Prozedurales 3D-Design mit OpenSCAD und Python-SDF

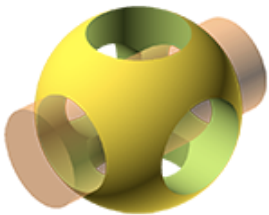
Yann Büchau

Uni Tübingen, Umweltphysik

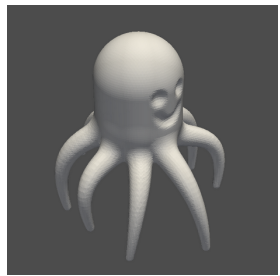
Tübix 2023 – Tübingen

Vorträge – 12:30-12:50 – V4/A302

01.07.2023



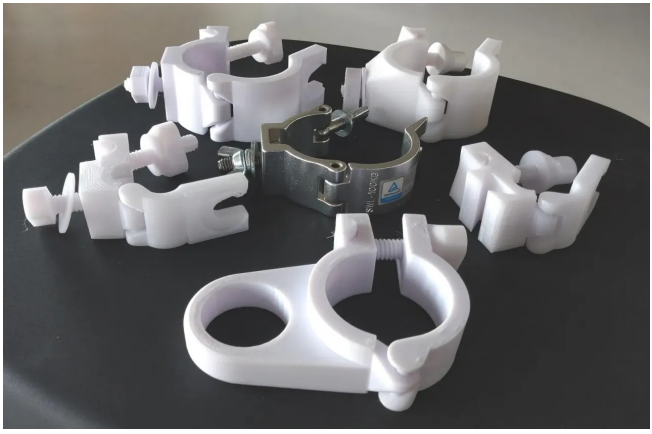
openscad.org



gitlab.com/nobodyinperson/sdf

Warum 3D-Design?

- 3D-Druck
- CNC-Fräsen
- Reparaturen
- Unkaufbares (Adapter, etc.)
- **Vor allem: Anpassbarkeit**
- Inspiration: Printables.com



https://www.printables.com/model/189670-customizable-clamp

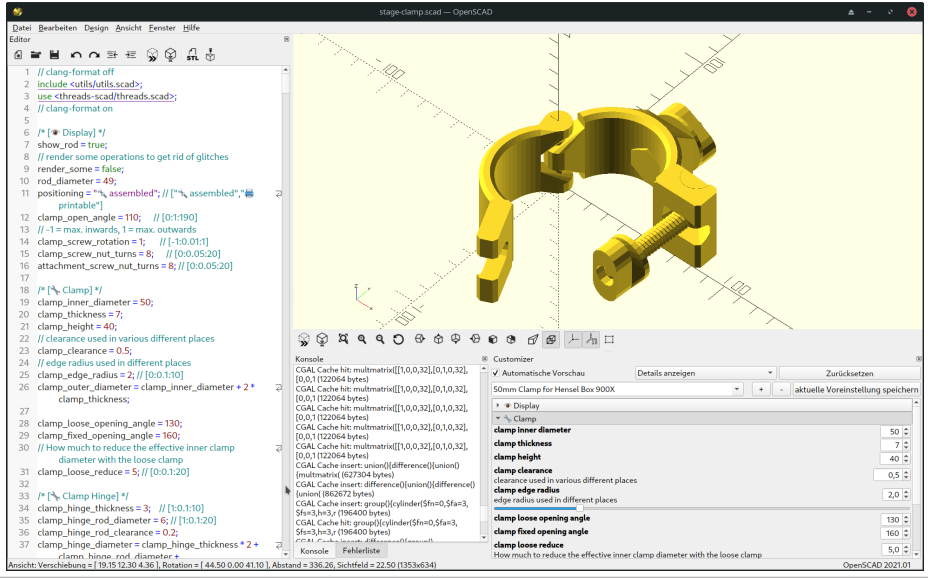
Warum Code für 3D-Design?

- `git`/Git Annex trackbar
  - Komplette Versionshistorie / Backup
  - Synchronisation zwischen Maschinen
  - Kollaboration mit Anderen
  - Klare Verhältnisse, wer was beigetragen hat
  - Branching / Versuche verwerfen etc.
- sehr wenig Speicherplatz für ein Design
- Beliebige Design-interne Logik möglich
- Inherent parametrisierbar

1 stage-clamp.scad

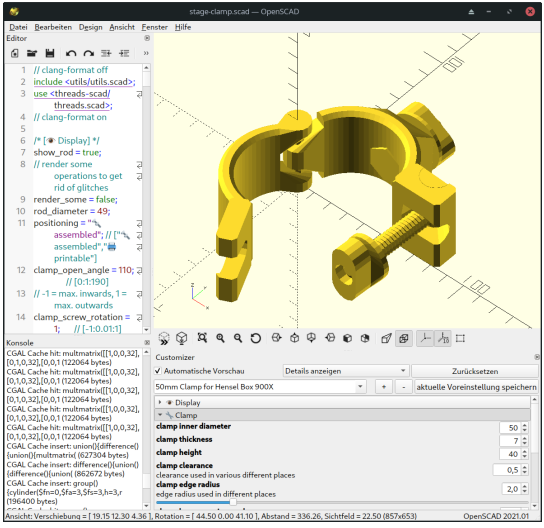
```
46
45 /// 🔧 The fixed part ///
44 render_if(render_some) difference()
43 {
42  union()
41  {
40    render_if(attachment_type == "🔺plug", convexity = 5) difference()
39    {
38      union()
37      {
36        translate([ clamp_outer_diameter / 2, 0, 0 ]) translate(clamp_offset)
35          pipe_part(angle = clamp_fixed_opening_angle);
34
33        ////////////////////
32        /// ➕ Attachment ///
31        ////////////////////
30        if (attachment_type != "🚫none" && attachment_type != "🐊croc") {
29          // transition to attachment
28          difference()
27          {
26            // „transition“ including the complete attachment shape and
25            // filling part of the clamp inners
24            intersection()
23            {
22              hull()
21              {
20                move_to_attachment_position() fixed_part_attachment();
19                translate([
18                  clamp_outer_diameter / 2,
```

OpenSCAD

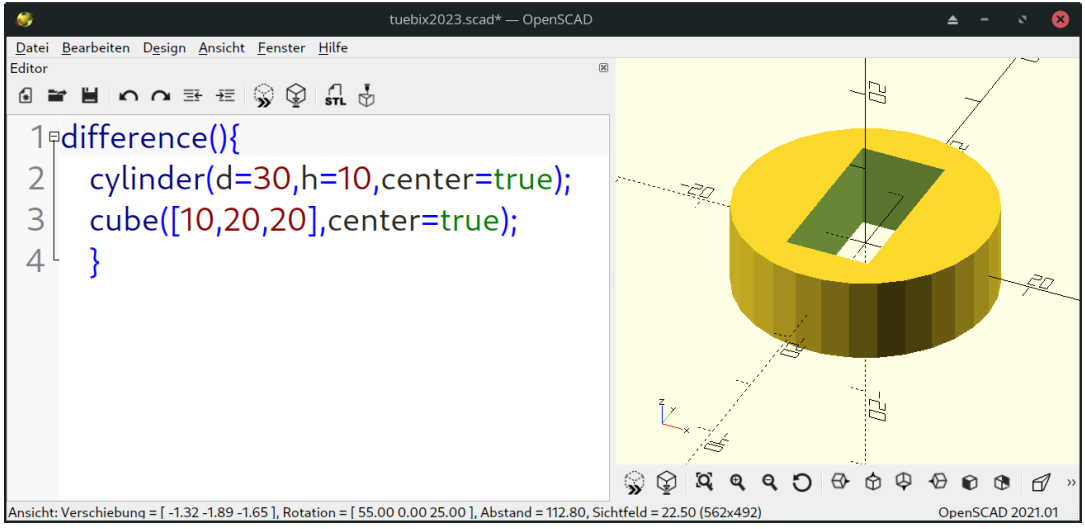

stage-clamp.scad — OpenSCAD
Datei Bearbeiten Design Ansicht Fenster Hilfe
Editor
// clang-format off
include <utils/utils.scad>;
use <threads-scad/threads.scad>;
// clang-format on
/* [Display] */
show_rod = true;
// render some operations to get rid of glitches
render_some = false;
rod_diameter = 49;
positioning = "assembled"; // ["assembled","printable"]
clamp_open_angle = 110; // [0:1:190]
// -1 = max. inwards, 1 = max. outwards
clamp_screw_rotation = 1; // [-1:0.01:1]
clamp_screw_nut_turns = 8; // [0:0.05:20]
attachment_screw_nut_turns = 8; // [0:0.05:20]
/* [Clamp] */
clamp_inner_diameter = 50;
clamp_thickness = 7;
clamp_height = 40;
// clearance used in various different places
clamp_clearance = 0.5;
// edge radius used in different places
clamp_edge_radius = 2; // [0:0.1:10]
clamp_outer_diameter = clamp_inner_diameter + 2 * clamp_thickness;
clamp_loose_opening_angle = 130;
clamp_fixed_opening_angle = 160;
// How much to reduce the effective inner clamp diameter with the loose clamp
clamp_loose_reduce = 5; // [0:0.1:20]
/* [Clamp Hinge] */
clamp_hinge_thickness = 3; // [1:0.1:10]
clamp_hinge_rod_diameter = 6; // [1:0.1:20]
clamp_hinge_rod_clearance = 0.2;
clamp_hinge_diameter = clamp_hinge_thickness * 2 + clamp_hinge_rod_diameter +
Konsole
CGAL Cache hit: multimatrix([[1,0,0,32],[0,1,0,32],[0,0,1,122064 bytes)
CGAL Cache hit: multimatrix([[1,0,0,32],[0,1,0,32],[0,0,1,122064 bytes)
CGAL Cache hit: multimatrix([[1,0,0,32],[0,1,0,32],[0,0,1,122064 bytes)
CGAL Cache hit: multimatrix([[1,0,0,32],[0,1,0,32],[0,0,1,122064 bytes)
CGAL Cache hit: multimatrix([[1,0,0,32],[0,1,0,32],[0,0,1,122064 bytes)
CGAL Cache insert: union(){difference(){union(){multimatrix((627304 bytes)
CGAL Cache insert: difference(){union(){difference(){union((862672 bytes)
CGAL Cache insert: group(){cylinder($fn=0,$fa=3,$fs=3,h=3,r (196400 bytes)
CGAL Cache hit: group(){cylinder($fn=0,$fa=3,$fs=3,h=3,r (196400 bytes)
Konsole
Fehlerliste
Customizer
Automatische Vorschau
Details anzeigen
Zurücksetzen
50mm Clamp for Hensel Box 900X
aktuelle Voreinstellung speichern
Display
Clamp
clamp inner diameter 50
clamp thickness 7
clamp height 40
clamp clearance 0,5
clearance used in various different places
clamp edge radius 2,0
edge radius used in different places
clamp loose opening angle 130
clamp fixed opening angle 160
clamp loose reduce 5,0
How much to reduce the effective inner clamp diameter with the loose clamp
Ansicht: Verschiebung = [19.15 12.30 4.36], Rotation = [44.50 0.00 41.10], Abstand = 336.26, Sichtfeld = 22.50 (1353x634)
OpenSCAD 2021.01

OpenSCAD

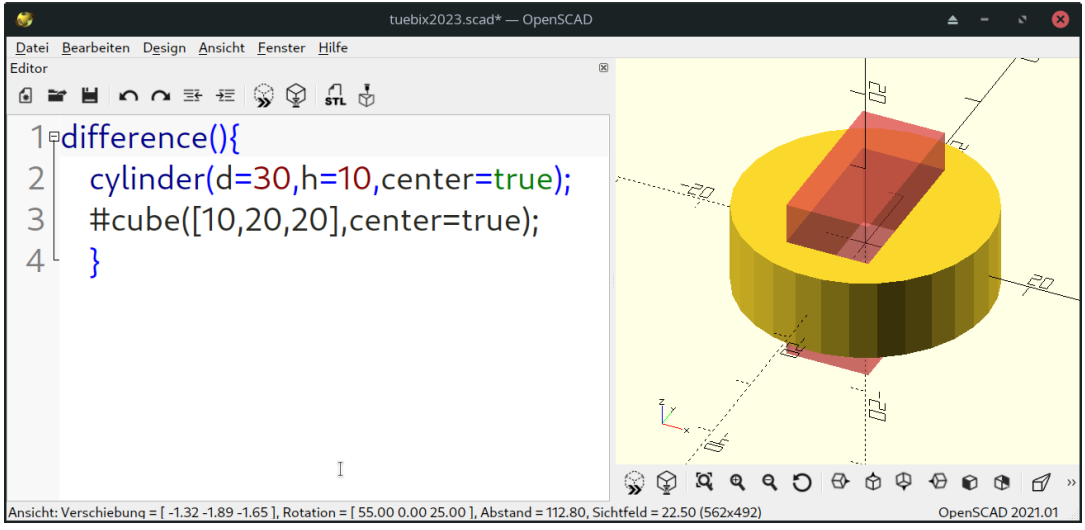
- mit die bekannteste Software für parametrisches 3D-Design
- C++, unkritische Abhängigkeiten, läuft überall
- All-in-one GUI mit Editor, Renderer und Customizer
- Headless Bedienung vom Terminal möglich
- Funktionale eigene Sprache (etwas merkwürdig)
- Produziert STL-Dateien (Meshes, immer diskret)
- Auch durch den Thingiverse.com *Customizer* bekannt geworden



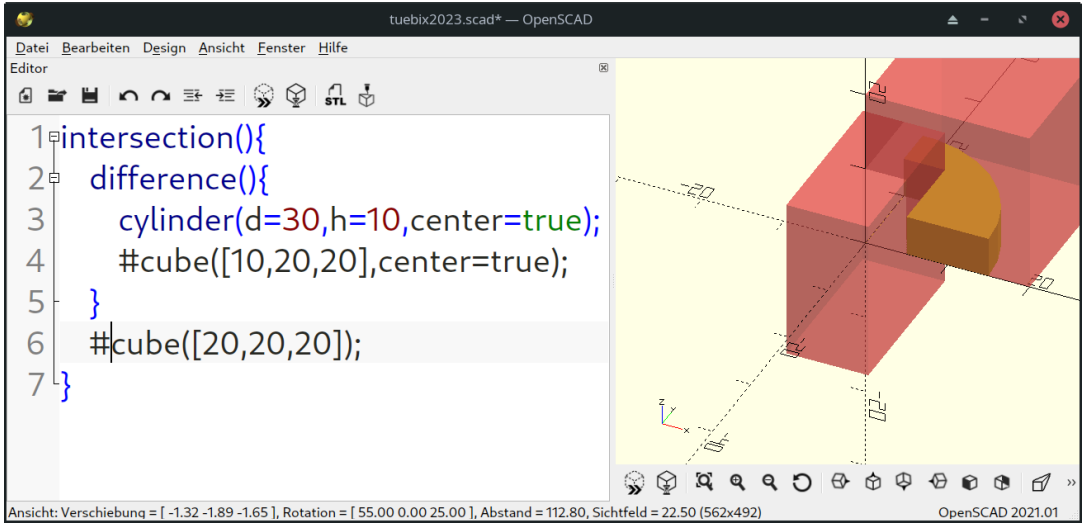
OpenSCAD Basics: Löcher machen



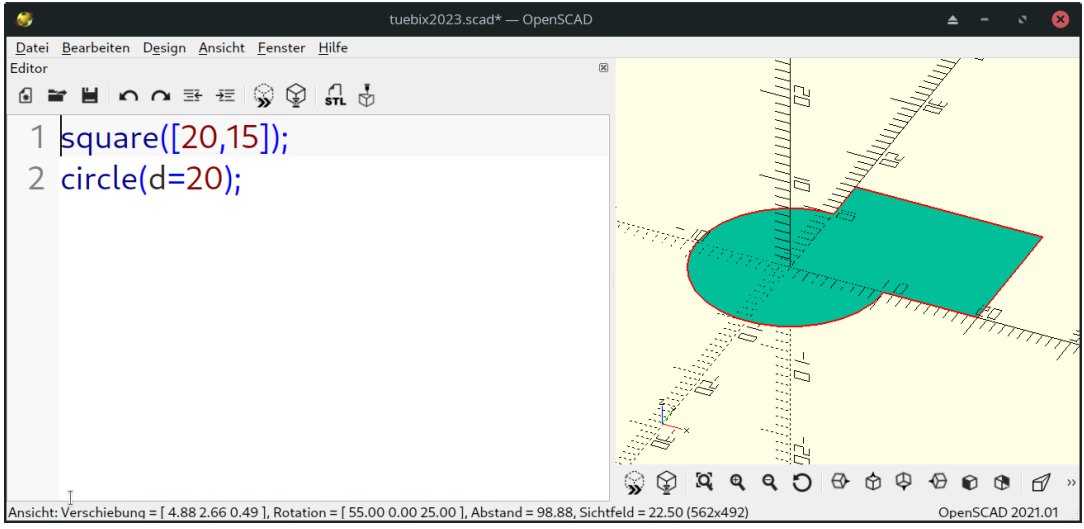
OpenSCAD Basics: Debuggen



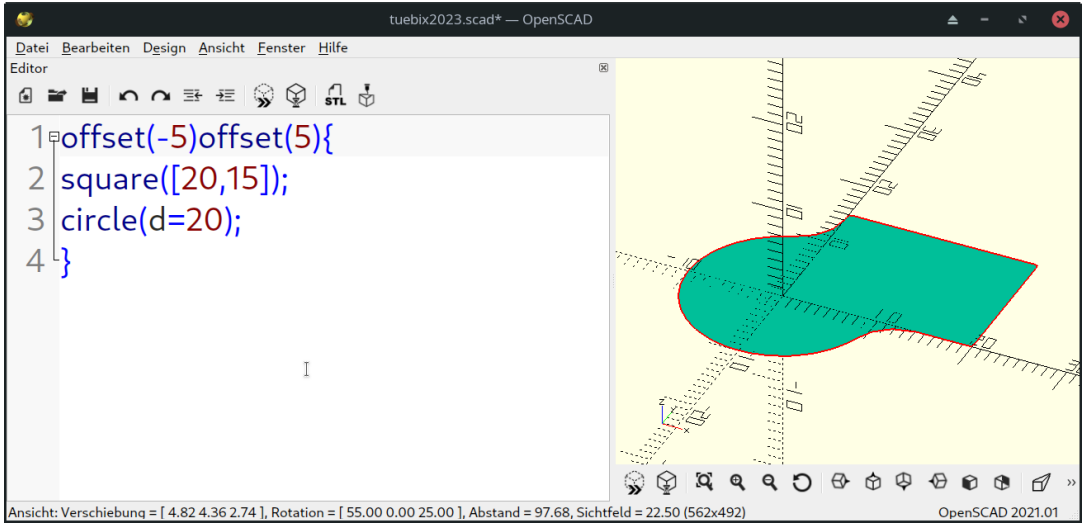
OpenSCAD Basics: Intersection



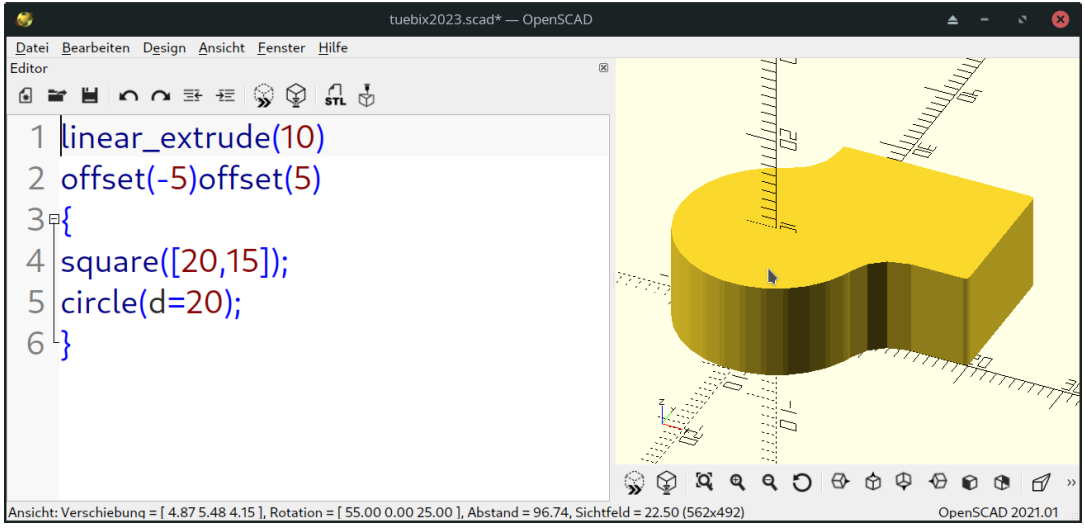
OpenSCAD Basics: 2D-Objekte



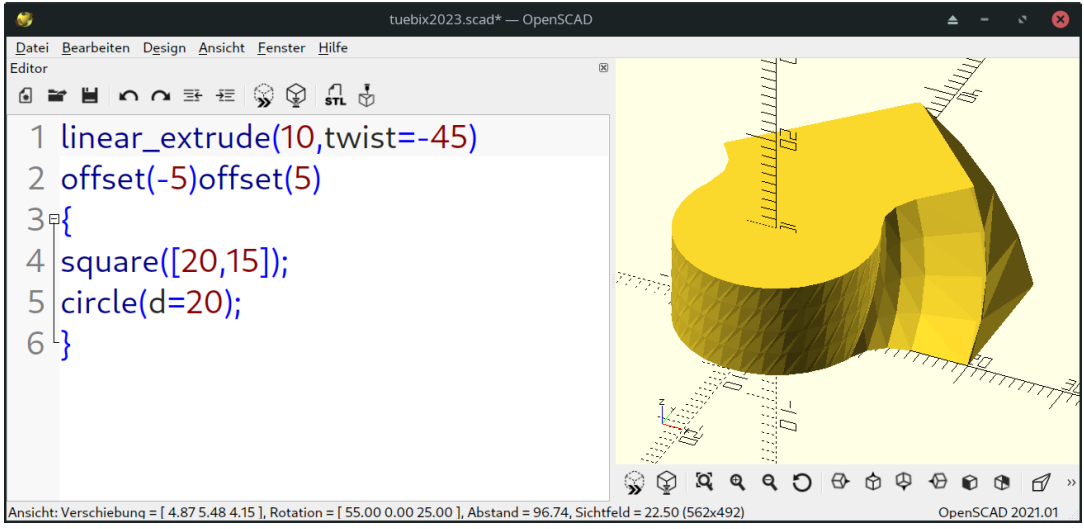
OpenSCAD Basics: 2D-Objekte Abrunden



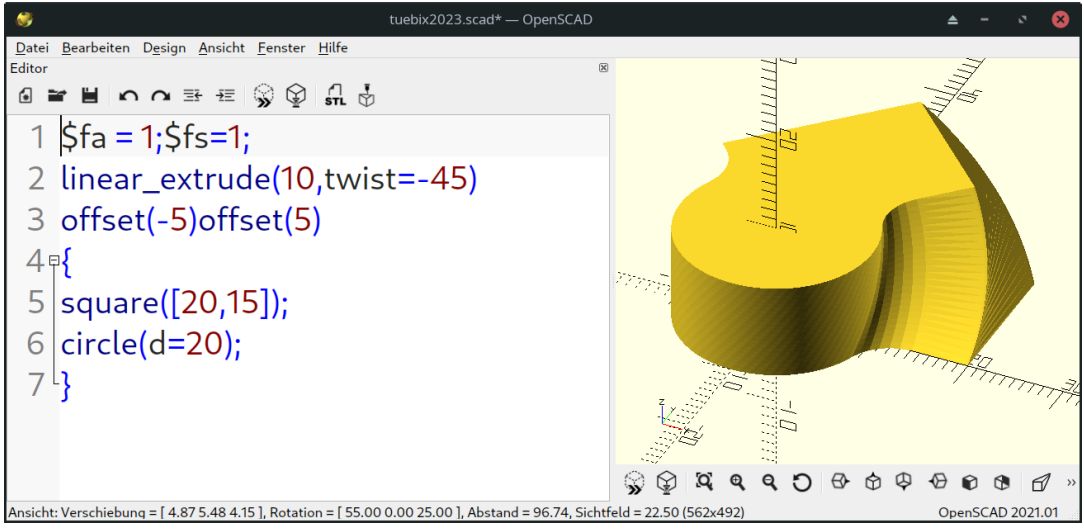
OpenSCAD Basics: 2D-Objekte Extrudieren



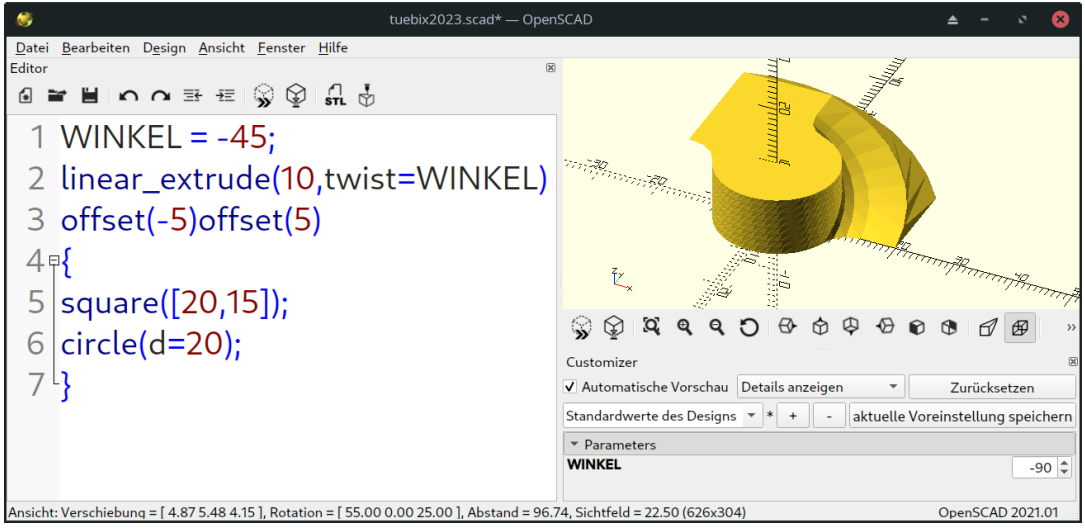
OpenSCAD Basics: 2D-Objekte Extrudieren und Drehen



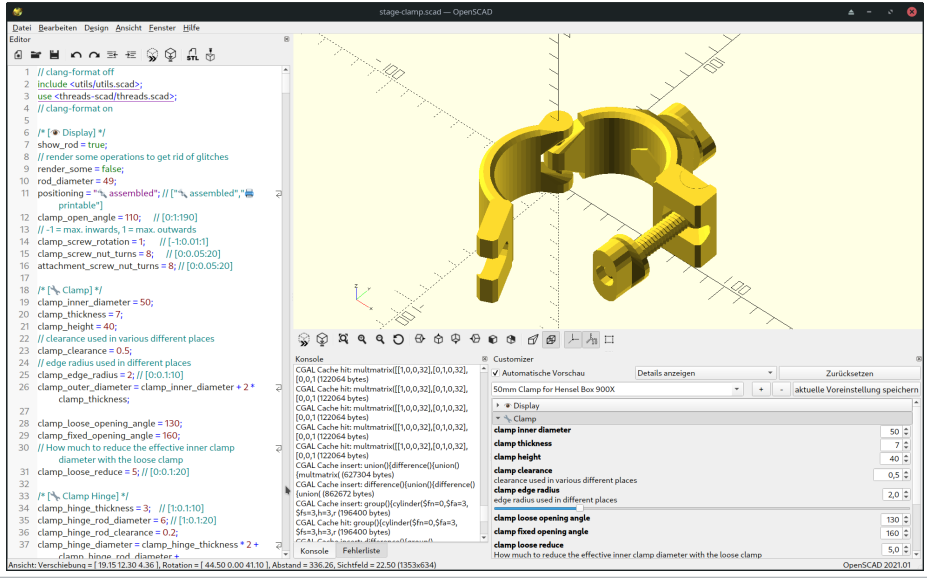
OpenSCAD Basics: Auflösung



OpenSCAD Basics: Parametrisieren



OpenSCAD: Mit einfachen Operationen komplexe Designs machen



OpenSCAD: Stärken und Schwächen

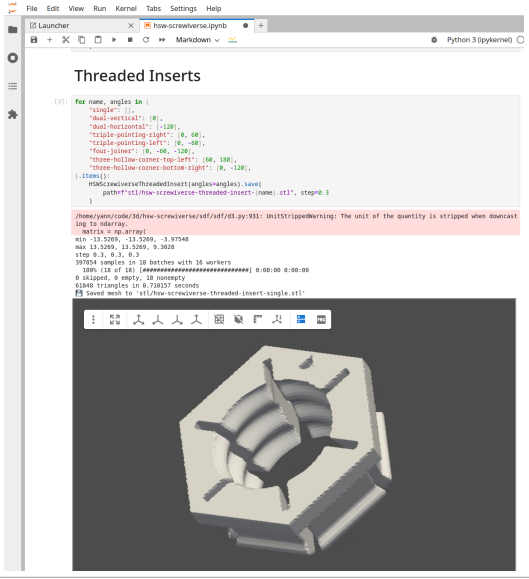
Stark

- recht **effiziente Meshes**, immer **präzise**
- All-in-One **GUI**, eigener Editor möglich
- große Community, viel Material und Libraries

Schwach

- merkwürdige Sprache
- **Rundungen** äußerst schwierig
- neue Features kaum selber hinzuzufügen
- nur plumpe Meshes ohne Farbinfo etc.
- Rendern sehr langsam bei größeren Designs, weil single-core (aber Update kommt!)

- Ursprünglich von Michael Fogleman auf Basis von Inigo Quilez' SDF-Material entwickelt: fogleman/sdf
- Mir gefiel die Idee dahinter so gut, also Fork: nobodyinperson/sdf
- Python, kann in Jupyter oder Konsole verwendet werden



SDF: Signed Distance Function

- Idee: Jedes Objekt ist eine mathematische Funktion (**S**igned **D**istance **F**unction), die für jeden Ort im Raum den **kürzesten Abstand zur Oberfläche** zurückgibt (innen negativ)
- Dadurch ergeben sich interessante Operationen:
  - Minimum(SDF1,SDF2) = Vereinigung
  - Maximum(SDF1,SDF2) = Überschneidung
  - Vorzeichen umdrehen = Inverse Form
  - Überschneidung mit Inverser Form = Differenz
- Raumtransformationen for dem Auswerten der SDF sorgen für Rotation, Verschiebung, Verdrehung, Dehnung, etc.
- Code für eine Kugel:

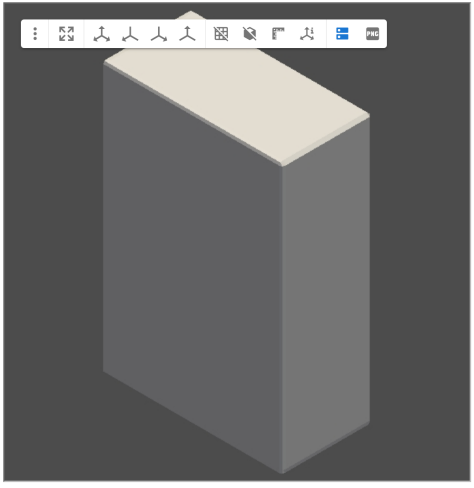
```
@sdf3
def sphere(radius,center=ORIGIN):
    def f(p):
        return _length(p - center) - radius
    return f
```

- Außerdem möglich: unendliche Objekte!

Python-SDF Basics

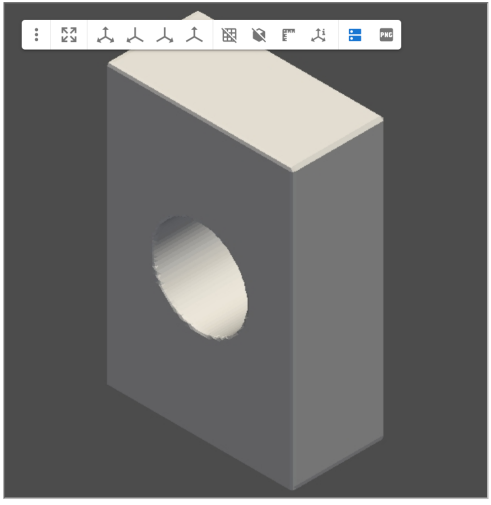
[4]:
```
from sdf import *
block = box([10,20,30])
block.save()
```

Saved mesh to 'out.stl'



[6]:
```
block -= cylinder(diameter=10).orient(X)
block.save()
```

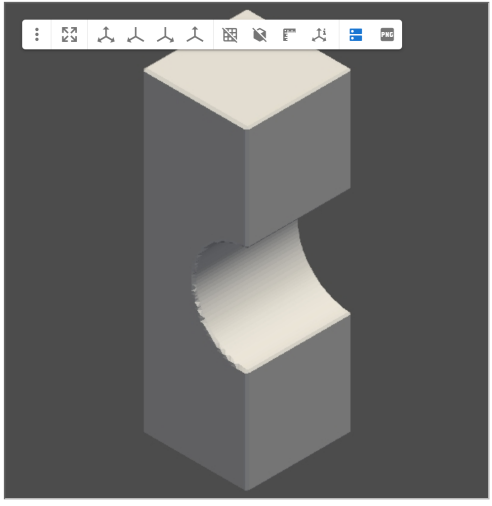
Saved mesh to 'out.stl'



Python-SDF: Operationen „rund“ durchführen mit k

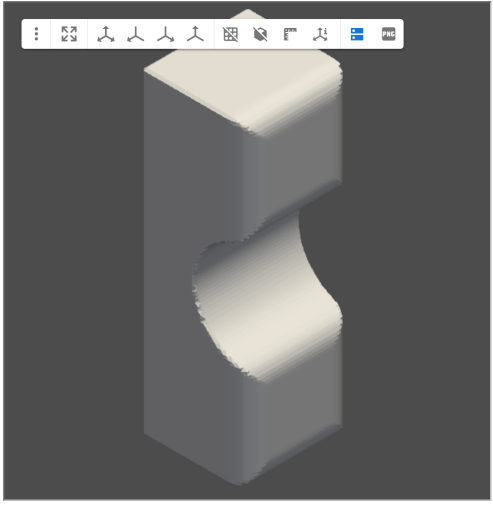
[13]:
```
(block & slab(y1=0)).save()
```

```
Saved mesh to 'out.stl'
```



[15]:
```
(block & slab(y1=0).k(2)).save()
```

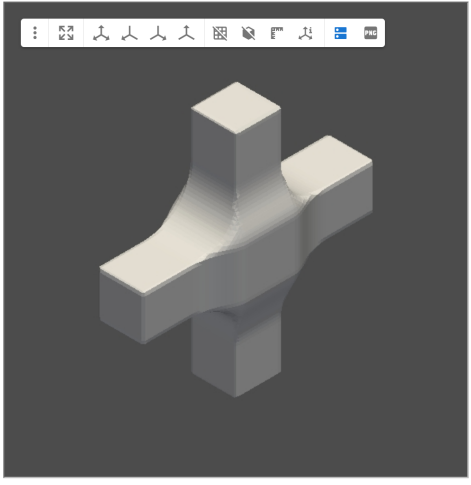
```
Saved mesh to 'out.stl'
```



Python-SDF: Operationen „rund“ durchführen mit k

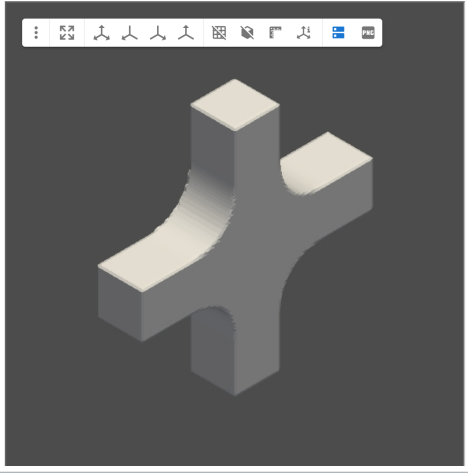
[28]:
```
cross = box([10,10,50])
cross |= cross.orient(X).k(10)
cross.save()
```

Saved mesh to 'out.stl'

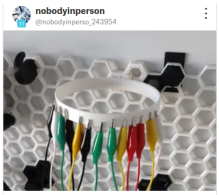


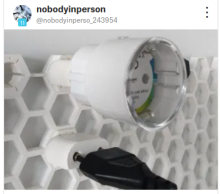
[30]:
```
cross = box([10,10,50])
cross |= cross.orient(X).k(10)
cross &= slab(dy=10)
cross.save()
```

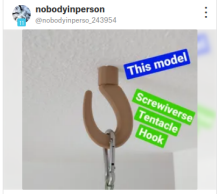
Saved mesh to 'out.stl'

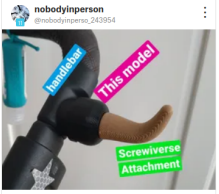


Python-SDF: Mit Python 3D-Objekte Modellieren









Python-SDF: Stärken und Schwächen

Stark

- Es ist Python!
- „einfache“ Mathematik dahinter
- neue Features selber hinzufügbar
- Rundungen einfach möglich
- multi-core

Schwach

- **globale Mesh-Auflösung**
  (momentan: Marching-Cubes)
  - große Designs mit kleinen Details produzieren **gigantische** Meshes
  - **Ecken nie präzise**
- kein Debugging/Hervorheben von Operationen
- auch nur plumpe Meshes ohne Farbinfo etc.
- kennt niemand. Keine Community.

Kontakt

Yann Büchau

GUZ, Uni Tübingen, Schnarrenbergstr. 94-96, 72076 Tübingen

+49 7071 29 74777

nobodyinperson@posteo.de

nobodyinperson.de

fosstodon.org/@nobodyinperson

@nobodyinperson