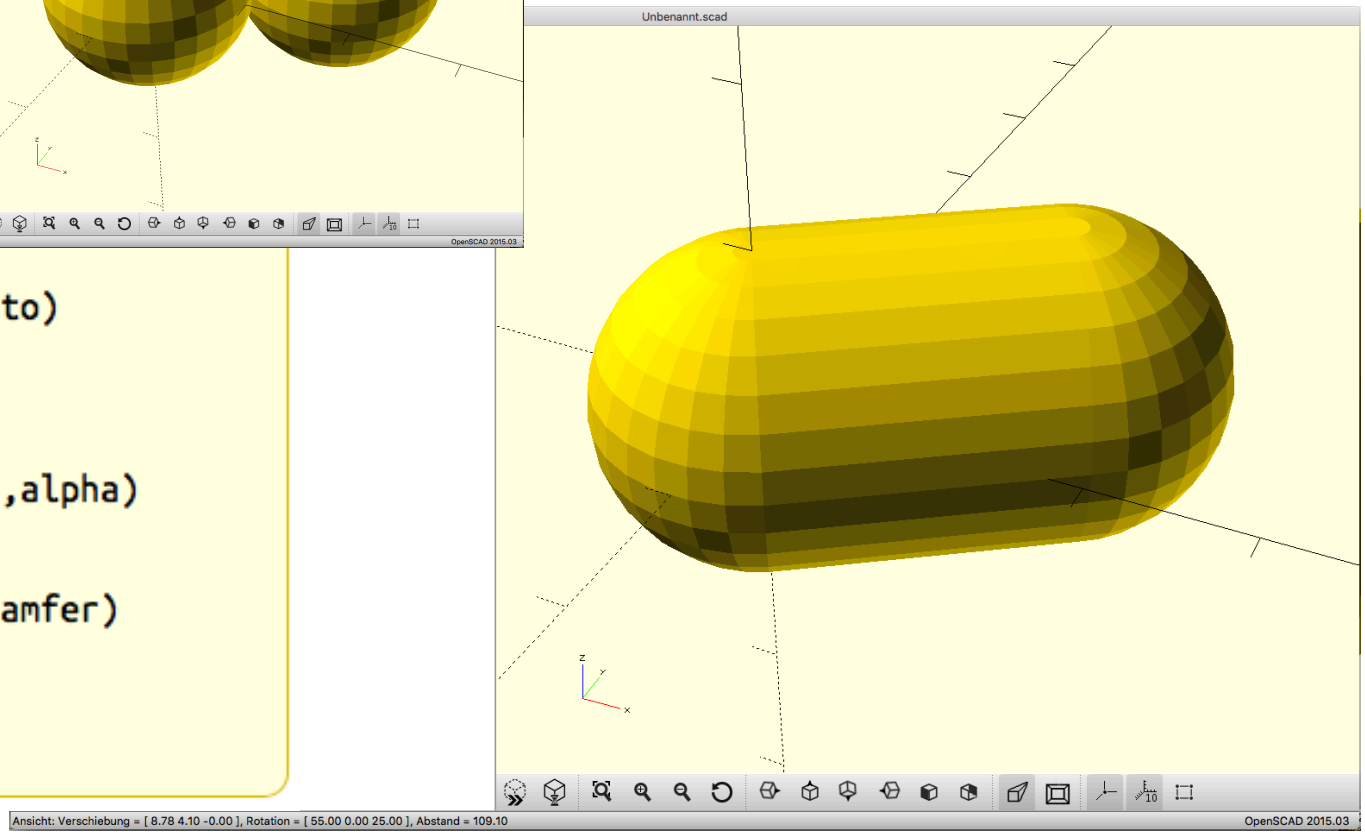


UNIVERSITÄT SALZBURG

OpenSCAD: `hull`

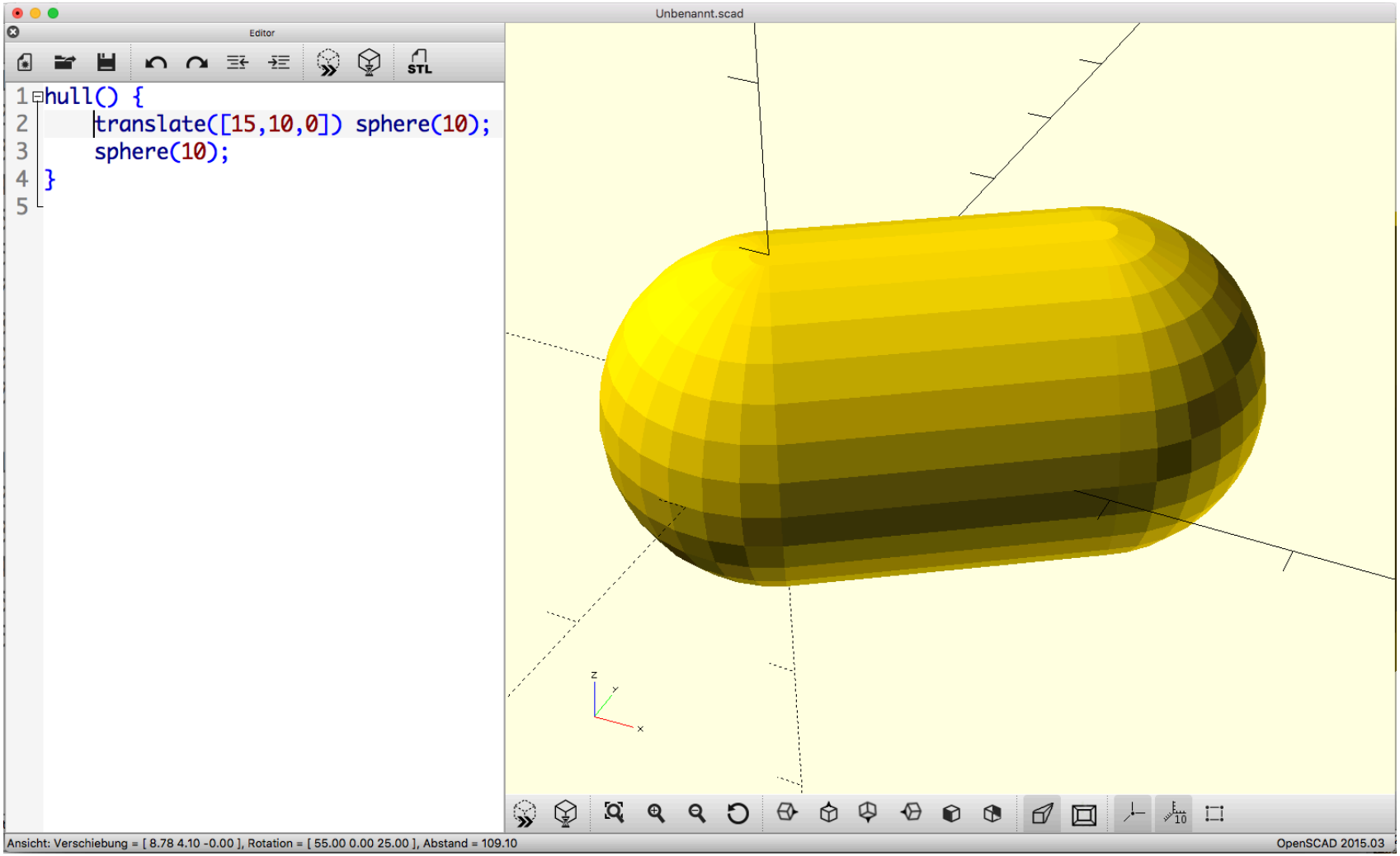



```
resize([x,y,z],auto)
mirror([x,y,z])
multmatrix(m)
color("colorname",alpha)
color([r,g,b,a])
offset(r|delta,chamfer)
hull()
minkowski()
```



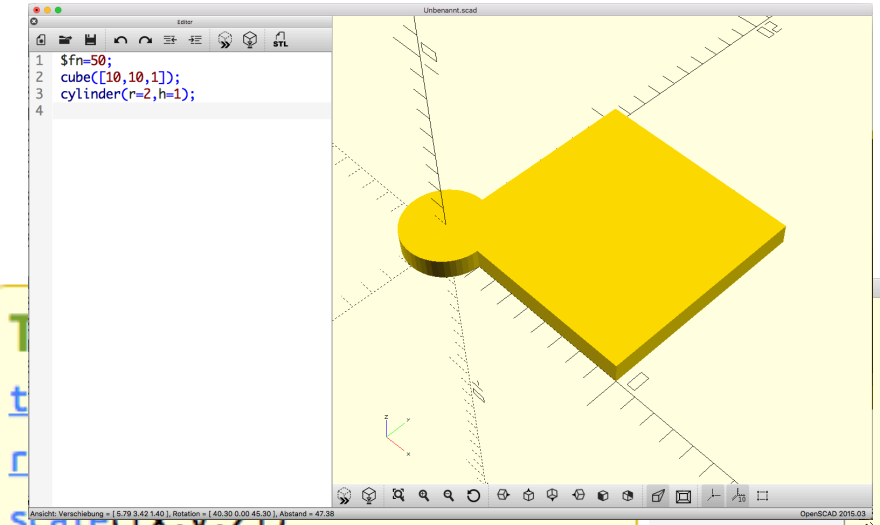
UNIVERSITÄT SALZBURG

OpenSCAD: `hull`

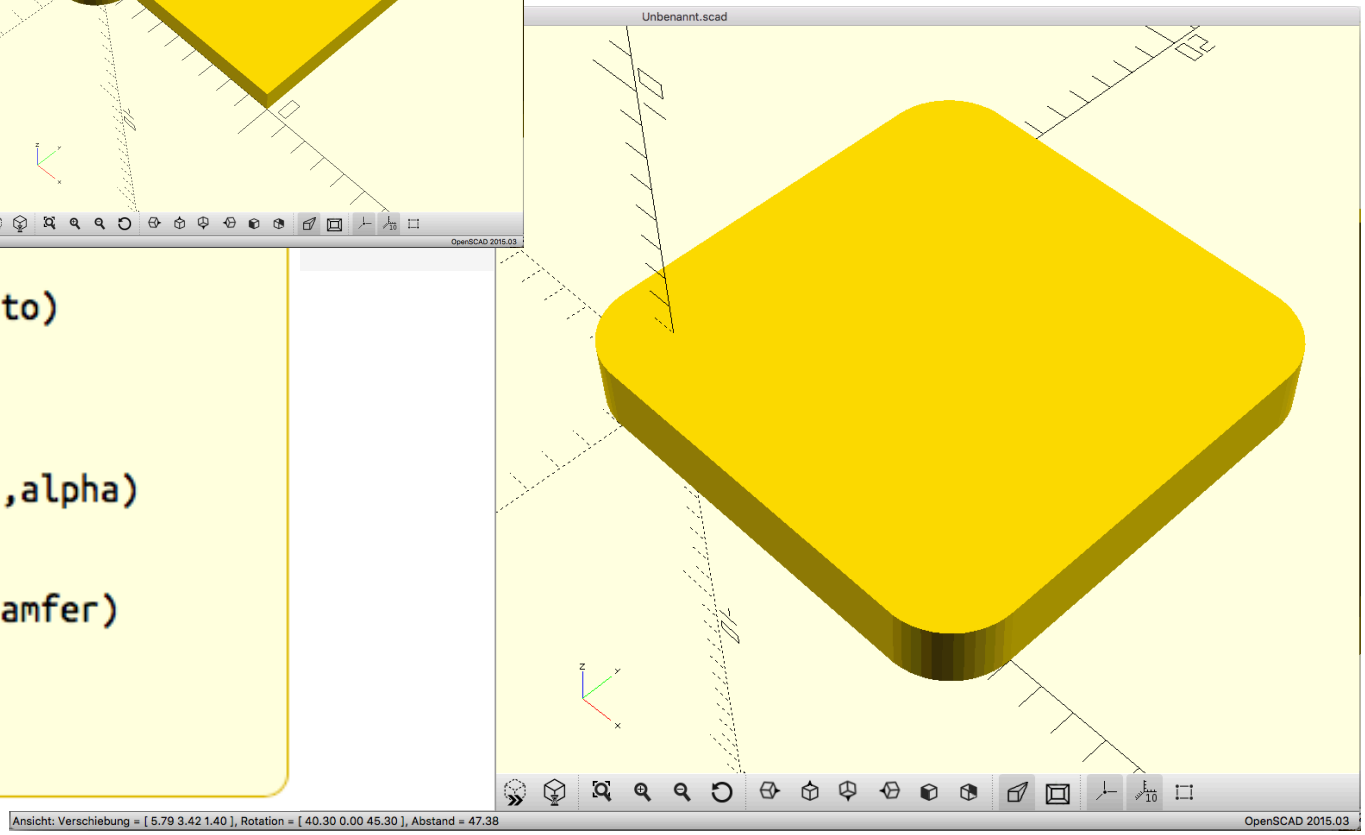


UNIVERSITÄT SALZBURG

OpenSCAD: minkowski

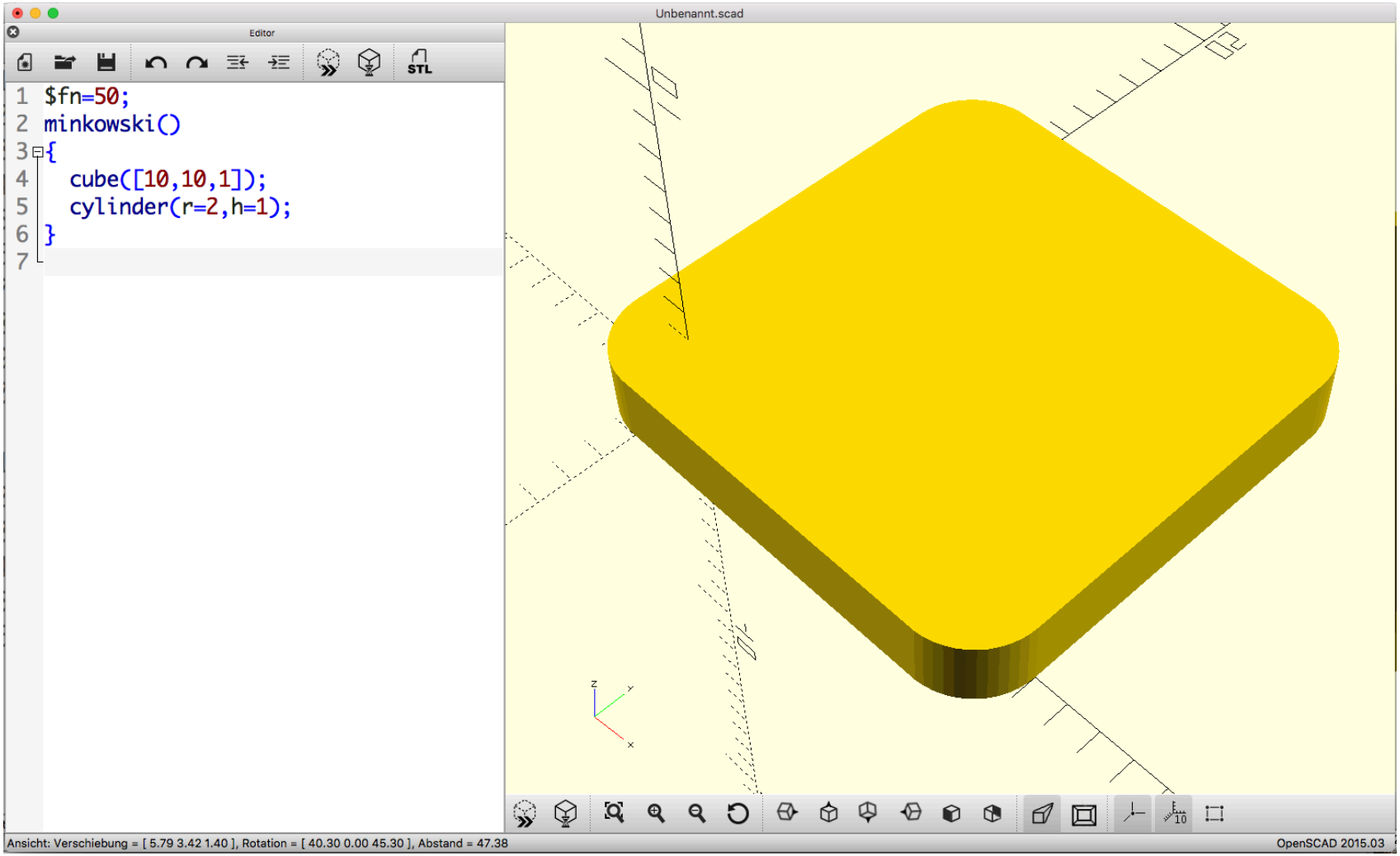



```
resize([x,y,z],auto)
mirror([x,y,z])
multmatrix(m)
color("colorname",alpha)
color([r,g,b,a])
offset(r|delta,chamfer)
hull()
minkowski()
```



UNIVERSITÄT SALZBURG

OpenSCAD: minkowski



UNIVERSITÄT SALZBURG

Übersicht

UNIVERSITÄT SALZBURG

OpenSCAD: Schlüsselanhänger

Unbenannt.scad

Editor

```
module cylinderh() {
    translate([0,0,2])
        linear_extrude(height = 3)
            translate([2,0,3])
                resize([35,22]) circle(10);
}
module ellipsoid() {
    intersection(){
        resize([50,30,10]) sphere(20);
        cube([60,40,6], center=true);
    }
}
module texth(){
    translate([-9,-4.5,2])
        text("C.H.", center=true);
}
module hole(){
    translate([-17.5,0,-5])
        cylinder(d=3, h=40);
}
difference(){
    difference(){
       ellipsoid();
       cylinderh();}
    hole();
}
texth();
```

Ansicht: Verschiebung = [0.81 -1.47 1.12], Rotation = [48.00 356.50 41.80], Abstand = 134.22

OpenSCAD 2015.03

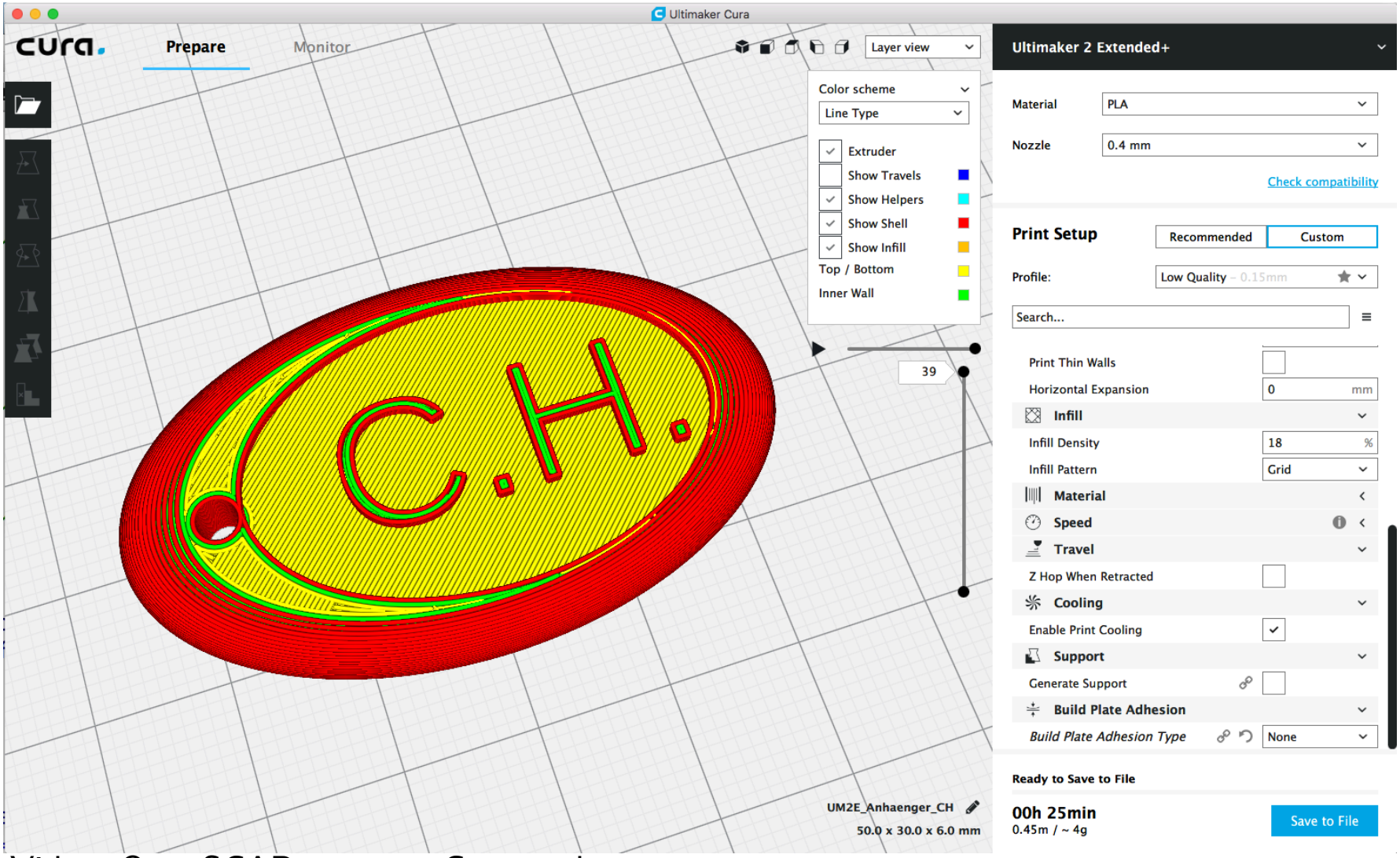
Video: OpenSCAD_anhaenger

UNIVERSITÄT SALZBURG

Übersicht

UNIVERSITÄT SALZBURG

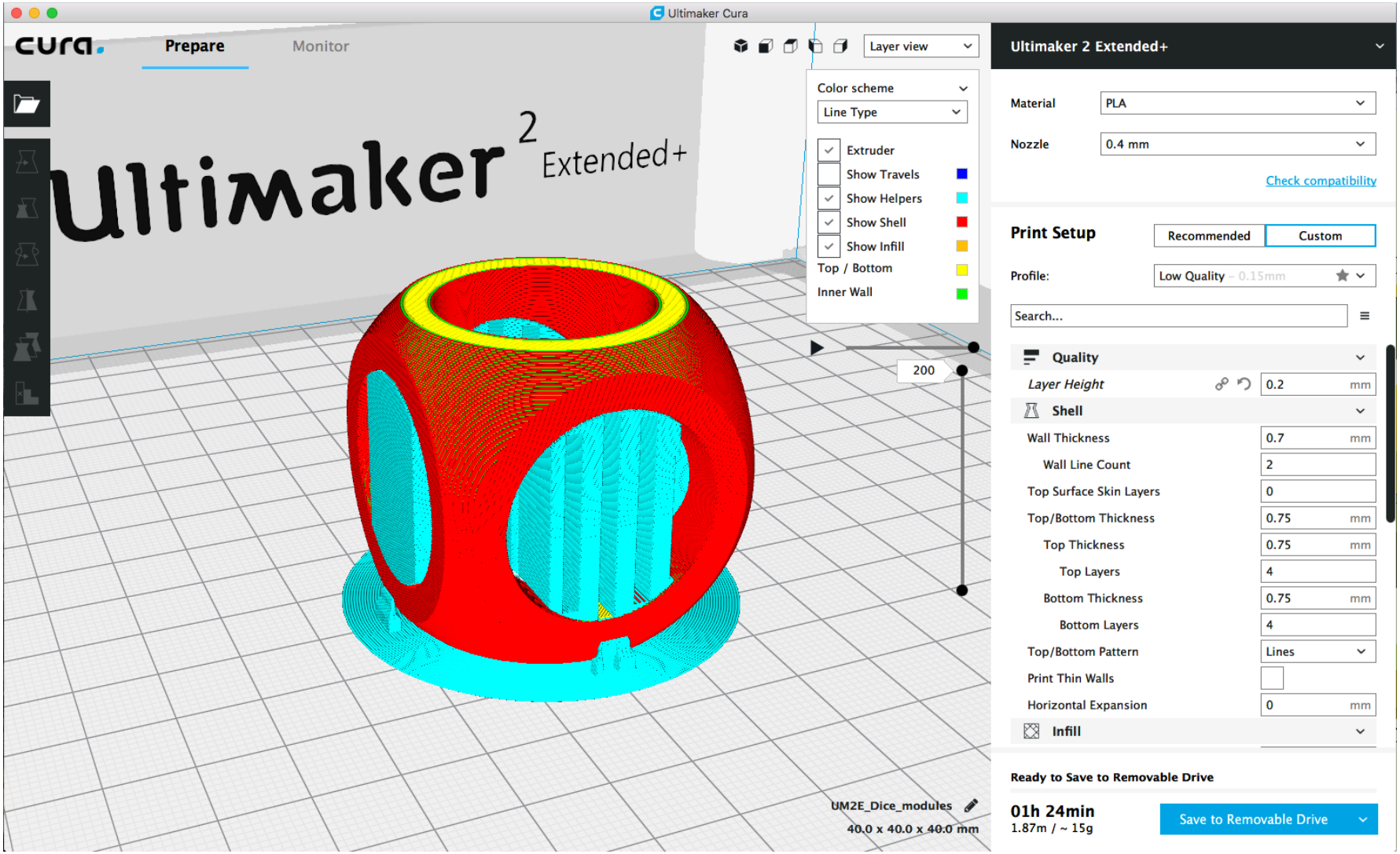
Export STL to Cura



Video: OpenSCAD_export_Cura_anhaenger

UNIVERSITÄT SALZBURG

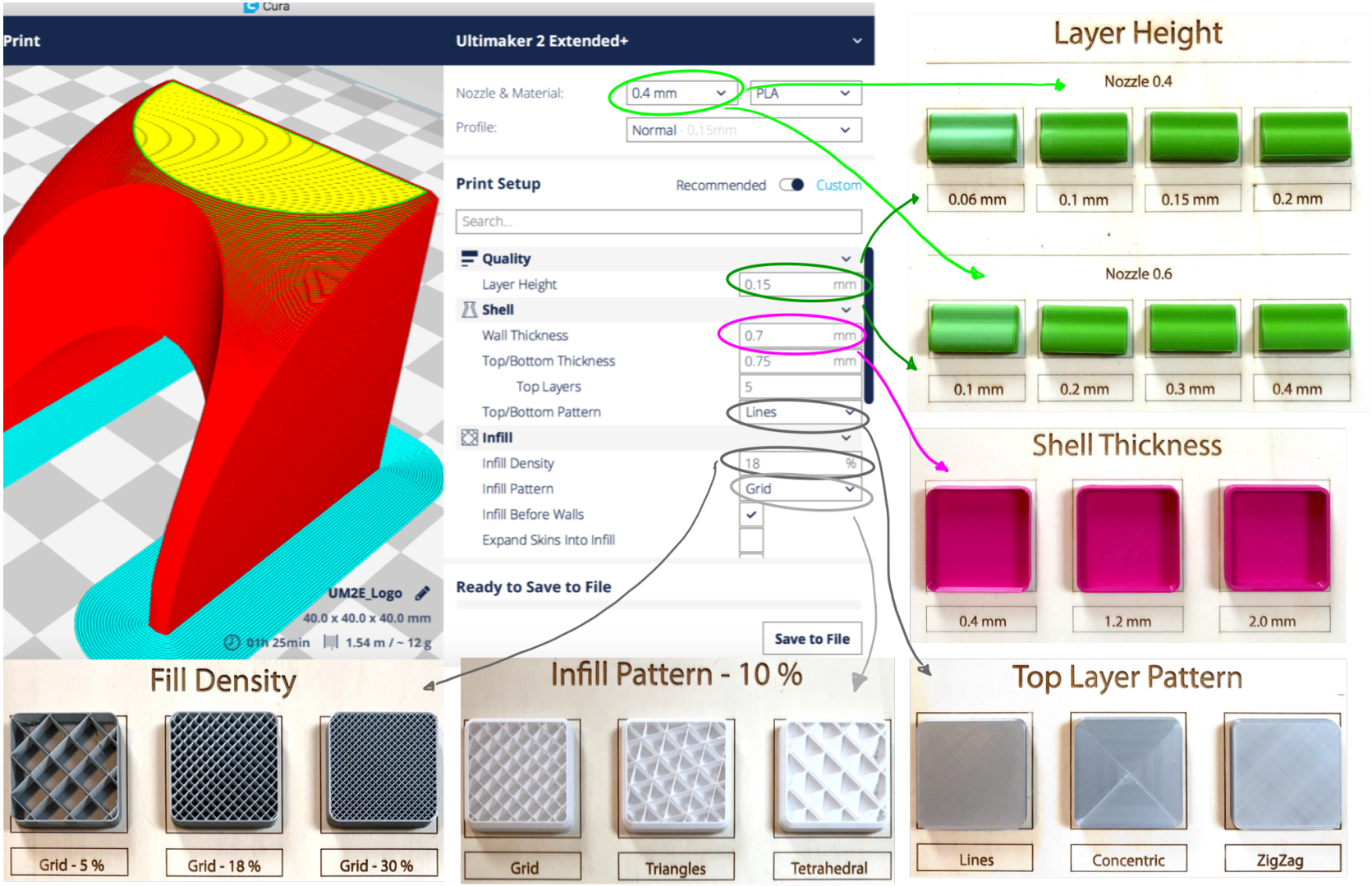
Export STL to Cura



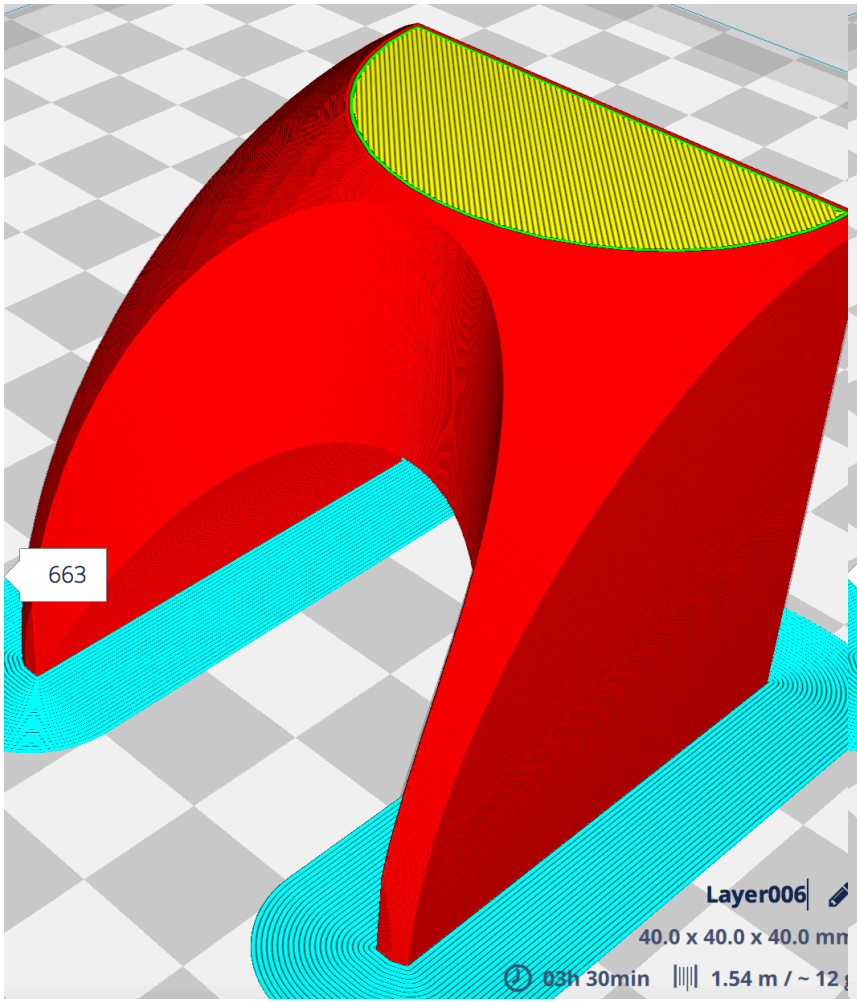
Video: OpenSCAD_export_Cura

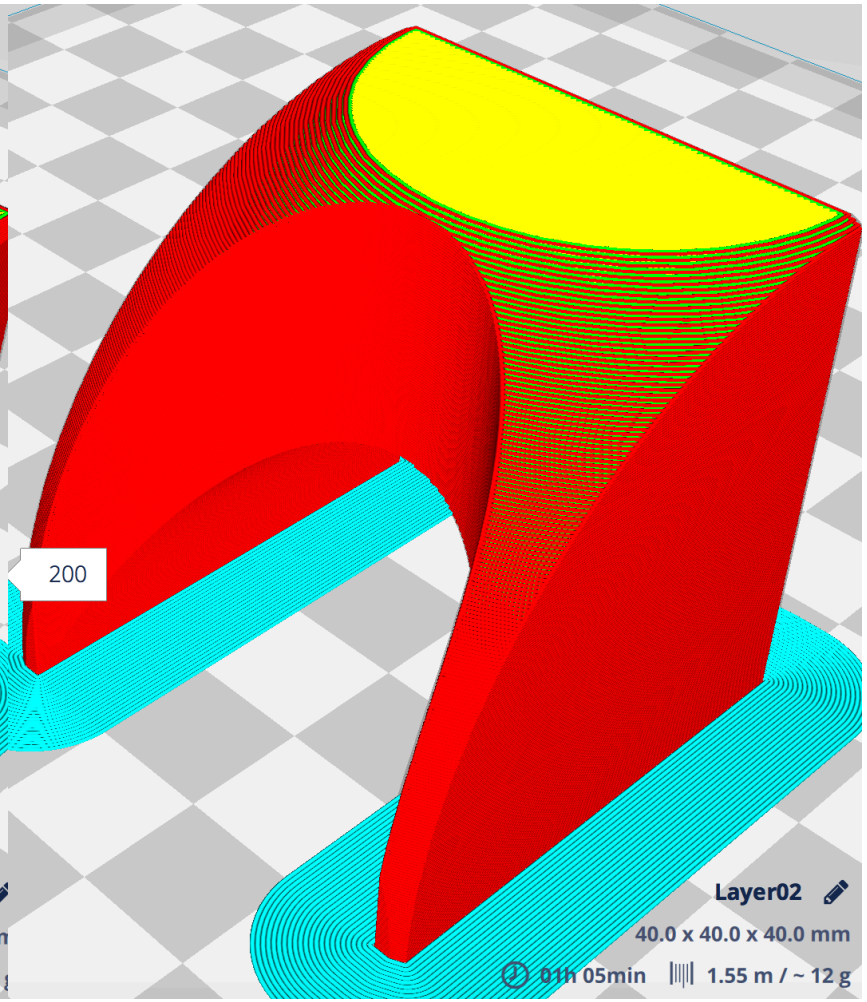
UNIVERSITÄT SALZBURG

Slicing-Software Cura: Einstellungen



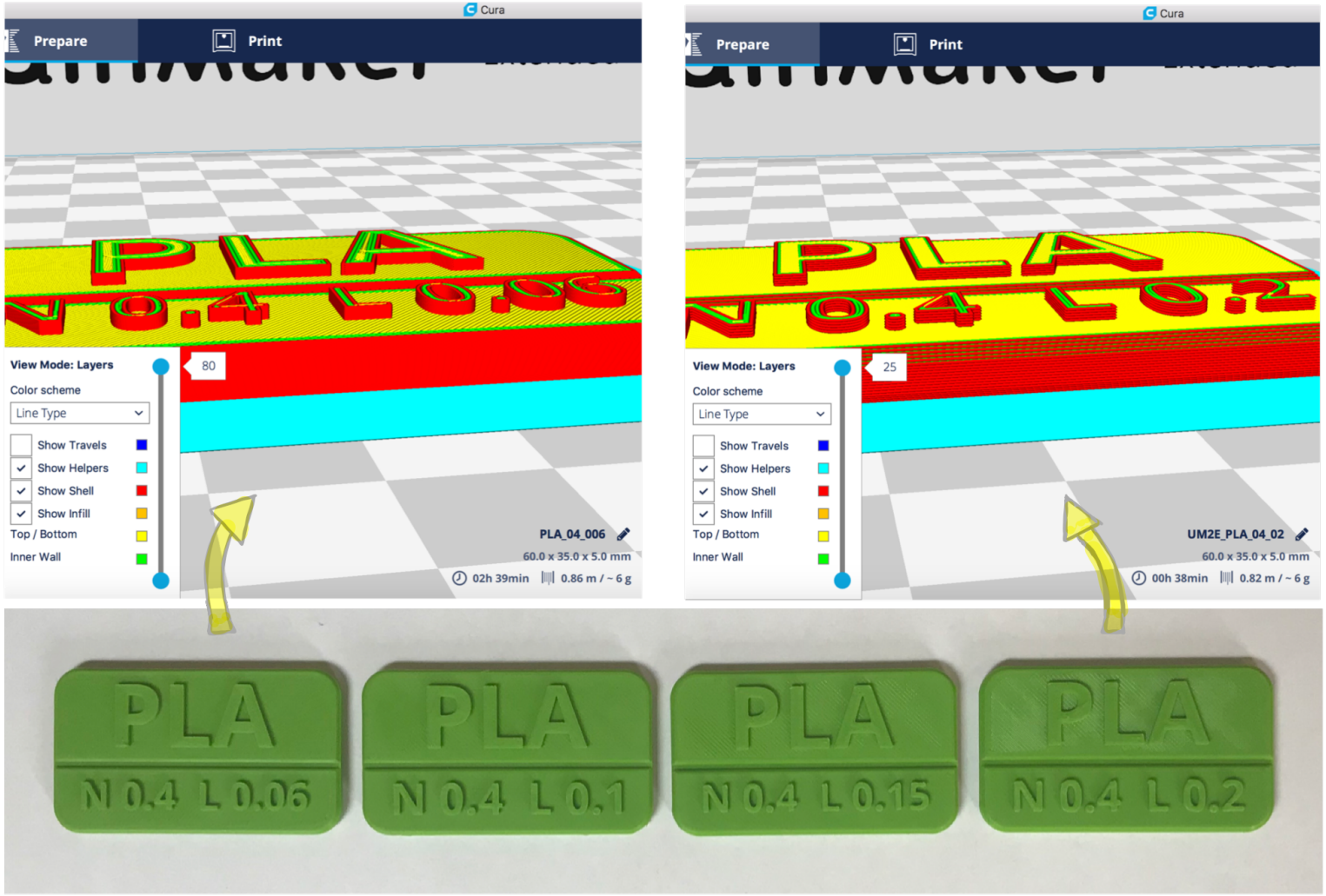
Quality: Layer Height





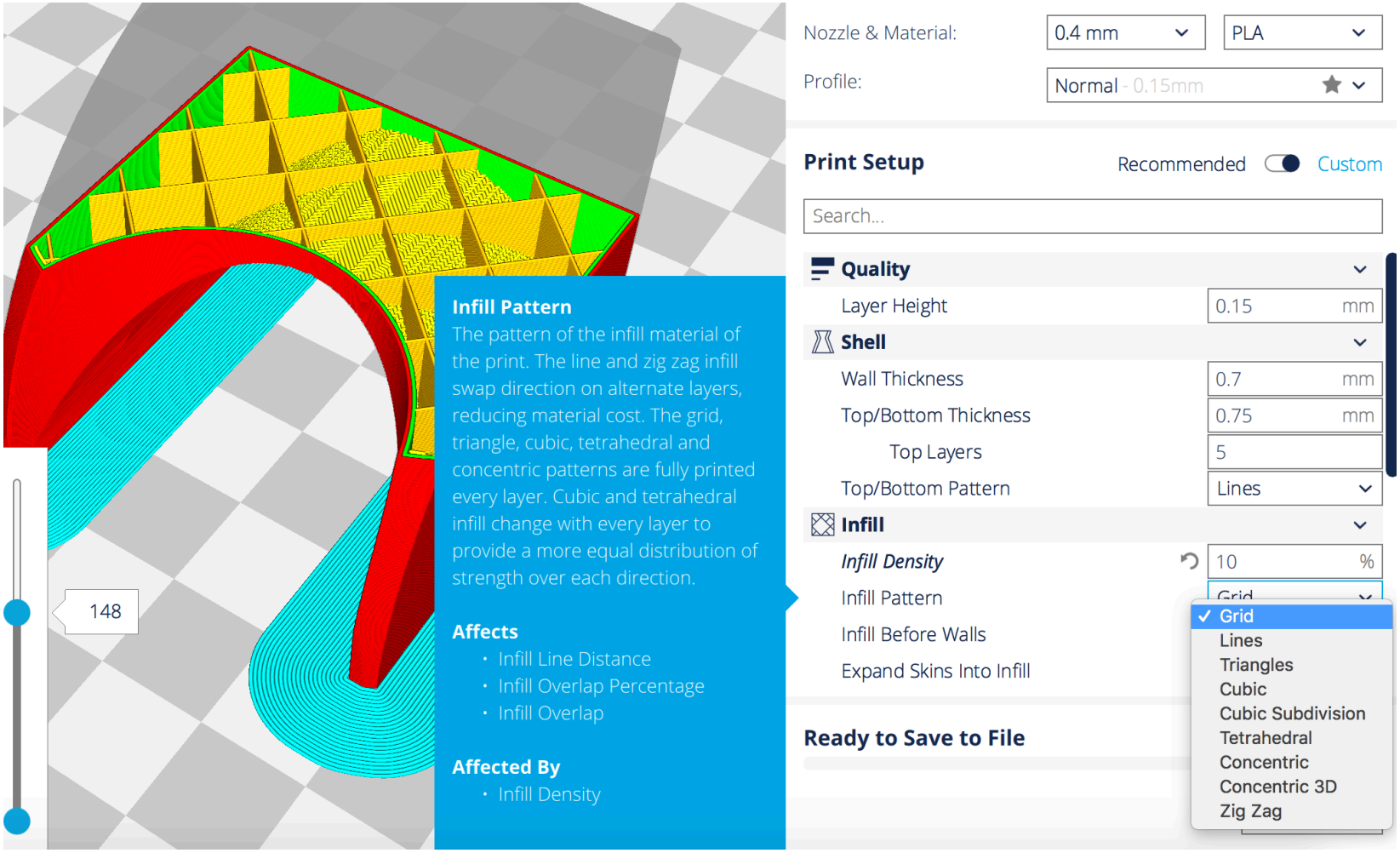
UNIVERSITÄT SALZBURG

Quality: Layer Height

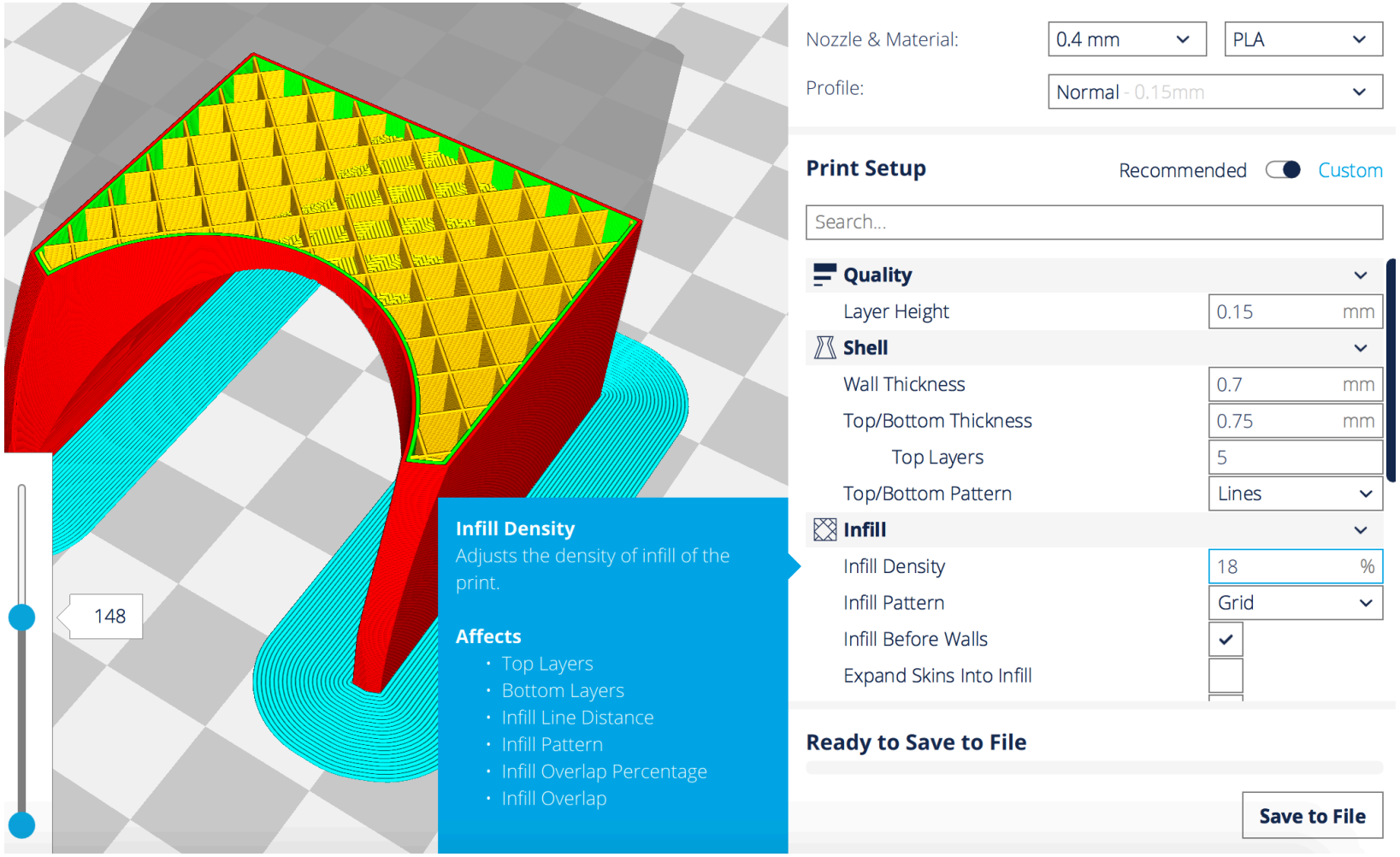


UNIVERSITÄT SALZBURG

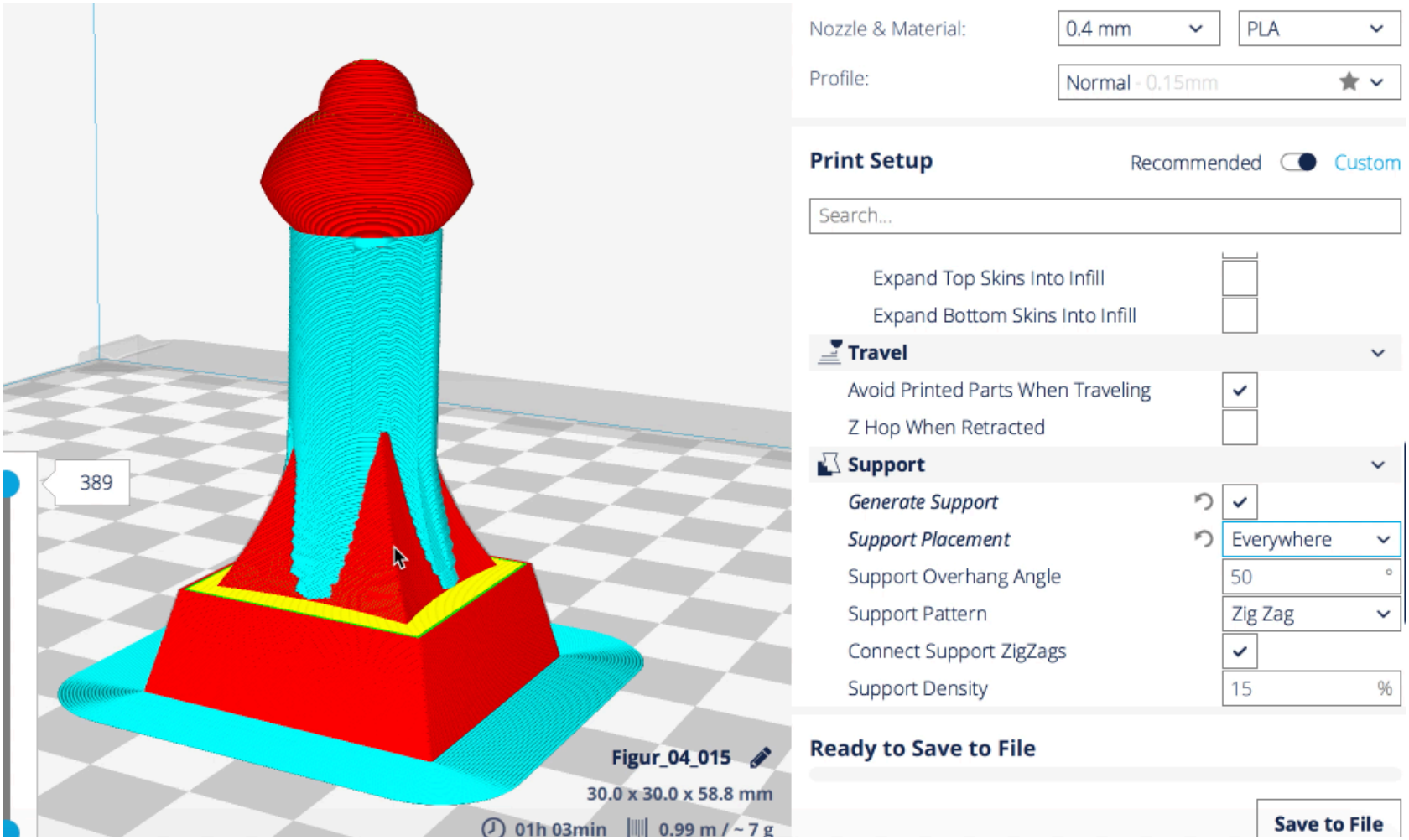
Infill Pattern



Infill Density

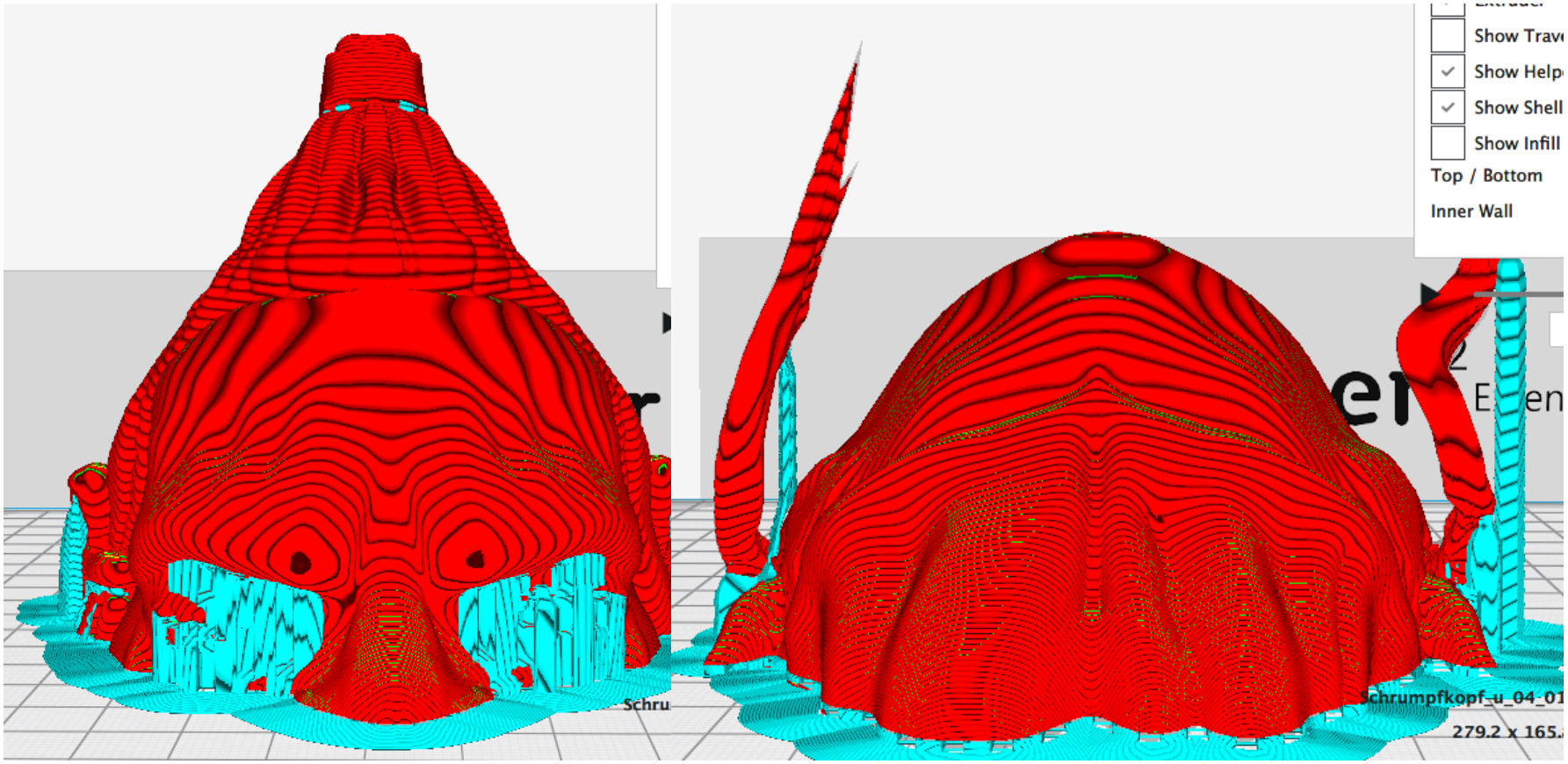


Support

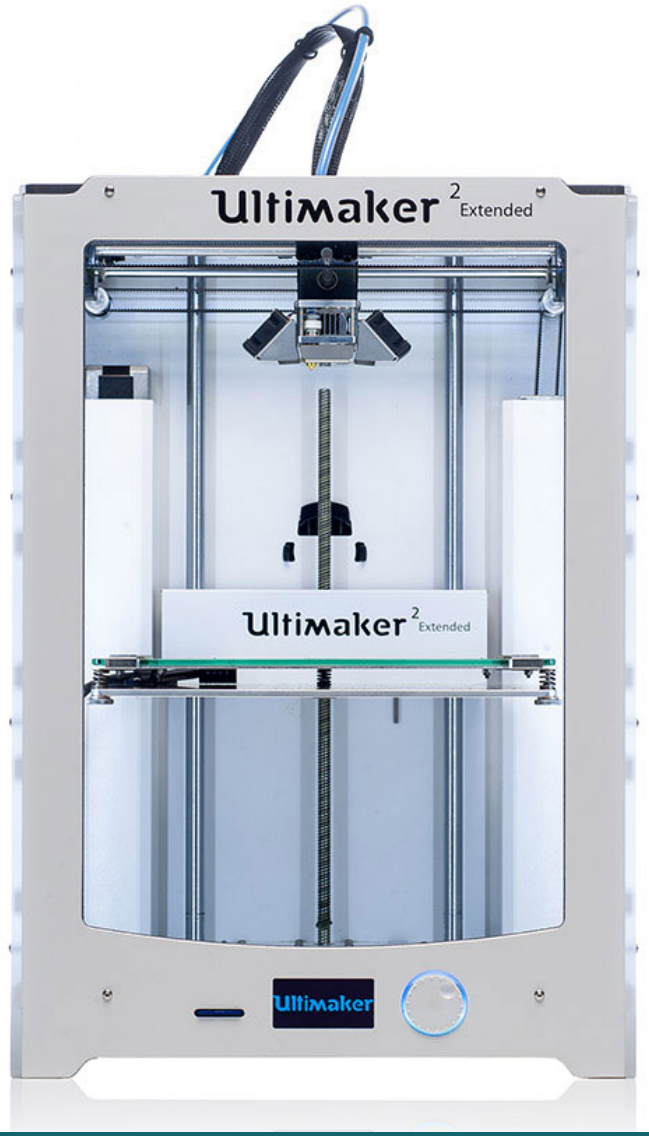

Nozzle & Material:
0.4 mm
PLA
Profile:
Normal - 0.15mm
Print Setup
Recommended
Custom
Search...
Expand Top Skins Into Infill
Expand Bottom Skins Into Infill
Travel
Avoid Printed Parts When Traveling
Z Hop When Retracted
Support
Generate Support
Support Placement
Everywhere
Support Overhang Angle
50
Support Pattern
Zig Zag
Connect Support ZigZags
Support Density
15
%
Ready to Save to File
Save to File
389
Figur_04_015
30.0 x 30.0 x 58.8 mm
01h 03min
0.99 m / ~ 7 g

UNIVERSITÄT SALZBURG

Support



SD → 3DPrint



Video: 3DP

Danke für die Aufmerksamkeit