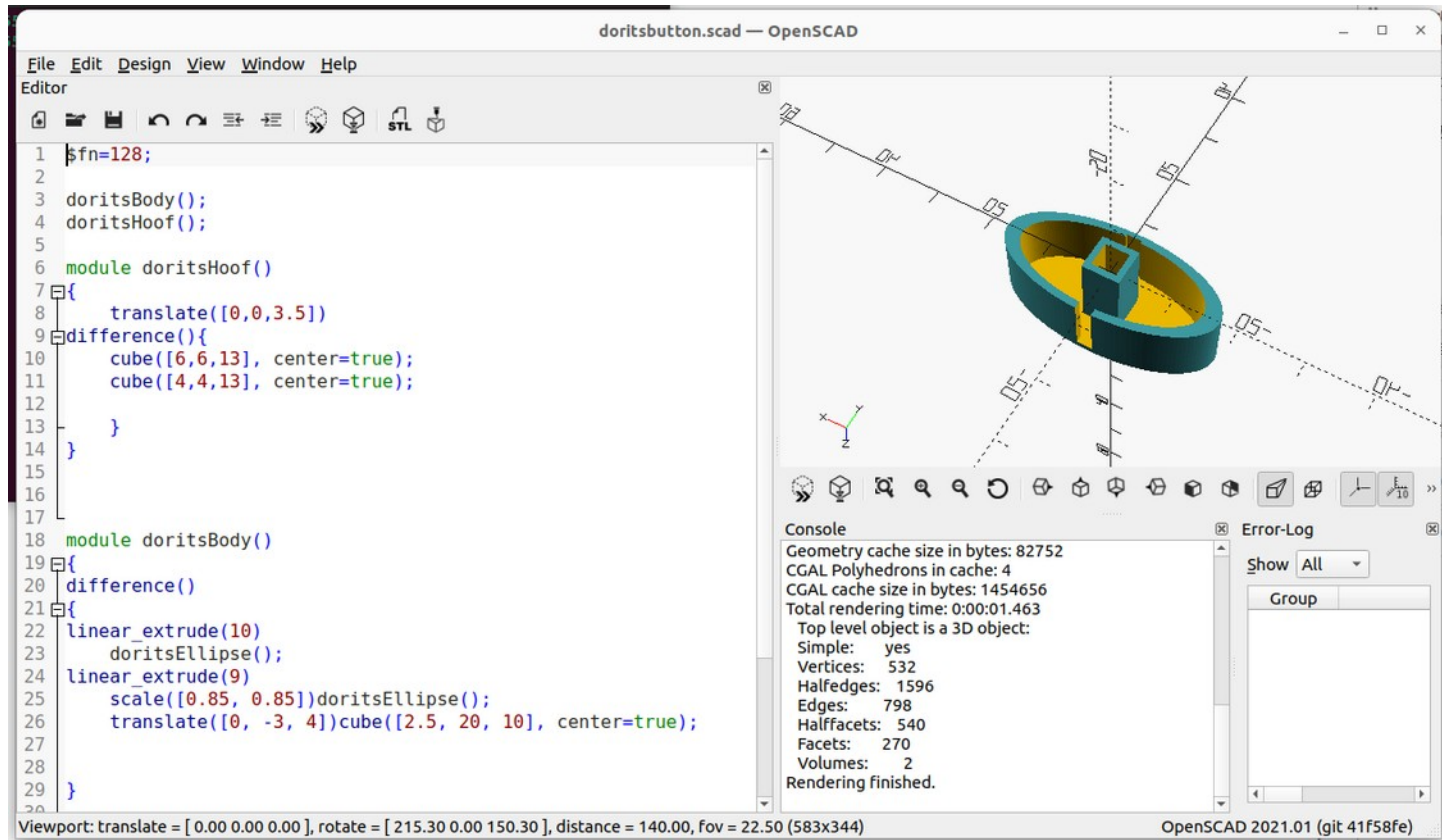


OpenSCAD – or – 3D modeling for the creatively challenged



Some house rules

- This is a workshop
 - If you can have OpenSCAD open, do so
 - <http://www.tamoggemon.com/test/2025/scad.zip>
- ...and more than a workshop
 - You can also just think along
 - (and play along later in your office with a cigar)



About /me

- Tam HANNA
- Tamoggemon Holding k.s.
- Instagram: tam.hanna
- YouTube
<https://www.youtube.com/@MrTamhan>



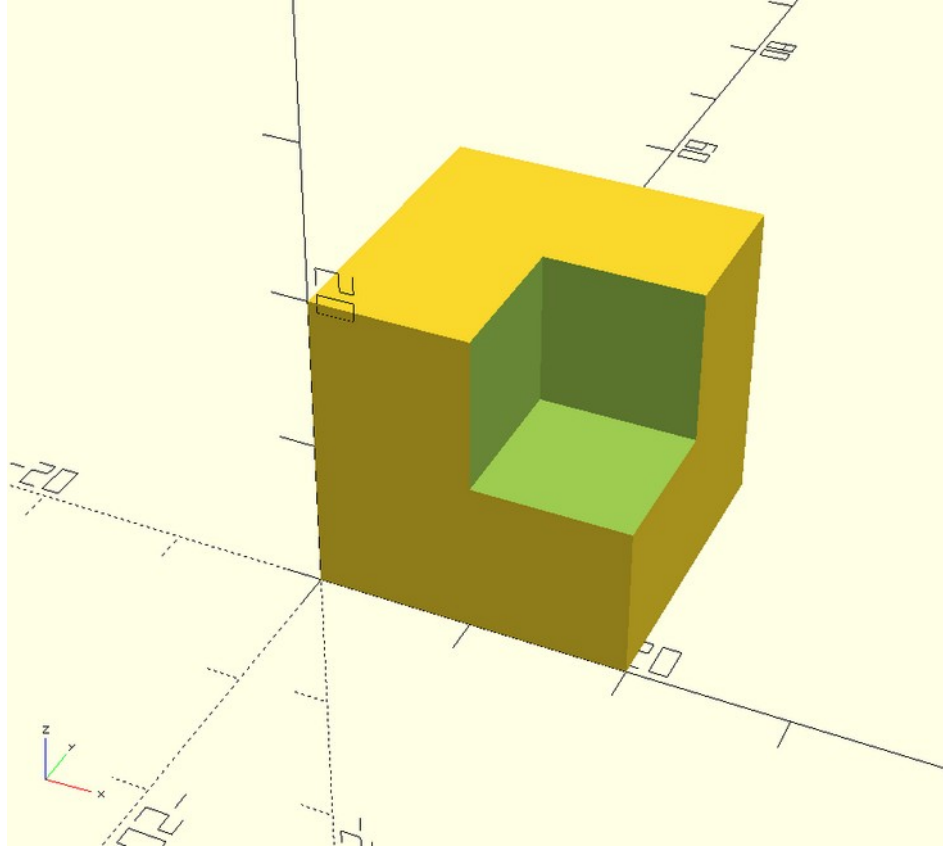
Buy my book

- English and German
- Print and PDF
- Available from Amazon

Order of Battle

- Basic things
- Round things
- Magical, two dimensional things...
- Textual things
- Mathematical things
- Profitable and random things

The basics



What is OpenSCAD?

How did you get by the laser field in the Great Hall?



Customize this YARN by adding your own text and voice to it. Add text on top, bottom, or both.

Image:
yarn

What is OpenSCAD? - II

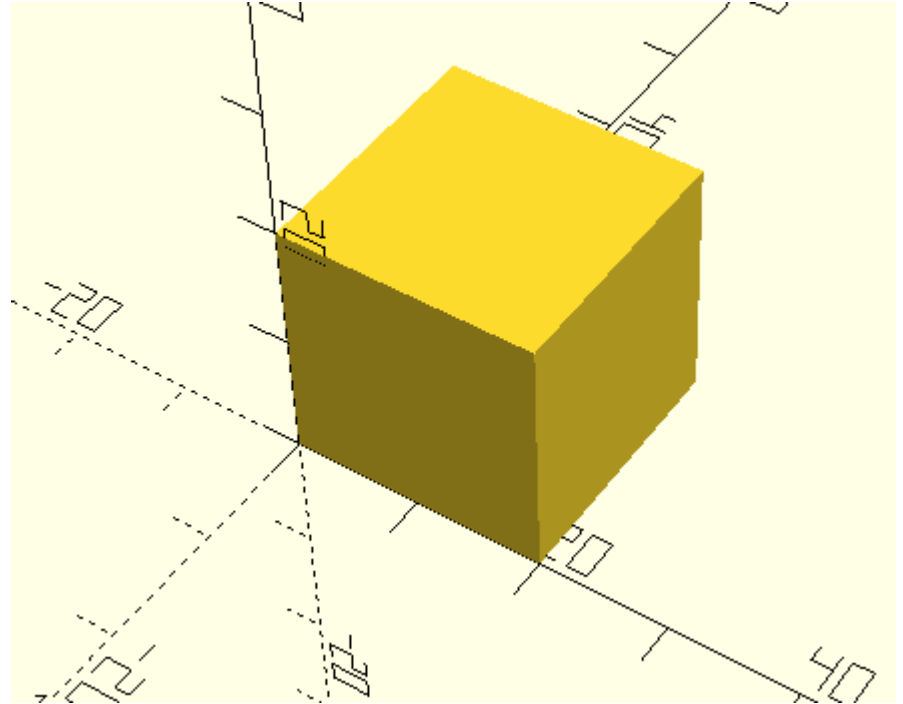
OpenSCAD is a free, open-source 3D CAD software that creates solid 3D models using **a scripting language (yes, we must code things...a lot of things)** It's ideal for programmers and engineers who want to create precise, technical, and parameter-driven objects, like mechanical parts, rather than artistic or organic models.

What is OpenSCAD? - III



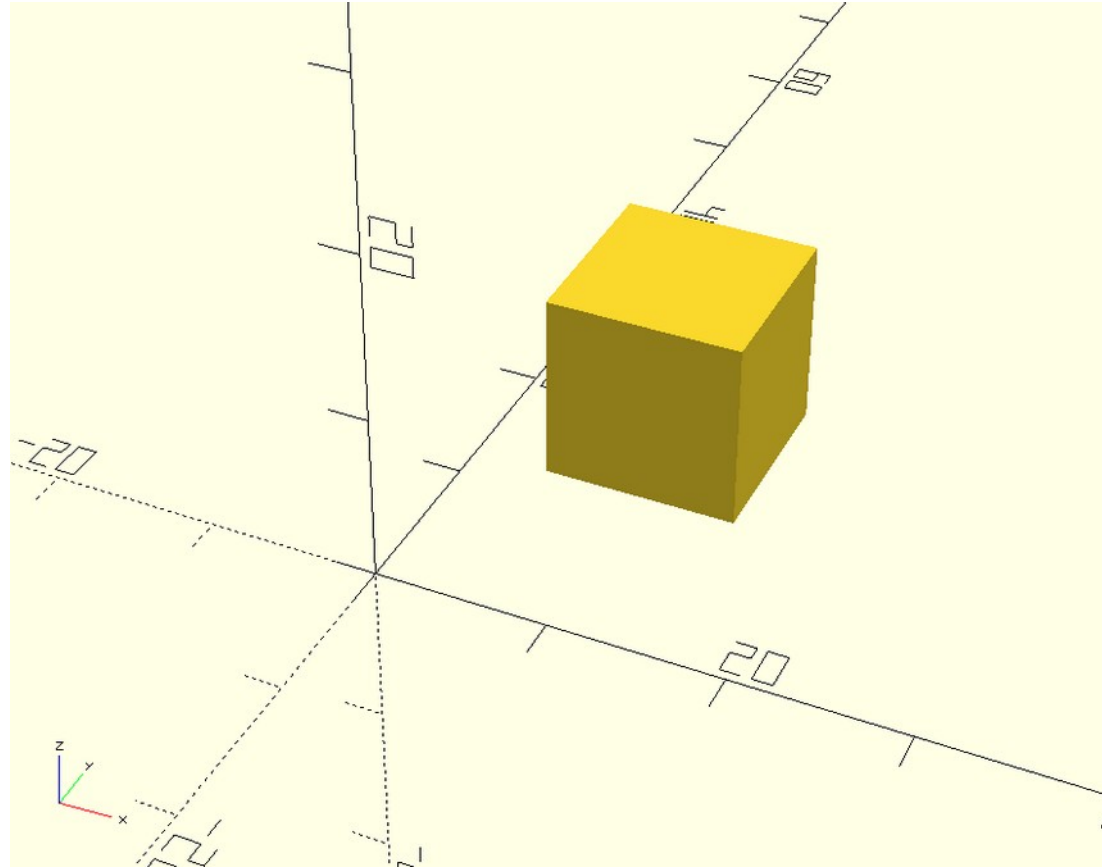
Machinist's cube – I

```
cube(10);
```



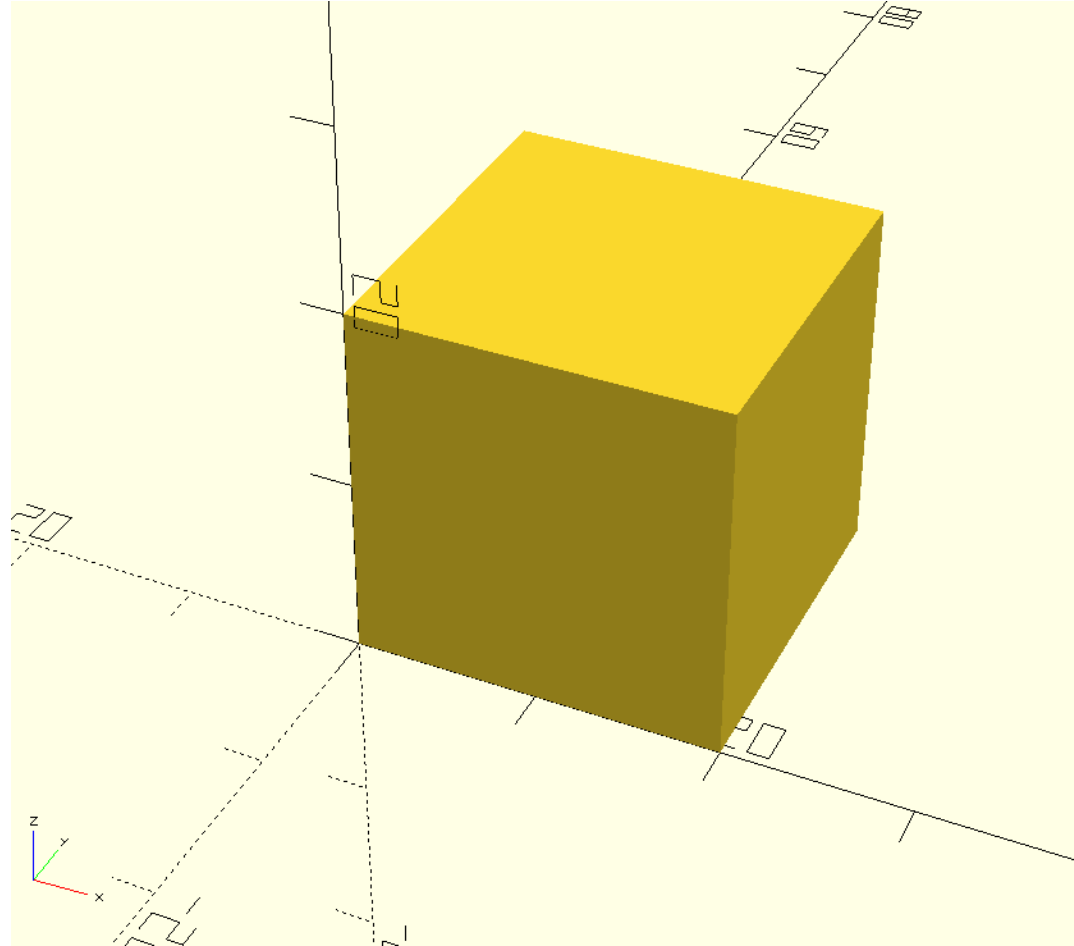
Machinist's cube - II

```
translate([10,0,10])  
cube(10);
```



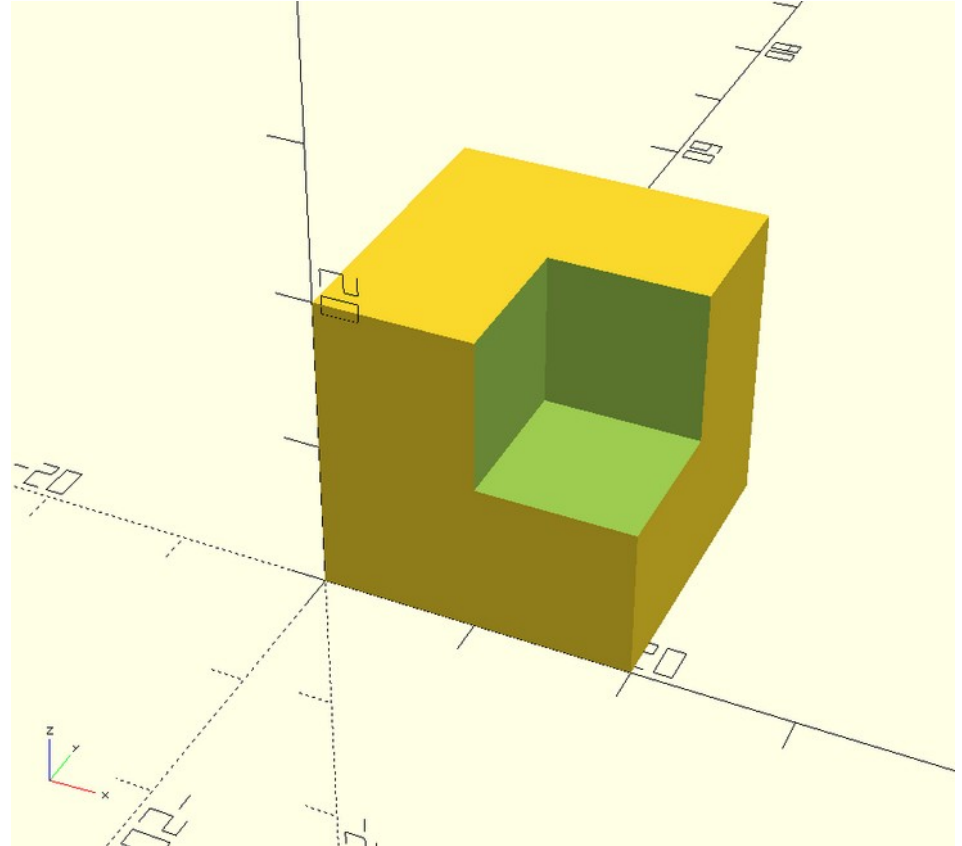
Machinist's cube - III

```
cube(20);  
translate([10,0,10])  
  cube(10);
```

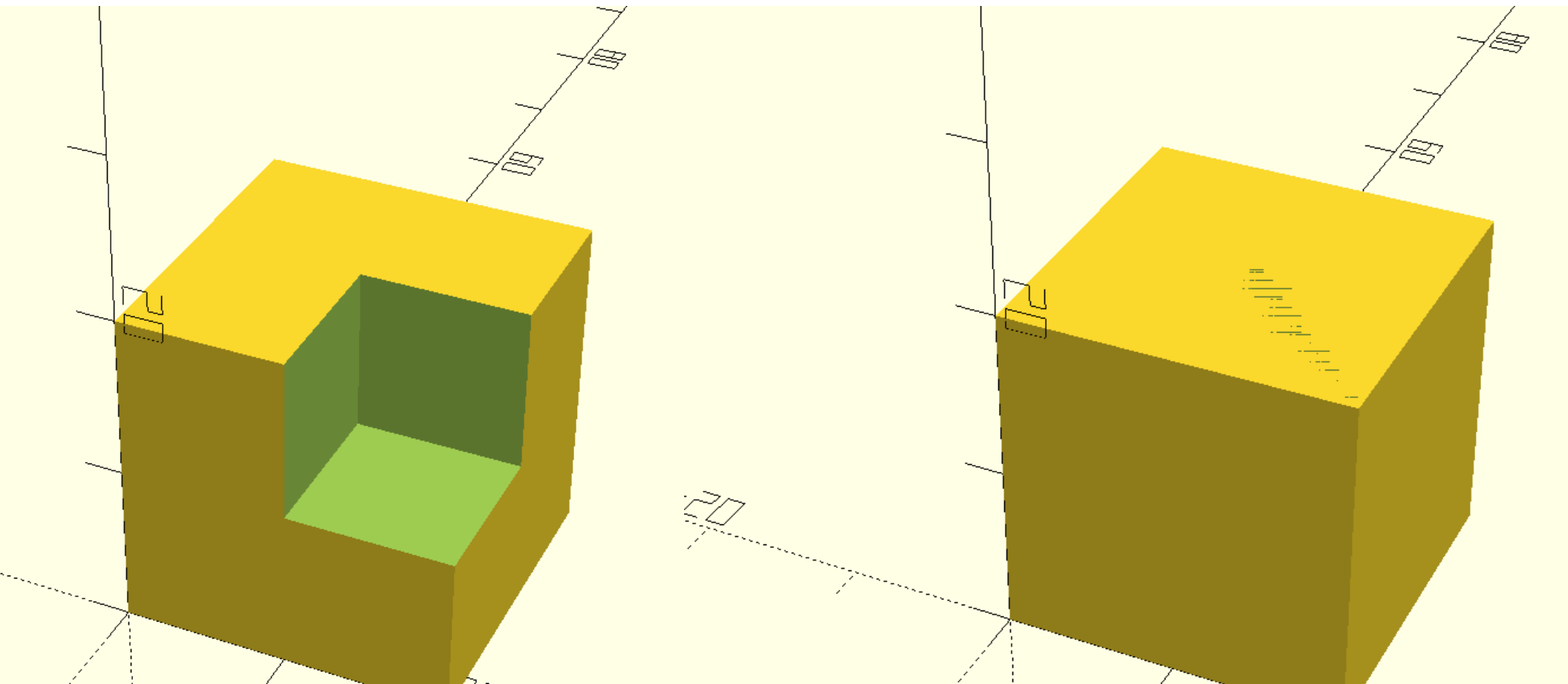


Machinist's cube - IV

```
difference(){  
    cube(20);  
    translate([10,0,10])  
    cube(10);  
}
```



Preview or Rendering?

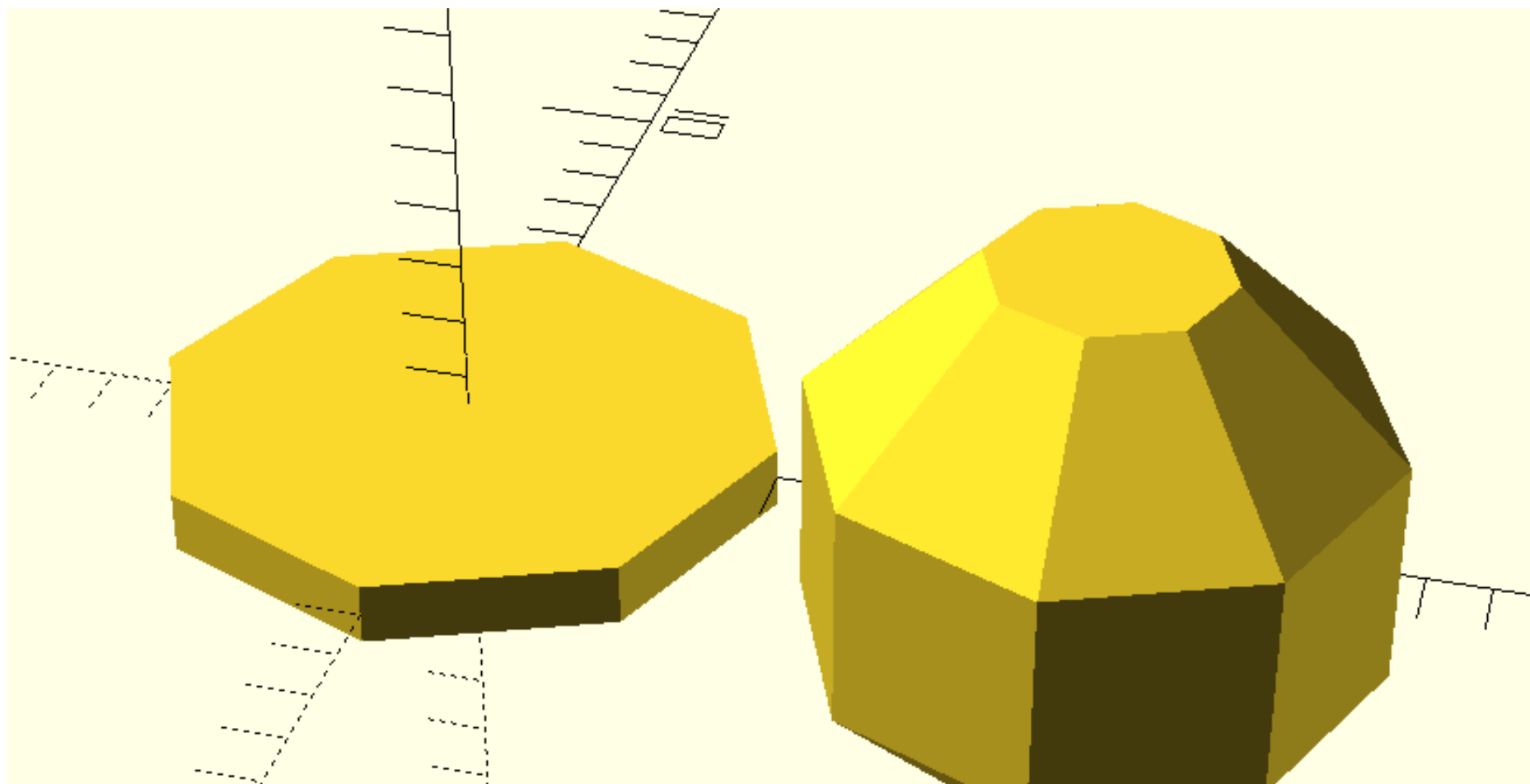


Round things

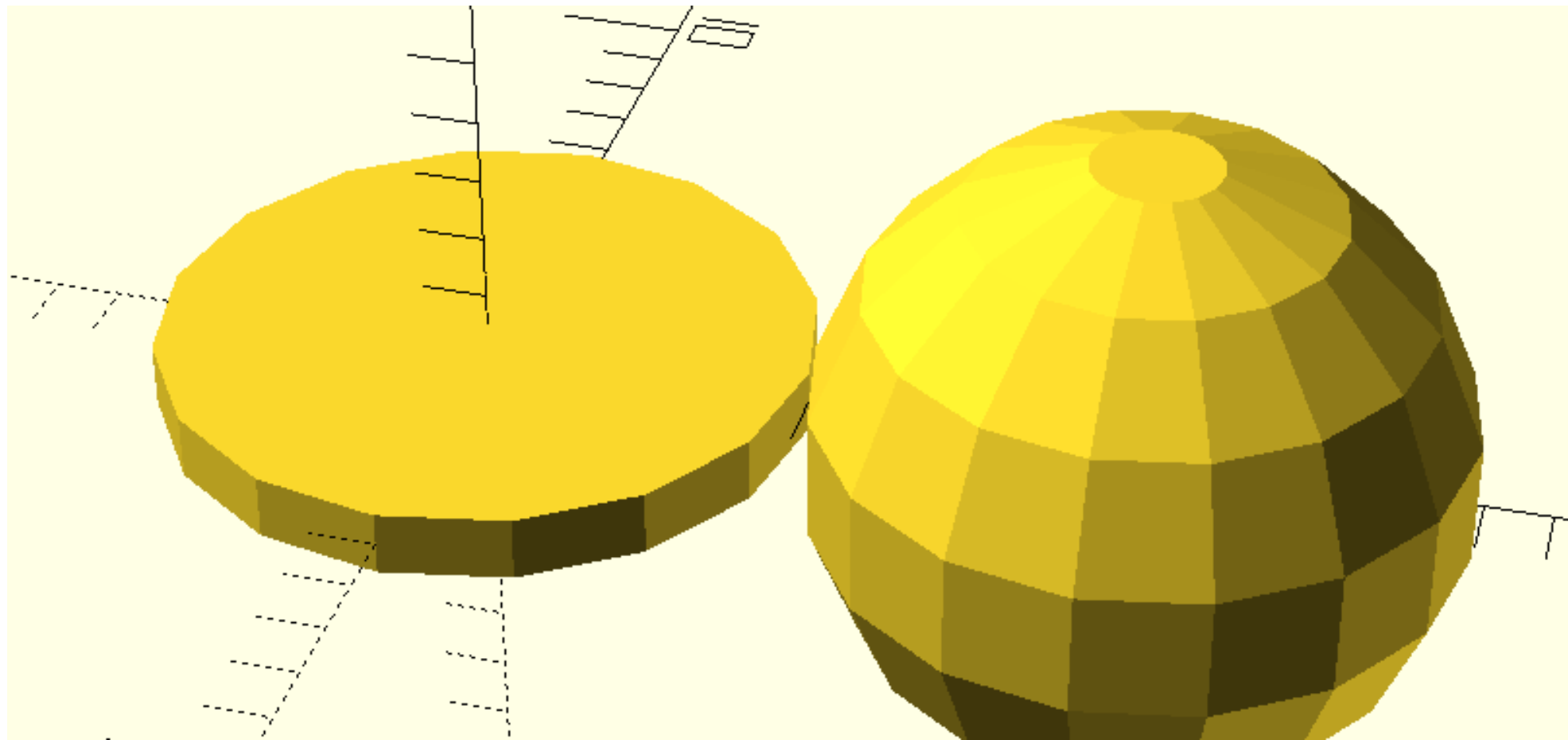
Problem analysis

- OpenSCAD uses a geometry parser
- Round things => how many points?
 - CPU limited !!!
- Solution: linear approximation

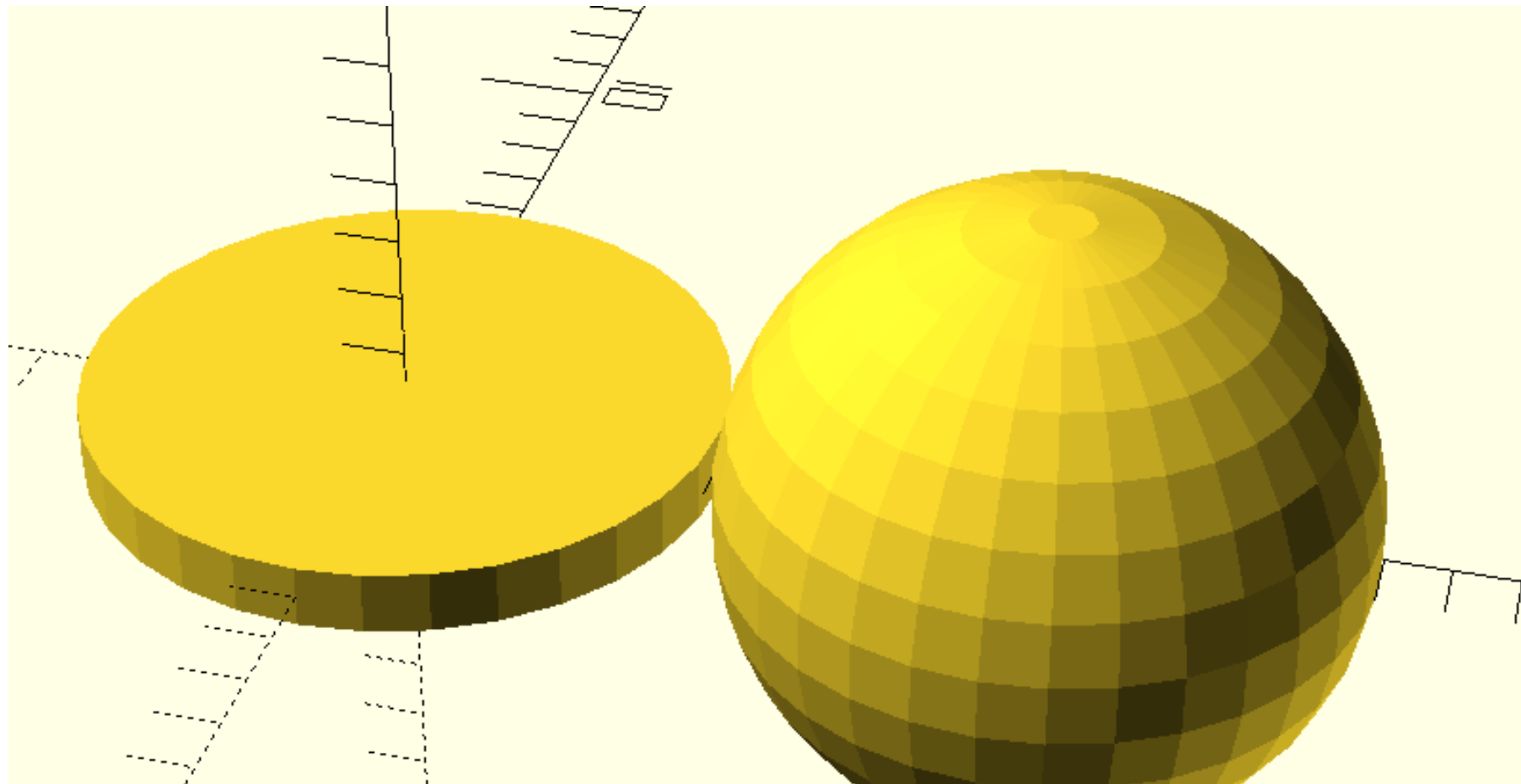
$\$fn=8$



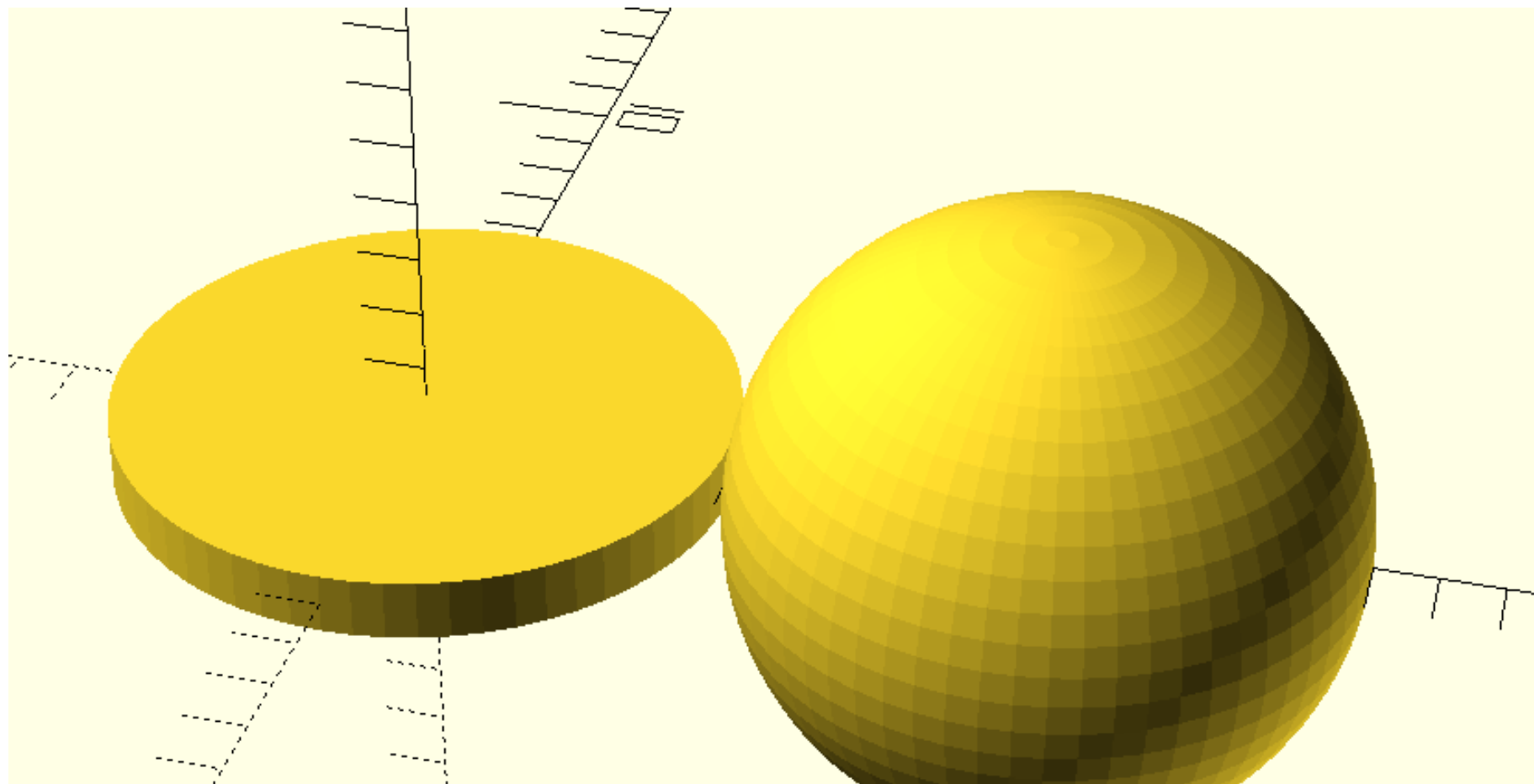
$\$fn=16$



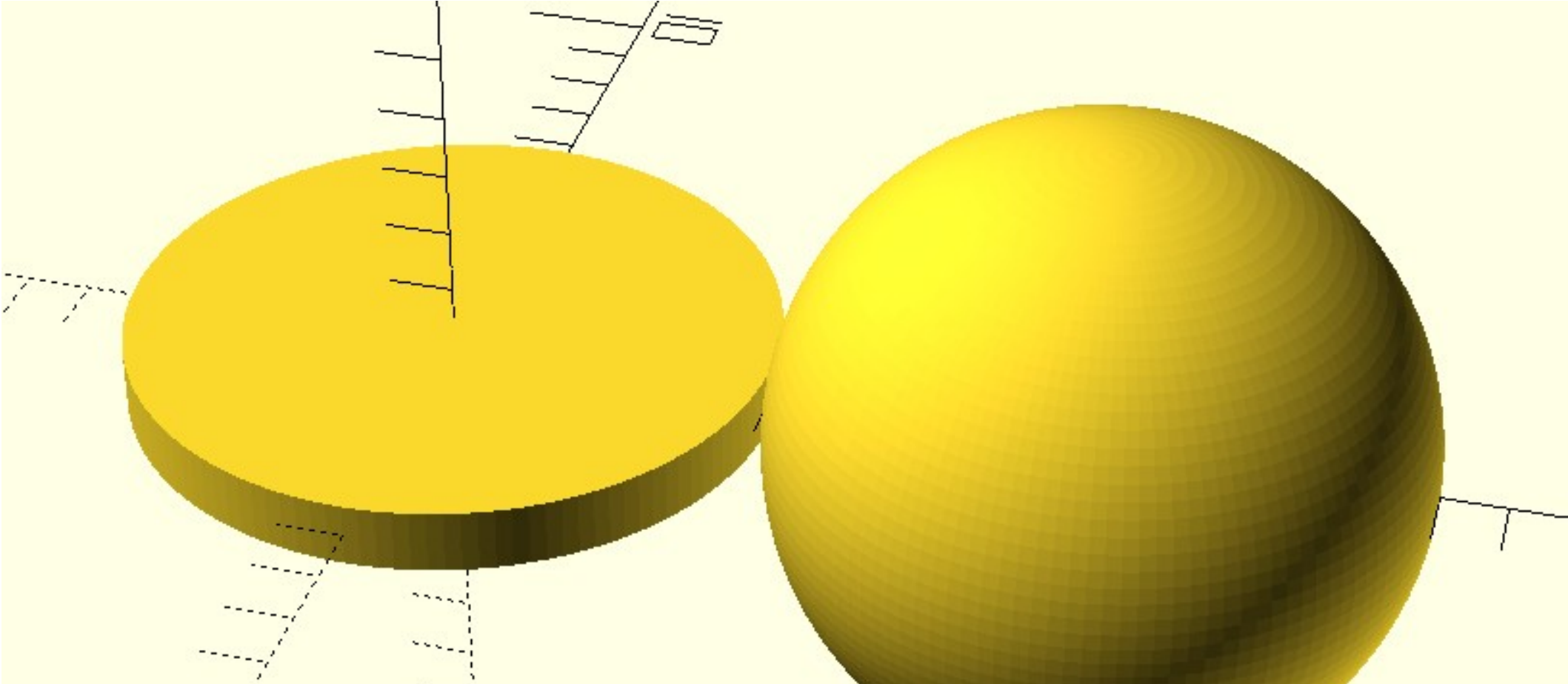
\$fn=32



$\$fn=64$



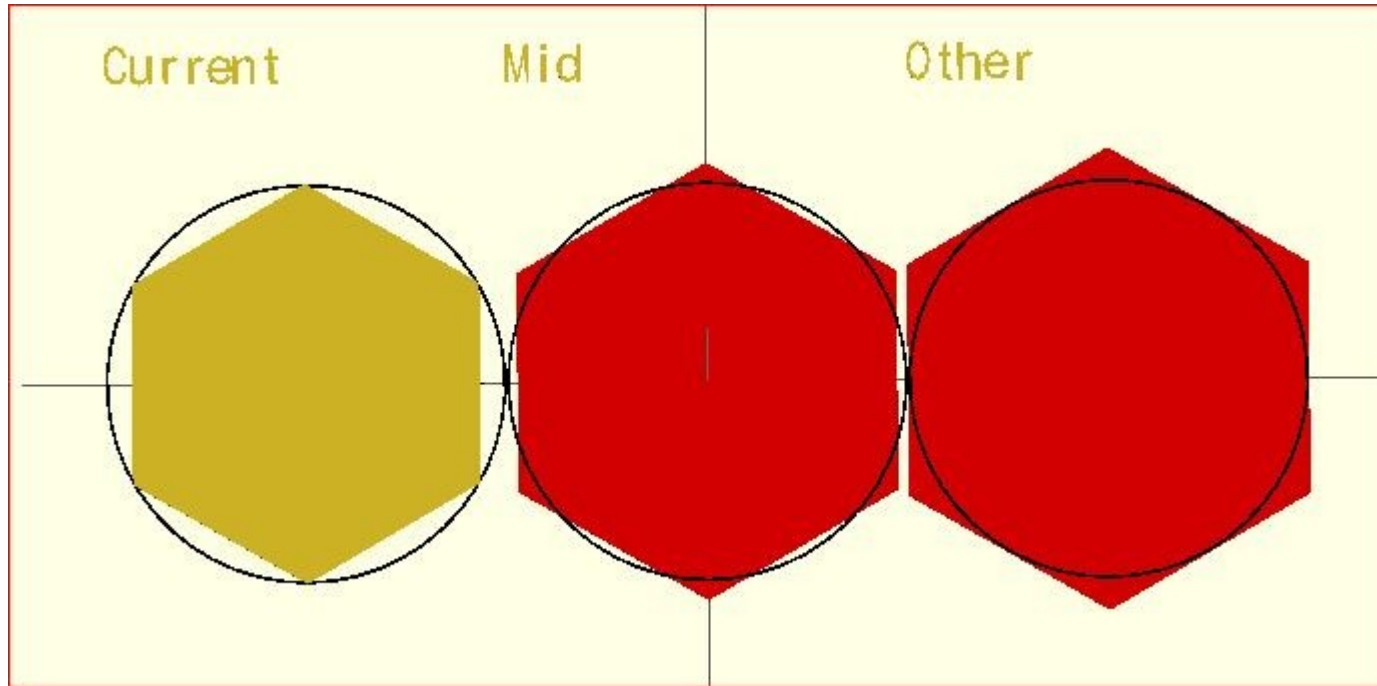
$\$fn=128$



CPU power used

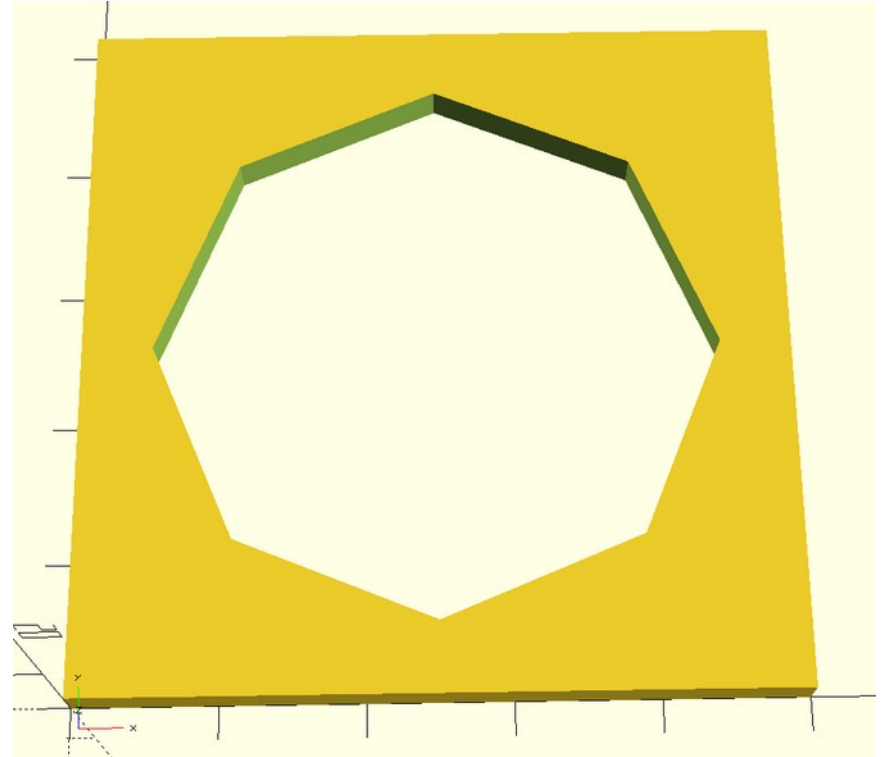
\$fn=8	0:00:02.571
\$fn=16	0:00:06.089
\$fn=32	0:00:18.814
\$fn=64	0:01:06.317
\$fn=128	0:04:18.864
\$fn=256	0:17:24.232

Beware the hole



Let us test

```
$fn=8;  
difference({  
  cube([50,50,3]);  
  
  translate([25,25,-1])  
    cylinder(30,40/2,40/2);  
})
```

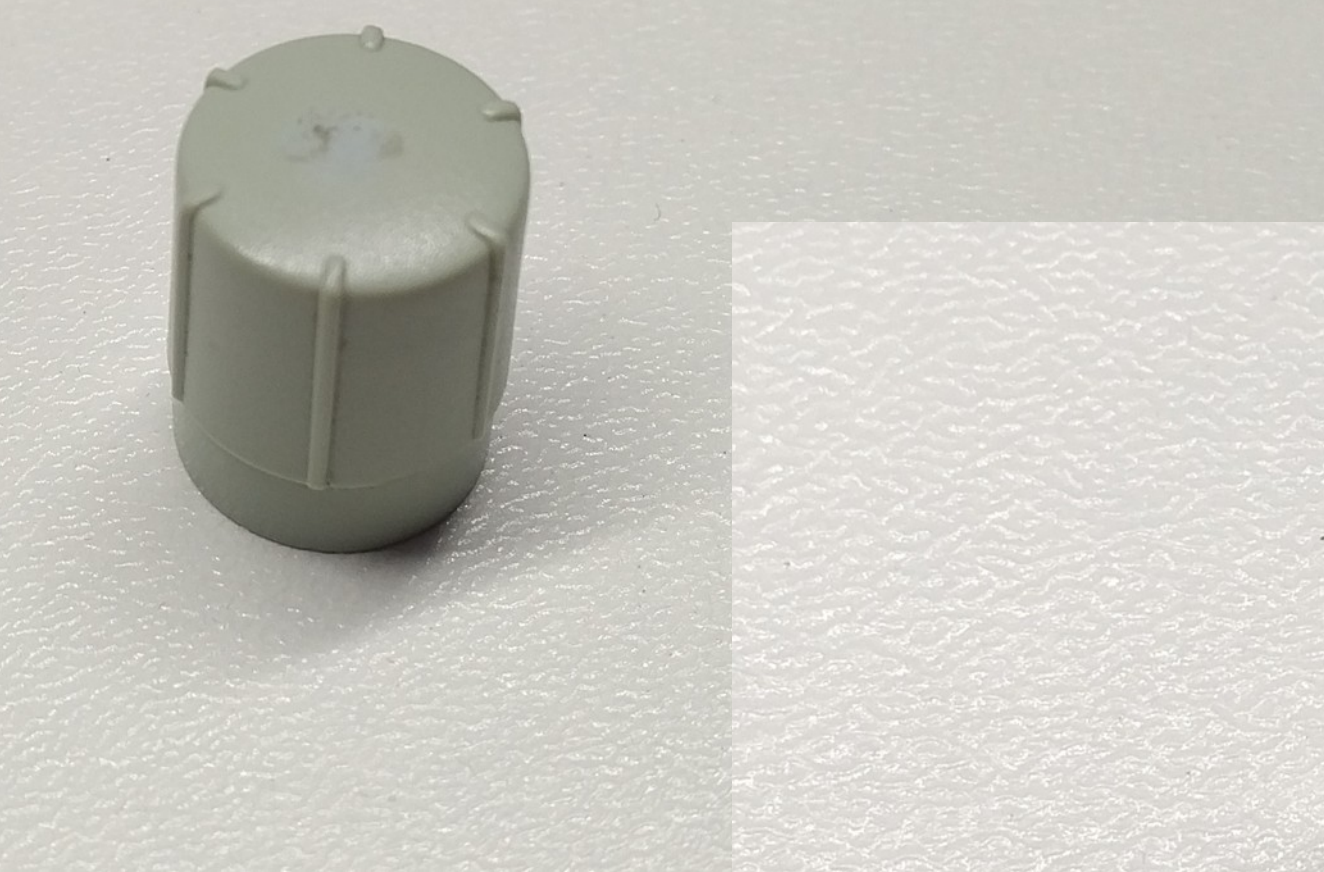


Real life results

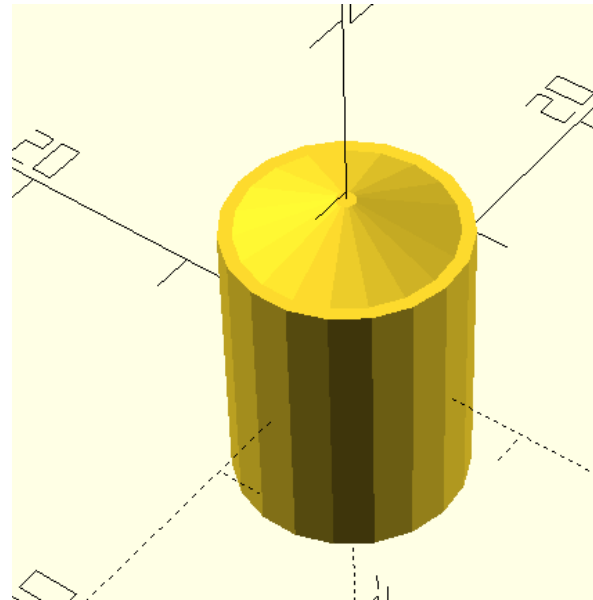
fn	Minimal	Maximal
8		
16	38.64	39.56
32	39.52	39.69



The Swiss patient

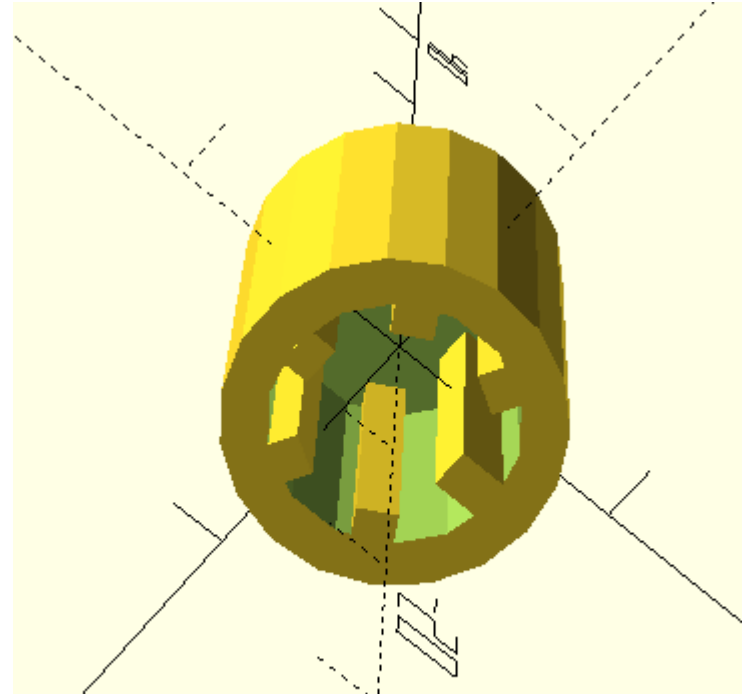


```
tamsmodule();  
module tamsmodule(){  
    difference(){  
        cylinder( 15.5, 12/2, 12/2, true);  
        translate([0,0,-2])cylinder(13, 8.8/2, 8.8/2, true);  
    }  
  
    //some kind of lid  
    translate([0,0,8.5])cylinder(2, 12/2, 1/2, true);  
}
```

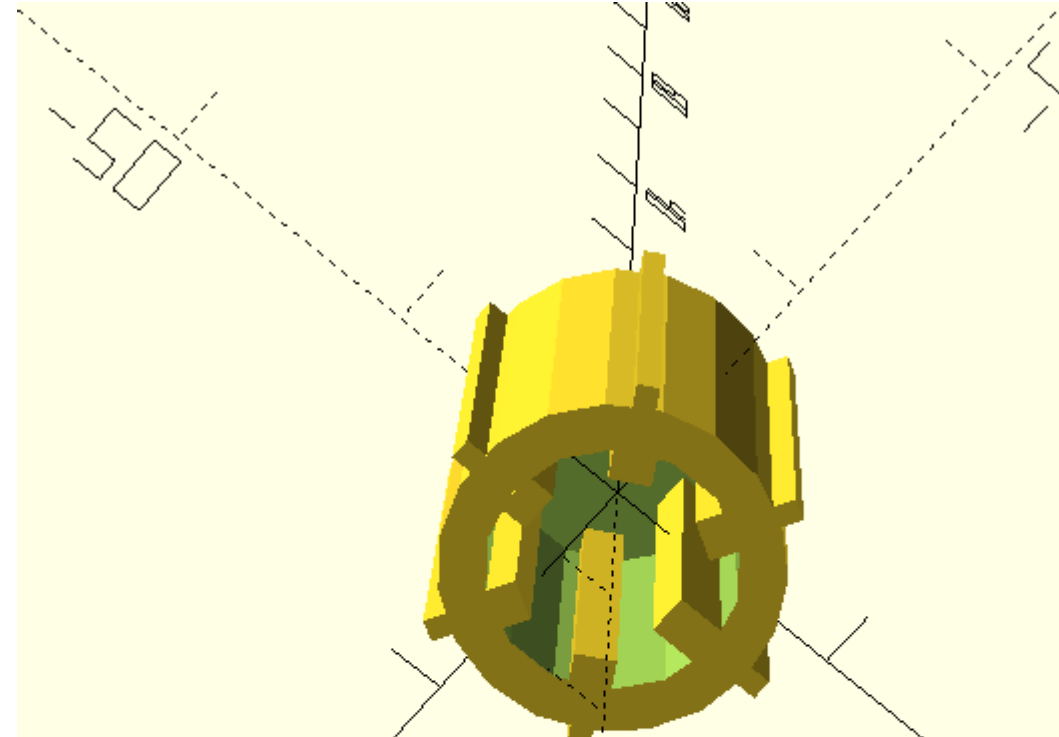


//inner detents

```
rotate([0,0,0])translate([12/2-2.2,0,0])cube([3.1-0.4,1.5,15.5],true);  
rotate([0,0,360/6])translate([12/2-2.2,0,0])cube([1.5-0.4,1.5,15.5],true);  
rotate([0,0,360/6*2])translate([12/2-2.2,0,0])cube([1.5-0.4,1.5,15.5],true);  
rotate([0,0,360/6*3])translate([12/2-2.2,0,0])cube([1.5-0.4,1.5,15.5],true);  
rotate([0,0,360/6*4])translate([12/2-2.2,0,0])cube([1.5-0.4,1.5,15.5],true);  
rotate([0,0,360/6*5])translate([12/2-2.2,0,0])cube([1.5-0.4,1.5,15.5],true);
```



```
rotate([0,0,0])translate([12/2,0,0])cube([1.5,0.8,15.5],true);  
rotate([0,0,360/6])translate([12/2,0,0])cube([1.5,0.8,15.5],true);  
rotate([0,0,360/6*2])translate([12/2,0,0])cube([1.5,0.8,15.5],true);  
rotate([0,0,360/6*3])translate([12/2,0,0])cube([1.5,0.8,15.5],true);  
rotate([0,0,360/6*4])translate([12/2,0,0])cube([1.5,0.8,15.5],true);  
rotate([0,0,360/6*5])translate([12/2,0,0])cube([1.5,0.8,15.5],true);
```



Outer Detents, simplified

```
for(var = [0:6])
```

```
rotate([0,0,360/6*var])translate([12/2,0,0])cube([1.  
5,0.8,15.5],true);
```

Inner detents

```
for(var = [0:5]) {  
  if(var==0){  
    rotate([0,0,0])translate([12/2-2.2,0,0])cube([3.1-0.4,1.5,15.5],true);  
  }  
  else{  
    rotate([0,0,360/6 * var])translate([12/2-2.2,0,0])cube([1.5-0.4,1.5,15.5],true);  
  }  
}
```

Short mention: modules

```
module tamsDreidel(dia=22)
{
    cylinder(dia, dia, dia, true);
    cylinder(dia*2, dia/9, dia/4, true);
}
```

Magical, two dimensional things...

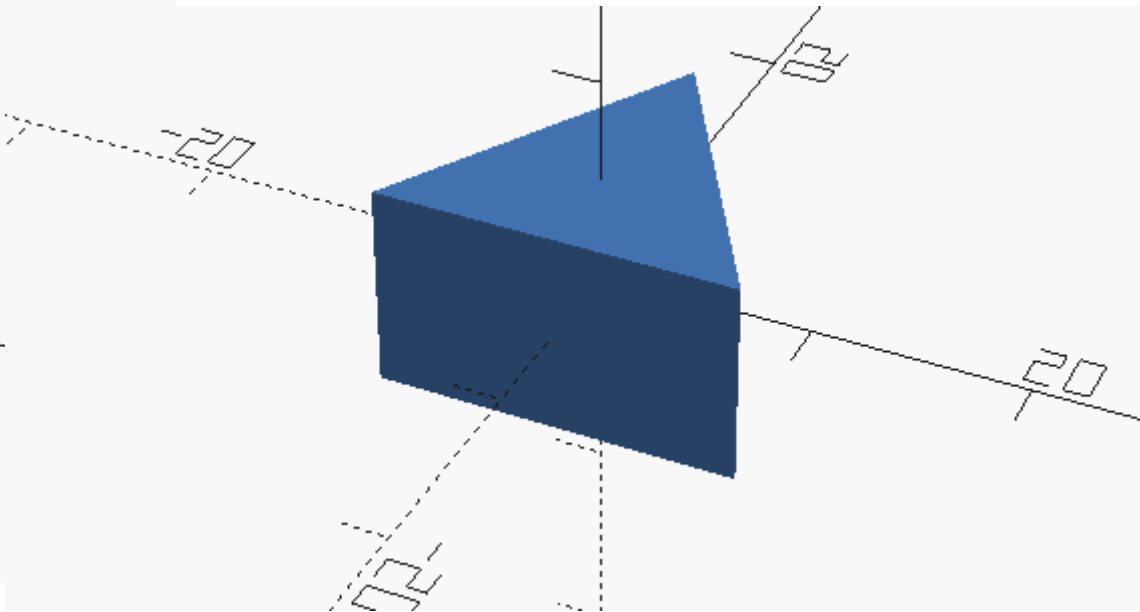
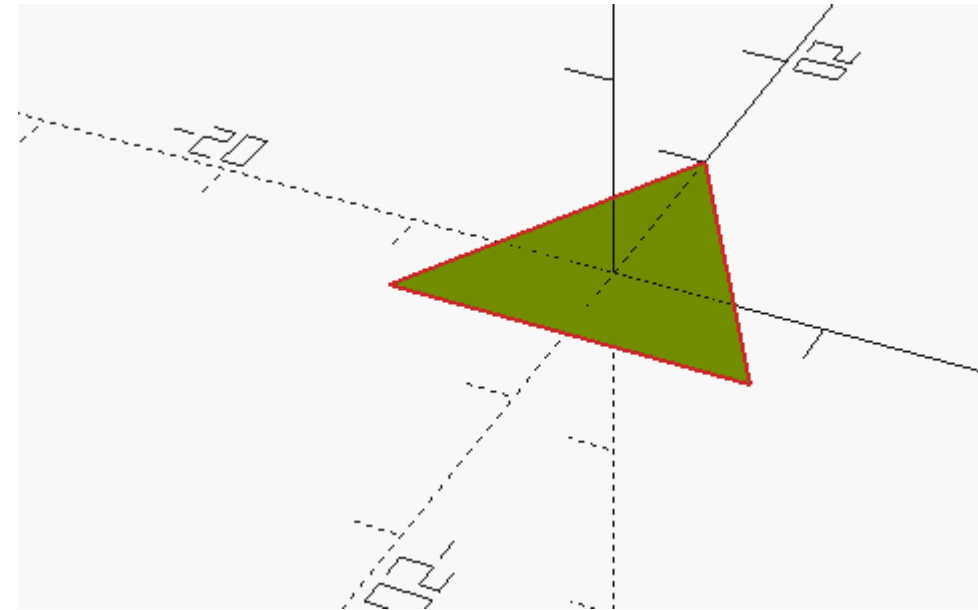


2D objects as “base” of 3D objects

- OpenSCAD is NOT AutoCAD / FreeCAD
- 2D export is limited
 - Used for “sectio” diagrams
 - Rarely used in general
- Primary use: extrusion to 3D

```
centeredtri();
```

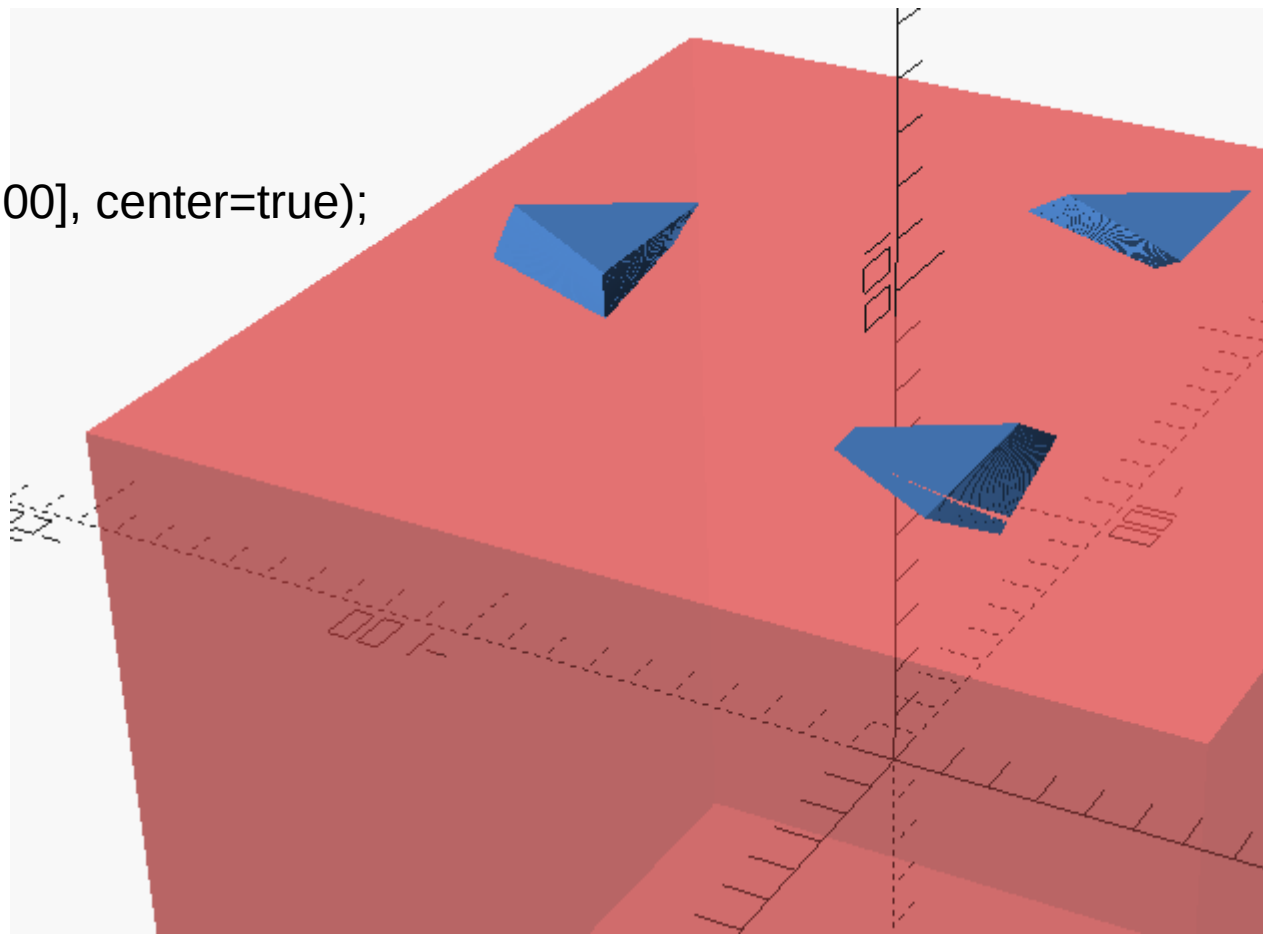
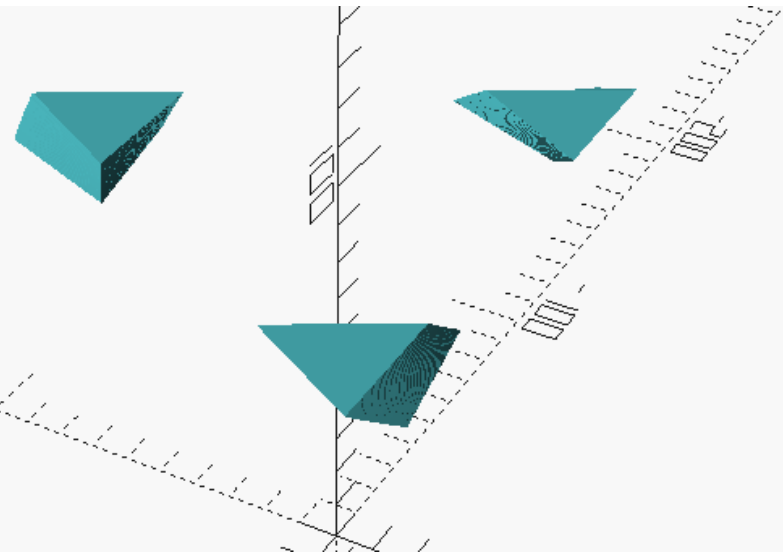
```
module centeredtri(){  
  scale(10)  
  polygon([[ -0.866,-0.5], [0.866, -0.5], [0, 1]]);  
}
```



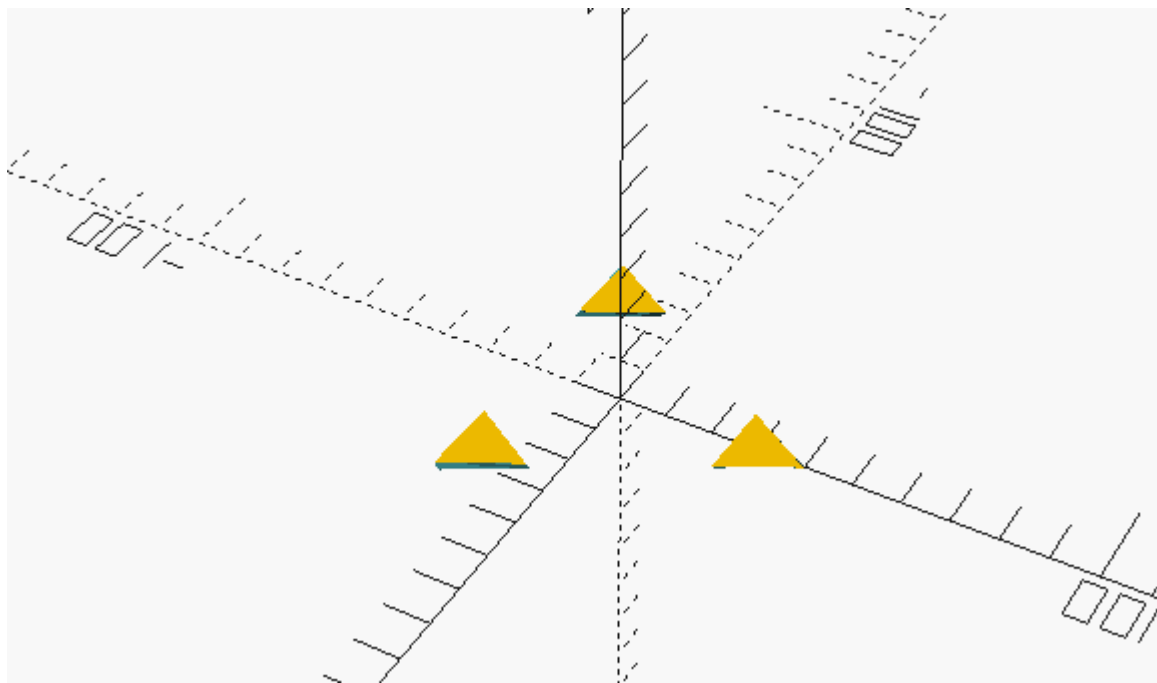
```
module vase()  
{  
  linear_extrude(100, scale=2, twist=180, slices=1000)  
  {  
    r = 30;  
    translate([-0.866*r,-0.5*r])centeredtri();  
    translate([0.866*r, -0.5*r])centeredtri();  
    translate([0, 1*r])centeredtri();  
  }  
}
```

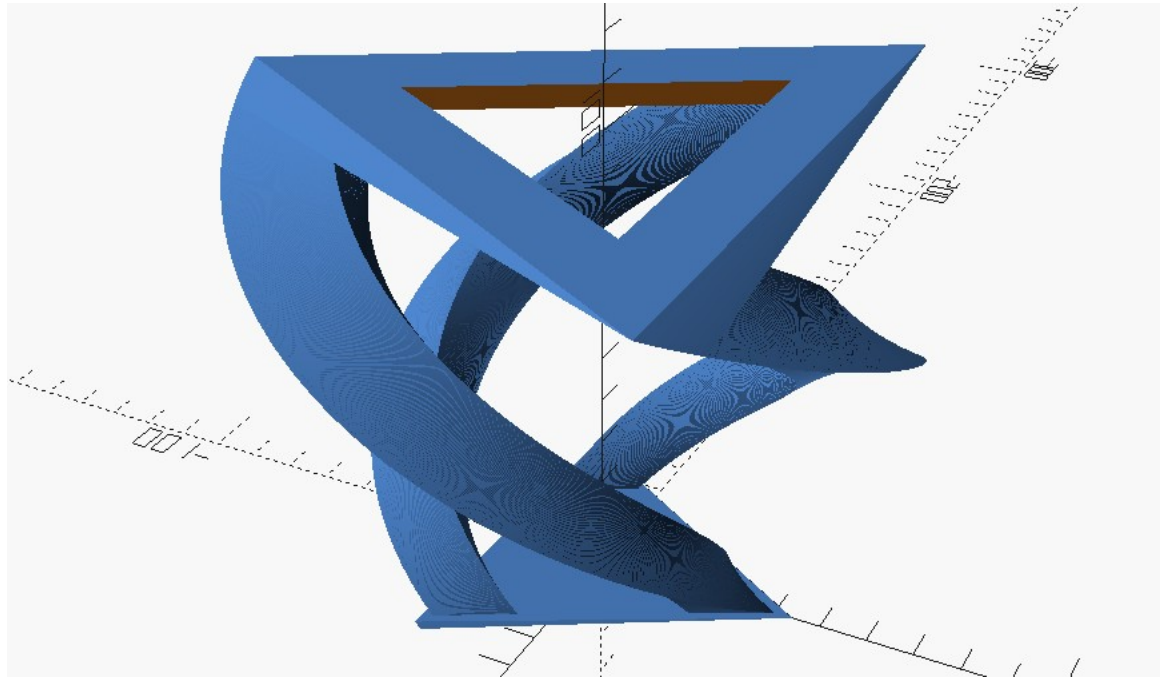


```
module roof()  
{  
  difference()  
  {  
    vase();  
    translate([0,0,-5])#cube([200,200,200], center=true);  
  }  
}
```



```
module foot()
{
    difference()
    {
        vase();
        translate([0,0,101])cube([200,200,200], center=true);
    }
}
```





```
scale(1.1)hull()foot();  
vase();  
difference()  
{  
  hull()roof();  
  linear_extrude(2300, center=true)rotate(180)scale(4.5) centeredtri();  
}
```

Textual things



Render a text

```
$fn=64;  
for ( i = [0 : 5 : 40] ){  
    translate([i*5, i*5]) {  
        text("Tamoggemon", size = i);  
    }  
}
```

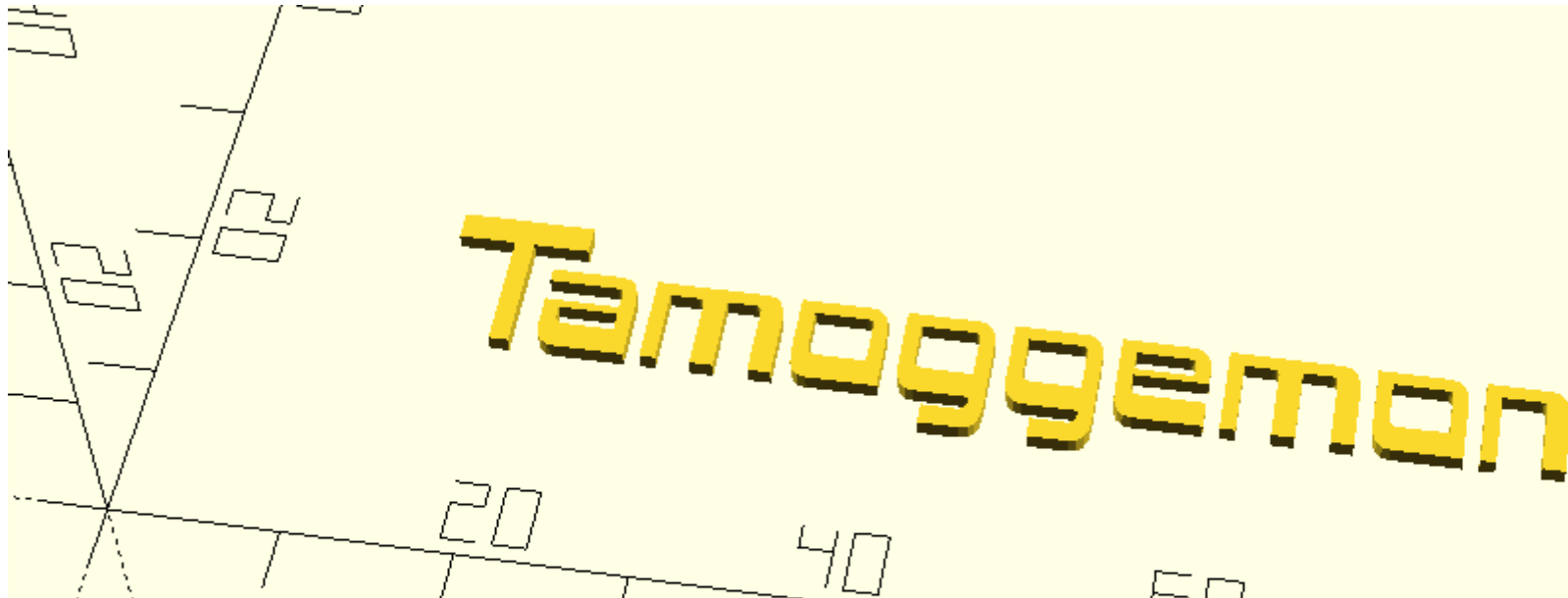


List fonts

OpenSCAD Font List — OpenSCAD		
This list shows the fonts currently registered with OpenSCAD.		
Example:		
<pre>text("OpenSCAD", font = "DejaVu Sans"); text("OpenSCAD", font = "Liberation Sans:style=Italic");</pre>		
Font name	Font style	Filename
137 Abyssinica SIL	Regular	/usr/share/fonts/truetype/abyssinica/AbyssinicaSIL-R.ttf
276 Ani	Regular	/usr/share/fonts/truetype/fonts-beng-extra/ani.ttf
9 AnjaliOldLipi	Regular	/usr/share/fonts/truetype/malayalam/AnjaliOldLipi.ttf
15 Bitstream Charter	Italic	/usr/share/fonts/X11/Type1/c0649bt_.pfb
51 Bitstream Charter	Bold Italic	/usr/share/fonts/X11/Type1/c0633bt_.pfb
54 Bitstream Charter	Regular	/usr/share/fonts/X11/Type1/c0648bt_.pfb
180 Bitstream Charter	Bold	/usr/share/fonts/X11/Type1/c0632bt_.pfb
66 Century Schoolbook L	Roman	/usr/share/fonts/type1/gsfonts/c059013l.pfb
184 Century Schoolbook L	Italic	/usr/share/fonts/type1/gsfonts/c059033l.pfb
225 Century Schoolbook L	Bold	/usr/share/fonts/type1/gsfonts/c059016l.pfb

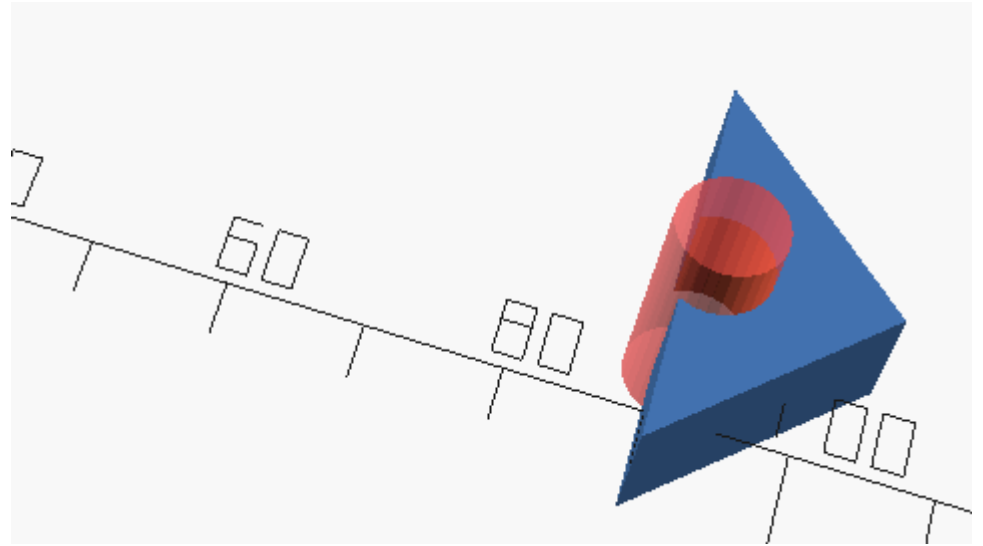
More fonts

```
use </media/tamhan/781A-F80B/OpenSCADBook/stuff/astron boy.ttf>  
translate([15, 15]) {  
  text("Tamoggemon", font = "Astron Boy");  
}
```

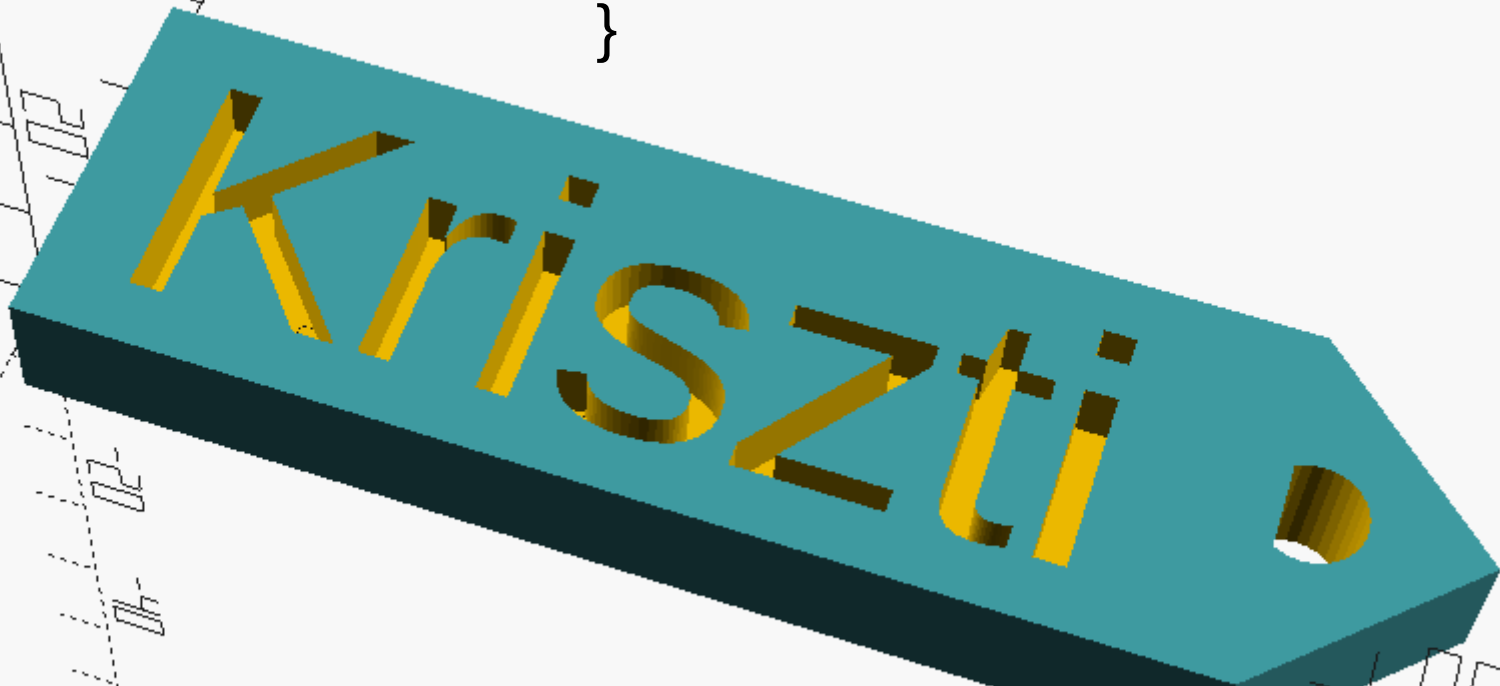


```
$fn=32;  
doortags();
```

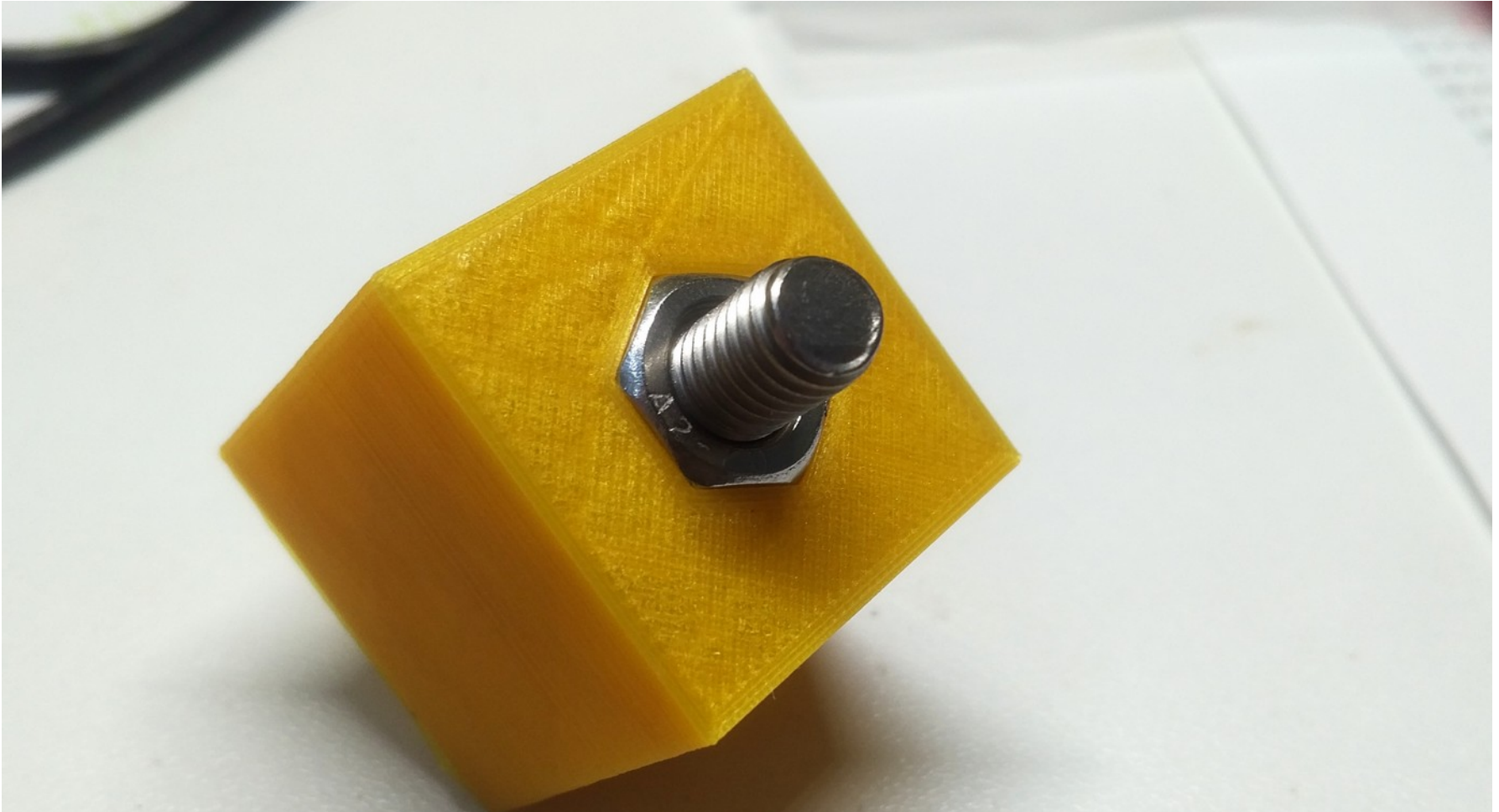
```
module doortags(){  
  translate([90,25,-5])  
  difference(){  
    rotate([0,0,270])  
    linear_extrude(10)  
    polygon(points=[[0,0],[30,0],[15,15]]);  
    translate([2,-15,-5])cylinder(d=8, h=20);  
  }  
}
```



```
module doortags(){  
  
  difference()  
  {  
    translate([0,-5,-5])cube([90,30,10]);  
    translate([5,0,0])linear_extrude(5)text("Kriszti", 20);  
  }  
}
```

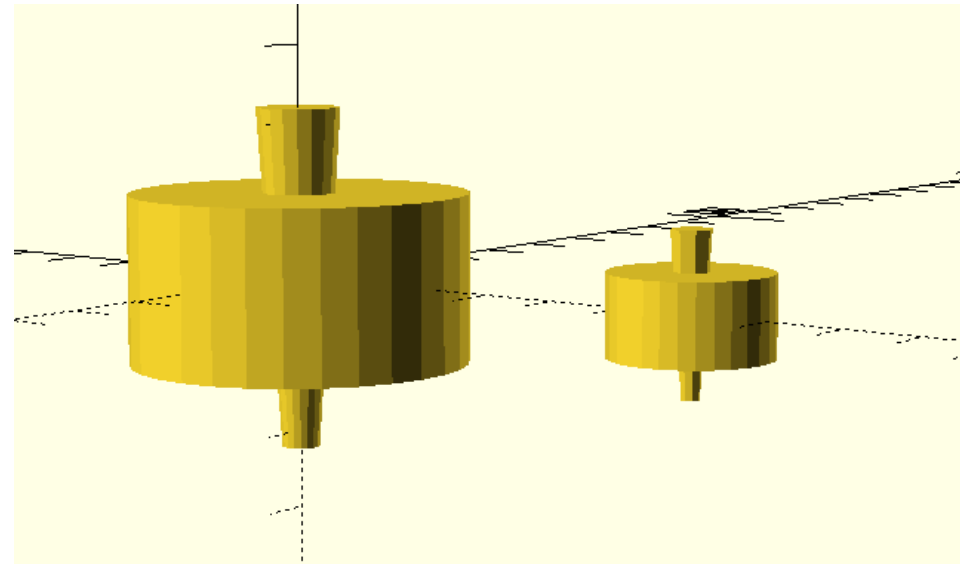


Technical things



Modules, reloaded

- Modules reproduce geometry
- Parametrization adjusts behavior
- Example: Tam's Dreidel
 - Allow for structuring
 - And for reuse



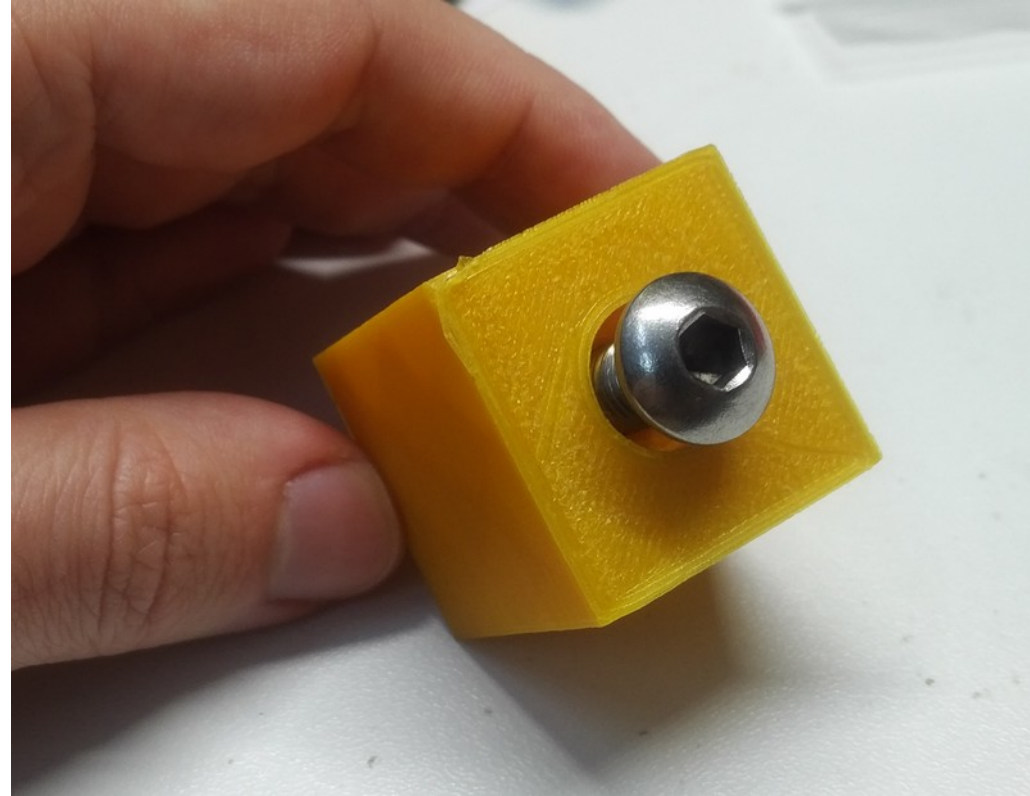
Modules, reloaded - II

```
module tamsDreidel(dia=22)
{
    cylinder(dia, dia, dia, true);
    cylinder(dia*2, dia/9, dia/4, true);
}
tamsDreidel();
```

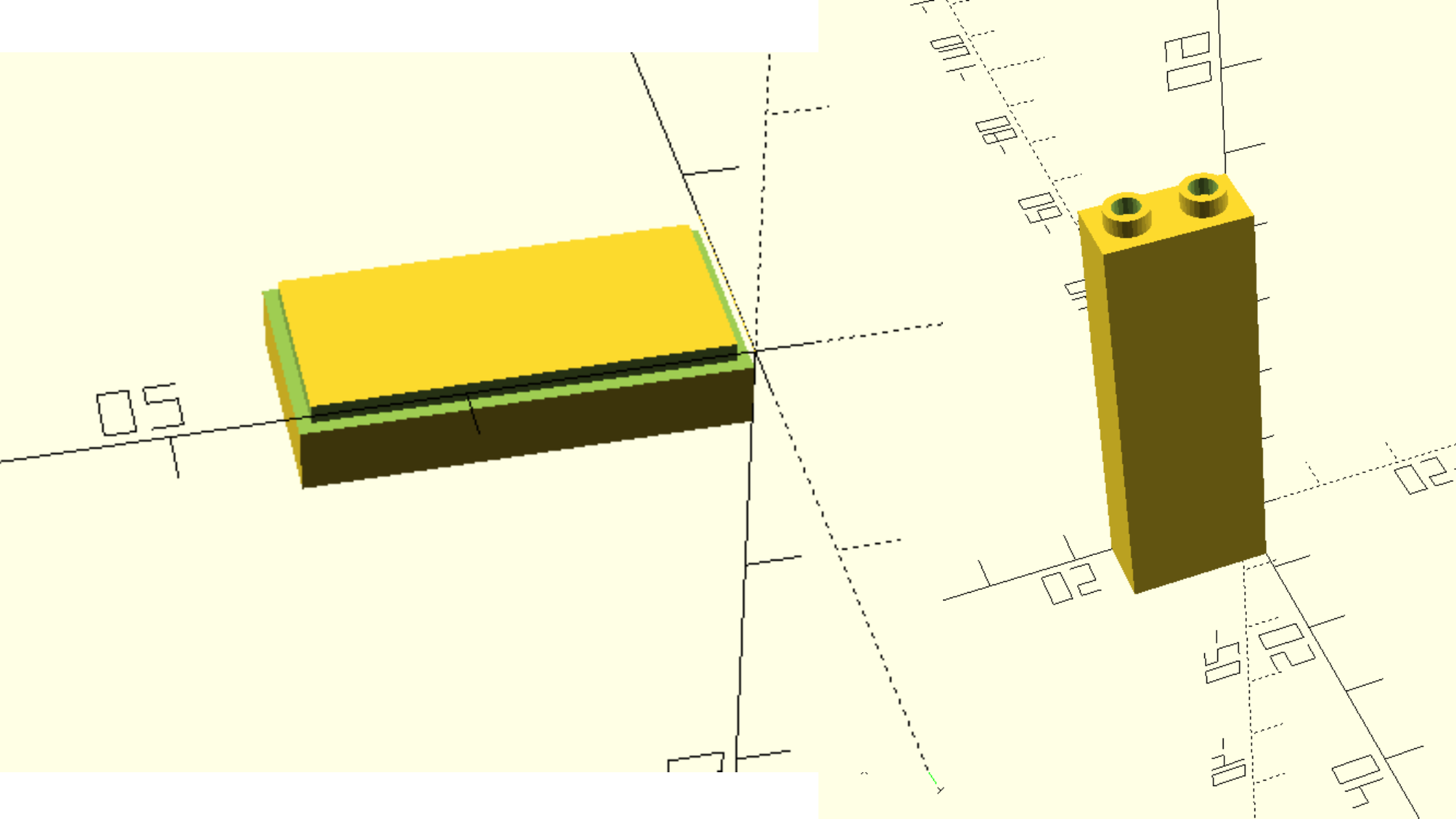
Modules, reloaded - III

```
module tamsDreidel(dia) {  
    if(dia==undef) {  
        echo ("ERROR");  
    }  
    cylinder(dia, dia, dia, true);  
    cylinder(dia*2, dia/9, dia/4, true);  
}  
tamsDreidel();
```

A couple of screws, loose

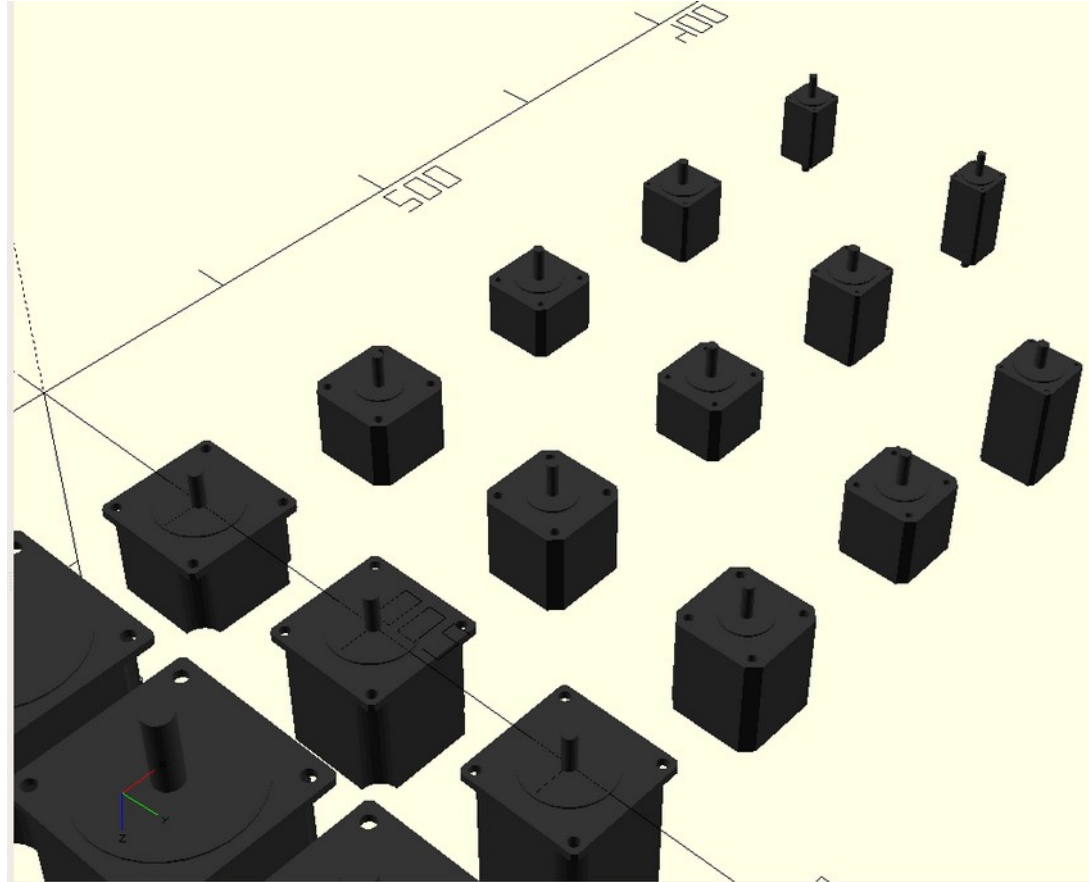


```
$fn=64;  
use <MCAD/nuts_and_bolts.scad>  
difference(){  
    difference()  
    {  
        translate([0,0,7.5])  
            cube(30, center=true);  
        boltHole(8, units="MM", length= 30);  
    }  
  
    translate([0,0,16.5])  
        nutHole(8, units="MM", tolerance = +0.0001, proj = -1);  
}
```

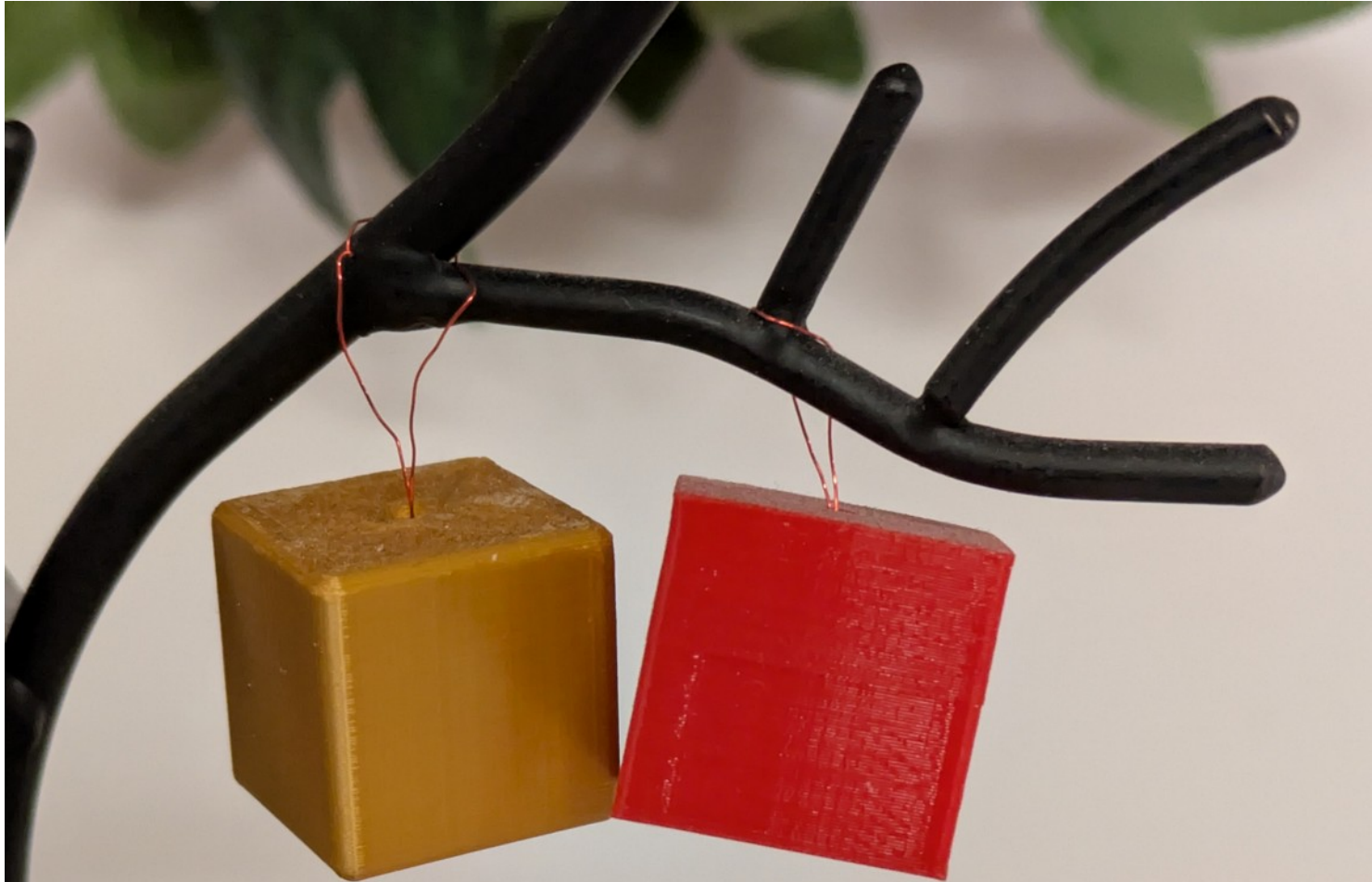


```
use <MCAD/lego_compatibility.scad>  
  block(1,2,5,reinforcement=false,hollow_knob=true);
```

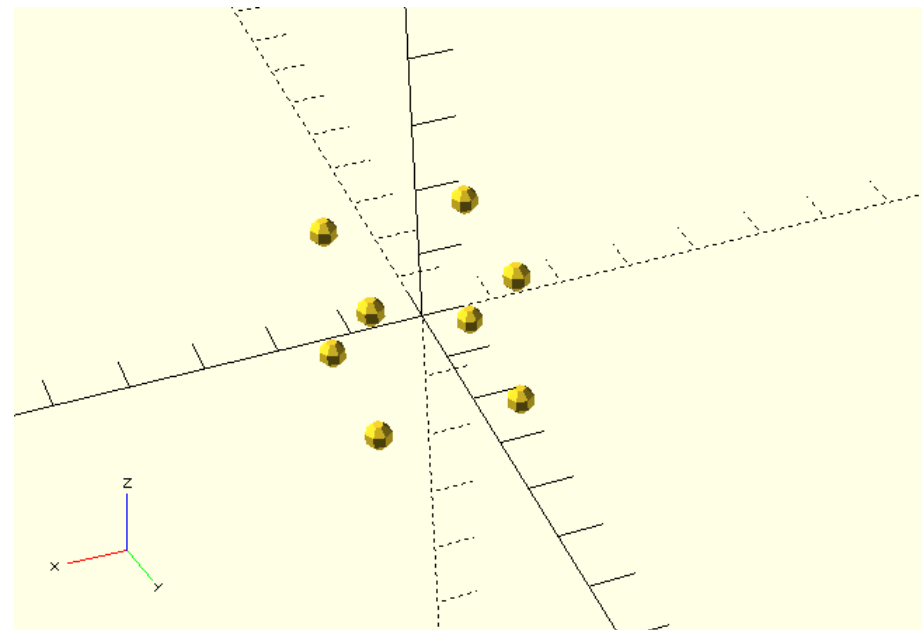
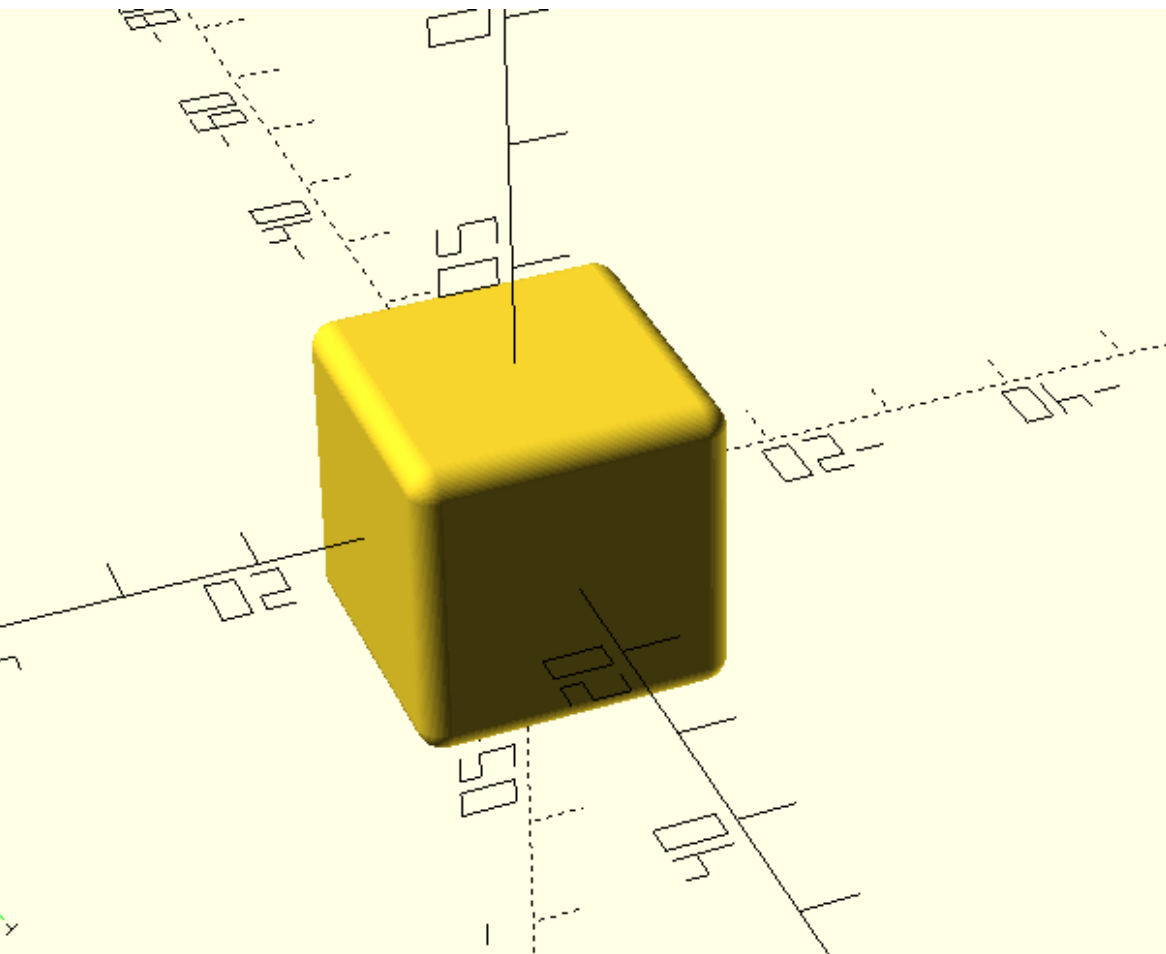
NEMA-Steppers



Profitable and random things

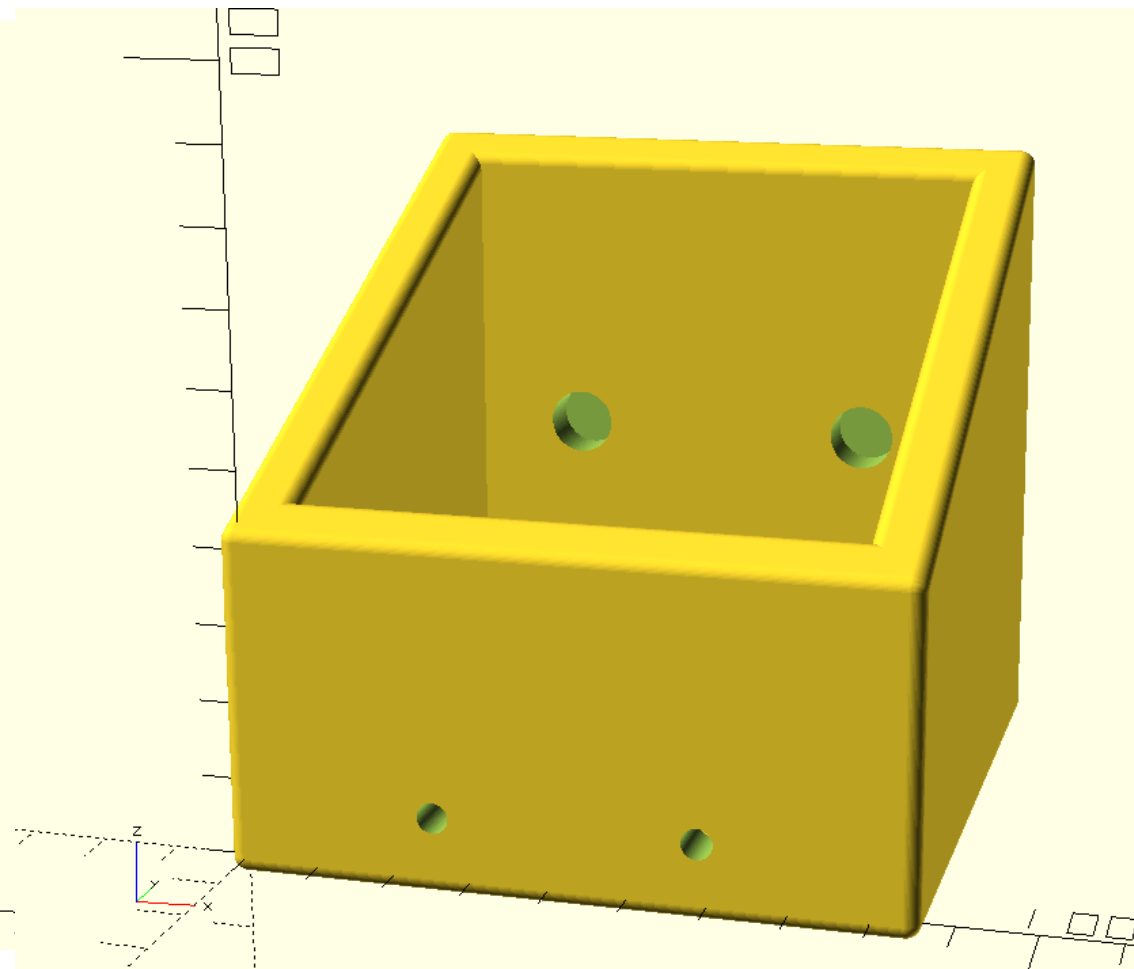
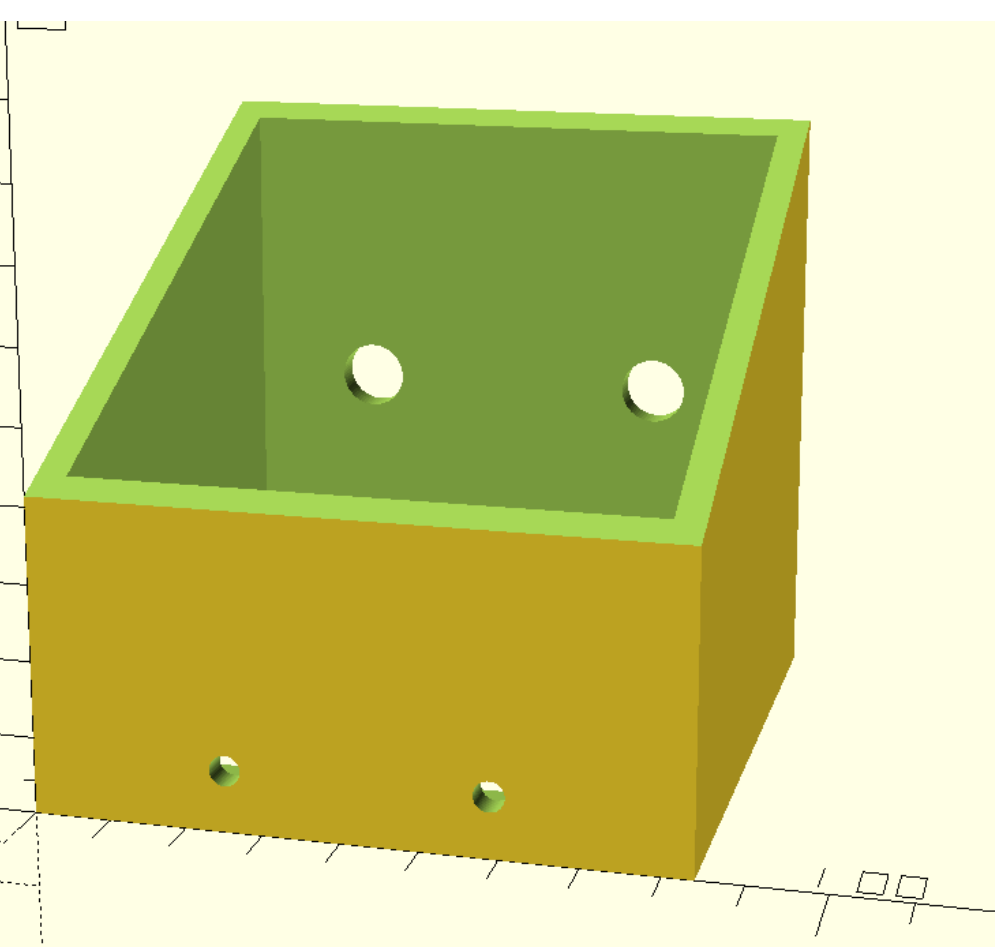


Hull

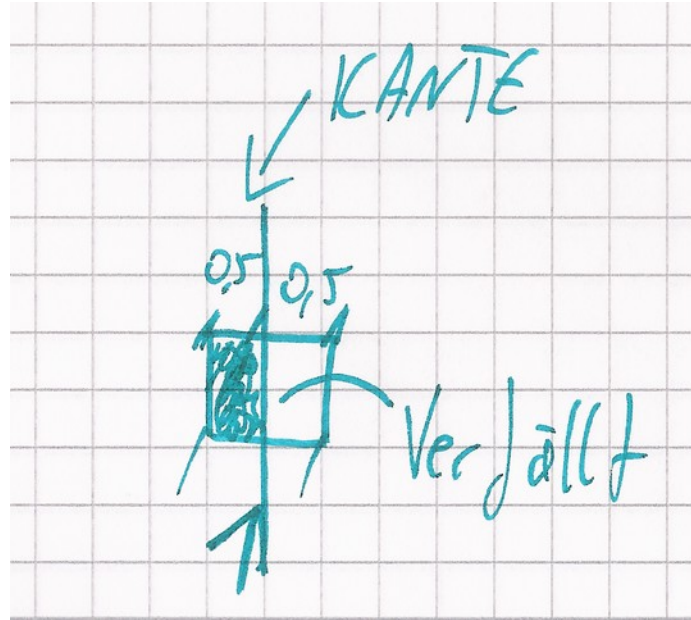


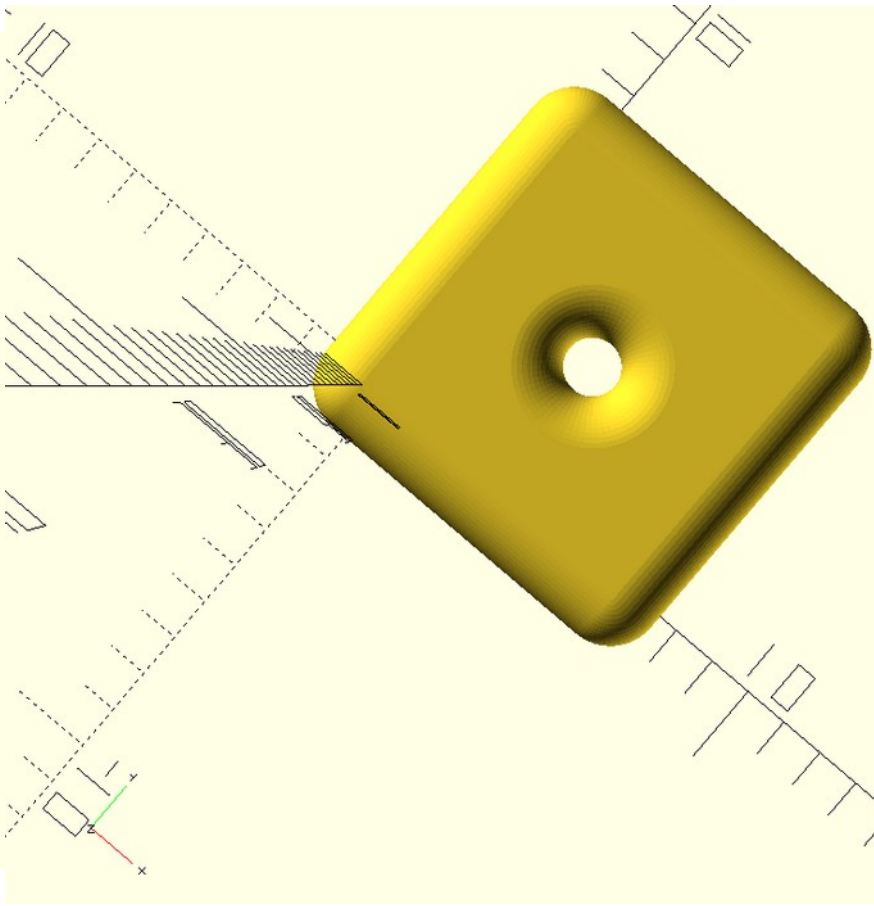
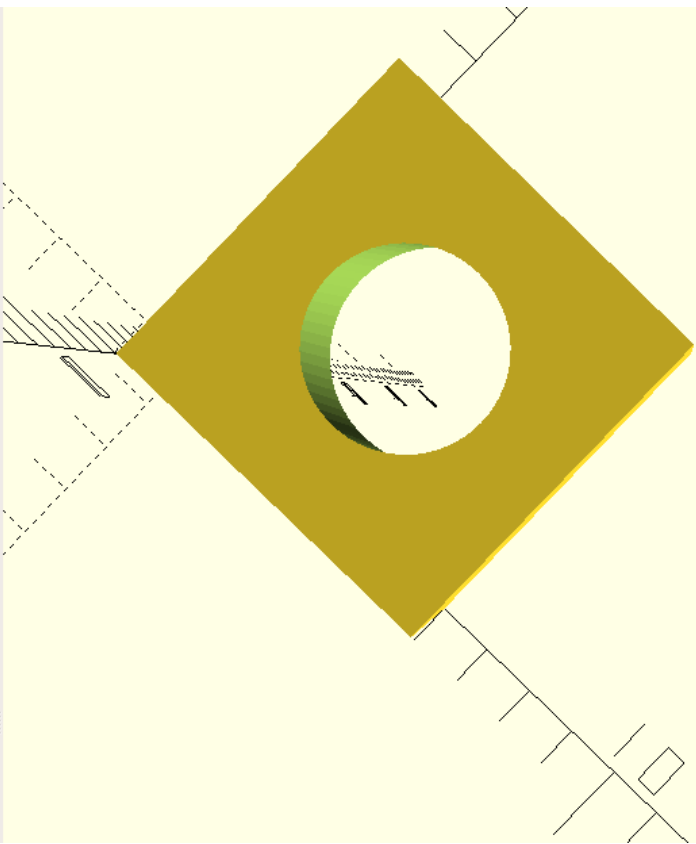
```
hull(){  
  translate([-10,-10,-10])  
    sphere(r=2);  
  translate([10,-10,-10])  
    sphere(r=2);  
  translate([-10,10,-10])  
    sphere(r=2);  
  translate([10,10,-10])  
    sphere(r=2);  
  translate([-10,-10,10])  
    sphere(r=2);  
  translate([10,-10,10])  
    sphere(r=2);  
  translate([-10,10,10])  
    sphere(r=2);  
  translate([10,10,10])  
    sphere(r=2);  
}
```

Minkowski



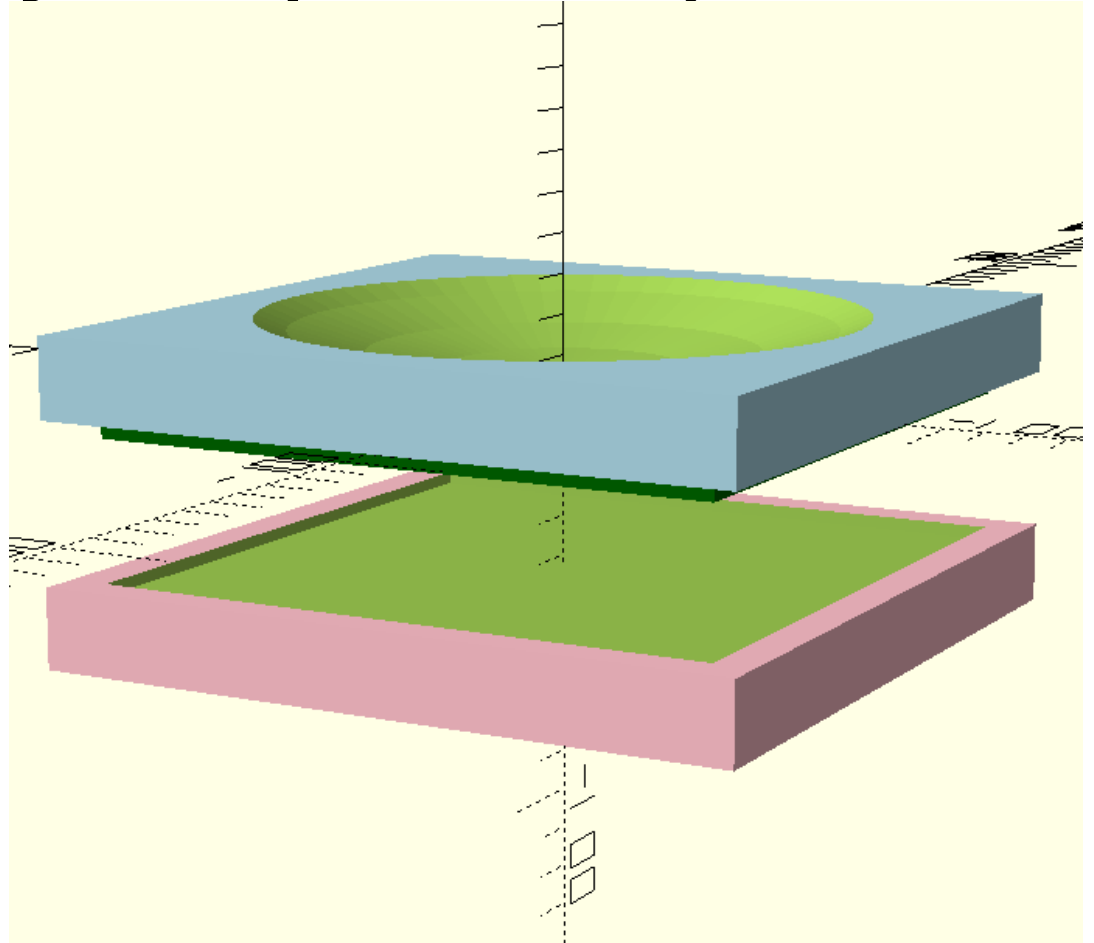
Minkowski - II

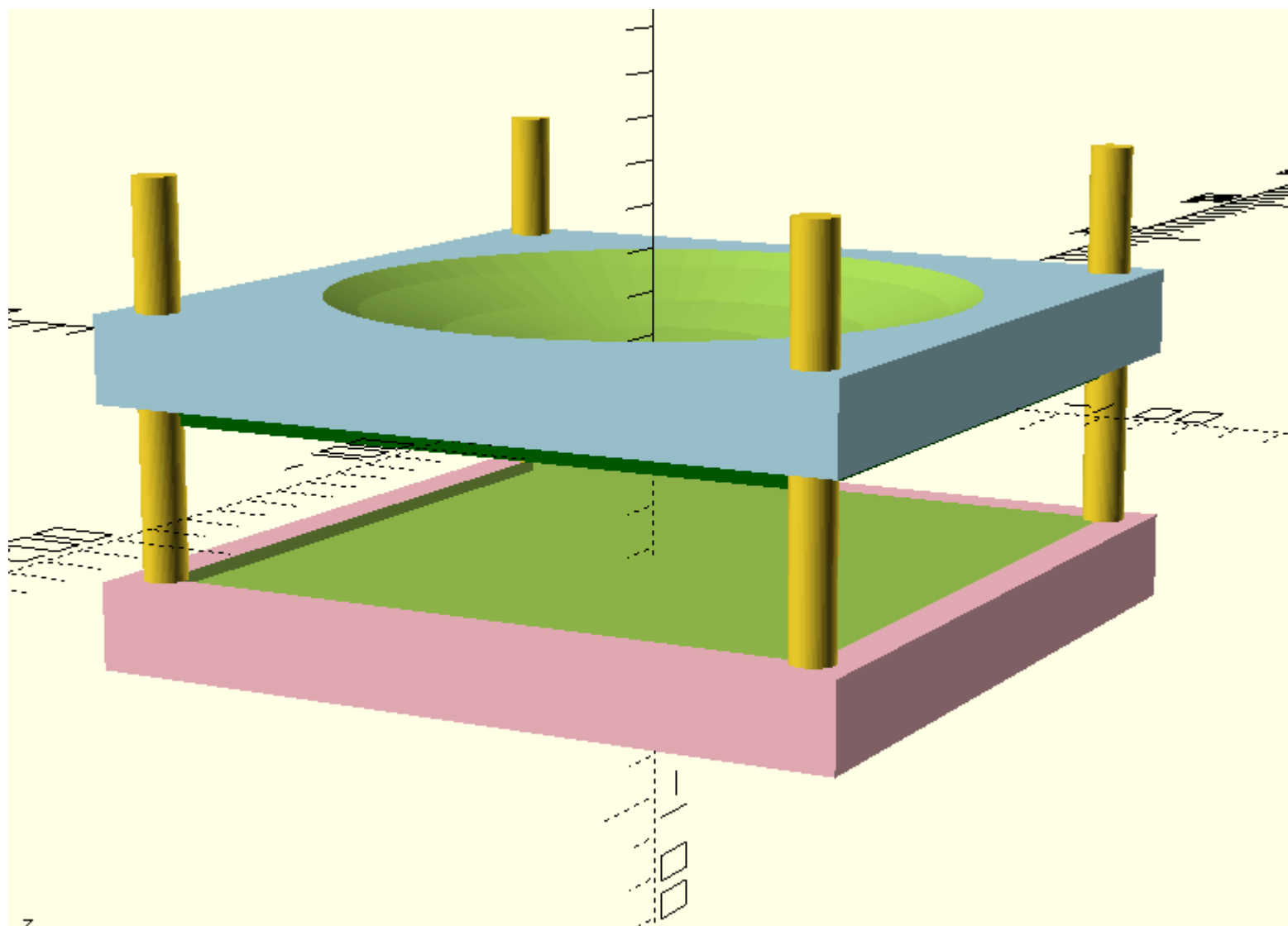




Colors (only for preview)

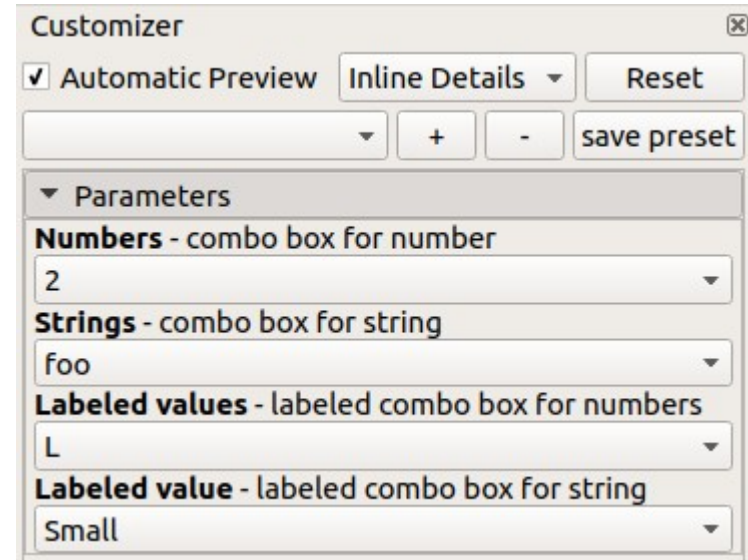
```
$fn=64;  
difference(){  
  color("lightblue")  
  translate([-90,-90,0])  
    cube([180, 180, 20]);  
  translate([0,0,205])  
    sphere(200);  
}
```





OpenSCAD customizer

- Expose parameters
- Users can adjust
- Web interface “exists”
 - Was Thingiverse
 - Dr. Vax also gave up



https://en.wikibooks.org/wiki/OpenSCAD_User_Manual/Customizer

?!?! → tamhan@tamoggemon.com

Instagram

Search

Log In

Sign Up



tam.hanna

Follow

1,620 posts

19.2k followers

5,046 following

Crazy Electronics Lab

Random snapshots from my #electronics lab, #testgear, #raspberrypi, #arduino, #pcbdesign and #engineering. Questions to tamhan@tamoggemon.com please!
www.elektor.com/technical-modeling-with-openscad



OpenSCA...



My Book



Video!



My Book (...)

POSTS

IGTV

TAGGED

