

Prozedurales 3D-Design mit OpenSCAD und Python-SDF

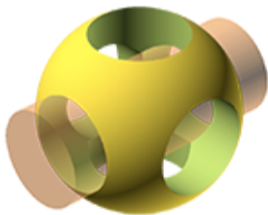
Yann Büchau

Uni Tübingen, Umweltphysik

Tübix 2023 – Tübingen


Vorträge – 12:30-12:50 – V4/A302

01.07.2023 



 openscad.org



 gitlab.com/nobodyinperson/sdf



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



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



- ▶ 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,
```



stage-clamp.scad — OpenSCAD

Editor

```

1 // clang-format off
2 include <utils/utils.scad>;
3 use <threads-scad/threads.scad>;
4 // clang-format on
5
6 /* [Display] */
7 show_rod = true;
8 // render some operations to get rid of glitches
9 render_some = false;
10 rod_diameter = 49;
11 positioning = "\n assembled"; // ["\n assembled", "\n
    printable"]
12 clamp_open_angle = 110; // [0:1:190]
13 // -1 = max. inwards, 1 = max. outwards
14 clamp_screw_rotation = 1; // [-1:0.01:1]
15 clamp_screw_nut_turns = 8; // [0:0.05:20]
16 attachment_screw_nut_turns = 8; // [0:0.05:20]
17
18 /* [Clamp] */
19 clamp_inner_diameter = 50;
20 clamp_thickness = 7;
21 clamp_height = 40;
22 // clearance used in various different places
23 clamp_clearance = 0.5;
24 // edge radius used in different places
25 clamp_edge_radius = 2; // [0:0.1:10]
26 clamp_outer_diameter = clamp_inner_diameter + 2 *
    clamp_thickness;
27
28 clamp_loose_opening_angle = 130;
29 clamp_fixed_opening_angle = 160;
30 // How much to reduce the effective inner clamp
    diameter with the loose clamp
31 clamp_loose_reduce = 5; // [0:0.1:20]
32
33 /* [Clamp Hinge] */
34 clamp_hinge_thickness = 3; // [1:0.1:10]
35 clamp_hinge_rod_diameter = 6; // [1:0.1:20]
36 clamp_hinge_rod_clearance = 0.2;
37 clamp_hinge_diameter = clamp_hinge_thickness * 2 +
    clamp_hinge_rod_diameter +
  
```

Konsole

```

CGAL Cache hit: multimatric[[[1,0,0,32],[0,1,0,32],
[0,0,1,122064 bytes)
CGAL Cache hit: multimatric[[[1,0,0,32],[0,1,0,32],
[0,0,1,122064 bytes)
CGAL Cache hit: multimatric[[[1,0,0,32],[0,1,0,32],
[0,0,1,122064 bytes)
CGAL Cache hit: multimatric[[[1,0,0,32],[0,1,0,32],
[0,0,1,122064 bytes)
CGAL Cache hit: multimatric[[[1,0,0,32],[0,1,0,32],
[0,0,1,122064 bytes)
CGAL Cache insert: union(){difference(){union()
(multimatric( 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)
CGAL Cache insert: difference(){group(){
  
```

Customizer

Automatische Vorschau Details anzeigen Zurücksetzen

50mm Clamp for Hensel Box 900X aktuelle Voreinstellung speichern

► Display

► Clamp

CGAL Cache hit: multimatric[[[1,0,0,32],[0,1,0,32],
[0,0,1,122064 bytes)

CGAL Cache hit: multimatric[[[1,0,0,32],[0,1,0,32],
[0,0,1,122064 bytes)

CGAL Cache hit: multimatric[[[1,0,0,32],[0,1,0,32],
[0,0,1,122064 bytes)

CGAL Cache insert: union(){difference(){union()
(multimatric(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)

CGAL Cache insert: difference(){group(){

cl^{amp} inner diameter 50

cl^{amp} thickness 7

cl^{amp} height 40

cl^{amp} clearance clearance used in various different places 0,5

cl^{amp} edge radius edge radius used in different places 2,0

cl^{amp} loose opening angle 130

cl^{amp} fixed opening angle 160

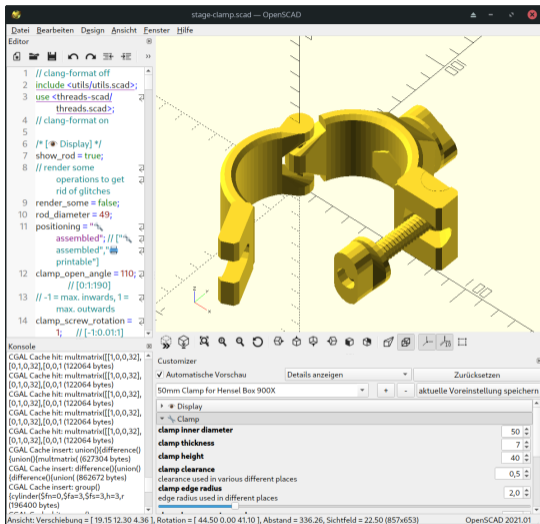
cl^{amp} loose reduce How much to reduce the effective inner clamp diameter with the loose clamp 5,0

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



- ▶ 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





tuebix2023.scad* — OpenSCAD

Datei Bearbeiten Design Ansicht Fenster Hilfe

Editor

```
1 difference(){
2   cylinder(d=30,h=10,center=true);
3   cube([10,20,20],center=true);
4 }
```

Ansicht: Verschiebung = [-1.32 -1.89 -1.65], Rotation = [55.00 0.00 25.00], Abstand = 112.80, Sichtfeld = 22.50 (562x492)

OpenSCAD 2021.01



tuebix2023.scad* — OpenSCAD

Datei Bearbeiten Design Ansicht Fenster Hilfe

Editor

```
1 difference(){
2   cylinder(d=30,h=10,center=true);
3   #cube([10,20,20],center=true);
4 }
```

Ansicht: Verschiebung = [-1.32 -1.89 -1.65], Rotation = [55.00 0.00 25.00], Abstand = 112.80, Sichtfeld = 22.50 (562x492)

OpenSCAD 2021.01



The screenshot shows the OpenSCAD interface. On the left is the code editor with the following code:

```
1 intersection(){
2   difference(){
3     cylinder(d=30,h=10,center=true);
4     #cube([10,20,20],center=true);
5   }
6   #cube([20,20,20]);
7 }
```

On the right is the 3D view showing a red cube with a smaller orange cube inside it. The orange cube is a 10x20x20 cube centered within a larger 20x20x20 cube. The red cube is a 30x20x20 cube centered on the origin. The 3D view includes a coordinate system (x, y, z) and various tool icons at the bottom.

Ansicht: Verschiebung = [-1.32 -1.89 -1.65], Rotation = [55.00 0.00 25.00], Abstand = 112.80, Sichtfeld = 22.50 (562x492) OpenSCAD 2021.01



tuebix2023.scad* — OpenSCAD

Datei Bearbeiten Design Ansicht Fenster Hilfe

Editor

```
1 square([20,15]);  
2 circle(d=20);
```

Ansicht: Verschiebung = [4.88 2.66 0.49], Rotation = [55.00 0.00 25.00], Abstand = 98.88, Sichtfeld = 22.50 (562x492)

OpenSCAD 2021.01



tuebix2023.scad* — OpenSCAD

Datei Bearbeiten Design Ansicht Fenster Hilfe

Editor

```
1 offset(-5)offset(5){  
2 square([20,15]);  
3 circle(d=20);  
4 }
```

Ansicht: Verschiebung = [4.82 4.36 2.74], Rotation = [55.00 0.00 25.00], Abstand = 97.68, Sichtfeld = 22.50 (562x492)

OpenSCAD 2021.01



tuebix2023.scad* — OpenSCAD

Datei Bearbeiten Design Ansicht Fenster Hilfe

Editor

```
1 linear_extrude(10)
2 offset(-5)offset(5)
3 {
4 square([20,15]);
5 circle(d=20);
6 }
```

Ansicht: Verschiebung = [4.87 5.48 4.15], Rotation = [55.00 0.00 25.00], Abstand = 96.74, Sichtfeld = 22.50 (562x492)

OpenSCAD 2021.01



tuebix2023.scad* — OpenSCAD

Datei Bearbeiten Design Ansicht Fenster Hilfe

Editor

```
1 linear_extrude(10,twist=-45)
2 offset(-5)offset(5)
3 {
4 square([20,15]);
5 circle(d=20);
6 }
```

Ansicht: Verschiebung = [4.87 5.48 4.15], Rotation = [55.00 0.00 25.00], Abstand = 96.74, Sichtfeld = 22.50 (562x492)

OpenSCAD 2021.01



tuebix2023.scad* — OpenSCAD

Datei Bearbeiten Design Ansicht Fenster Hilfe

Editor

```
1 $fa = 1;$fs=1;
2 linear_extrude(10,twist=-45)
3 offset(-5)offset(5)
4 {
5 square([20,15]);
6 circle(d=20);
7 }
```

Ansicht: Verschiebung = [4.87 5.48 4.15], Rotation = [55.00 0.00 25.00], Abstand = 96.74, Sichtfeld = 22.50 (562x492)

OpenSCAD 2021.01



tuebix2023.scad* — OpenSCAD

Datei Bearbeiten Design Ansicht Fenster Hilfe

Editor

```
1 WINKEL = -45;  
2 linear_extrude(10,twist=WINKEL)  
3 offset(-5)offset(5)  
4 {  
5 square([20,15]);  
6 circle(d=20);  
7 }
```

Customizer

Automatische Vorschau Details anzeigen Zurücksetzen

Standardwerte des Designs * + - aktuelle Voreinstellung speichern

Parameters

WINKEL -90

Ansicht: Verschiebung = [4.87 5.48 4.15], Rotation = [55.00 0.00 25.00], Abstand = 96.74, Sichtfeld = 22.50 (626x304) OpenSCAD 2021.01



```

1 // clang-format off
2 include <utils/utils.scad>;
3 use <threads-scad/threads.scad>;
4 // clang-format on
5
6 /* [Display] */
7 show_rod = true;
8 // render some operations to get rid of glitches
9 render_some = false;
10 rod_diameter = 49;
11 positioning = "\n assembled"; // ["\n assembled", "\n
    printable"]
12 clamp_open_angle = 110; // [0:1:190]
13 // -1 = max. inwards, 1 = max. outwards
14 clamp_screw_rotation = 1; // [-1:0.01:1]
15 clamp_screw_nut_turns = 8; // [0:0.05:20]
16 attachment_screw_nut_turns = 8; // [0:0.05:20]
17
18 /* [Clamp] */
19 clamp_inner_diameter = 50;
20 clamp_thickness = 7;
21 clamp_height = 40;
22 // clearance used in various different places
23 clamp_clearance = 0.5;
24 // edge radius used in different places
25 clamp_edge_radius = 2; // [0:0.1:10]
26 clamp_outer_diameter = clamp_inner_diameter + 2 *
    clamp_thickness;
27
28 clamp_loose_opening_angle = 130;
29 clamp_fixed_opening_angle = 160;
30 // How much to reduce the effective inner clamp
    diameter with the loose clamp
31 clamp_loose_reduce = 5; // [0:0.1:20]
32
33 /* [Clamp Hinge] */
34 clamp_hinge_thickness = 3; // [1:0.1:10]
35 clamp_hinge_rod_diameter = 6; // [1:0.1:20]
36 clamp_hinge_rod_clearance = 0.2;
37 clamp_hinge_diameter = clamp_hinge_thickness * 2 +
    clamp_hinge_rod_diameter +
            
```

Konsole

```

CGAL Cache hit: multimatix[[[1,0,0,32],[0,1,0,32],
[0,0,1] (122064 bytes)
CGAL Cache hit: multimatix[[[1,0,0,32],[0,1,0,32],
[0,0,1] (122064 bytes)
CGAL Cache hit: multimatix[[[1,0,0,32],[0,1,0,32],
[0,0,1] (122064 bytes)
CGAL Cache hit: multimatix[[[1,0,0,32],[0,1,0,32],
[0,0,1] (122064 bytes)
CGAL Cache insert: union(){difference(){union()
(multimatix( (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)
CGAL Cache insert: difference(){union()
            
```

Customizer

Automatische Vorschau Zurücksetzen

50mm Clamp for Hensel Box 900X

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

clamp edge radius: 2.0

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

 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

The screenshot shows a Jupyter Notebook interface with the following content:

Threaded Inserts

```
[3]: for name, angles in {
    "single": [],
    "dual-vertical": [0],
    "dual-horizontal": [-120],
    "triple-pointing-right": [0, 60],
    "triple-pointing-left": [0, -60],
    "four-joiner": [0, -60, -120],
    "three-hollow-corner-top-left": [60, 180],
    "three-hollow-corner-bottom-right": [0, -120],
}.items():
    hsw-screwverseThreadedInsert(angles=angles).save(
        path=f'stl/hsw-screwverse-threaded-insert-{name}.stl', step=0.3
    )
```

Output:

```
/home/yann/code/3d/hsw-screwverse/sdf/sdf/d3.py:931: UnitStrippedWarning: The unit of the quantity is stripped when downcast
ing to ndarray.
matrix = np.array(
min -13.5269, -13.5269, -3.97548
max 13.5269, 13.5269, 9.3028
step 0.3, 0.3, 0.3
397854 samples in 10 batches with 16 workers
100% (18 of 18) [#####] 0:00:00 0:00:00
0 skipped, 0 empty, 18 nonempty
61848 triangles in 0.718157 seconds
Saved mesh to 'stl/hsw-screwverse-threaded-insert-single.stl'
```

The 3D model shows a grey, hexagonal threaded insert with a central hole and six radial slots. The rendering is done in a dark environment with a white background.



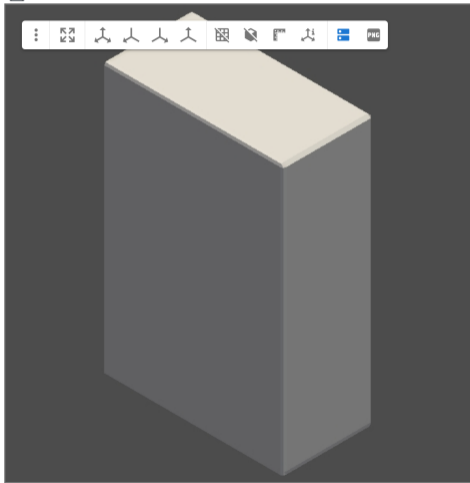
- ▶ Idee: Jedes Objekt ist eine mathematische Funktion (**Signed Distance Function**), die für jeden Ort im Raum den **kürzesten Abstand zur Oberfläche** zurückgibt (innen negativ)
- ▶ Dadurch ergeben sich interessante Operationen:
 - ▶ $\text{Minimum}(\text{SDF1}, \text{SDF2}) = \text{Vereinigung}$
 - ▶ $\text{Maximum}(\text{SDF1}, \text{SDF2}) = \text{Überschneidung}$
 - ▶ Vorzeichen umdrehen = Inverse Form
 - ▶ Überschneidung mit Inverser Form = Differenz
- ▶ Raumtransformationen für dem Auswerten der SDF sorgen für Rotation, Verschiebung, Verdrehung, Dehnung, etc.

```
1 @sdf3
2 def sphere(radius, center=ORIGIN):
3     def f(p):
4         return _length(p - center) - radius
5     return f
6
```

- ▶ Code für eine Kugel:
- ▶ Außerdem möglich: unendliche Objekte!

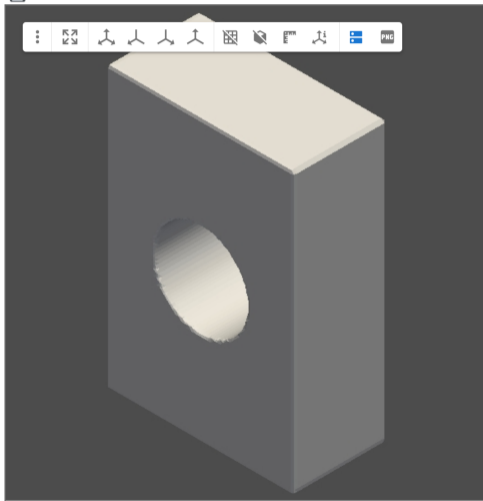


```
[4]: from sdf import *
      block = box([10,20,30])
      block.save()
      Saved mesh to 'out.stl'
```



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

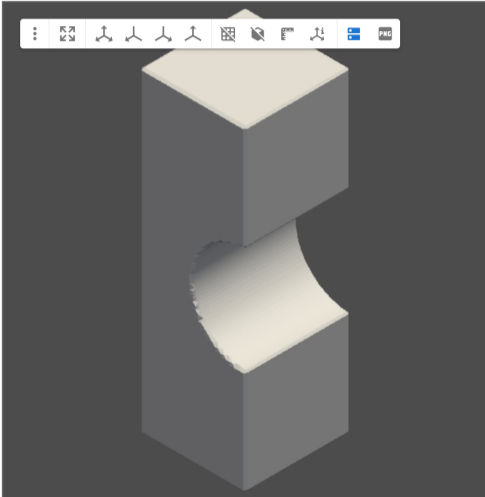
Saved mesh to 'out.stl'





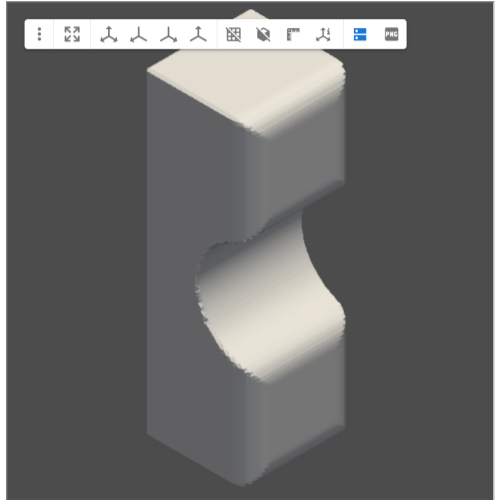
```
[13]: (block & slab(y1=0)).save()
```

Saved mesh to 'out.stl'



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

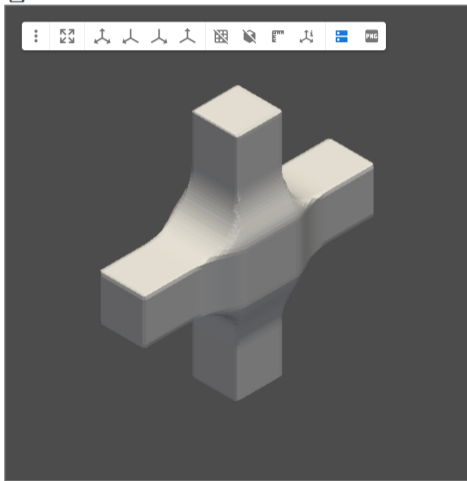
Saved mesh to 'out.stl'





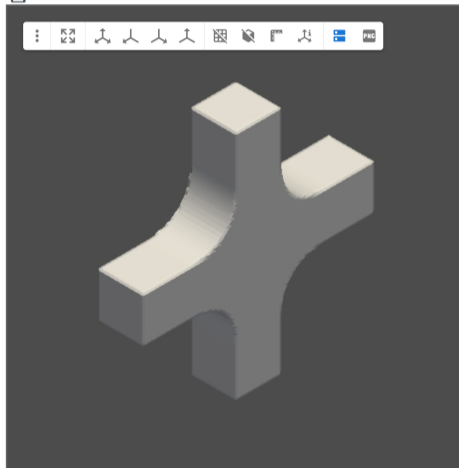
```
[28]: cross = box([10,10,50])
cross |= cross.orient(X).k(10)
cross.save()
```

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'





nobodyinperson
@nobodyinperso_243954

HSW Screwiverse Flat Wall Tape/...
Hobby & Makers > Organizers

30 5 52

nobodyinperson
@nobodyinperso_243954

HSW Screwiverse Rings
Hobby & Makers > Organizers

10 0 15

nobodyinperson
@nobodyinperso_243954

HSW Screwiverse Tentacle Hooks
Hobby & Makers > Organizers

9 0 32

nobodyinperson
@nobodyinperso_243954

HSW Screwiverse EU Power Socket
Hobby & Makers > Organizers

5 0 9

nobodyinperson
@nobodyinperso_243954

Screwiverse Ceiling Socket
Hobby & Makers > Organizers

2 0 3

nobodyinperson
@nobodyinperso_243954

ABC Design Salsa 4 Screwiverse ...
Hobby & Makers > Organizers

2 0 1

 Stark

- ▶ Es ist Python!
- ▶ „einfache“ Mathematik dahinter
- ▶ neue Features selber hinzufügbare
- ▶ 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.



Yann Büchau

- ✉ GUZ, Uni Tübingen, Schnarrenbergstr. 94-96, 72076 Tübingen
- ☎ +49 7071 29 74777
- @ nobodyinperson@posteo.de
- 🌐 nobodyinperson.de
- 📺 fosstodon.org/@nobodyinperson
- 🐙 @nobodyinperson

