3D Modeling and Printing by Python

EuroPython 2016@Bilbao, Spain

2016/07/22

Takuro Wada

Hi!

和 田　　拓 朗

Takuro Wada

Kabuku Inc. (Japanese Startup)

Software Engineer

Providing services
related to 3D printing

taxpon

taxpon

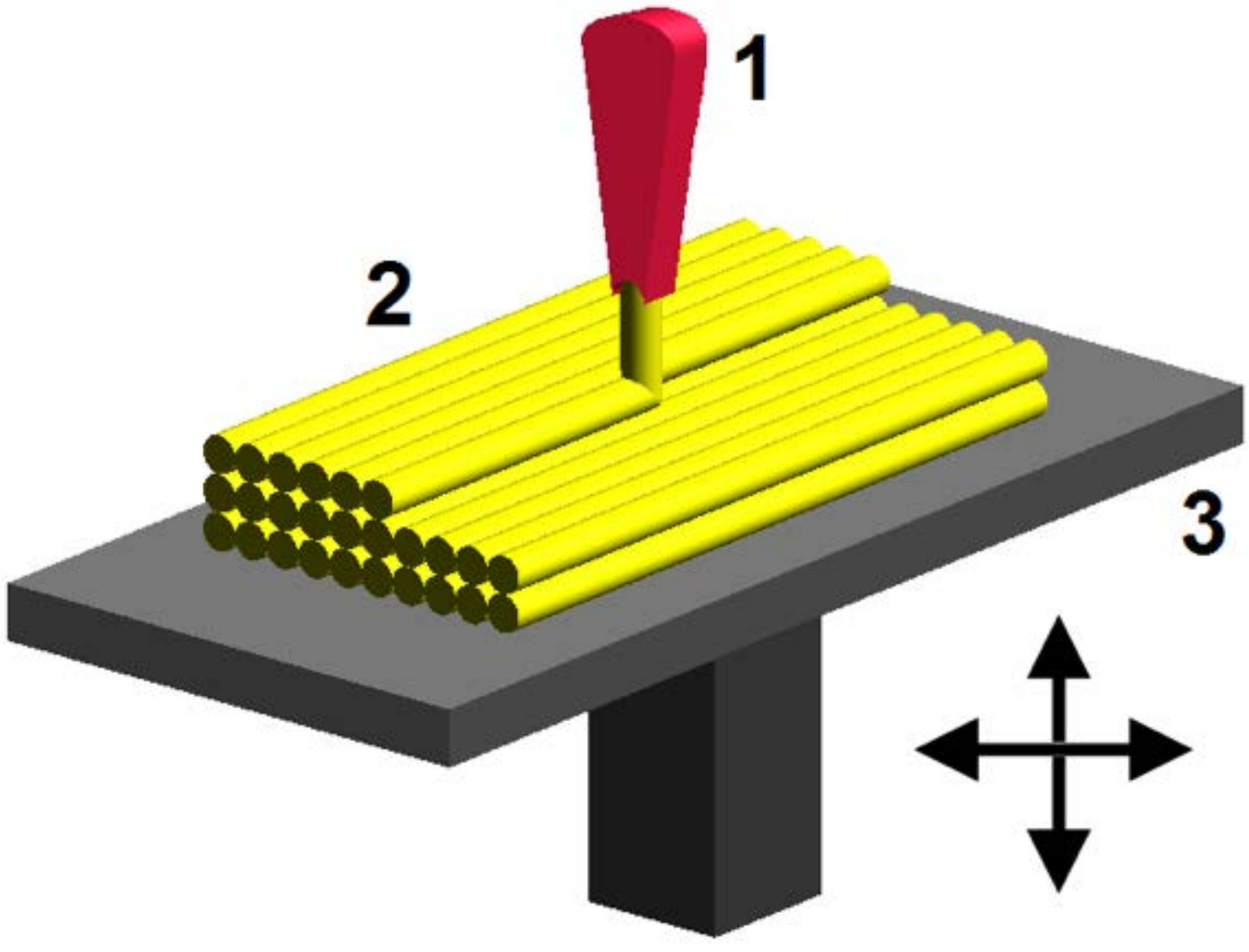
http://takuro.ws

What is 3D printing?

What is 3D printing?

Creating products by forming successive layers of material based on 3D Data

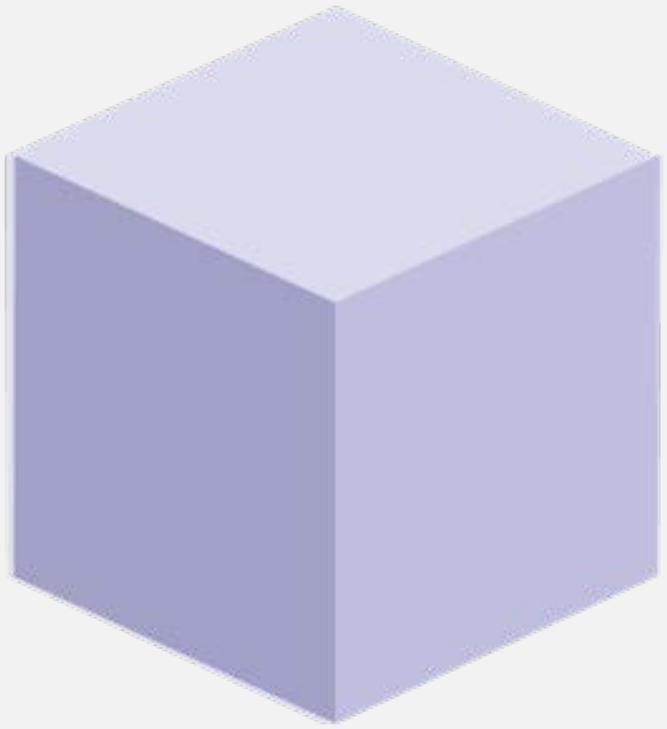
Example



https://en.wikipedia.org/wiki/3D_printing

What is required for 3D Printing?

What is required for 3D printing?



3D Data

3D Printer

How to create 3D Data?

3D Data Example
STL(ASCII)

```
solid Exported from Blender-2.75 (sub 0)
facet normal -1.000000 0.000000 0.000000
outer loop
vertex -1.000000 1.000000 -1.000000
vertex -1.000000 -1.000000 -1.000000
vertex -1.000000 -1.000000 1.000000
endloop
endfacet
facet normal -1.000000 0.000000 0.000000
outer loop
vertex -1.000000 -1.000000 1.000000
vertex -1.000000 1.000000 1.000000
vertex -1.000000 1.000000 -1.000000
endloop
endfacet
facet normal 0.000000 1.000000 0.000000
outer loop
vertex -1.000000 1.000000 1.000000
vertex 1.000000 1.000000 1.000000
vertex 1.000000 1.000000 -1.000000
endloop
endfacet
```

```
solid Exported from Blender-2.75 (sub 0)
facet normal -1.000000 0.000000 0.000000
outer loop
vertex -1.000000 1.000000 -1.000000
vertex -1.000000 -1.000000 -1.000000
vertex -1.000000 -1.000000 1.000000
endloop
endfacet
facet normal -1.000000 0.000000 0.000000
outer loop
vertex -1.000000 -1.000000 1.000000
vertex -1.000000 1.000000 1.000000
vertex -1.000000 1.000000 -1.000000
endloop
endfacet
facet normal 0.000000 1.000000 0.000000
outer loop
vertex -1.000000 1.000000 1.000000
vertex 1.000000 1.000000 1.000000
vertex 1.000000 1.000000 -1.000000
endloop
endfacet
```

Definition of Triangle (polygon)

```
solid Exported from Blender-2.75 (sub 0)
facet normal -1.000000 0.000000 0.000000
outer loop
vertex -1.000000 1.000000 -1.000000
vertex -1.000000 -1.000000 -1.000000
vertex -1.000000 -1.000000 1.000000
endloop
endfacet
facet normal -1.000000 0.000000 0.000000
outer loop
vertex -1.000000 -1.000000 1.000000
vertex -1.000000 1.000000 1.000000
vertex -1.000000 1.000000 -1.000000
endloop
endfacet
facet normal 0.000000 1.000000 0.000000
outer loop
vertex -1.000000 1.000000 1.000000
vertex 1.000000 1.000000 1.000000
vertex 1.000000 1.000000 -1.000000
endloop
endfacet
```

Definition of Triangle (polygon)

Successive triangle definitions

Format is so simple

But
creating 3D Data
from scratch
is so hard task

3DCG/CAD software

MAYA

CATIA

AUTODESK 3DS MAX 2015

SOLIDWORKS

AUTODESK AUTOCAD

Shade 3D

blender

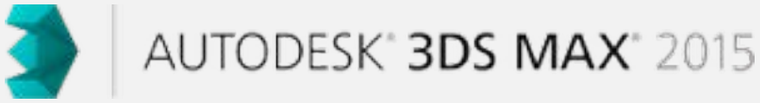
AUTODESK FUSION 360

MODO

OpenScad

Rhinoceros
modeling tools for designers

Many 3D software provides Python API



Many 3D software provides Python API

MAYA

CATIA

AUTODESK 3DS MAX 2015

SOLIDWORKS

AUTODESK AUTOCAD

Shade 3D

blender

AUTODESK FUSION 360

MODO

Rhinoceros
modeling tools for designers

You can manipulate 3D data via Python

Script Modeling

- You can create 3D model even if you are not familiar with 3DCG/CAD software
- You can create shapes that are difficult for hand creation
- Modifying shape is easy

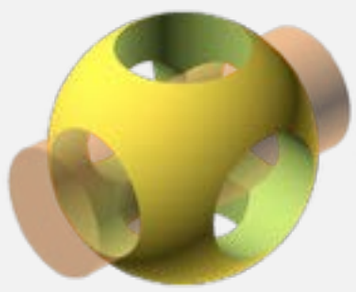
Script Modeling

- You can create 3D model even if you are not familiar with 3DCG/CAD software
- You can create shapes that are difficult for hand creation
- Modifying shape is easy

```
_人人人人人人人_
>  GREAT ! !  <
￣Y^Y^Y^Y^Y^Y^Y￣
```

```
_人人人人人人人_
>FANTASTIC ! ! <
￣Y^Y^Y^Y^Y^Y^Y￣
```

Today’s software



OpenScad

人人人人人人人

> FREE！！ <

￣Y^Y^Y^Y^Y^Y￣

Script Modeling with Blender and Python

What is Blender?

- Developed since 1995
- Open source software, Multi platform
- Language: C, C++, Python
  - Python is used as API interface
  - Blender has its own python interpreter in the software（3.5.1 in Blender2.77a） asyncio!!

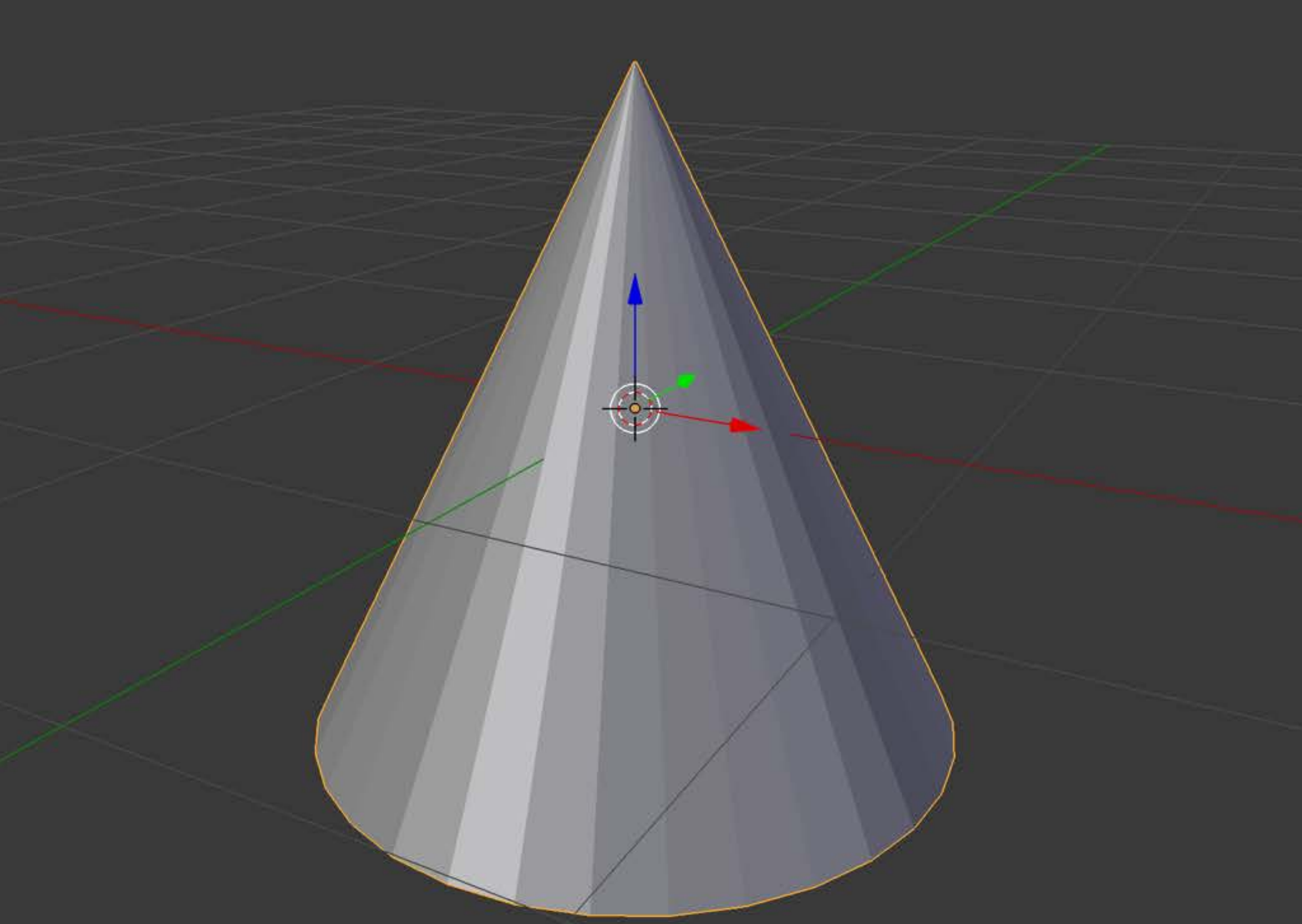
```python
import bpy


def delete_all():
    for item in bpy.context.scene.objects:
        bpy.context.scene.objects.unlink(item)

    for item in bpy.data.objects:
        bpy.data.objects.remove(item)

    for item in bpy.data.meshes:
        bpy.data.meshes.remove(item)

    for item in bpy.data.materials:
        bpy.data.materials.remove(item)


def add_cone():
    bpy.ops.mesh.primitive_cone_add()


if __name__ == "__main__":
    delete_all()
    add_cone()
```

Delete default Object

Add cone

```
>>> blender -P tut1.py
```

-P option and script name to execute



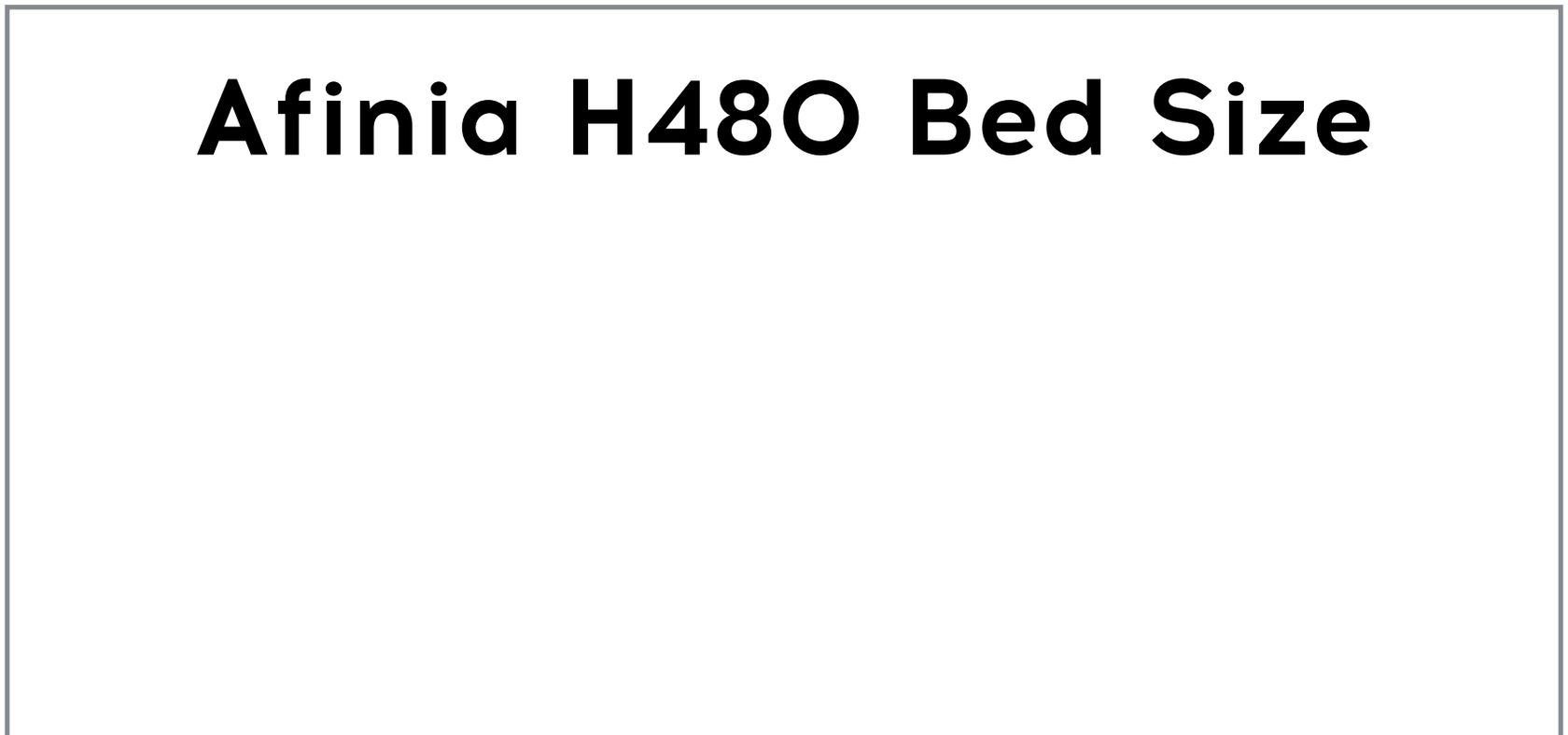
Making Chain

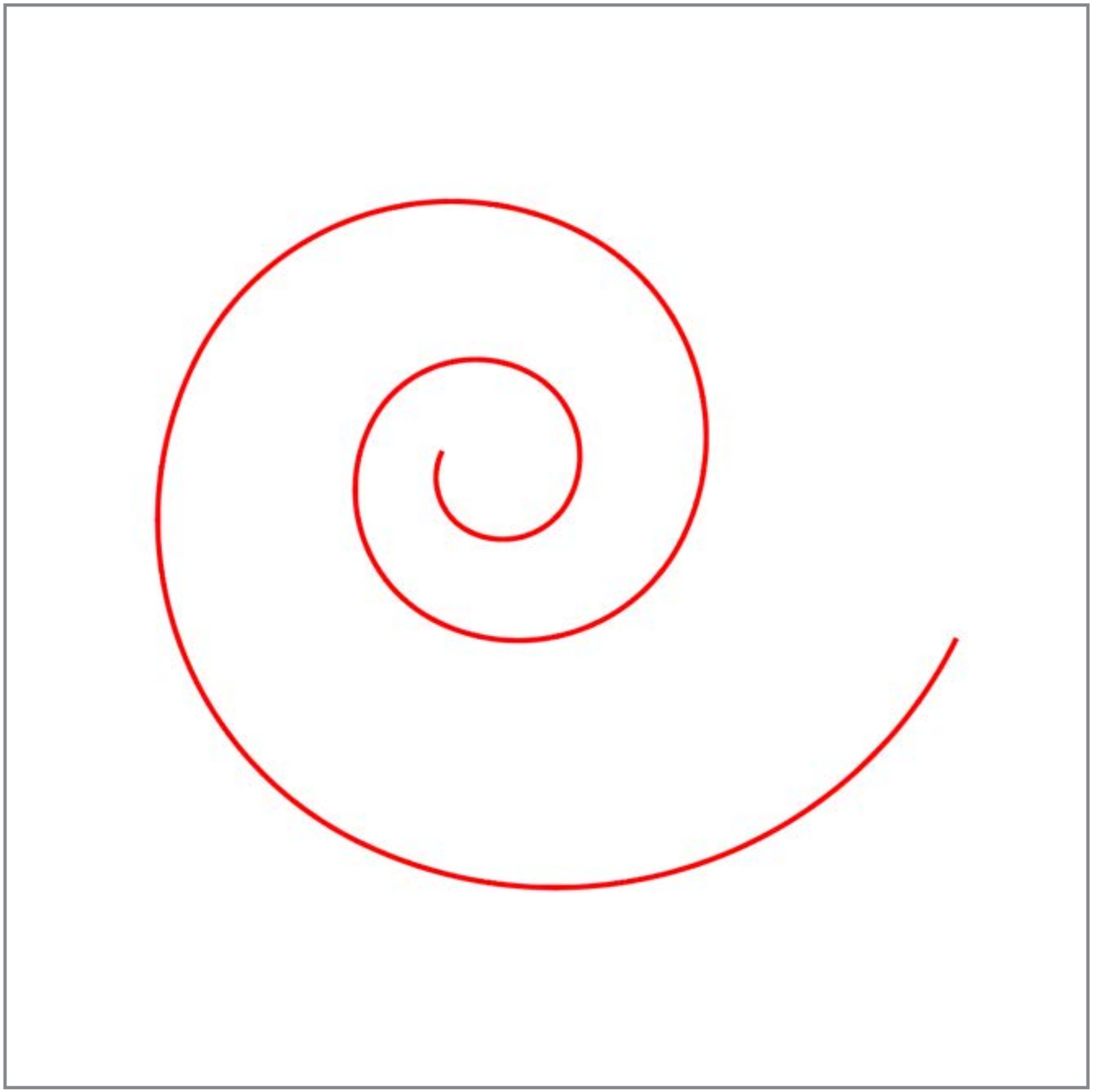
https://www.myminifactory.com/object/5943

https://www.youtube.com/watch?v=z1OSXnCG-jk

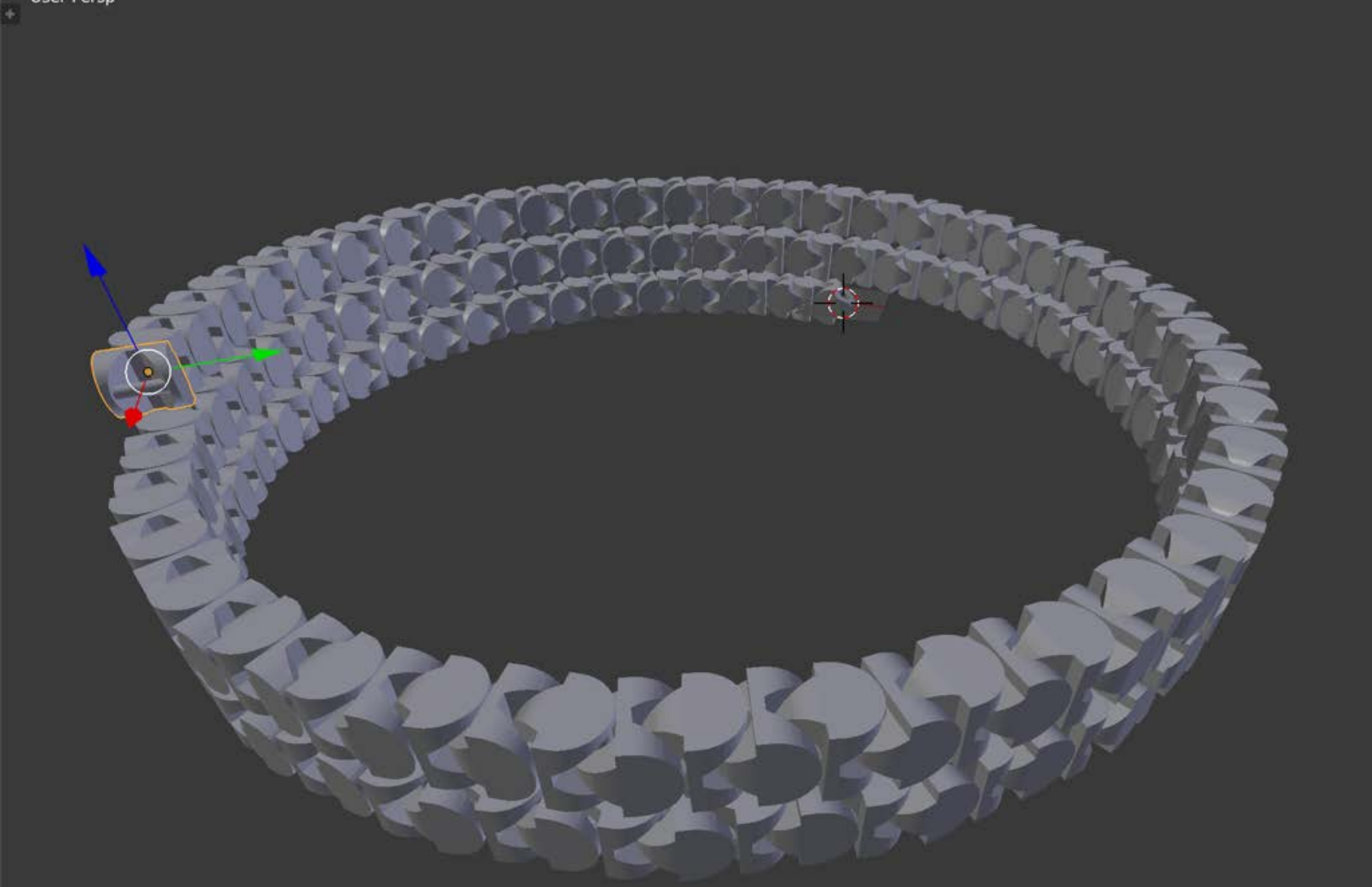
http://afinia.com/3d-printers/h480/

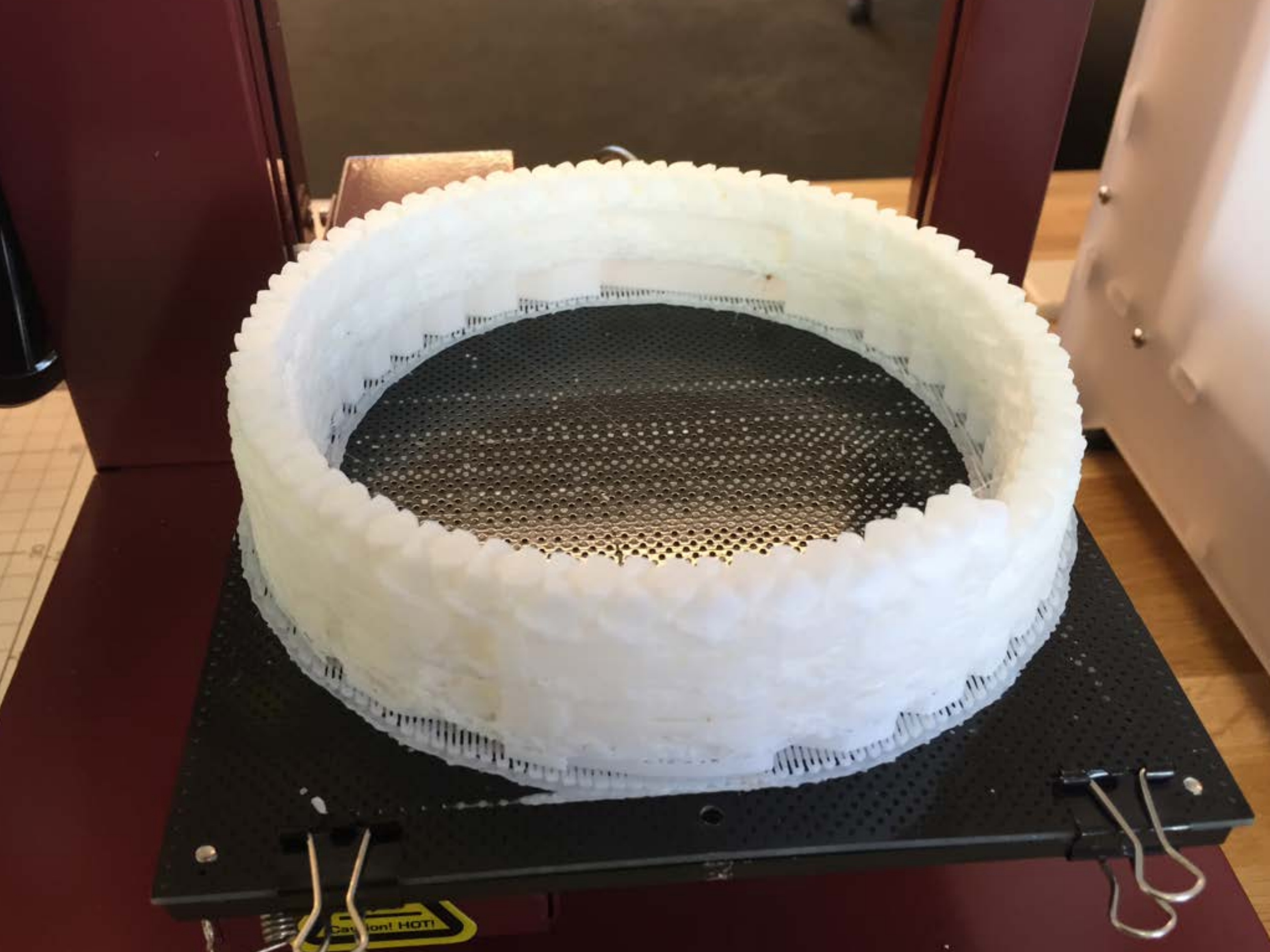
Afinia H480 Bed Size





```
55
56
57  def make_relation():
58      """
59      :return:
60      """
61      last_unit = None
62      for i, cur_unit in enumerate(reversed(bpy.context.scene.objects)):
63          bpy.ops.object.select_all(action='DESELECT')
64          if i == 0:
65              last_unit = cur_unit
66              continue
67          last_unit.select = True
68          cur_unit.select = True
69          bpy.context.scene.objects.active = last_unit
70
71          # Create parent relation
72          bpy.ops.object.parent_set()
73
74          # Save current unit
75          last_unit = cur_unit
76
77
78  def add_rotation():
79      """
80      Add rotation to each chain unit
81      :return:
82      """
83      horizontal_rot = 17
84      vertical_rot = -2.5
85
86      for area in bpy.context.screen.areas:
87          if area.type == 'VIEW_3D':
88              area.spaces[0].transform_orientation = 'LOCAL'
89
90      for i, cur_unit in enumerate(bpy.context.scene.objects):
91          bpy.ops.object.select_all(action='DESELECT')
92          if i == 0:
93              continue
94
95          cur_unit.select = True
```




Caution! HOT!

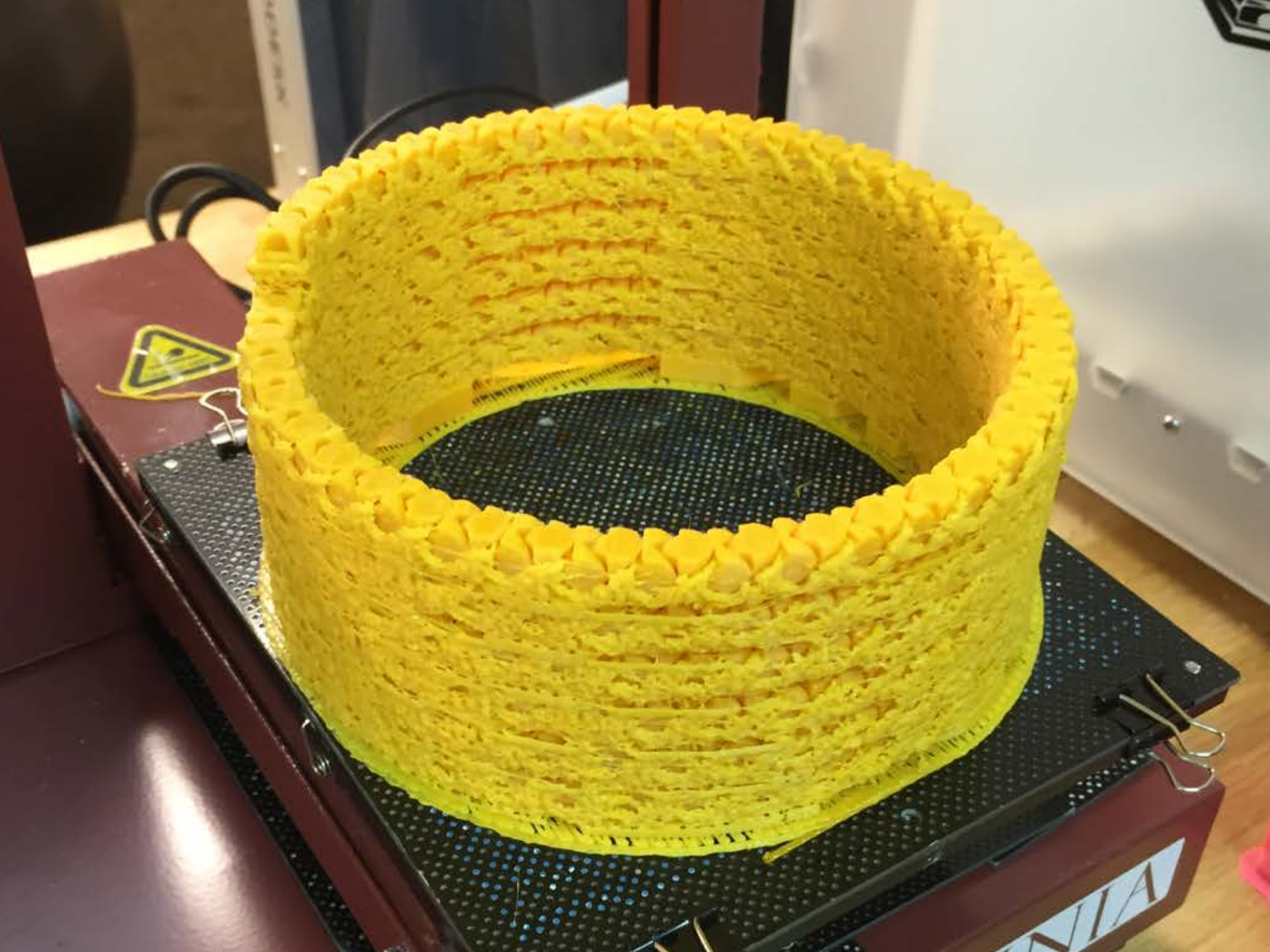
Longer one ...

jumping rope...

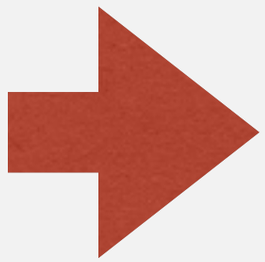
```
def add_rotation():
    """
    Add rotation to each chain unit
    :return:
    """
    horizontal_rot = 17
    vertical_rot = -2.5

    for area in bpy.context.screen.areas:
        if area.type == 'VIEW_3D':
            area.spaces[0].transform_orientation = 'LOCAL'

    for i, cur_unit in enumerate(bpy.context.scene.objects):
        bpy.ops.object.select_all(action='DESELECT')
        if i == 0:
            continue

        cur_unit.select = True
        bpy.context.scene.objects.active = cur_unit
        # continue

        if i % 4 == 1:
            cur_unit.rotation_euler[1] = deg_to_rad(vertical_rot)

        if i % 4 == 2:
            continue

        if i % 4 == 3:
            cur_unit.rotation_euler[1] = deg_to_rad(-vertical_rot)

        if i % 4 == 0:
            cur_unit.rotation_euler[2] = deg_to_rad(horizontal_rot)


if __name__ == '__main__':
    delete_all()
    load_stl()
    make_chain(500)
    make_relation()
    add_rotation()
```

```
def add_rotation():
    """
    Add rotation to each chain unit
    :return:
    """
    horizontal_rot = 17
    vertical_rot = -2.5

    for area in bpy.context.screen.areas:
        if area.type == 'VIEW_3D':
            area.spaces[0].transform_orientation = 'LOCAL'

    for i, cur_unit in enumerate(bpy.context.scene.objects):
        bpy.ops.object.select_all(action='DESELECT')
        if i == 0:
            continue

        cur_unit.select = True
        bpy.context.scene.objects.active = cur_unit
        # continue

        if i % 4 == 1:
            cur_unit.rotation_euler[1] = deg_to_rad(vertical_rot)

        if i % 4 == 2:
            continue

        if i % 4 == 3:
            cur_unit.rotation_euler[1] = deg_to_rad(-vertical_rot)

        if i % 4 == 0:
            cur_unit.rotation_euler[2] = deg_to_rad(horizontal_rot)


if __name__ == '__main_
    delete_all()
    load_stl()
    make_chain(500)
    make_relation()
    add_rotation()
```

```
make_chain(500)
```




MINECRAFT



Connect blender and minecraft world using Python

Convert this kind of 3D data
to Minecraft Blocks!!

Implementation Overview

Consists of two main pars.

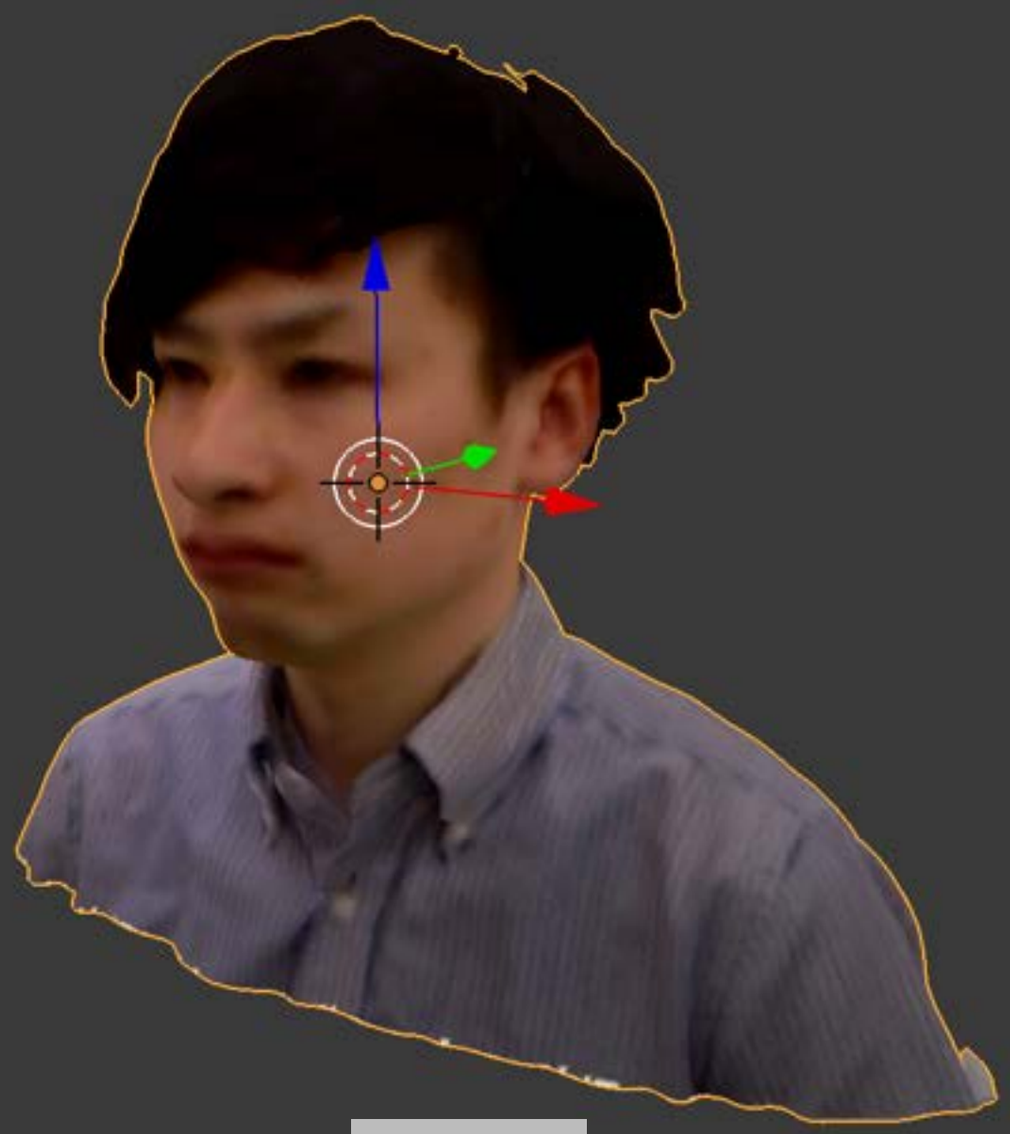
Convert 3D model into Blocks

Converting

Transport blocks into Minecraft

Transporting

Converting

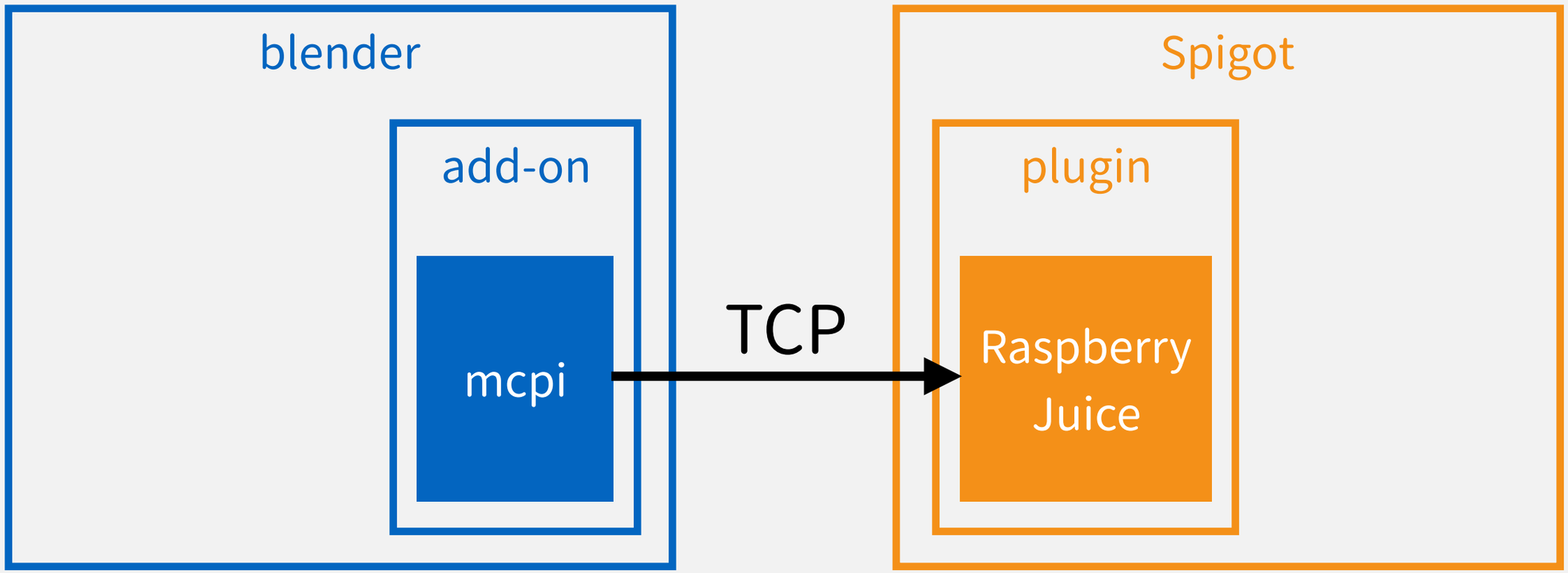


before

after

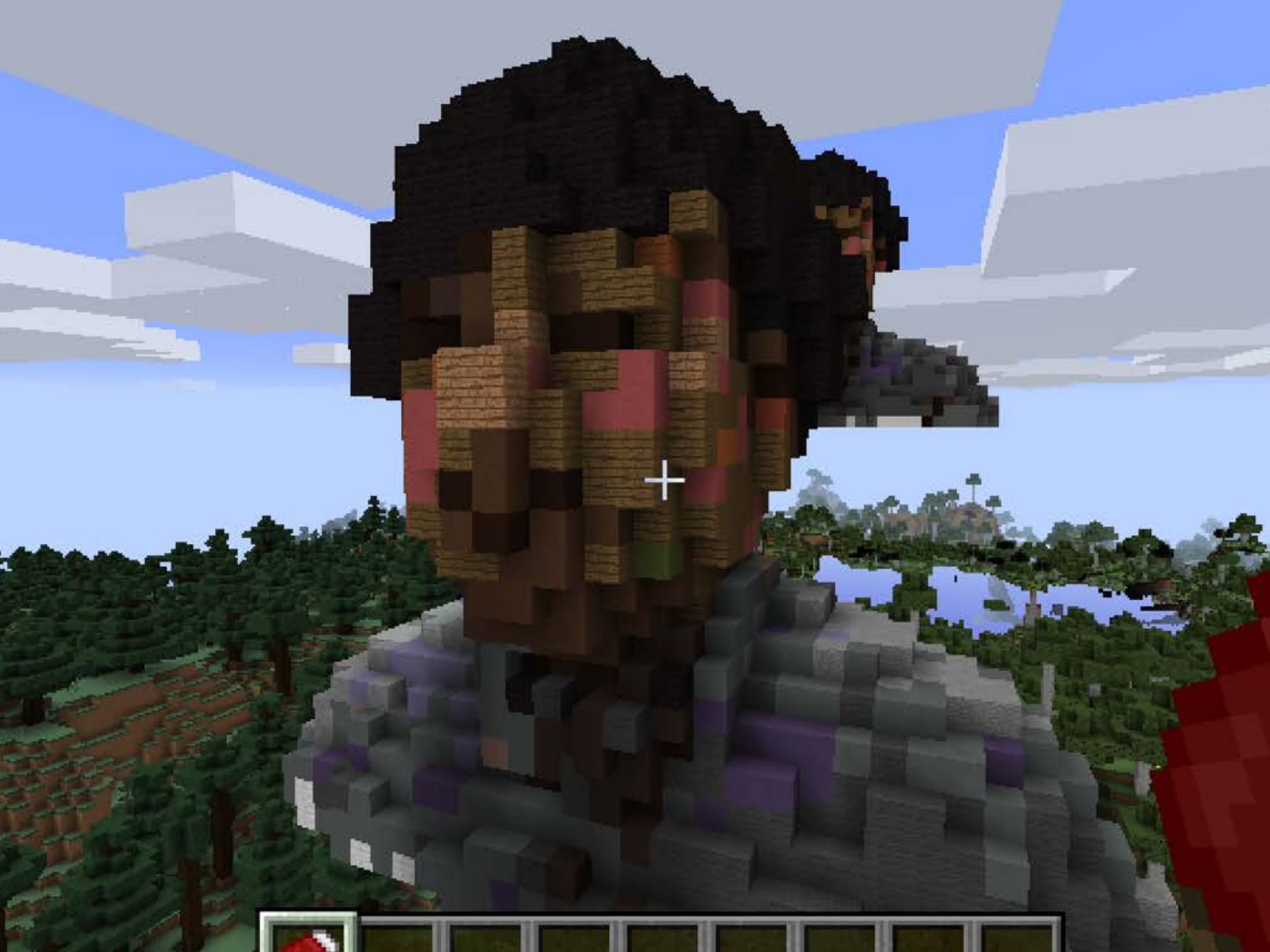
Level of Conversion

Transporting



- Use mcpi (python module for minecraft) module

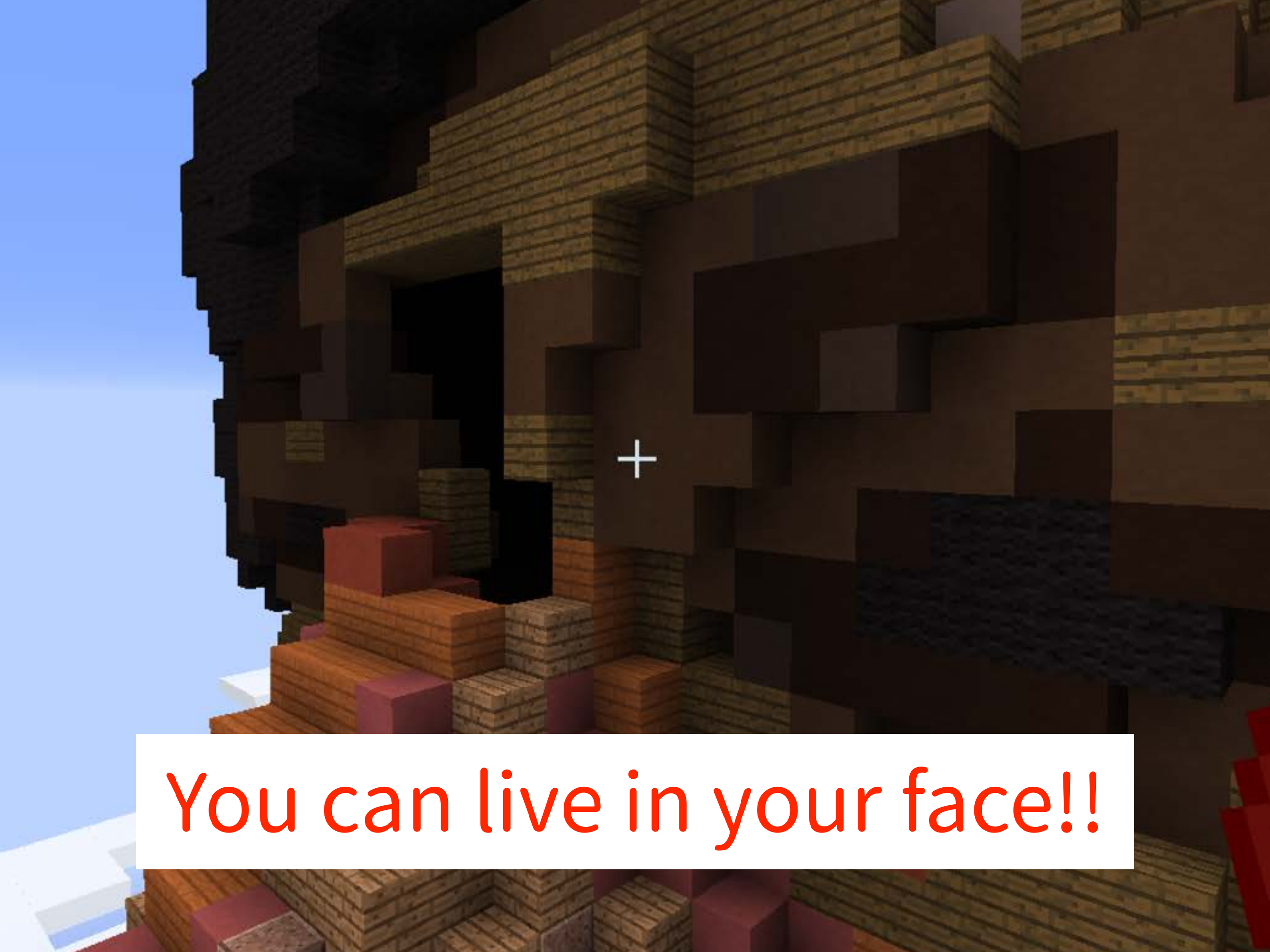
- Spigot=Minecraft mod server
- Raspberry Juice is plugin for Spigot



\(^o^)/

__人人人人人人人__
> Success!! <
￣Y^Y^Y^Y^Y^Y^Y^Y￣


Floating Faces!!


You can live in your face!!

Github

- b2mine
  - https://github.com/taxpon/b2mine

README.rst

b2mine

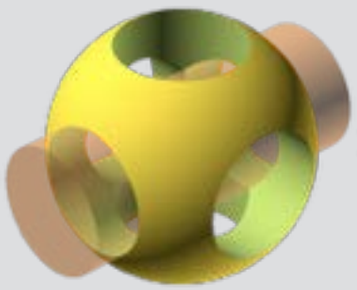
Blender add-on to transfer color object to minecraft.

What?

This add-on convert colored 3D model into block from. You can send the converted model to minecraft server embedding raspberry juice API.

Convert

Script Modeling
with OpenSCAD and Python

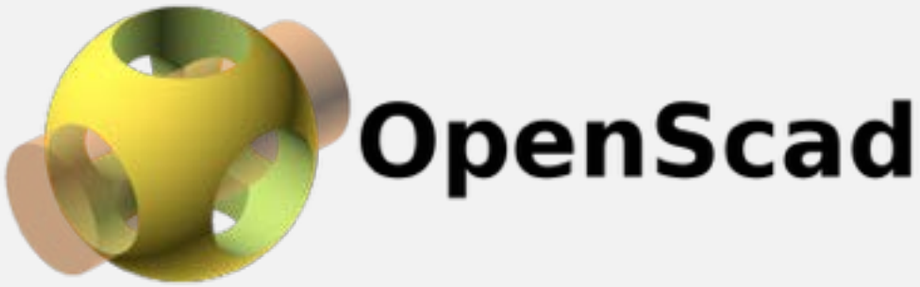


OpenScad

python™

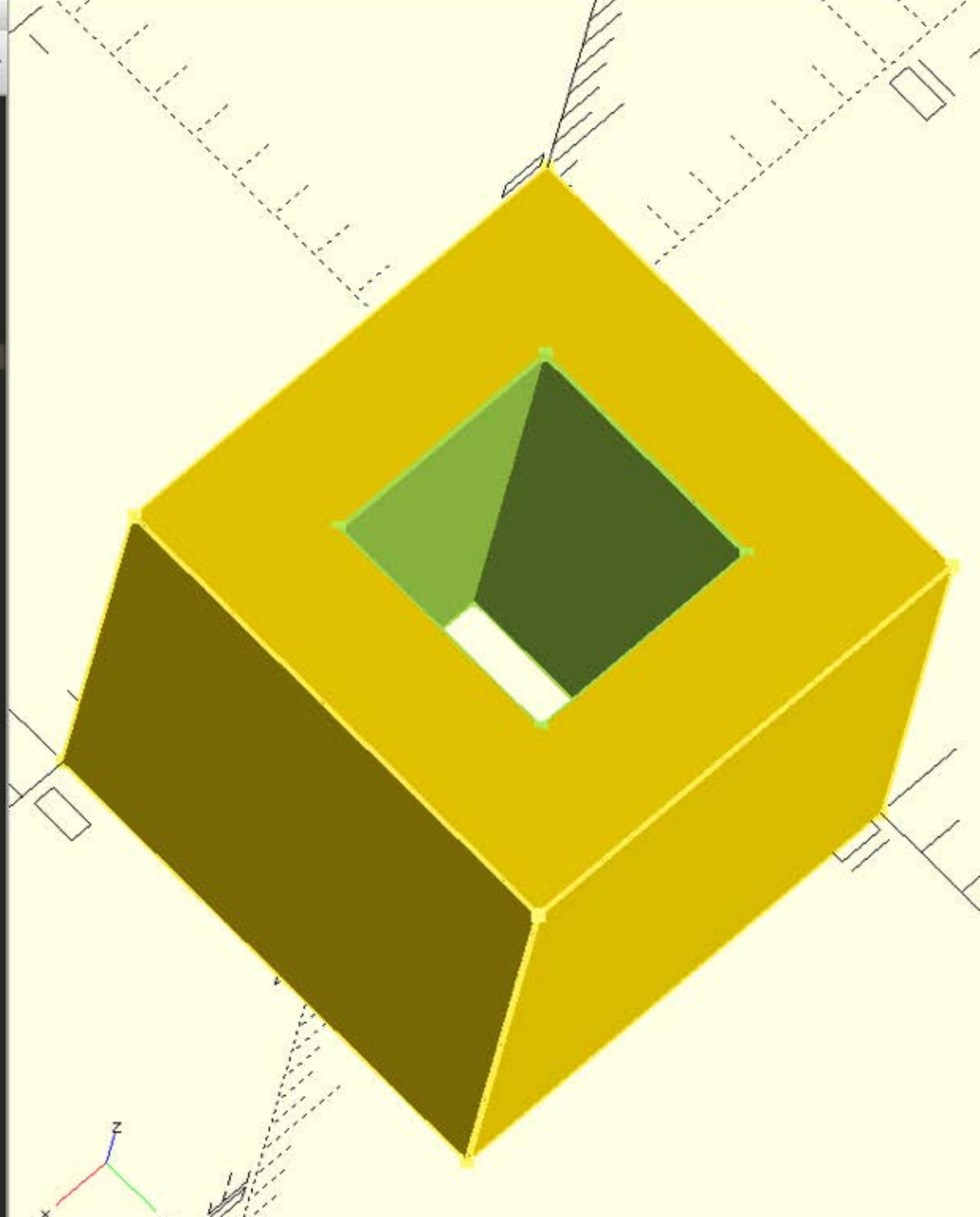
What is OpenSCAD?

- CAD software, Open source (GPL)
- Multi platform (Win, Mac, Linux)
- Implemented by C++(Qt)
- Create 3D model **using original programming language**
- No GUI to manipulate 3D data



Editor

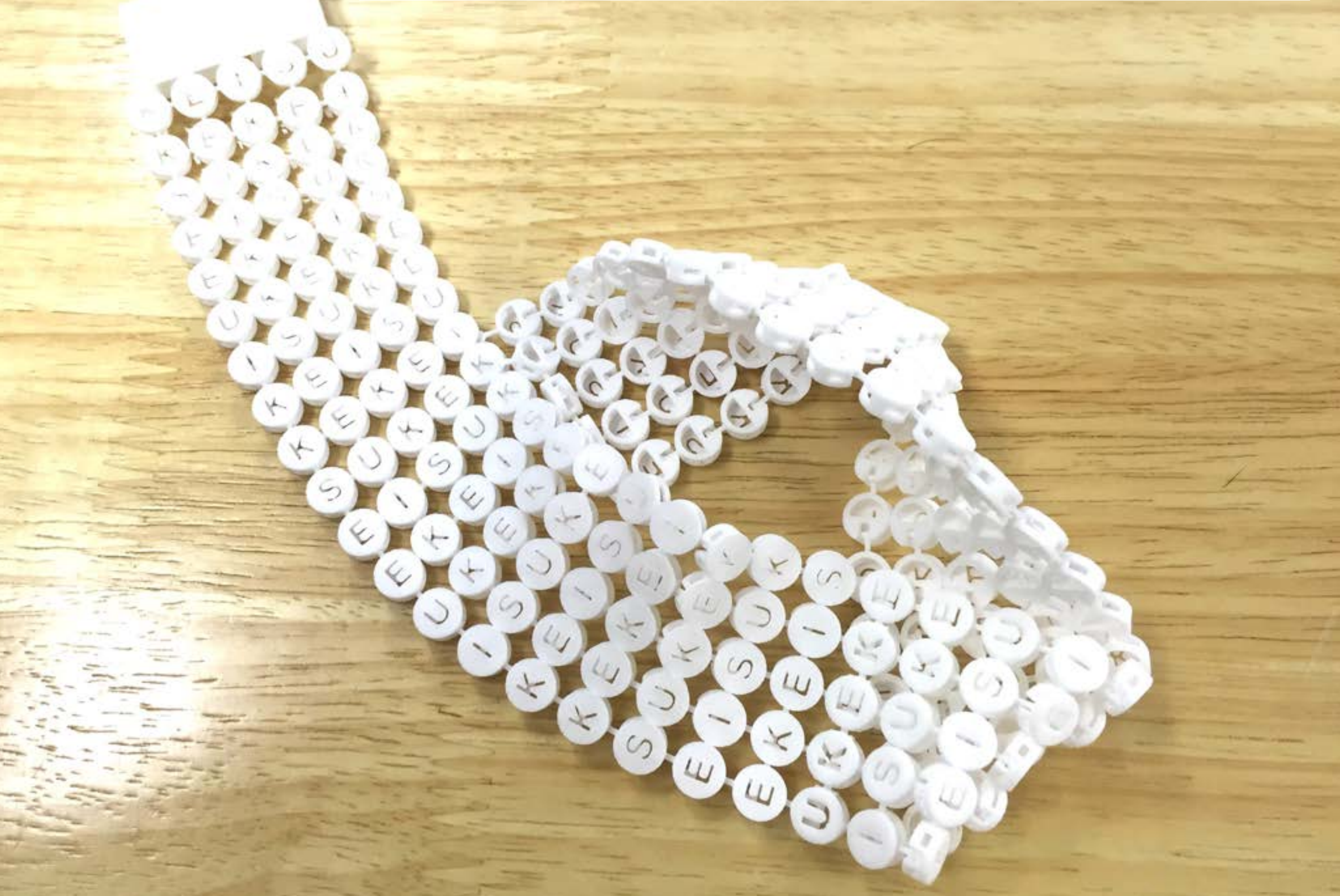
```
1
2  difference() {
3      cube(10);
4
5      translate([2.5, 2.5, -3]){
6          cube([5, 5, 20]);
7      }
8  }
9
10
11
```

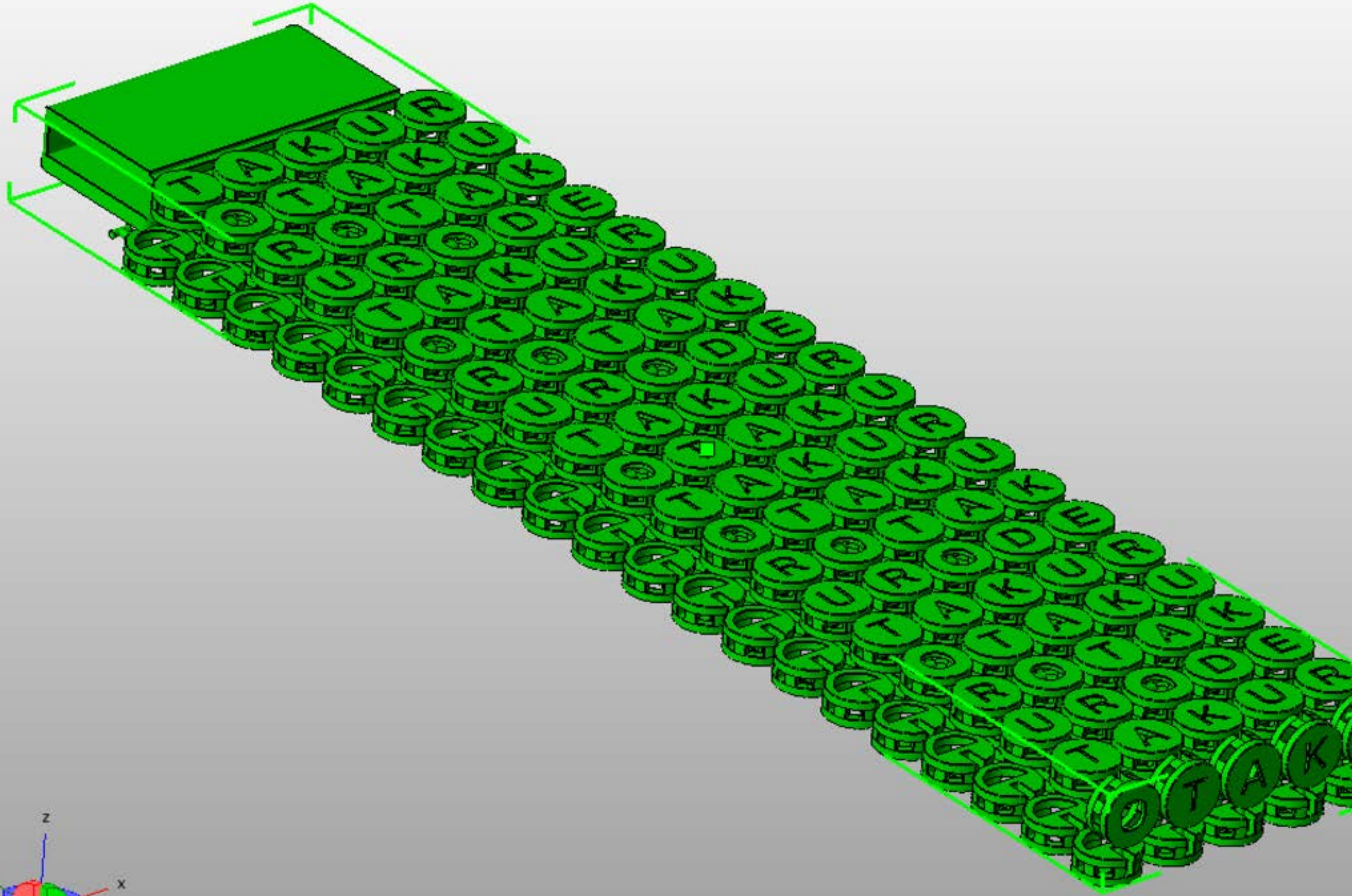


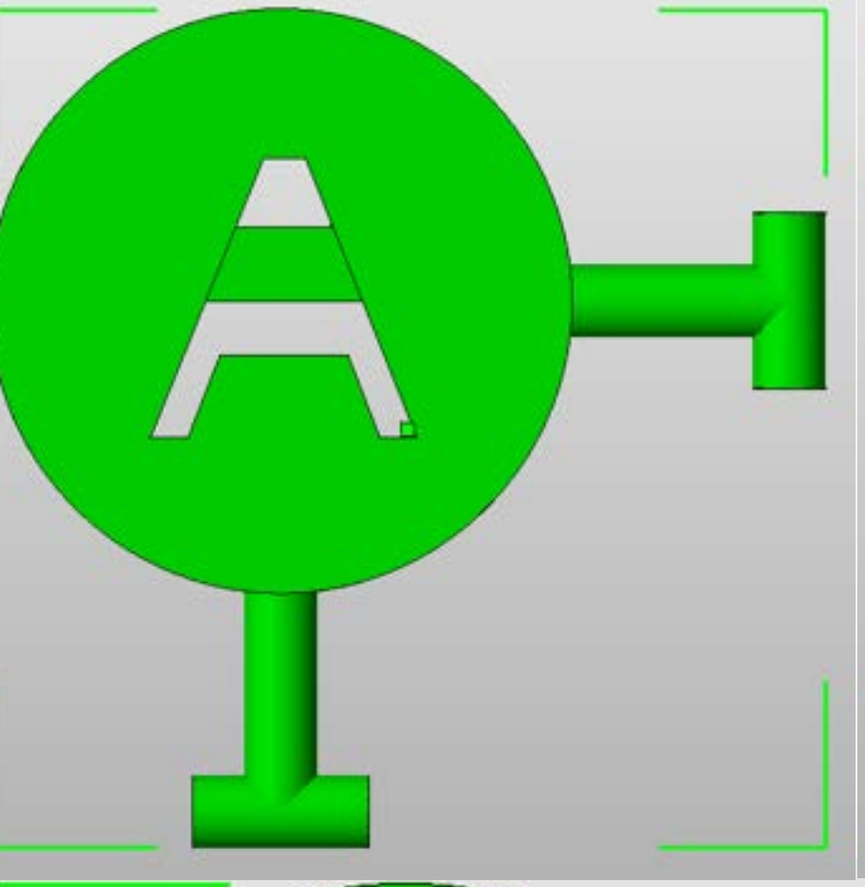
Feature of OpenSCAD

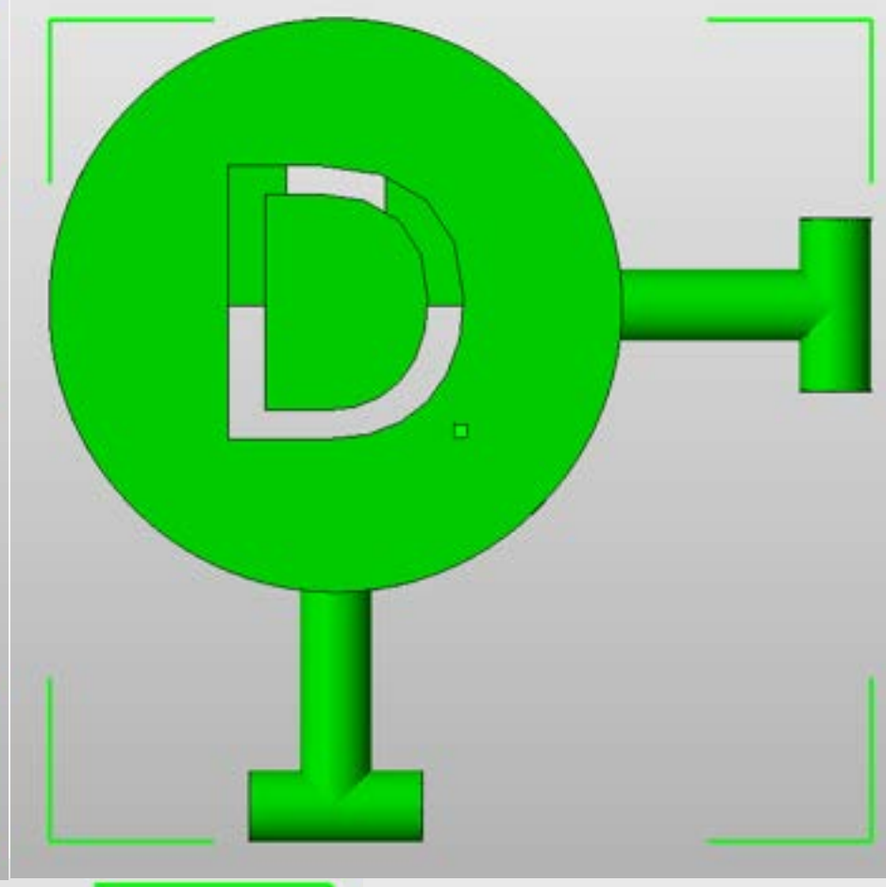
- Very good **for Software Engineer**
- Written script can be kicked from command line
  - Generate model in Server
  - Create multiple model with command

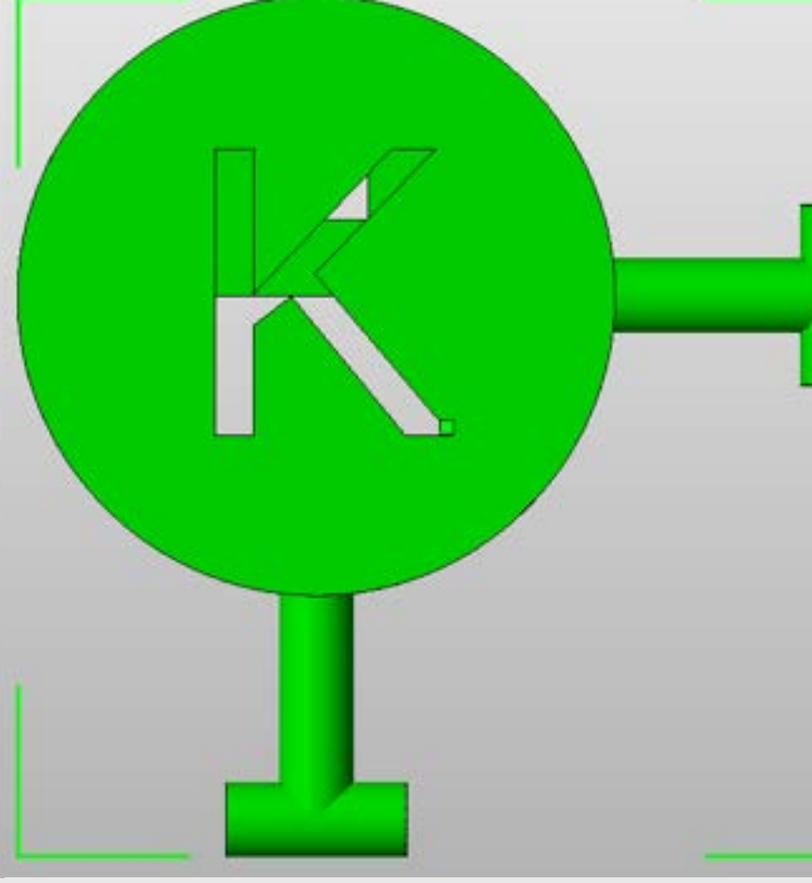
Neck tie composed of multiple characters

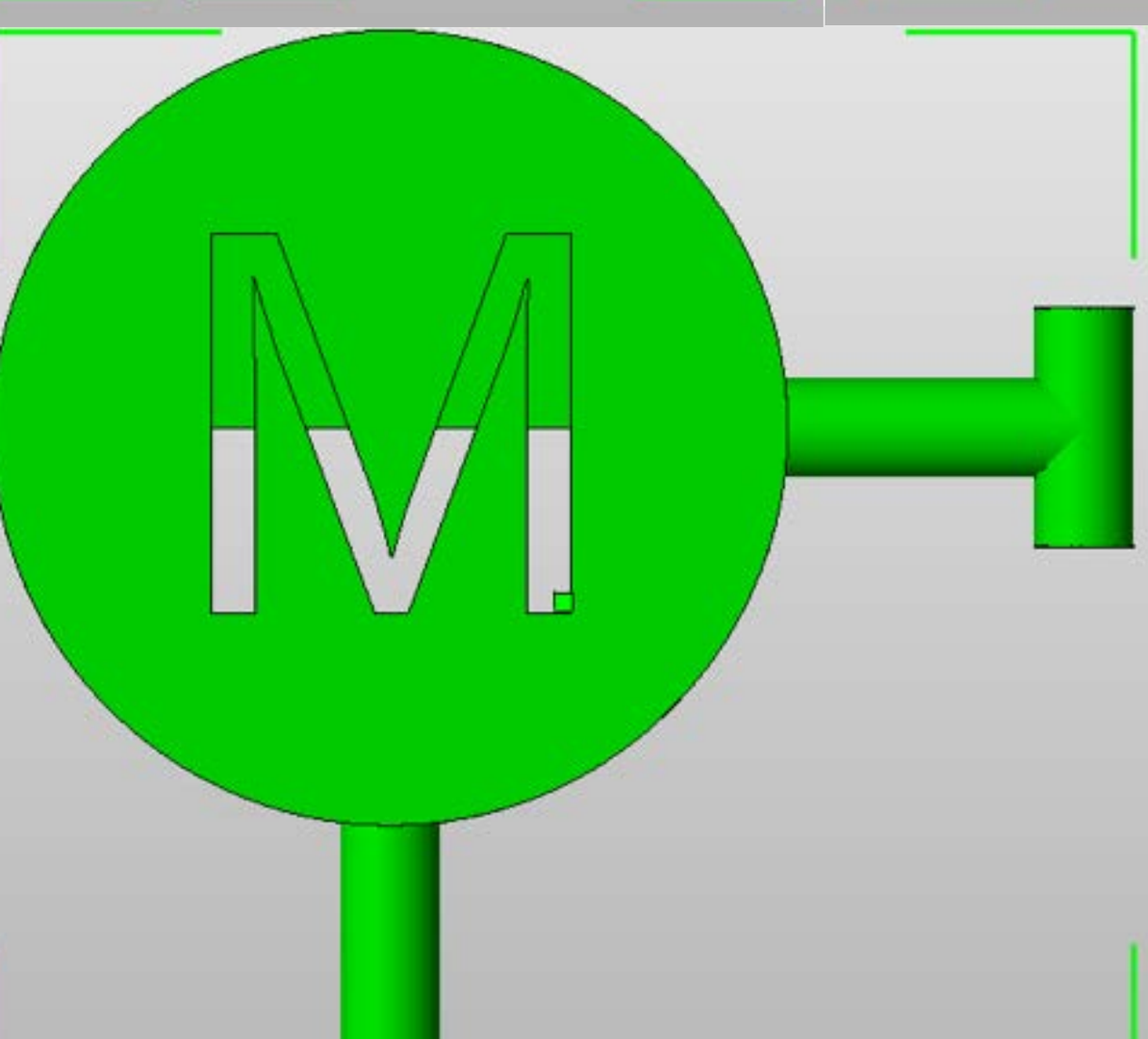


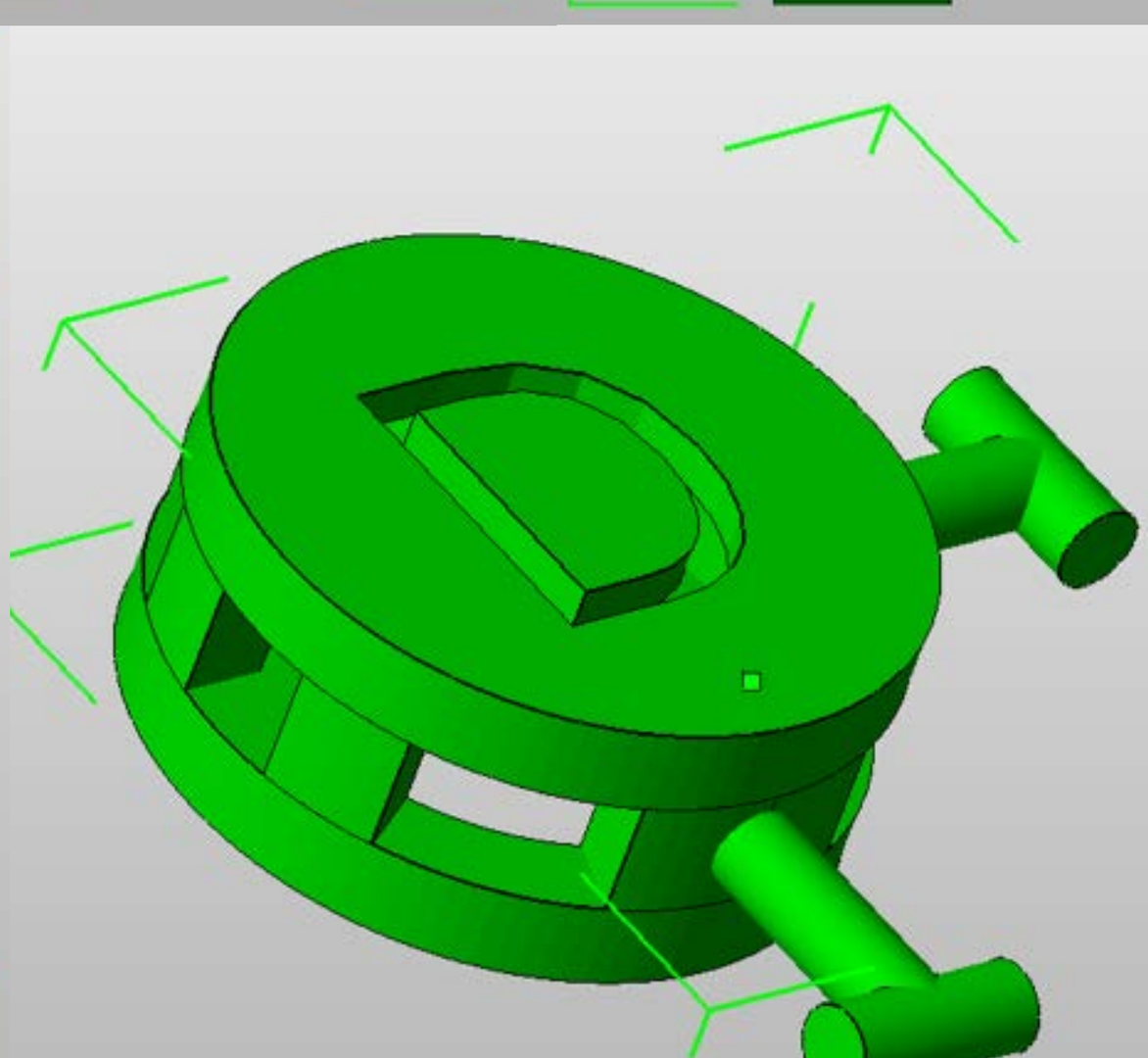

Z
X


A


D


K


M



A copy.stl

A_nr copy.stl

A_nr.stl

A.stl

B copy.stl

B_nr copy.stl

B_nr.stl

B.stl

C copy.stl

C_nr copy.stl

C_nr.stl

C.stl

D copy.stl

D_nr copy.stl

D_nr.stl

D.stl

E copy.stl

E_nr copy.stl

E_nr.stl

E.stl

F copy.stl

F_nr copy.stl

F_nr.stl

F.stl

G copy.stl

G_nr copy.stl

G_nr.stl

G.stl

create_new_unit.scad

Editor

```
16
17  ////////////////////////////////////////////////////////
18  // Common Func
19
20  // Base Arc
21  module arc(height, in_radius, out_radius, degrees){
22      difference(){
23          rotate_extrude($fn=100)
24              translate([in_radius, 0, 0])
25                  square([out_radius - in_radius, height]);
26
27          if (degrees > 0){
28              translate([0, -(out_radius + 1), -.5])
29                  cube([out_radius+1, (out_radius+1)*2, height + 1]);
30
31              rotate([0, 0, 180-degrees])
32                  translate([0, -(out_radius + 1), -.5])
33                      cube([out_radius+1, (out_radius+1)*2, height + 1]);
34          }
35      }
36  }
37
38  // Joint Receive
39  module joint_receive(){
40      difference(){
41          rotate([0, 0, 210])
42              arc(middle_height, in_rad, out_rad, 60);
43          translate([-joint_slit_width/2.0, 0, -middle_height])
44              cube([joint_slit_width, 10, middle_height*3]);
45      }
46  }
47
48  module text_part(letter){
49      translate([0, 0, -5])
50          linear_extrude(height = 10) {
51              text(text=letter, halign="center", valign="center", size=font_size);
52          }
53  }
54
55  ////////////////////////////////////////////////////////
56  // Unit part gen
57
58  module top_part(){
59      difference(){
60          cylinder(top_height, out_rad, out_rad, $fn=100);
61          translate([0, 0, -top_height/2.0]){
62              cylinder(top_height, in_rad, in_rad, $fn=100);
63          }
64      }
65  }
66
67  module bottom_part(){
68      difference(){
69          union(){
70              arc(bottom_height, in_rad, out_rad, 0);
71              rotate([0, 0, -90])
72                  arc(bottom_height, 0, out_rad, 180);
73          }
74
75          // Bottom Slit
76          translate([-joint_slit_width/2.0, 1, -bottom_height])
77              cube([joint_slit_width, 5, bottom_height*3]);
78      }
```

Console

```
Compiling design (CSG Tree generation)...
Compiling design (CSG Products generation)...
Geometries in cache: 36
Geometry cache size in bytes: 251272
CGAL Polyhedrons in cache: 86
CGAL cache size in bytes: 37325536
Compiling design (CSG Products normalization)...
Normalized CSG tree has 26 elements
Compile and preview finished.
Total rendering time: 0 hours, 7 minutes, 7 seconds
```

Viewport: translate = [-0.00 -0.00 -0.00], rotate = [39.60 0.00 43.20], distance = 54.24 (886x882)

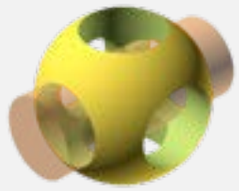
OpenSCAD 2016.01.27 (git 176a1e8)

Generate by 1 command

- You can pass arguments from command line to OpenSCAD script

gen_unit.py

```
#!/usr/local/bin/python

def main():
    import subprocess
    for l in [unichr(i) for i in xrange(65, 91)]:
        cmd = "/Applications/OpenSCAD.app/Contents/MacOS/OpenSCAD -o data/{letter}.stl -D 'create_letter=
            \"{letter}\"' create_new_unit.scad".format(letter=l)
        subprocess.call(cmd, shell=True)

if __name__ == '__main__':
    main()
```

OpenPySCAD  

- Generate OpenSCAD code from Python code
  - https://github.com/taxpon/openpyscad

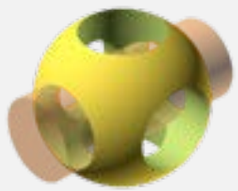
- Install via pip command

```
$ pip install openpyscad
```

OpenPySCAD

- Union operation

```
Cube([20, 10, 10]) + Cube([10, 20, 10])
```

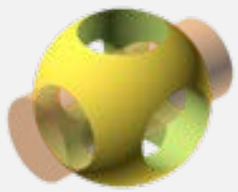


```
union(){
    cube([20, 10, 10])
    cube([10, 20, 10])
};
```

OpenPySCAD

- Difference operation

```
Cube([20, 10, 10]) - Cube([10, 20, 10])
```

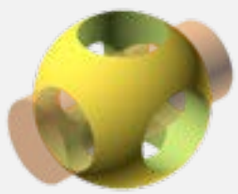


```
difference(){
    cube([20, 10, 10])
    cube([10, 20, 10])
};
```

OpenPySCAD

- Rotation

```
c1 = Cube([20, 10, 10])
c1.rotate([0, 0, 45])
```



```
rotate([0, 0, 45]){
    cube([20, 10, 10])
};
```

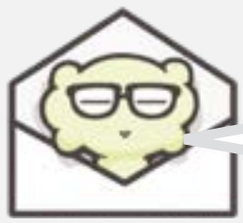
Conclusion

Script Modeling is FUN!!

Let's start today

We are Hiring!!

- Python Developer
- C++ Developer
- Frontend Developer
  - Angular/React
- You can use 3D printer
- International members
- 3 Google Developer Experts



http://www.kabuku.co.jp/#jobs

Engineer team