3D printing

programmer's perspective

Bartek 'BaSz' Szurgot

https://baszerr.eu

November 7, 2021

- YAWN (Yet Another Weird Nerd)
- C++
- Linux
- DevOps
- Networks
- Algo + data structures
- Cybersecurity
- Some electronics
- 3D printing

@work

QIAGEN

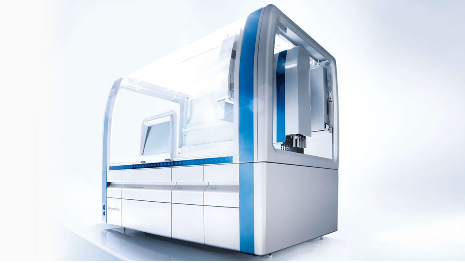
https://upload.wikimedia.org/wikipedia/commons/9/9c/Qiagen_Logo.svg
https://biotech-today.com/wp-content/uploads/2020/08/Qiagen-research.jpg
https://go.wroclaw.pl/api/download/img-1a694cda2df7bd0817363e1d47f285cf/sky-tower-wroclaw-1-jpg.jpg
https://www.verywellhealth.com/thmb/laboratory-with-nurse-taking-a-blood-sample-from-patient-599486370-5aeb1e2730371300362c67cb.jpg

@work

https://upload.wikimedia.org/wikipedia/commons/9/9c/Qiagen_Logo.svg
https://biotech-today.com/wp-content/uploads/2020/08/Qiagen-research.jpg
https://go.wroclaw.pl/api/download/img-1a694cda2df7bd0817363e1d47f285cf/sky-tower-wroclaw-1-jpg.jpg
https://www.verywellhealth.com/thmb/laboratory-with-nurse-taking-a-blood-sample-from-patient-599486370-5aeb1e2730371300362c67cb.jpg

@work

QIAGEN

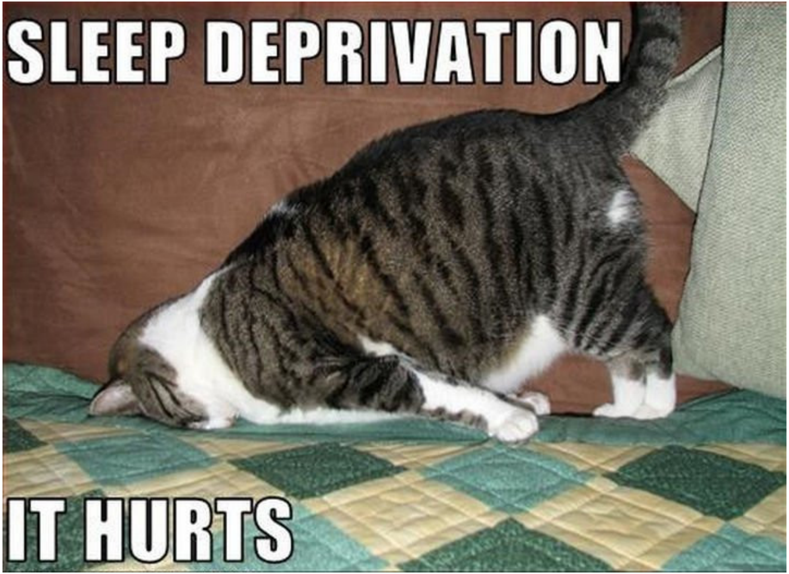


https://upload.wikimedia.org/wikipedia/commons/9/9c/Qiagen_Logo.svg
https://biotech-today.com/wp-content/uploads/2020/08/Qiagen-research.jpg
https://go.wroclaw.pl/api/download/img-1a694cda2df7bd0817363e1d47f285cf/sky-tower-wroclaw-1-jpg.jpg
https://www.verywellhealth.com/thmb/laboratory-with-nurse-taking-a-blood-sample-from-patient-599486370-5aeb1e2730371300362c67cb.jpg

After hours. . .

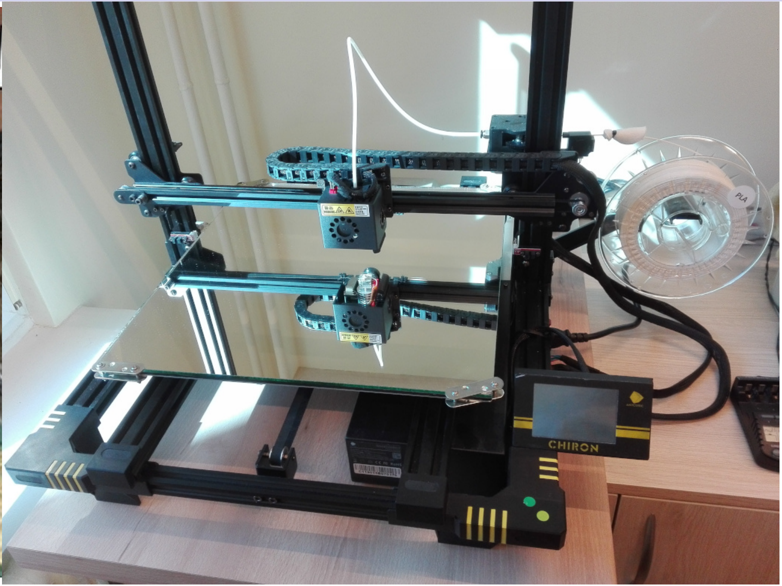
https://winkgo.com/wp-content/uploads/2017/03/25-Sleep-Deprived-Moms-Share-Hilarious-Unfortunate-Things-Theyve-Done-01.jpg

After hours. . .



https://winkgo.com/wp-content/uploads/2017/03/25-Sleep-Deprived-Moms-Share-Hilarious-Unfortunate-Things-Theyve-Done-01.jpg

After hours. . .

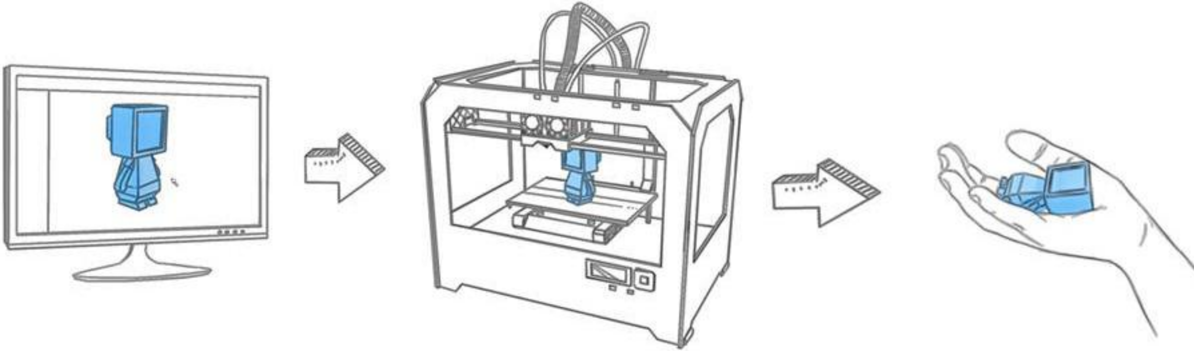


https://winkgo.com/wp-content/uploads/2017/03/25-Sleep-Deprived-Moms-Share-Hilarious-Unfortunate-Things-Theyve-Done-01.jpg

Time for. . .

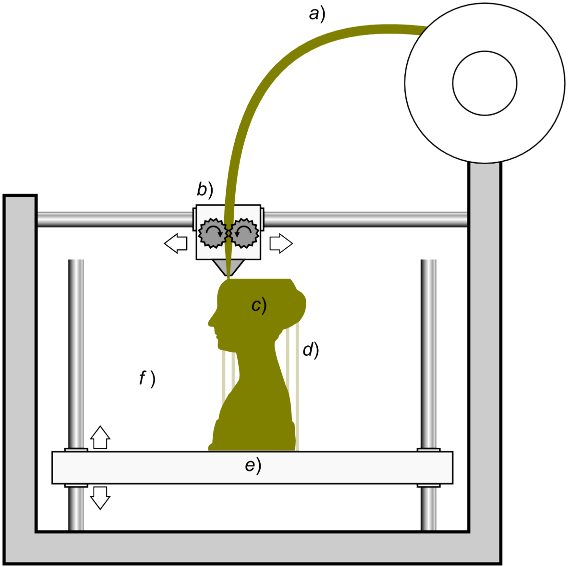
1. 3D printing 101
2. Modeling in OpenSCAD
3. Slicing
4. ETA
5. Pipeline
6. Ending

Process



https://i2.wp.com/www.technobyte.org/wp-content/uploads/2016/12/3d-printing-process.jpg

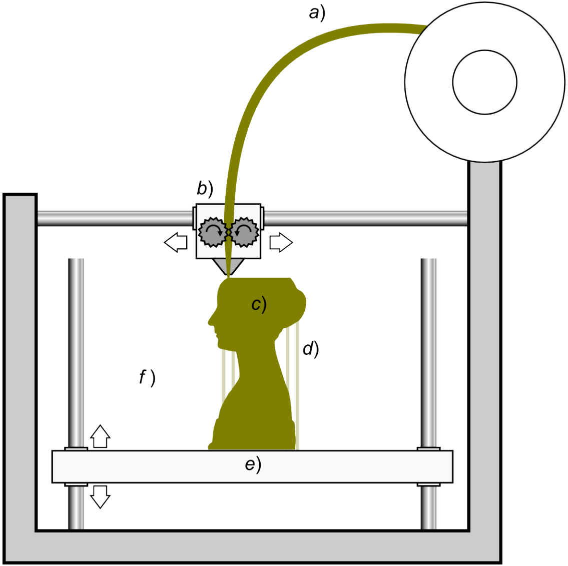
FDM: Fused Deposition Modeling

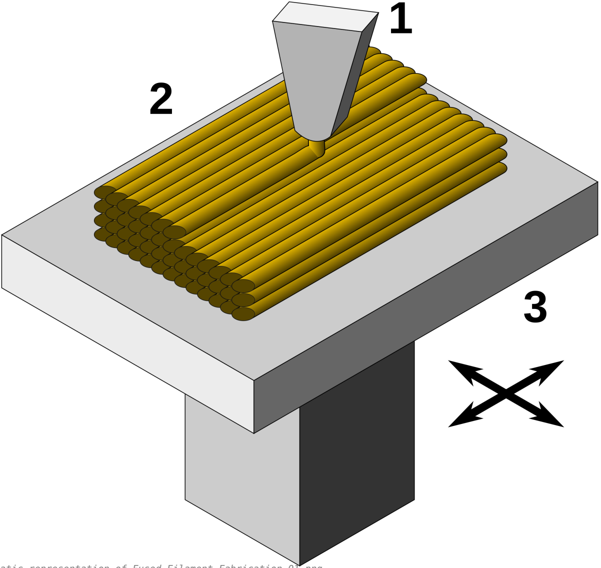


https://upload.wikimedia.org/wikipedia/commons/c/c8/Schematic_representation_of_Fused_Filament_Fabrication_01.png

https://upload.wikimedia.org/wikipedia/commons/7/70/FDM_printing_diagram.svg

FDM: Fused Deposition Modeling

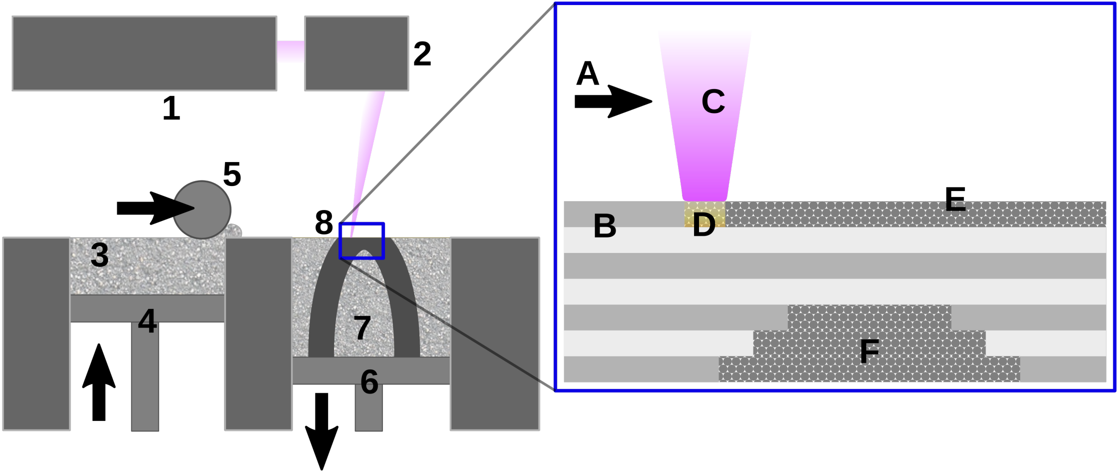




https://upload.wikimedia.org/wikipedia/commons/c/c8/Schematic_representation_of_Fused_Filament_Fabrication_01.png

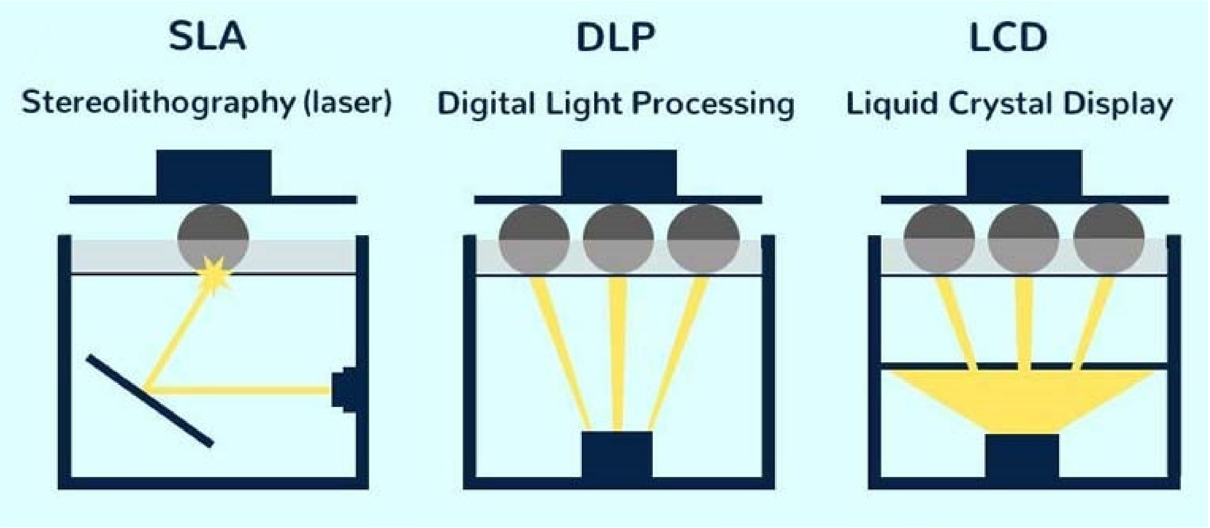
https://upload.wikimedia.org/wikipedia/commons/7/70/FDM_printing_diagram.svg

SLS: Selective Laser Sintering



https://upload.wikimedia.org/wikipedia/commons/f/fb/SLS_schematic.svg

SLA: Stereolithography (resin)



https://pic2-cdn.creality.com/website/34f19099-9206-4eca-ad17-a06c60f7b302

Comparison

- **FDM**
- **SLS**
- **SLA(ish)**

Comparison

- **FDM**

+ Cheap
+ RepRap :)
+ Mulimaterial
+ Clean
+ Compact

– Slow

– Complex

- **SLS**

- **SLA(ish)**

Comparison

- **FDM**

+ Cheap
+ RepRap :)
+ Mulimaterial
+ Clean
+ Compact
– Slow
– Complex

- **SLS**

+ Precise
+ Prints metal
+ Strong
– Large
– Expensive
– Slow
– Hazardous

- **SLA(ish)**

Comparison

- **FDM**

+ Cheap
+ RepRap :)
+ Mulimaterial
+ Clean
+ Compact

– Slow

– Complex

- **SLS**

+ Precise
+ Prints metal
+ Strong

– Large

– Expensive

– Slow

– Hazardous

- **SLA(ish)**

+ Fairly cheap
+ Super fast
+ Precise

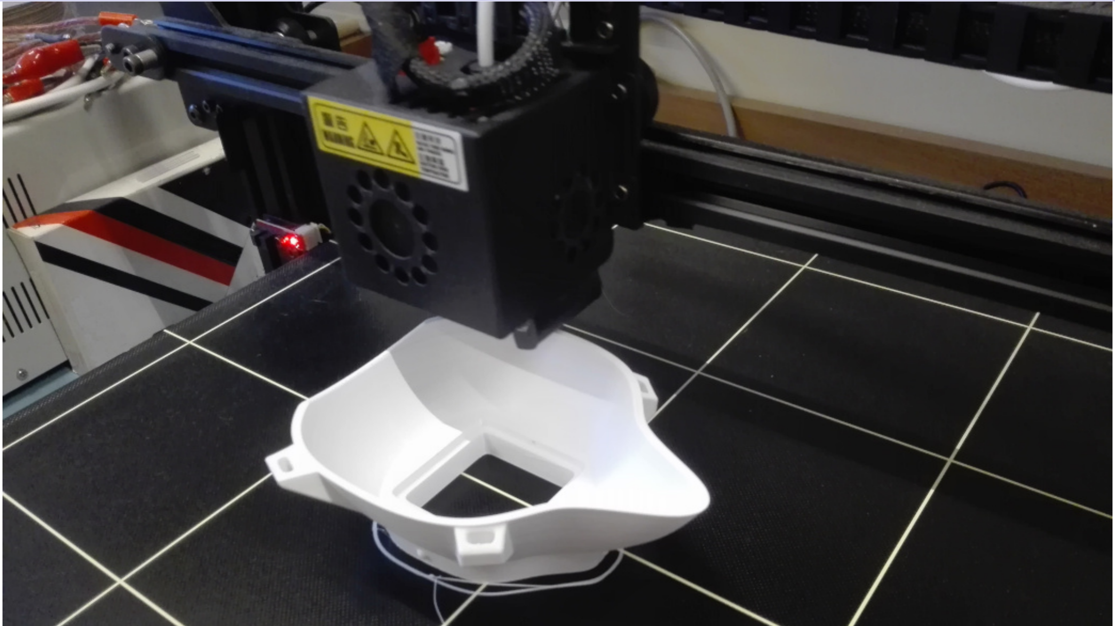
– 2-3 devices

– Toxic

– Dirty

– Smelly

FDM from now on!

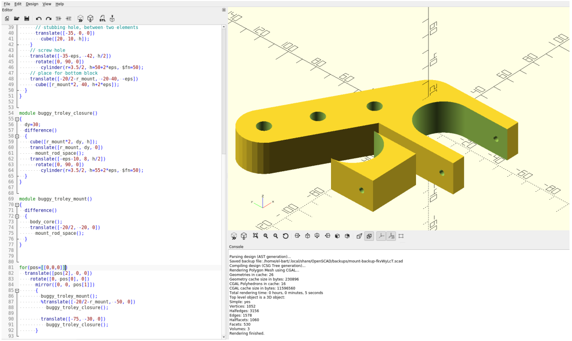


Pipeline

- **CAD model**
- **STL**
- **G-code**

Pipeline

- **CAD model**
- **STL**
- **G-code**



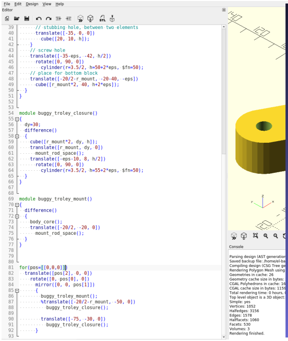
```
      // stubbing hole, between two elements
      translate([-35, 0, 0])
        cube([20, 10, h]);
    }
    // screw hole
    translate([-35-eps, -42, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=50+2*eps, $fn=50);
    // place for bottom block
    translate([-20/2-r_mount, -20-40, -eps])
      cube([r_mount*2, 40, h+2*eps]);
  }
}


module buggy_troley_closure()
{
  dy=30;
  difference()
  {
    cube([r_mount*2, dy, h]);
    translate([r_mount, dy, 0])
      mount_rod_space();
    translate([-eps-10, 8, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=55+2*eps, $fn=50);
  }
}


module buggy_troley_mount()
{
  difference()
  {
    body_core();
    translate([-20/2, -20, 0])
      mount_rod_space();
  }
}



for(pos=[[0,0,0]])
  translate([pos[2], 0, 0])
    rotate([0, pos[0], 0])
      mirror([0, 0, pos[1]])
      {
        buggy_troley_mount();
        %translate([-20/2-r_mount, -50, 0])
          buggy_troley_closure();

        translate([-75, -30, 0])
          buggy_troley_closure();
      }
```

```
Parsing design (AST generation)...
Saved backup file: /home/el-bart/.local/share/OpenSCAD/backups/mount-backup-fkvWyLcT.scad
Compiling design (CSG Tree generation)...
Rendering Polygon Mesh using CGAL...
Geometries in cache: 26
Geometry cache size in bytes: 230896
CGAL Polyhedrons in cache: 16
CGAL cache size in bytes: 11596560
Total rendering time: 0 hours, 0 minutes, 5 seconds
Top level object is a 3D object:
Simple: yes
Vertices: 1052
Halfedges: 3156
Edges: 1578
Halffacets: 1060
Facets: 530
Volumes: 3
Rendering finished.
```

"Java source"

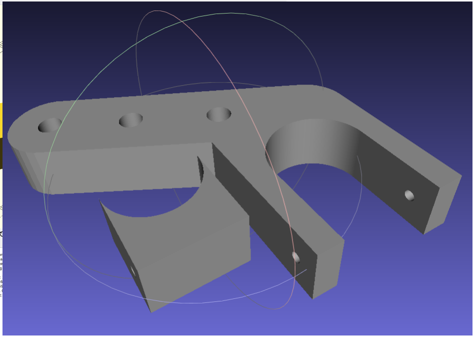
Pipeline

- **CAD model**
- **STL**
- **G-code**



```
      // stubbing hole, between two elements
      translate([-35, 0, 0])
        cube([20, 10, h]);
    }
    // screw hole
    translate([-35-eps, -42, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=50+2*eps, $fn=50);
    // place for bottom block
    translate([-20/2-r_mount, -20-40, -eps])
      cube([r_mount*2, 40, h+2*eps]);
  }
}


module buggy_troley_closure()
{
  dy=30;
  difference()
  {
    cube([r_mount*2, dy, h]);
    translate([r_mount, dy, 0])
      mount_rod_space();
    translate([-eps-10, 8, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=55+2*eps, $fn=50);
  }
}


module buggy_troley_mount()
{
  difference()
  {
    body_core();
    translate([-20/2, -20, 0])
      mount_rod_space();
  }
}



for(pos=[[0,0,0]])
  translate([pos[2], 0, 0])
    rotate([0, pos[0], 0])
      mirror([0, 0, pos[1]])
      {
        buggy_troley_mount();
        %translate([-20/2-r_mount, -50, 0])
          buggy_troley_closure();

        translate([-75, -30, 0])
          buggy_troley_closure();
      }

```

```
Parsing design (AST generation
Saved backup file: /home/el-ba
Compiling design (CSG Tree ge
Rendering Polygon Mesh using
Geometries in cache: 26
Geometry cache size in bytes:
CGAL Polyhedrons in cache: 16
CGAL cache size in bytes: 1159
Total rendering time: 0 hours, 0
Top level object is a 3D object:
Simple: yes
Vertices: 1052
Halfedges: 3156
Edges: 1578
Halffacets: 1060
Facets: 530
Volumes: 3
Rendering finished.
```

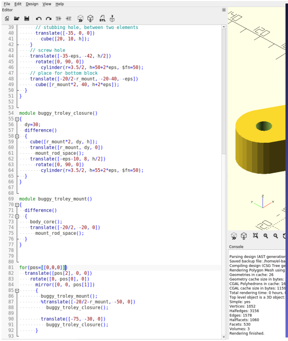


"Java source"

"Java IR"

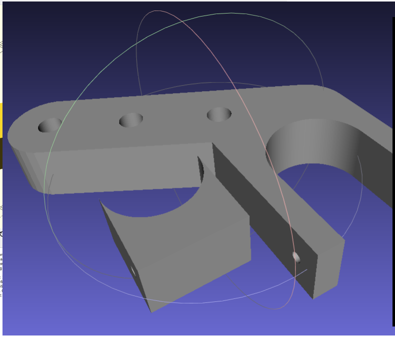
Pipeline

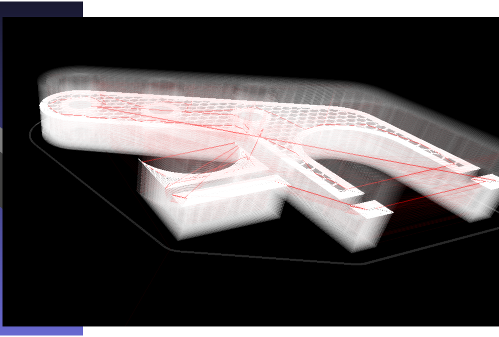
- **CAD model**
- **STL**
- **G-code**



```
      // stubbing hole, between two elements
      translate([-35, 0, 0])
        cube([20, 10, h]);
    }
    // screw hole
    translate([-35-eps, -42, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=50+2*eps, $fn=50);
    // place for bottom block
    translate([-20/2-r_mount, -20-40, -eps])
      cube([r_mount*2, 40, h+2*eps]);
  }
}


module buggy_troley_closure()
{
  dy=30;
  difference()
  {
    cube([r_mount*2, dy, h]);
    translate([r_mount, dy, 0])
      mount_rod_space();
    translate([-eps-10, 8, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=55+2*eps, $fn=50);
  }
}


module buggy_troley_mount()
{
  difference()
  {
    body_core();
    translate([-20/2, -20, 0])
      mount_rod_space();
  }
}



for(pos=[[0,0,0]])
  translate([pos[2], 0, 0])
    rotate([0, pos[0], 0])
      mirror([0, 0, pos[1]])
      {
        buggy_troley_mount();
        %translate([-20/2-r_mount, -50, 0])
          buggy_troley_closure();

        translate([-75, -30, 0])
          buggy_troley_closure();
      }

```

```
Parsing design (AST generation
Saved backup file: /home/el-ba
Compiling design (CSG Tree ge
Rendering Polygon Mesh using
Geometries in cache: 26
Geometry cache size in bytes:
CGAL Polyhedrons in cache: 16
CGAL cache size in bytes: 1159
Total rendering time: 0 hours, 0
Top level object is a 3D object:
Simple: yes
Vertices: 1052
Halfedges: 3156
Edges: 1578
Halffacets: 1060
Facets: 530
Volumes: 3
Rendering finished.
```





"Java source"

"Java IR"

"Machine code"

https://github.com/el-bart/mini/tree/master/buggy_troley_mount

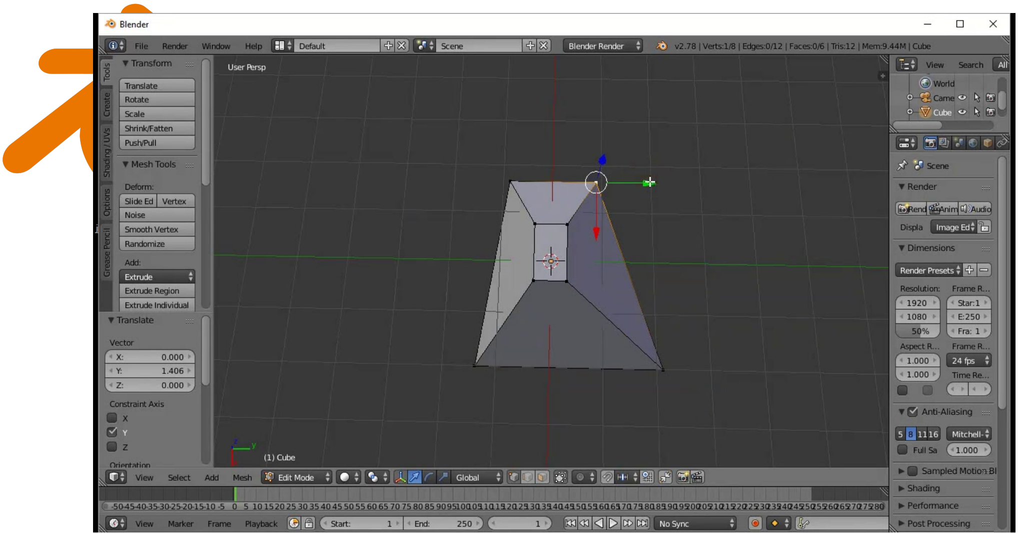
Got idea?

https://upload.wikimedia.org/wikipedia/commons/9/97/OpenSCAD-logo.png
https://www.mysolidbox.com/wp-content/uploads/2017/02/Autodesk-Fusion-360-logo.png
https://upload.wikimedia.org/wikipedia/commons/3/3c/Logo_Blender.svg
https://i1.wp.com/total3dprinting.org/wp-content/uploads/2020/07/fusion-360.png
https://i.ytimg.com/vi/7Ve5Fmc6D3s/maxresdefault.jpg

Got idea?

https://upload.wikimedia.org/wikipedia/commons/9/97/OpenSCAD-logo.png
https://www.mysolidbox.com/wp-content/uploads/2017/02/Autodesk-Fusion-360-logo.png
https://upload.wikimedia.org/wikipedia/commons/3/3c/Logo_Blender.svg
https://i1.wp.com/total3dprinting.org/wp-content/uploads/2020/07/fusion-360.png
https://i.ytimg.com/vi/7Ve5Fmc6D3s/maxresdefault.jpg

Got idea?

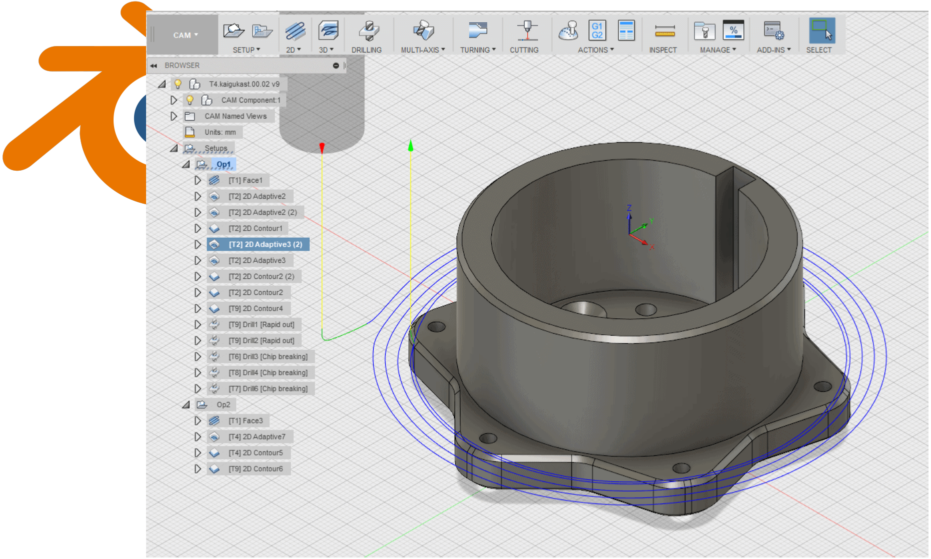


https://upload.wikimedia.org/wikipedia/commons/9/97/OpenSCAD-logo.png
https://www.mysolidbox.com/wp-content/uploads/2017/02/Autodesk-Fusion-360-logo.png
https://upload.wikimedia.org/wikipedia/commons/3/3c/Logo_Blender.svg
https://i1.wp.com/total3dprinting.org/wp-content/uploads/2020/07/fusion-360.png
https://i.ytimg.com/vi/7Ve5Fmc6D3s/maxresdefault.jpg

Got idea?

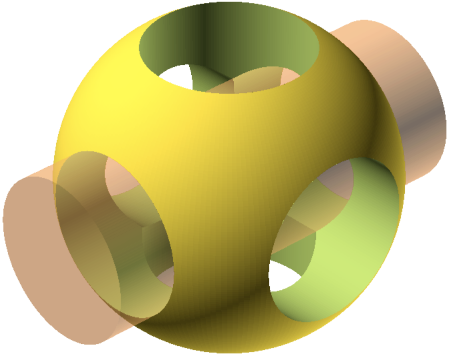
https://upload.wikimedia.org/wikipedia/commons/9/97/OpenSCAD-logo.png
https://www.mysolidbox.com/wp-content/uploads/2017/02/Autodesk-Fusion-360-logo.png
https://upload.wikimedia.org/wikipedia/commons/3/3c/Logo_Blender.svg
https://i1.wp.com/total3dprinting.org/wp-content/uploads/2020/07/fusion-360.png
https://i.ytimg.com/vi/7Ve5Fmc6D3s/maxresdefault.jpg

Got idea?



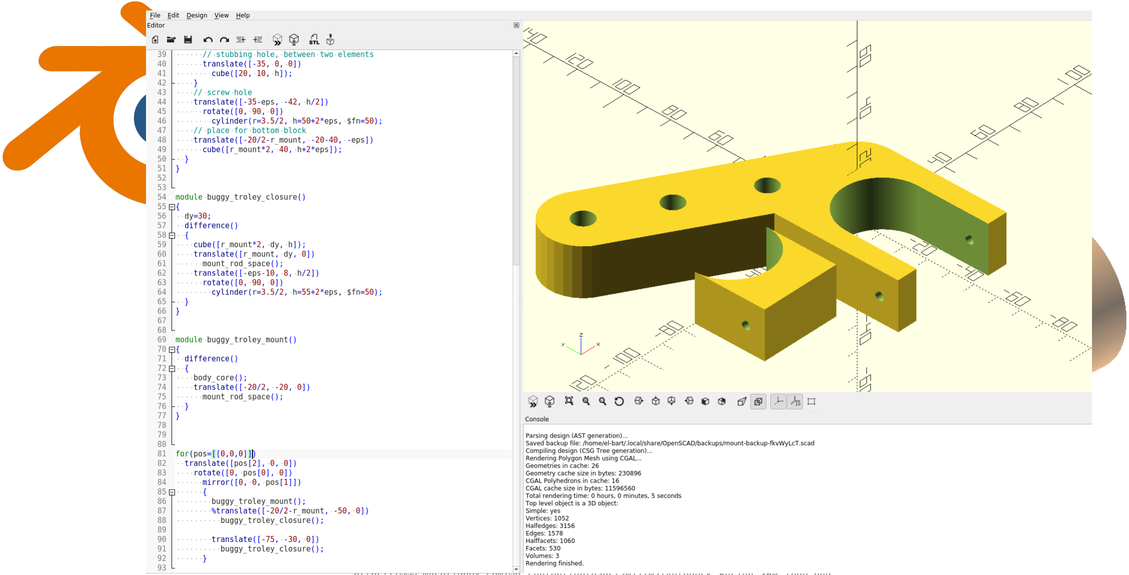
https://upload.wikimedia.org/wikipedia/commons/9/97/OpenSCAD-logo.png
https://www.mysolidbox.com/wp-content/uploads/2017/02/Autodesk-Fusion-360-logo.png
https://upload.wikimedia.org/wikipedia/commons/3/3c/Logo_Blender.svg
https://i1.wp.com/total3dprinting.org/wp-content/uploads/2020/07/fusion-360.png
https://i.ytimg.com/vi/7Ve5Fmc6D3s/maxresdefault.jpg

Got idea?



https://upload.wikimedia.org/wikipedia/commons/9/97/OpenSCAD-logo.png
https://www.mysolidbox.com/wp-content/uploads/2017/02/Autodesk-Fusion-360-logo.png
https://upload.wikimedia.org/wikipedia/commons/3/3c/Logo_Blender.svg
https://i1.wp.com/total3dprinting.org/wp-content/uploads/2020/07/fusion-360.png
https://i.ytimg.com/vi/7Ve5Fmc6D3s/maxresdefault.jpg

Got idea?



```
      // stubbing hole, between two elements
      translate([-35, 0, 0])
        cube([20, 10, h]);
    }
    // screw hole
    translate([-35-eps, -42, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=50+2*eps, $fn=50);
    // place for bottom block
    translate([-20/2-r_mount, -20-40, -eps])
      cube([r_mount*2, 40, h+2*eps]);
  }
}


module buggy_troley_closure()
{
  dy=30;
  difference()
  {
    cube([r_mount*2, dy, h]);
    translate([r_mount, dy, 0])
      mount_rod_space();
    translate([-eps-10, 8, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=55+2*eps, $fn=50);
  }
}


module buggy_troley_mount()
{
  difference()
  {
    body_core();
    translate([-20/2, -20, 0])
      mount_rod_space();
  }
}



for(pos=[[0,0,0]])
  translate([pos[2], 0, 0])
    rotate([0, pos[0], 0])
      mirror([0, 0, pos[1]])
      {
        buggy_troley_mount();
        %translate([-20/2-r_mount, -50, 0])
          buggy_troley_closure();

        translate([-75, -30, 0])
          buggy_troley_closure();
      }
```

```
Parsing design (AST generation)...
Saved backup file: /home/el-bart/.local/share/OpenSCAD/backups/mount-backup-fkvWyLcT.scad
Compiling design (CSG Tree generation)...
Rendering Polygon Mesh using CGAL...
Geometries in cache: 26
Geometry cache size in bytes: 230896
CGAL Polyhedrons in cache: 16
CGAL cache size in bytes: 11596560
Total rendering time: 0 hours, 0 minutes, 5 seconds
Top level object is a 3D object:
Simple: yes
Vertices: 1052
Halfedges: 3156
Edges: 1578
Halffacets: 1060
Facets: 530
Volumes: 3
Rendering finished.
```

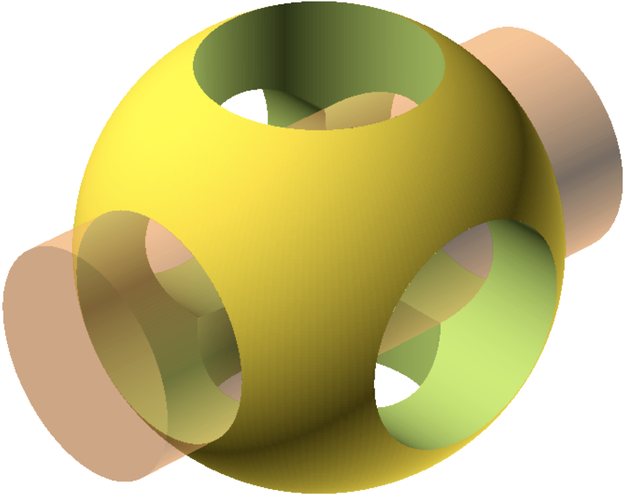
https://www.mysolidbox.com/wp-content/uploads/2017/02/Autodesk-Fusion-360-Logo.png
https://upload.wikimedia.org/wikipedia/commons/3/3c/Logo_Blender.svg
https://i1.wp.com/total3dprinting.org/wp-content/uploads/2020/07/fusion-360.png
https://i.ytimg.com/vi/7Ve5Fmc6D3s/maxresdefault.jpg

Time for. . .

1. 3D printing 101
2. Modeling in OpenSCAD
3. Slicing
4. ETA
5. Pipeline
6. Ending

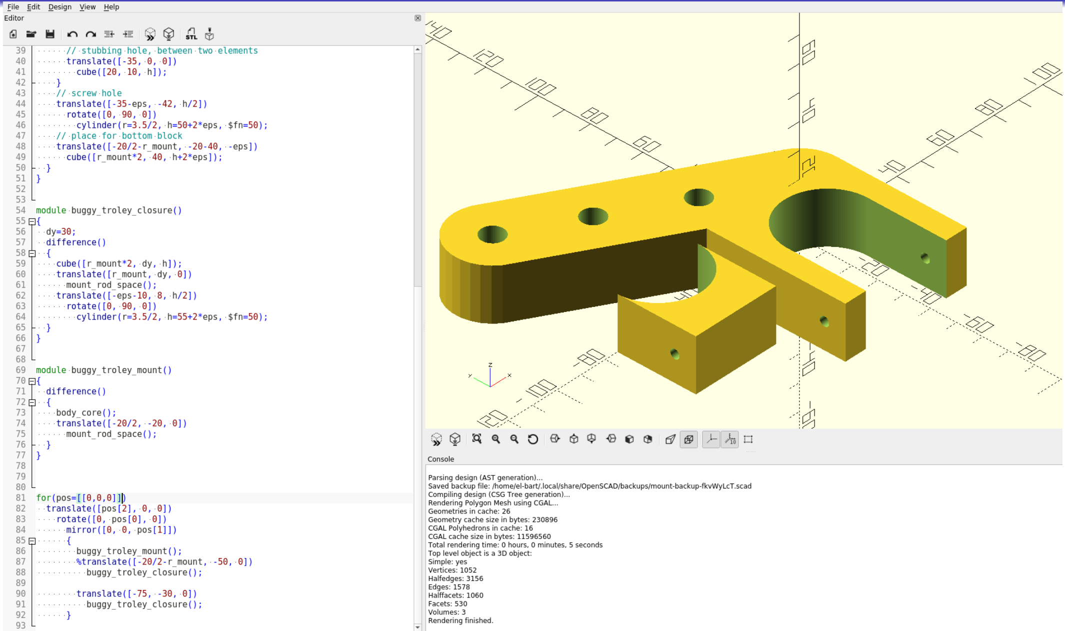
What is OpenSCAD?

What is OpenSCAD?



https://upload.wikimedia.org/wikipedia/commons/9/97/OpenSCAD-logo.png

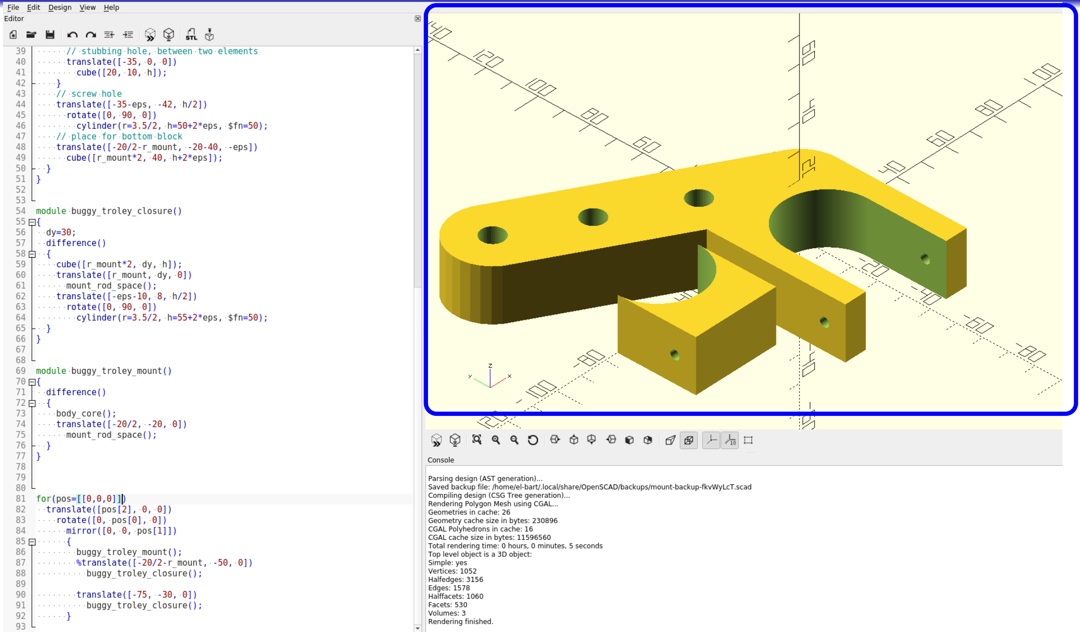
OpenSCAD’s IDE



```
      // stubbing hole, between two elements
      translate([-35, 0, 0])
        cube([20, 10, h]);
    }
    // screw hole
    translate([-35-eps, -42, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=50+2*eps, $fn=50);
    // place for bottom block
    translate([-20/2-r_mount, -20-40, -eps])
      cube([r_mount*2, 40, h+2*eps]);
  }
}


module buggy_troley_closure()
{
  dy=30;
  difference()
  {
    cube([r_mount*2, dy, h]);
    translate([r_mount, dy, 0])
      mount_rod_space();
    translate([-eps-10, 8, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=55+2*eps, $fn=50);
  }
}


module buggy_troley_mount()
{
  difference()
  {
    body_core();
    translate([-20/2, -20, 0])
      mount_rod_space();
  }
}



for(pos=[[0,0,0]])
  translate([pos[2], 0, 0])
    rotate([0, pos[0], 0])
      mirror([0, 0, pos[1]])
      {
        buggy_troley_mount();
        %translate([-20/2-r_mount, -50, 0])
          buggy_troley_closure();

        translate([-75, -30, 0])
          buggy_troley_closure();
      }

```

```
Parsing design (AST generation)...
Saved backup file: /home/el-bart/.local/share/OpenSCAD/backups/mount-backup-fkvWyLcT.scad
Compiling design (CSG Tree generation)...
Rendering Polygon Mesh using CGAL...
Geometries in cache: 26
Geometry cache size in bytes: 230896
CGAL Polyhedrons in cache: 16
CGAL cache size in bytes: 11596560
Total rendering time: 0 hours, 0 minutes, 5 seconds
Top level object is a 3D object:
Simple: yes
Vertices: 1052
Halfedges: 3156
Edges: 1578
Halffacets: 1060
Facets: 530
Volumes: 3
Rendering finished.
```

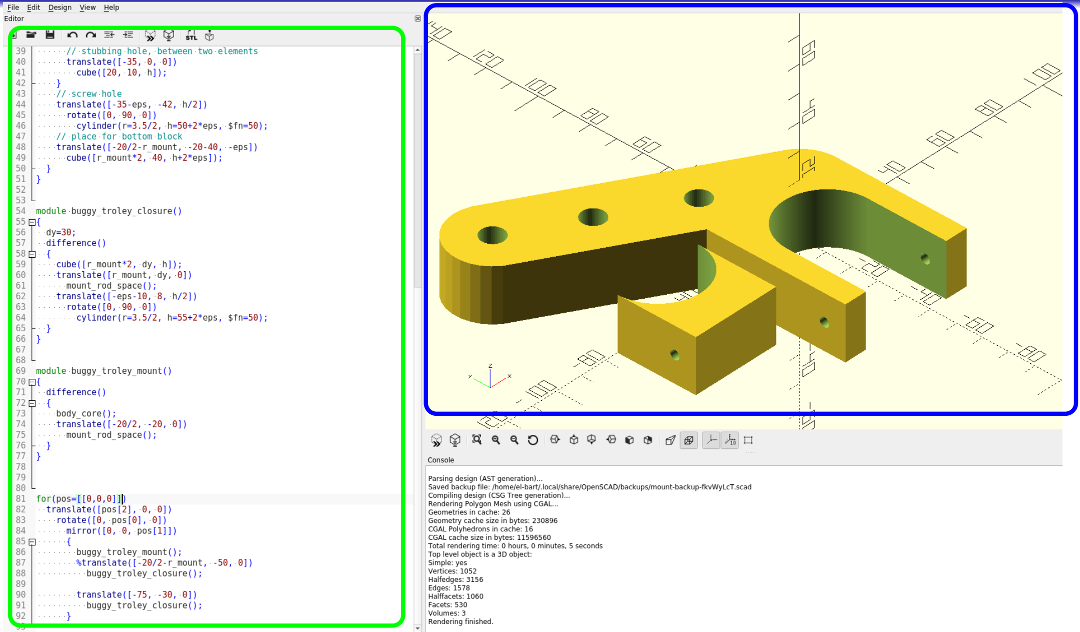
OpenSCAD's IDE



```
      // stubbing hole, between two elements
      translate([-35, 0, 0])
        cube([20, 10, h]);
    }
    // screw hole
    translate([-35-eps, -42, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=50+2*eps, $fn=50);
    // place for bottom block
    translate([-20/2-r_mount, -20-40, -eps])
      cube([r_mount*2, 40, h+2*eps]);
  }
}


module buggy_troley_closure()
{
  dy=30;
  difference()
  {
    cube([r_mount*2, dy, h]);
    translate([r_mount, dy, 0])
      mount_rod_space();
    translate([-eps-10, 8, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=55+2*eps, $fn=50);
  }
}


module buggy_troley_mount()
{
  difference()
  {
    body_core();
    translate([-20/2, -20, 0])
      mount_rod_space();
  }
}



for(pos=[[0,0,0]])
  translate([pos[2], 0, 0])
    rotate([0, pos[0], 0])
      mirror([0, 0, pos[1]])
      {
        buggy_troley_mount();
        %translate([-20/2-r_mount, -50, 0])
          buggy_troley_closure();

        translate([-75, -30, 0])
          buggy_troley_closure();
      }

```

```
Parsing design (AST generation)...
Saved backup file: /home/el-bart/.local/share/OpenSCAD/backups/mount-backup-fkvWyLcT.scad
Compiling design (CSG Tree generation)...
Rendering Polygon Mesh using CGAL...
Geometries in cache: 26
Geometry cache size in bytes: 230896
CGAL Polyhedrons in cache: 16
CGAL cache size in bytes: 11596560
Total rendering time: 0 hours, 0 minutes, 5 seconds
Top level object is a 3D object:
Simple: yes
Vertices: 1052
Halfedges: 3156
Edges: 1578
Halffacets: 1060
Facets: 530
Volumes: 3
Rendering finished.
```

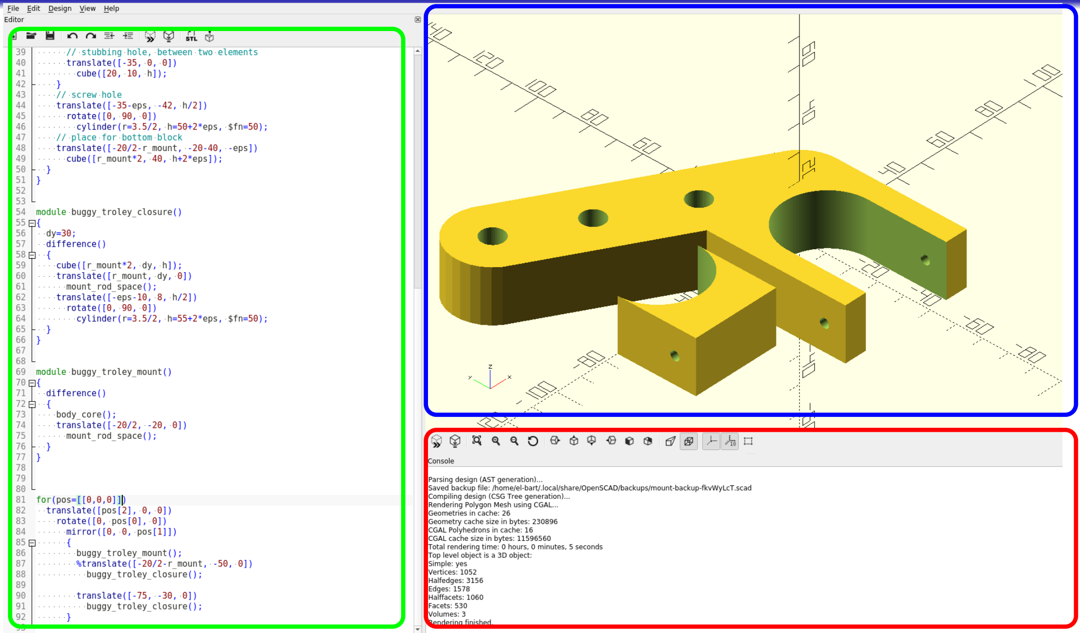
OpenSCAD’s IDE



```
      // stubbing hole, between two elements
      translate([-35, 0, 0])
        cube([20, 10, h]);
    }
    // screw hole
    translate([-35-eps, -42, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=50+2*eps, $fn=50);
    // place for bottom block
    translate([-20/2-r_mount, -20-40, -eps])
      cube([r_mount*2, 40, h+2*eps]);
  }
}


module buggy_troley_closure()
{
  dy=30;
  difference()
  {
    cube([r_mount*2, dy, h]);
    translate([r_mount, dy, 0])
      mount_rod_space();
    translate([-eps-10, 8, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=55+2*eps, $fn=50);
  }
}


module buggy_troley_mount()
{
  difference()
  {
    body_core();
    translate([-20/2, -20, 0])
      mount_rod_space();
  }
}



for(pos=[[0,0,0]])
  translate([pos[2], 0, 0])
    rotate([0, pos[0], 0])
      mirror([0, 0, pos[1]])
      {
        buggy_troley_mount();
        %translate([-20/2-r_mount, -50, 0])
          buggy_troley_closure();

        translate([-75, -30, 0])
          buggy_troley_closure();
      }
```

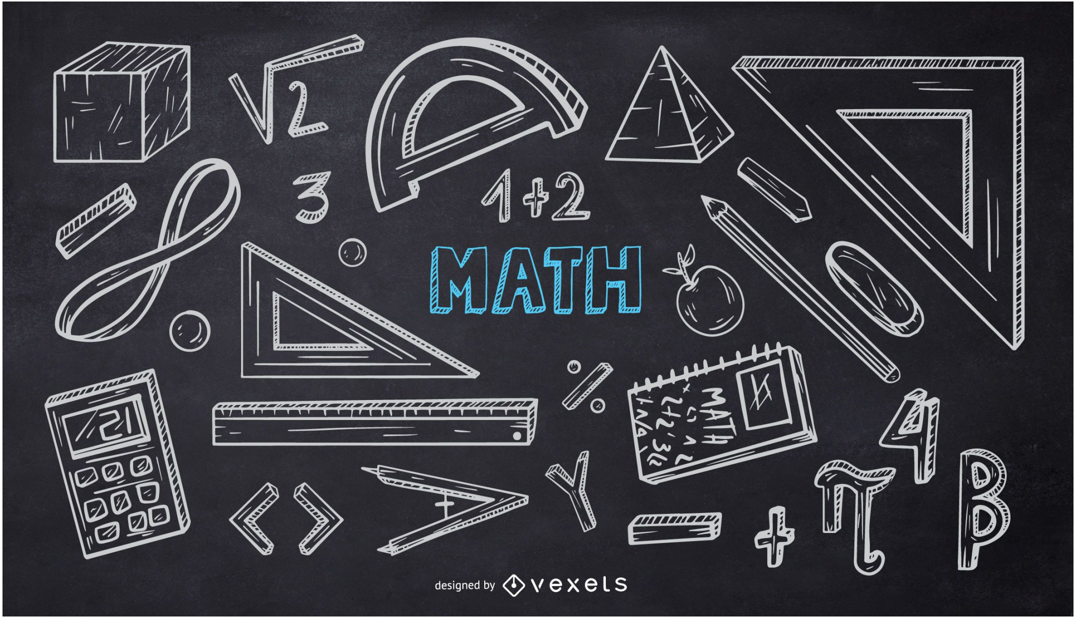
```
Parsing design (AST generation)...
Saved backup file: /home/el-bart/.local/share/OpenSCAD/backups/mount-backup-fkvWyLcT.scad
Compiling design (CSG Tree generation)...
Rendering Polygon Mesh using CGAL...
Geometries in cache: 26
Geometry cache size in bytes: 230896
CGAL Polyhedrons in cache: 16
CGAL cache size in bytes: 11596560
Total rendering time: 0 hours, 0 minutes, 5 seconds
Top level object is a 3D object:
Simple: yes
Vertices: 1052
Halfedges: 3156
Edges: 1578
Halffacets: 1060
Facets: 530
Volumes: 3
Rendering finished.
```

OpenSCAD’s IDE

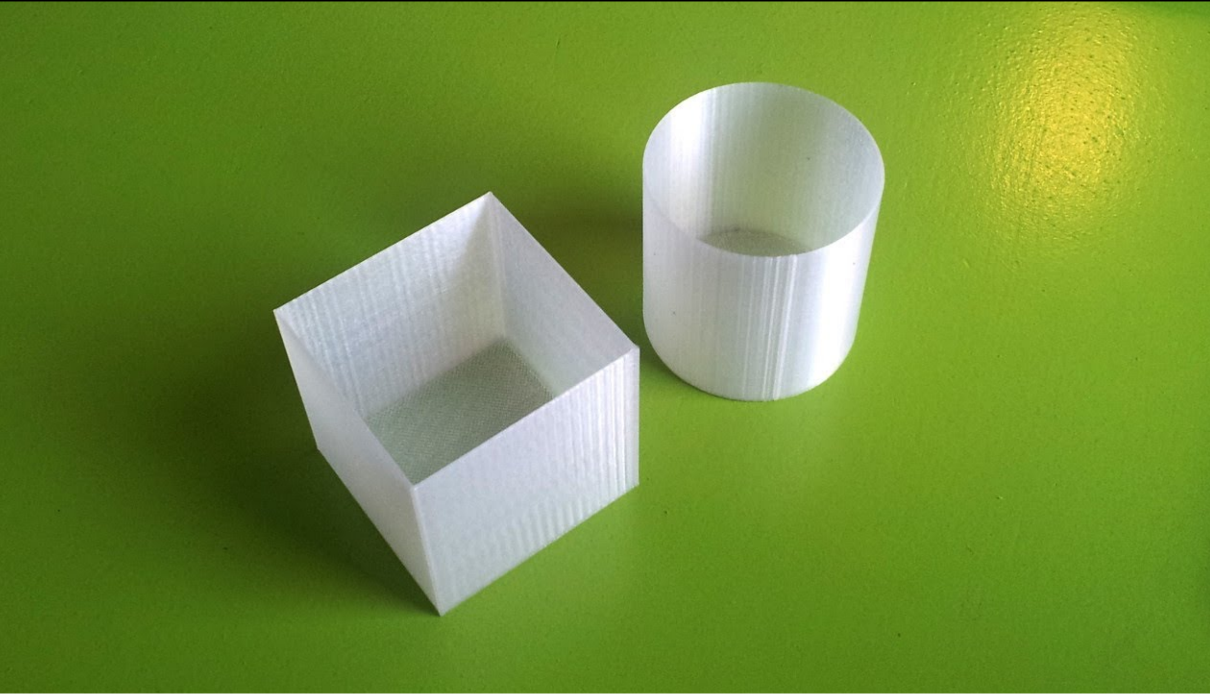


```
      // stubbing hole, between two elements
      translate([-35, 0, 0])
        cube([20, 10, h]);
    }
    // screw hole
    translate([-35-eps, -42, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=50+2*eps, $fn=50);
    // place for bottom block
    translate([-20/2-r_mount, -20-40, -eps])
      cube([r_mount*2, 40, h+2*eps]);
  }
}


module buggy_troley_closure()
{
  dy=30;
  difference()
  {
    cube([r_mount*2, dy, h]);
    translate([r_mount, dy, 0])
      mount_rod_space();
    translate([-eps-10, 8, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=55+2*eps, $fn=50);
  }
}


module buggy_troley_mount()
{
  difference()
  {
    body_core();
    translate([-20/2, -20, 0])
      mount_rod_space();
  }
}



for(pos=[[0,0,0]])
  translate([pos[2], 0, 0])
    rotate([0, pos[0], 0])
      mirror([0, 0, pos[1]])
      {
        buggy_troley_mount();
        %translate([-20/2-r_mount, -50, 0])
          buggy_troley_closure();

        translate([-75, -30, 0])
          buggy_troley_closure();
      }
```

```
Parsing design (AST generation)...
Saved backup file: /home/el-bart/.local/share/OpenSCAD/backups/mount-backup-fkvWyLcT.scad
Compiling design (CSG Tree generation)...
Rendering Polygon Mesh using CGAL...
Geometries in cache: 26
Geometry cache size in bytes: 230896
CGAL Polyhedrons in cache: 16
CGAL cache size in bytes: 11596560
Total rendering time: 0 hours, 0 minutes, 5 seconds
Top level object is a 3D object:
Simple: yes
Vertices: 1052
Halfedges: 3156
Edges: 1578
Halffacets: 1060
Facets: 530
Volumes: 3
Rendering finished.
```



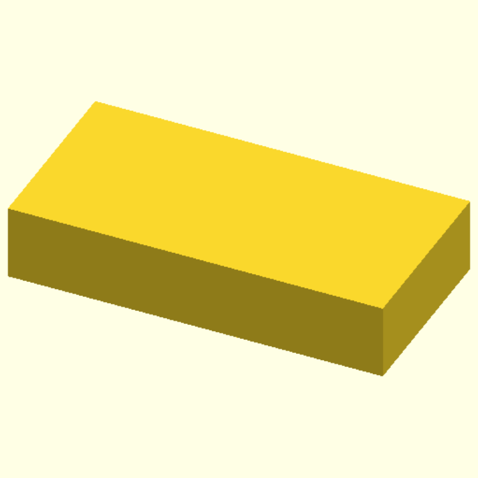
https://images.vexels.com/media/users/3/197149/raw/57966b1555f7c693b22fda0d34ca286b-math-school-chalk-doodle-pack.jpg



http://i.ytimg.com/vi/1iRlVtrrdow/maxresdefault.jpg

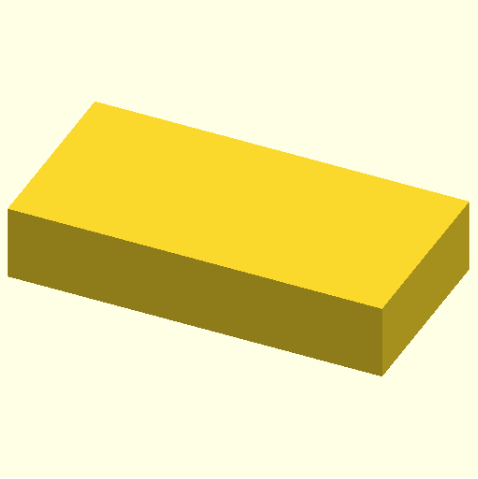
Empty box

```
//     OX  OY  OZ
cube([100, 50, 20]);
x="foo bar 42";
/* other */
// foo
```



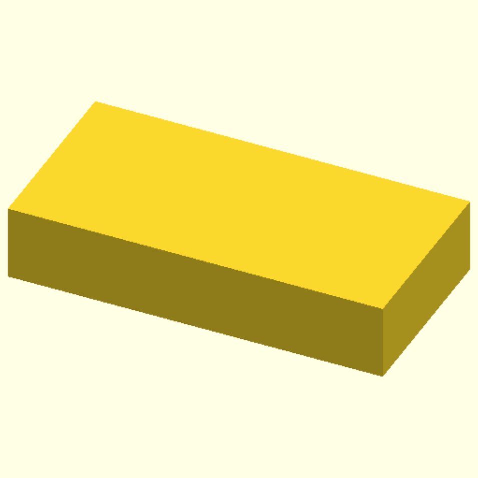
Empty box

```
//     OX  OY  OZ
cube([100, 50, 20]);
x="foo bar 42";
/* other */
// foo
```



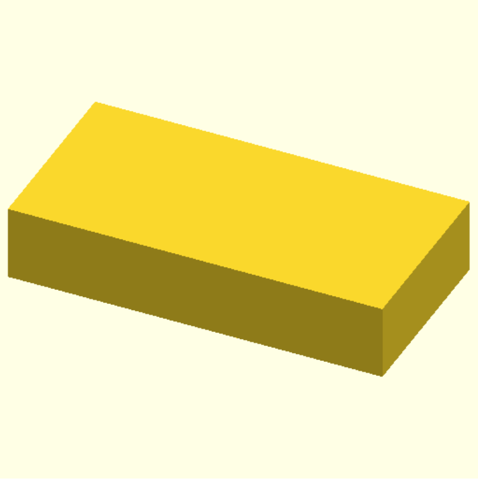
Empty box

```
//     OX  OY  OZ
cube([100, 50, 20]);
x="foo bar 42";
/* other */
// foo
```



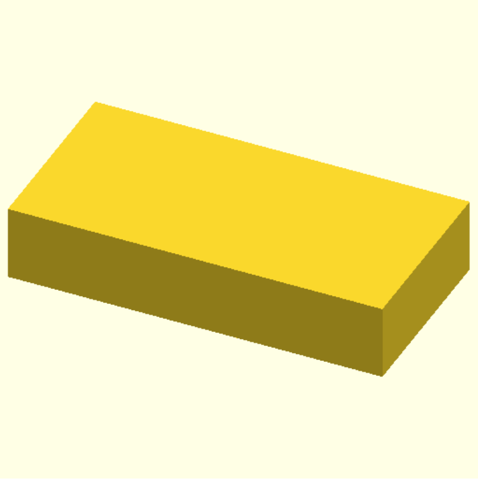
Empty box

```
//     OX  OY  OZ
cube([100, 50, 20]);
x="foo bar 42";
/* other */
// foo
```



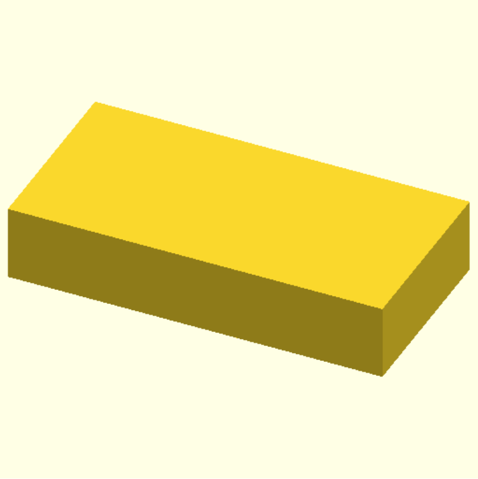
Empty box

```
//     OX  OY  OZ
cube([100, 50, 20]);
x="foo bar 42";
/* other */
// foo
```



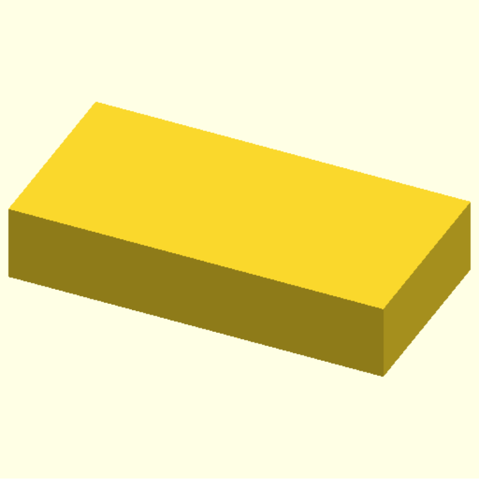
Empty box

```
//     OX  OY  OZ
cube([100, 50, 20]);
x="foo bar 42";
/* other */
// foo
```



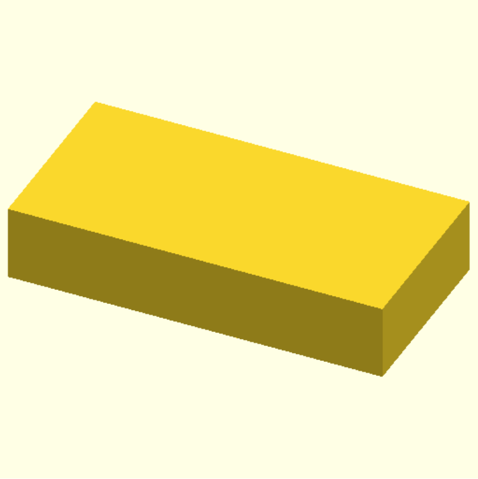
Defining constant

```
size = [100, 50, 20];
cube(size);
```



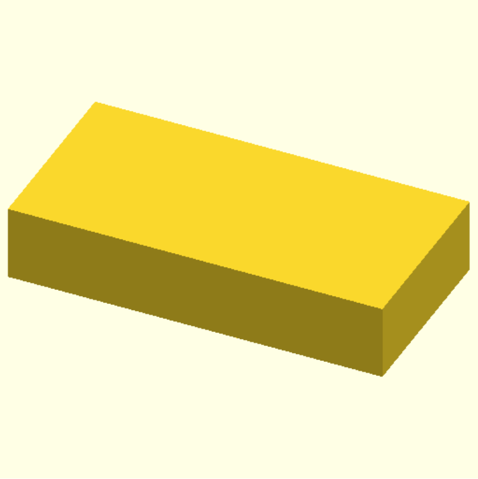
Defining constant

```
size = [100, 50, 20];
cube(size);
```



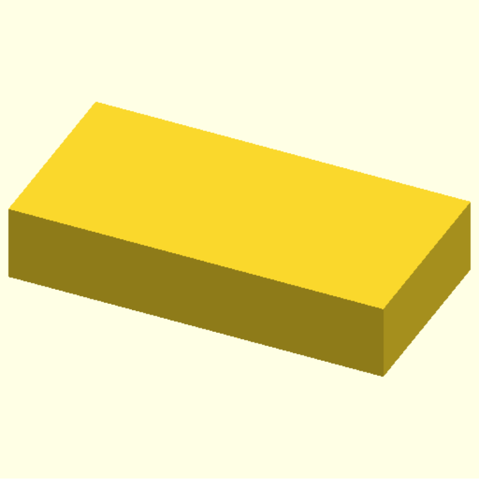
Defining constant

```
size = [100, 50, 20];
cube(size);
```



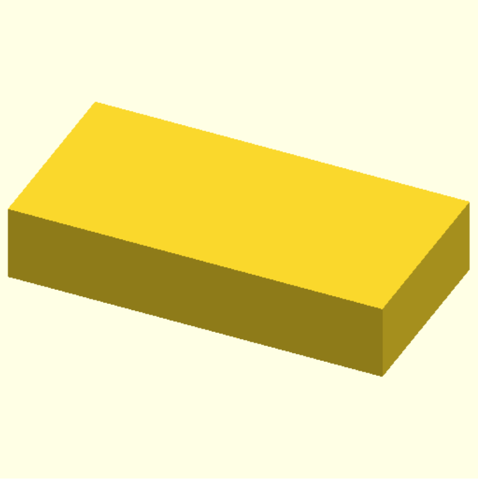
Object difference

```
size = [100, 50, 20];
difference()
{
  cube(size);
}
```



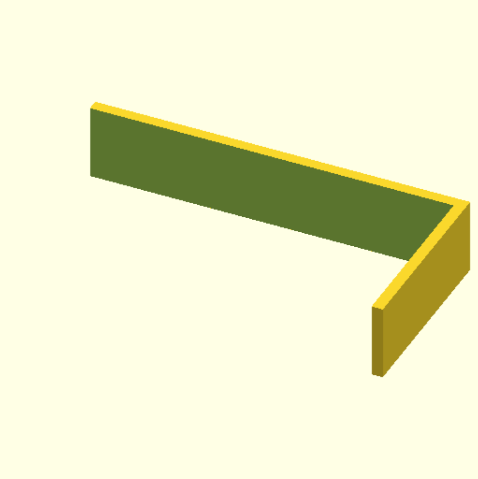
Object difference

```
size = [100, 50, 20];
difference()
{
  cube(size);
}
```



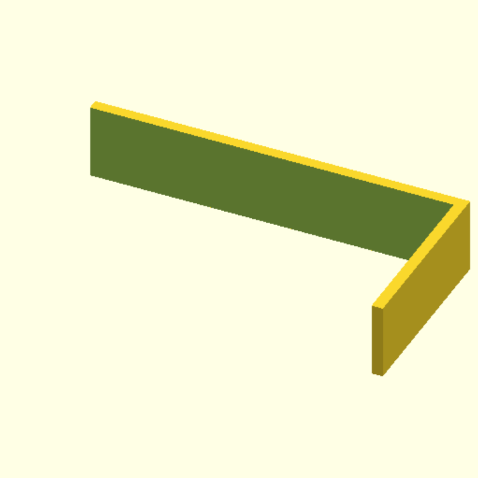
. . . this time for real!

```
wall = 1.5;
size = [100, 50, 20];
difference()
{
  cube(size);
  cube(size
       -2*wall*[1,1,0]);
}
```



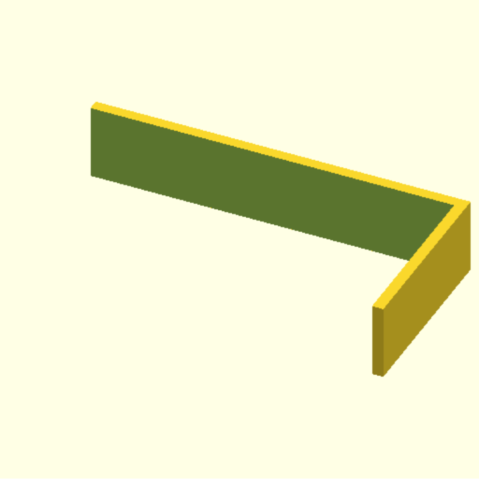
. . . this time for real!

```
wall = 1.5;
size = [100, 50, 20];
difference()
{
  cube(size);
  cube(size
       -2*wall*[1,1,0]);
}
```



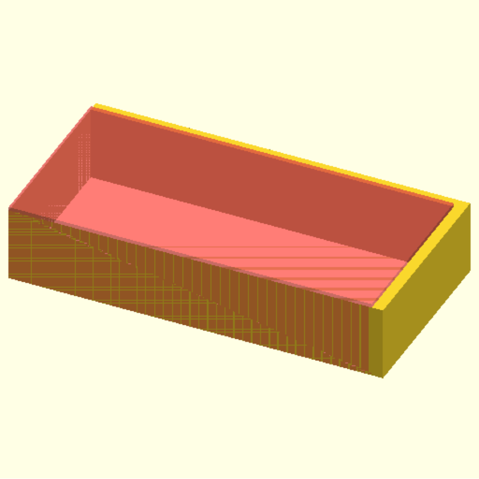
. . . this time for real!

```
wall = 1.5;
size = [100, 50, 20];
difference()
{
  cube(size);
  cube(size
       -2*wall*[1,1,0]);
}
```



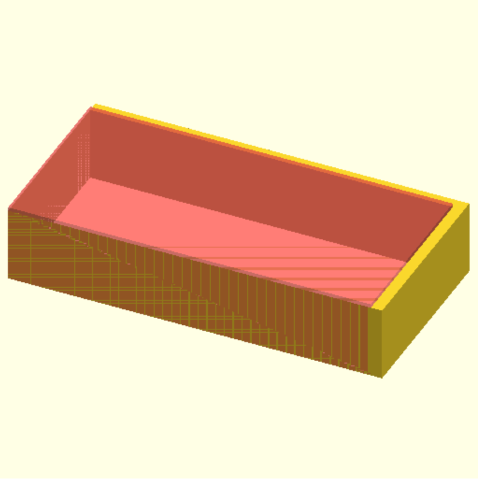
Debugging time. . .

```
wall = 1.5;
size = [100, 50, 20];
difference()
{
  cube(size);
  #cube(size
        - 2*wall*[1,1,0]
        + [0,0,1]);
}
```



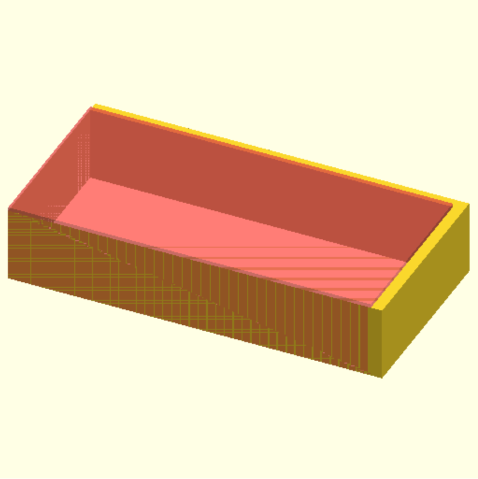
Debugging time. . .

```
wall = 1.5;
size = [100, 50, 20];
difference()
{
   cube(size);
   #cube(size
         - 2*wall*[1,1,0]
         + [0,0,1]);
}
```



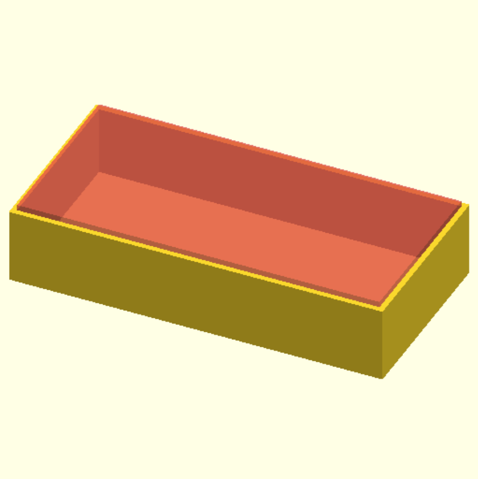
Debugging time. . .

```
wall = 1.5;
size = [100, 50, 20];
difference()
{
  cube(size);
  #cube(size
        - 2*wall*[1,1,0]
        + [0,0,1]);
}
```



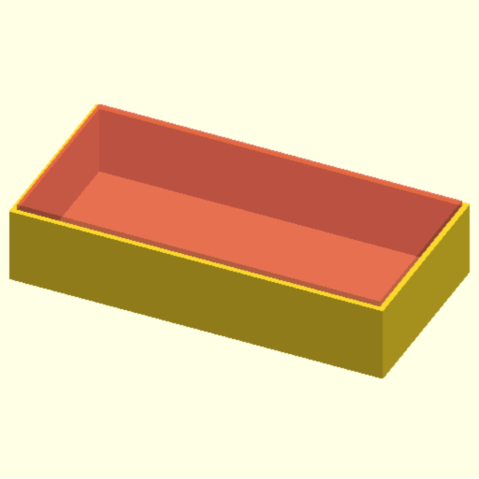
Re-positioning

```
wall = 1.5;
size = [100, 50, 20];
difference()
{
  cube(size);
  translate(wall*[1,1,1])
    #cube(size
         -2*wall*[1,1,0]);
}
```



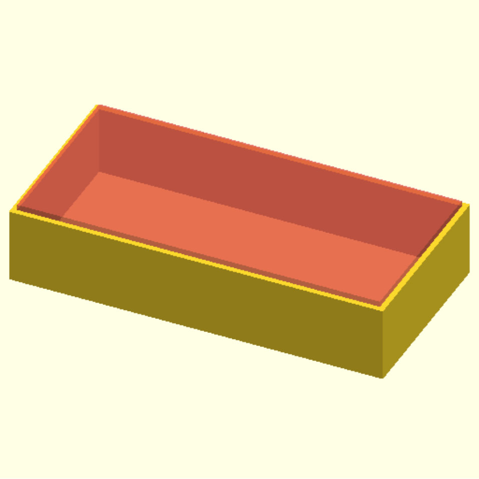
Re-positioning

```
wall = 1.5;
size = [100, 50, 20];
difference()
{
  cube(size);
  translate(wall*[1,1,1])
    #cube(size
         -2*wall*[1,1,0]);
}
```



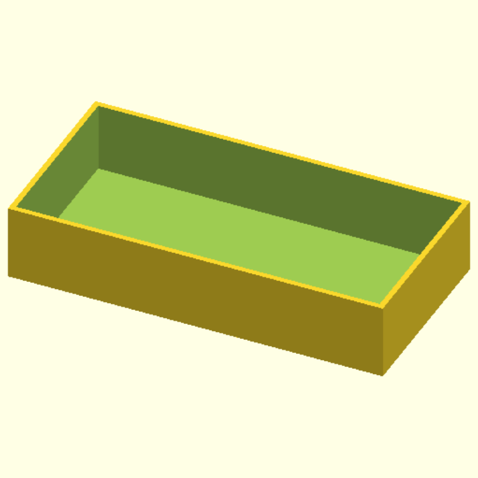
Re-positioning

```
wall = 1.5;
size = [100, 50, 20];
difference()
{
  cube(size);
  translate(wall*[1,1,1])
    #cube(size
         -2*wall*[1,1,0]);
}
```



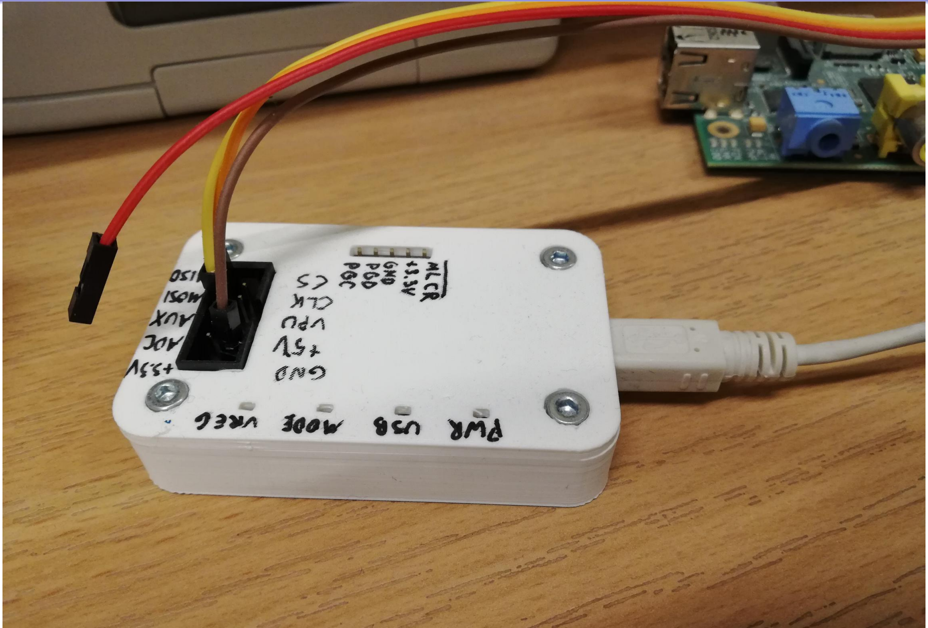
Final thing

```
wall = 1.5;
size = [100, 50, 20];
difference()
{
  cube(size);
  translate(wall*[1,1,1])
    cube(size
         -2*wall*[1,1,0]);
}
```

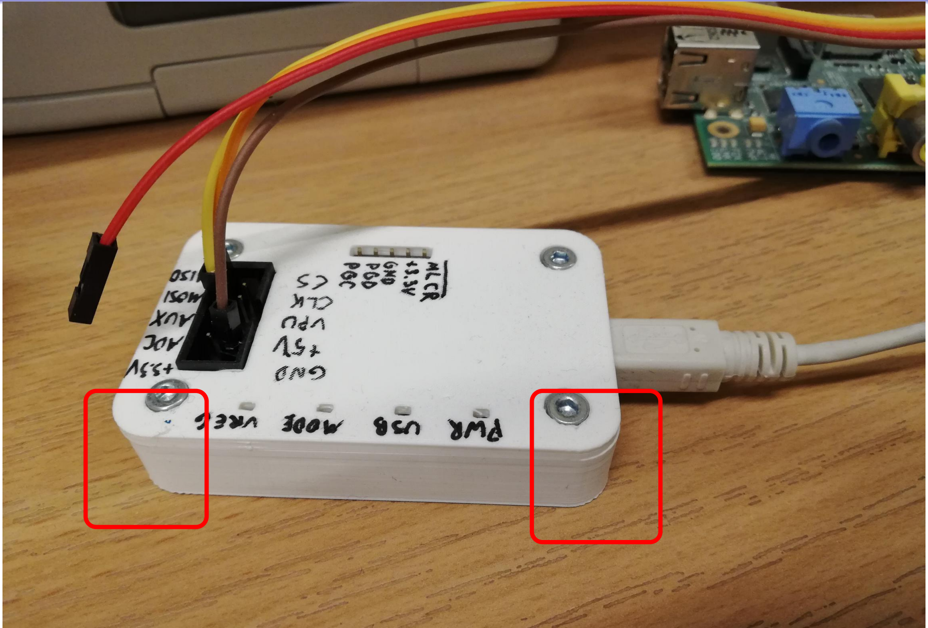


https://pbs.twimg.com/media/DSy9uQTVMAAyvI4.jpg

Rounded box

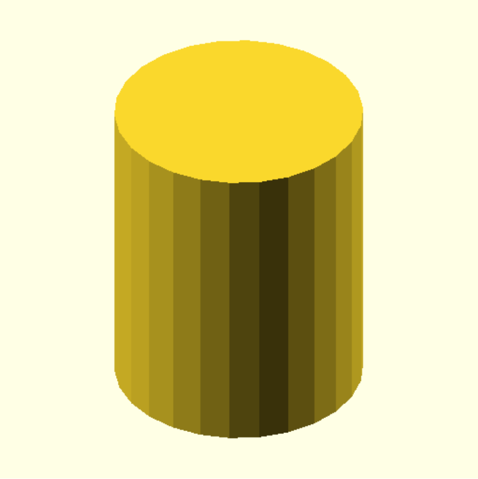

MISO
MOSI
AUX
ADC
+3.3V
CS
CLK
VPU
+5V
GND
MLCR
+3.3V
GND
PGD
PGC
VREG
MODE
USB
PWR

Rounded box


MLCR
+3.3V
GND
PGD
PGC
CS
CLK
VPU
+5V
GND
MISO
MOSI
AUX
ADC
+3.3V
VREG
MODE
USB
PWR

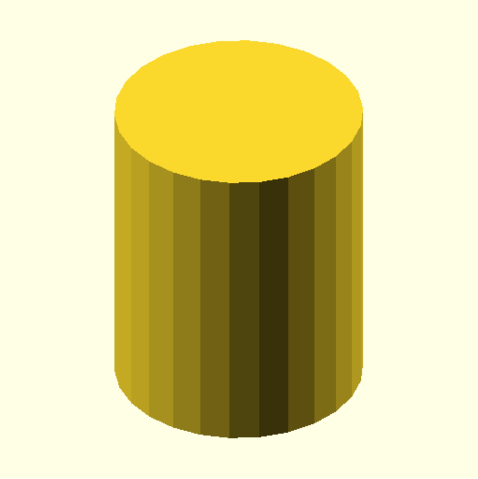
Cylinder

```
c_r = 8; // corrner R
size = [100, 50, 20];

cylinder(r=c_r,
         h=size[2]);
```



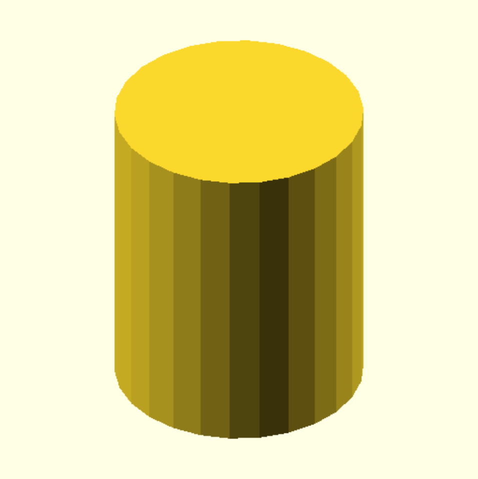
Cylinder

```
c_r = 8; // corrner R
size = [100, 50, 20];

cylinder(r=c_r,
         h=size[2]);
```



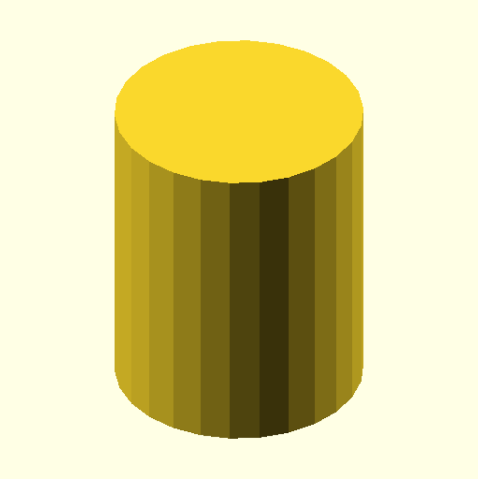
Cylinder

```
c_r = 8; // corrner R
size = [100, 50, 20];

cylinder(r=c_r,
         h=size[2]);
```



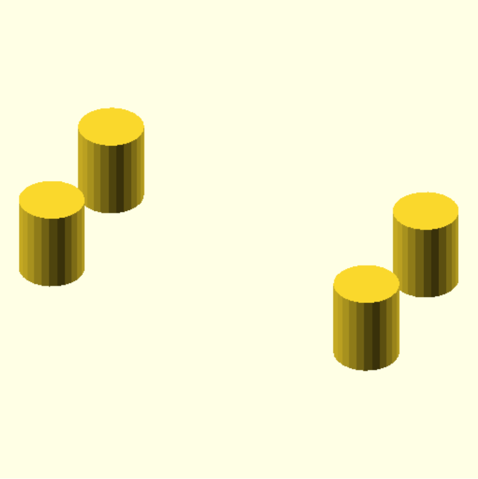
Cylinder

```
c_r = 8; // corrner R
size = [100, 50, 20];

cylinder(r=c_r,
         h=size[2]);
```



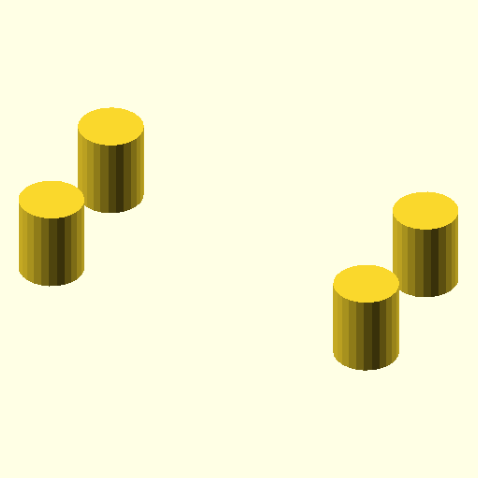
Looping me through!

```
c_r = 8; // corrner R
size = [100, 50, 20];
int_s = size -2*c_r*[1,1,0];

for(dx = [0:1])
  for(dy = [0:1])
    translate([dx*int_s[0],
               dy*int_s[1],
               0])
      cylinder(r=c_r,
               h=int_s[2]);
```



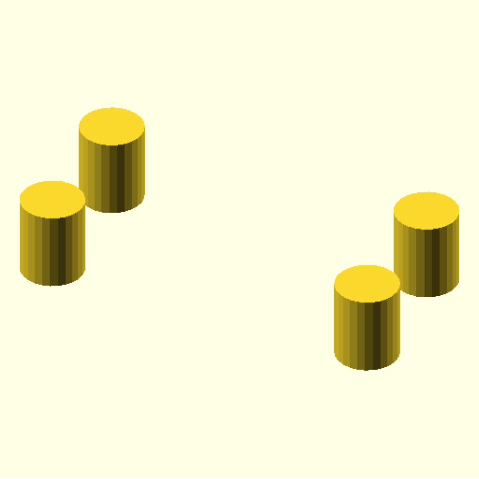
Looping me through!

```
c_r = 8; // corrner R
size = [100, 50, 20];
int_s = size -2*c_r*[1,1,0];

for(dx = [0:1])
  for(dy = [0:1])
    translate([dx*int_s[0],
               dy*int_s[1],
               0])
      cylinder(r=c_r,
               h=int_s[2]);
```



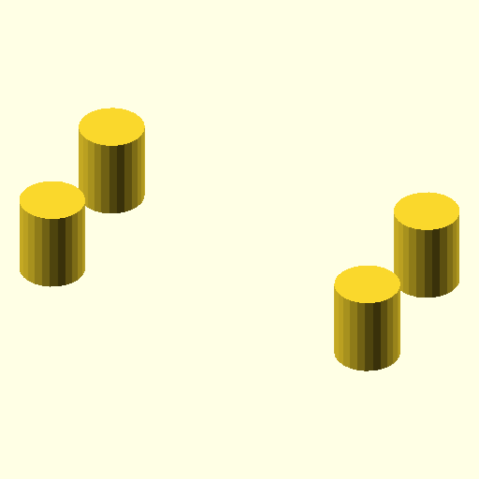
Looping me through!

```
c_r = 8; // corrner R
size = [100, 50, 20];
int_s = size -2*c_r*[1,1,0];

for(dx = [0:1])
  for(dy = [0:1])
    translate([dx*int_s[0],
               dy*int_s[1],
               0])
      cylinder(r=c_r,
               h=int_s[2]);
```



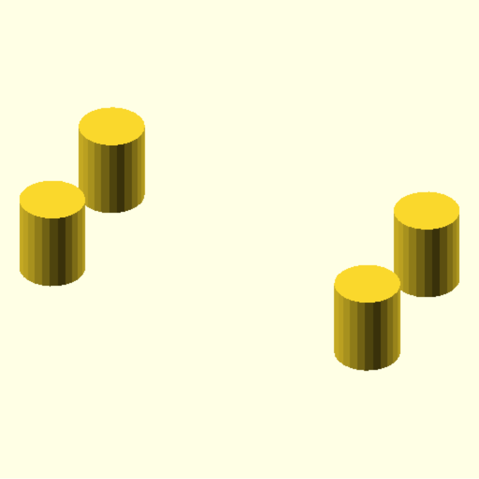
Looping me through!

```
c_r = 8; // corrner R
size = [100, 50, 20];
int_s = size -2*c_r*[1,1,0];

for(dx = [0:1])
  for(dy = [0:1])
    translate([dx*int_s[0],
               dy*int_s[1],
               0])
      cylinder(r=c_r,
               h=int_s[2]);
```



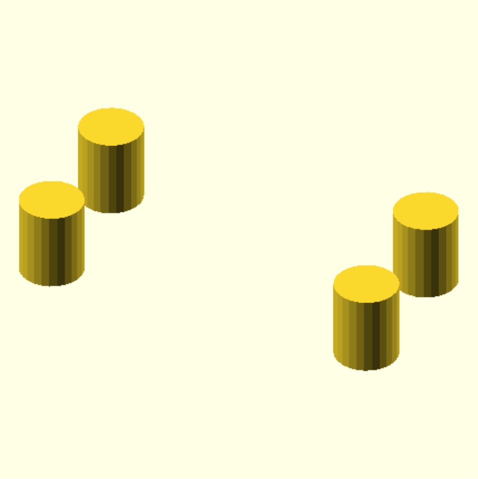
Looping me through!

```
c_r = 8; // corrner R
size = [100, 50, 20];
int_s = size -2*c_r*[1,1,0];

for(dx = [0:1])
  for(dy = [0:1])
    translate([dx*int_s[0],
               dy*int_s[1],
               0])
      cylinder(r=c_r,
               h=int_s[2]);
```



Looping me through!

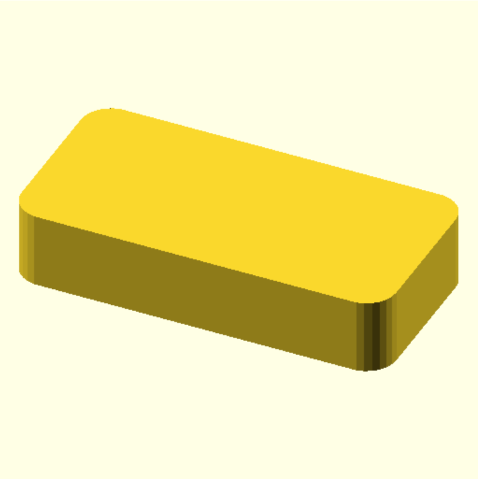
```
c_r = 8; // corrner R
size = [100, 50, 20];
int_s = size -2*c_r*[1,1,0];

for(dx = [0:1])
  for(dy = [0:1])
    translate([dx*int_s[0],
               dy*int_s[1],
               0])
      cylinder(r=c_r,
               h=int_s[2]);
```



CYLINDERS!
CYLINDERS EVERYWHERE!
imgflip.com

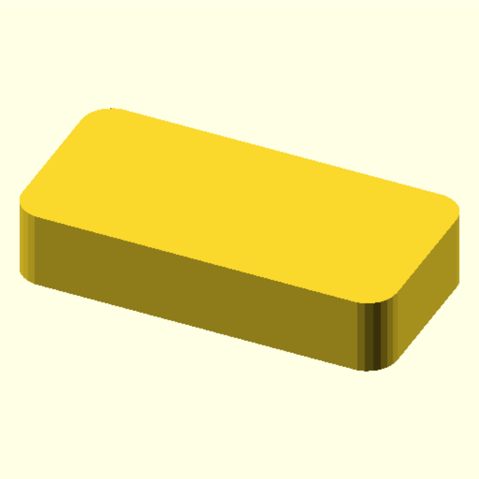
Creating hull

```
c_r = 8; // corrner R
size = [100, 50, 20];
int_s = size -2*c_r*[1,1,0];

hull()
for(dx = [0:1])
  for(dy = [0:1])
    translate([dx*int_s[0],
               dy*int_s[1],
               0])
      cylinder(r=c_r,
               h=int_s[2]);
```



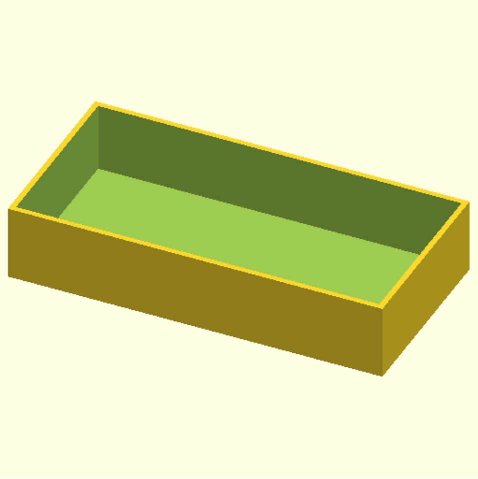
Creating hull

```
c_r = 8; // corrner R
size = [100, 50, 20];
int_s = size -2*c_r*[1,1,0];

hull()
for(dx = [0:1])
  for(dy = [0:1])
    translate([dx*int_s[0],
               dy*int_s[1],
               0])
      cylinder(r=c_r,
               h=int_s[2]);
```



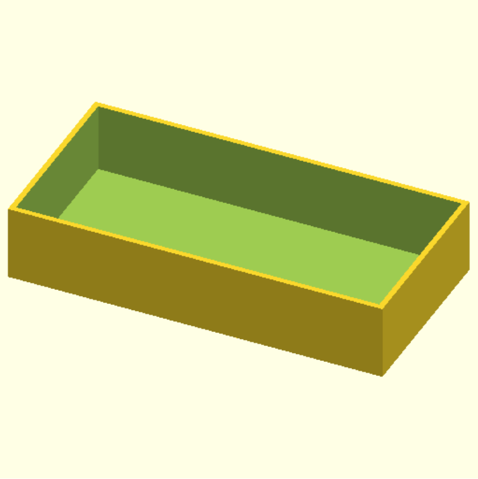
Work duplication...

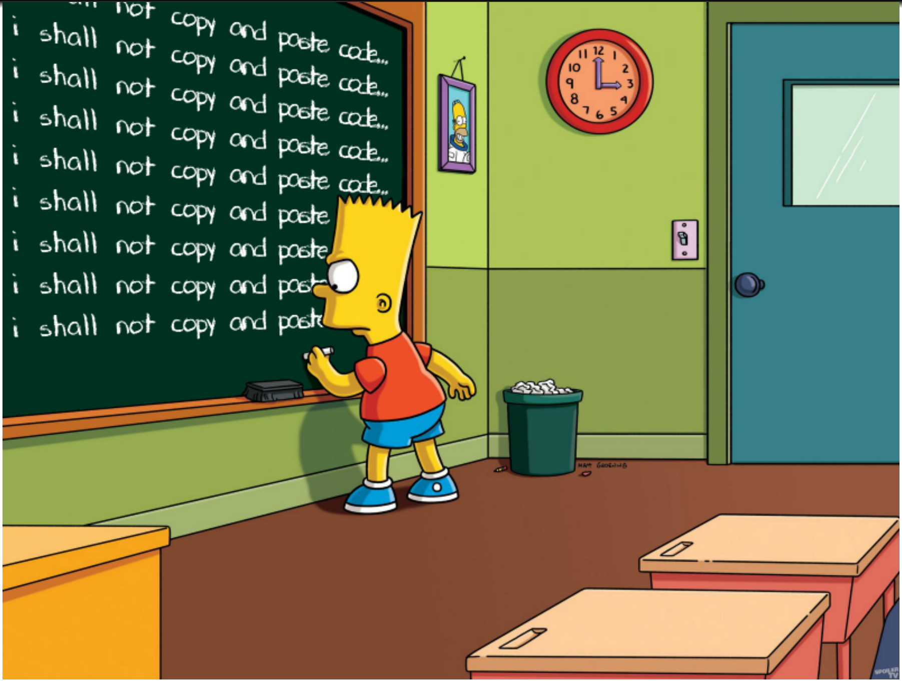
```
wall = 1.5;
size = [100, 50, 20];
difference()
{
  cube(size);
  translate(wall*[1,1,1])
    cube(size
         -2*wall*[1,1,0]);
}
```



Work duplication...

```
wall = 1.5;
size = [100, 50, 20];
difference()
{
  cube(size);
  translate(wall*[1,1,1])
    cube(size
         -2*wall*[1,1,0]);
}
```




i shall not copy and paste code...
i shall not copy and paste code...
i shall not copy and paste code...
i shall not copy and paste code...
i shall not copy and paste code...
i shall not copy and paste
i shall not copy and paste
i shall not copy and paste
i shall not copy and paste
12
11
1
10
2
9
3
8
4
7
6
5

https://www.ranzey.com/generators/bart/index.html

Module

```
module rounded_box(c_r, size)
{
  s = size -2*c_r*[1,1,0];

  hull()
    for(dx = [0:1])
      for(dy = [0:1])
        translate([dx*s[0], dy*s[1], 0])
          cylinder(r=c_r, h=s[2]);
}

rounded_box(c_r=8, size=[100,50,20]);
```

Module

```
module rounded_box(c_r, size)
{
  s = size -2*c_r*[1,1,0];

  hull()
    for(dx = [0:1])
      for(dy = [0:1])
        translate([dx*s[0], dy*s[1], 0])
          cylinder(r=c_r, h=s[2]);
}

rounded_box(c_r=8, size=[100,50,20]);
```

Module

```
module rounded_box(c_r, size)
{
  s = size -2*c_r*[1,1,0];

  hull()
    for(dx = [0:1])
      for(dy = [0:1])
        translate([dx*s[0], dy*s[1], 0])
          cylinder(r=c_r, h=s[2]);
}

rounded_box(c_r=8, size=[100,50,20]);
```

Module

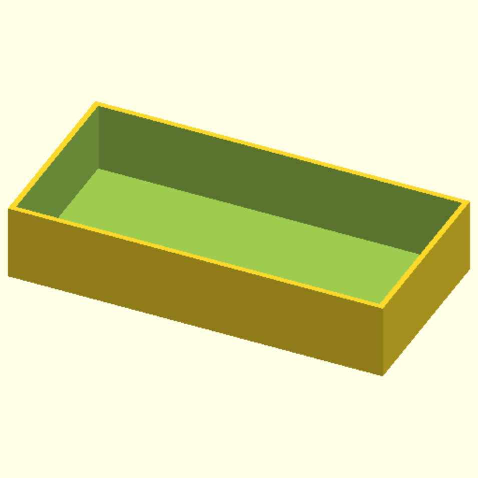
```
module rounded_box(c_r, size)
{
  s = size -2*c_r*[1,1,0];

  hull()
    for(dx = [0:1])
      for(dy = [0:1])
        translate([dx*s[0], dy*s[1], 0])
          cylinder(r=c_r, h=s[2]);
}

rounded_box(c_r=8, size=[100,50,20]);
```

Module

```
module rounded_box(c_r, size)
{
  s = size -2*c_r*[1,1,0];

  hull()
    for(dx = [0:1])
      for(dy = [0:1])
        translate([dx*s[0], dy*s[1], 0])
          cylinder(r=c_r, h=s[2]);
}

rounded_box(c_r=8, size=[100,50,20]);
```

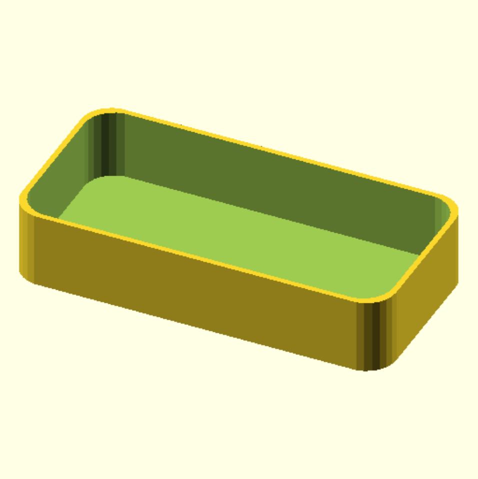
So we had. . .

```
wall = 1.5;
size = [100, 50, 20];
difference()
{
  cube(size);
  translate(wall*[1,1,1])
    cube(size
         -2*wall*[1,1,0]);
}
```



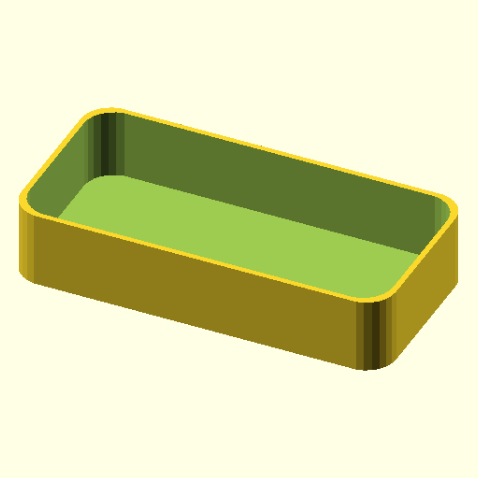
And now we have. . .

```
use <rounded_box.scad>
wall = 1.5;
size = [100, 50, 20];
r = 8;
difference()
{
  rounded_box(r, size);
  translate(wall*[1,1,1])
    rounded_box(r,
      size -2*wall*[1,1,0]);
}
```



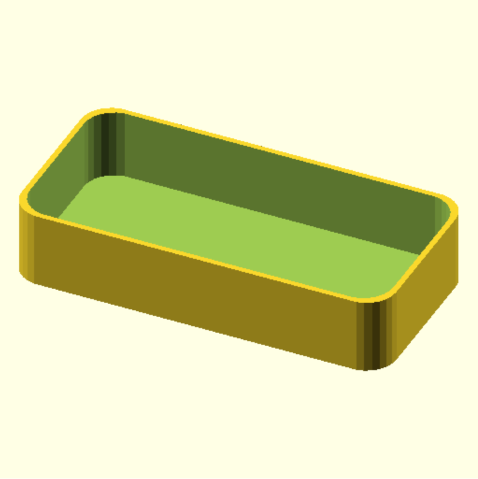
And now we have. . .

```
use <rounded_box.scad>
wall = 1.5;
size = [100, 50, 20];
r = 8;
difference()
{
  rounded_box(r, size);
  translate(wall*[1,1,1])
    rounded_box(r,
      size -2*wall*[1,1,0]);
}
```



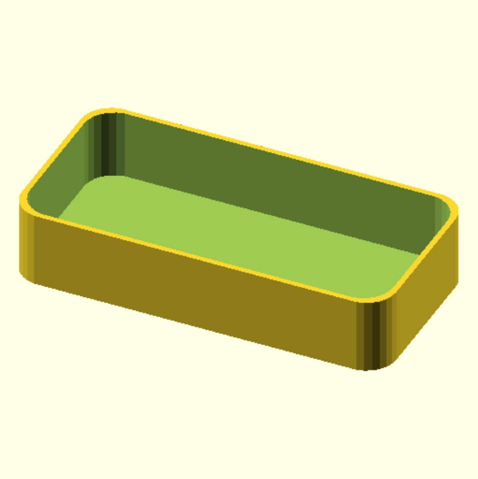
And now we have. . .

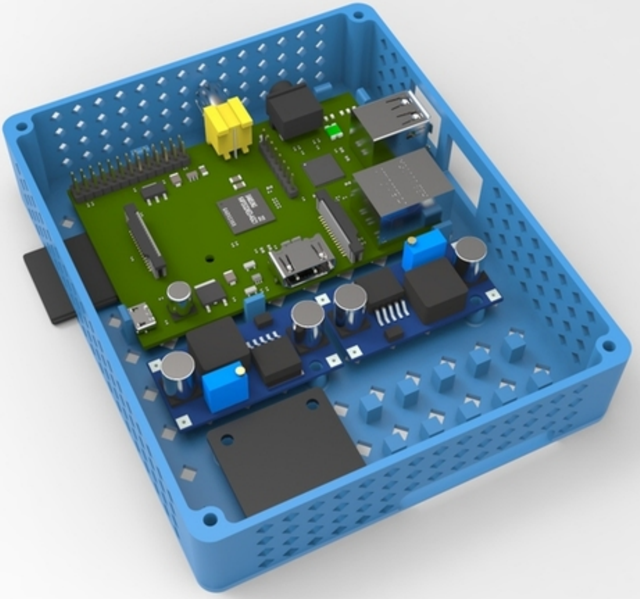
```
use <rounded_box.scad>
wall = 1.5;
size = [100, 50, 20];
r = 8;
difference()
{
  rounded_box(r, size);
  translate(wall*[1,1,1])
    rounded_box(r,
      size -2*wall*[1,1,0]);
}
```



And now we have. . .

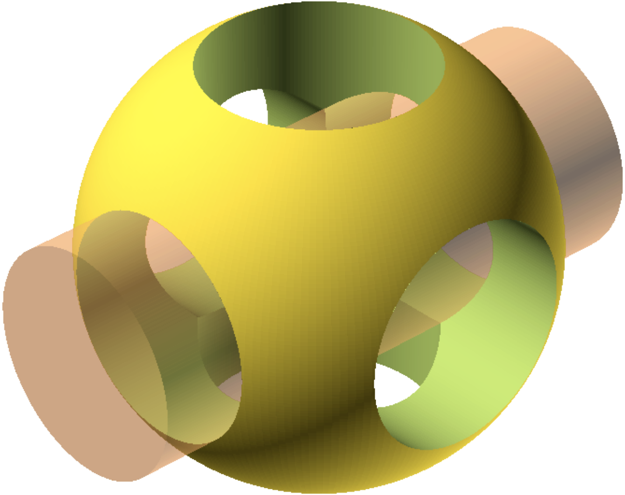
```
use <rounded_box.scad>
wall = 1.5;
size = [100, 50, 20];
r = 8;
difference()
{
  rounded_box(r, size);
  translate(wall*[1,1,1])
    rounded_box(r,
      size -2*wall*[1,1,0]);
}
```





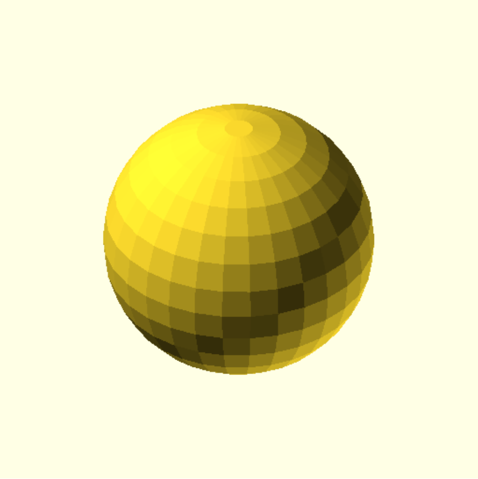
https://assets.pinshape.com/uploads/image/file/20145/container_electronics-box-for-3d-printer-3d-printing-20145.jpg

Logo time!



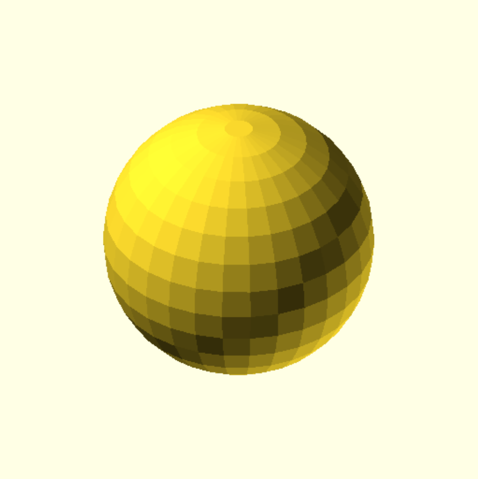
Starting point

```
r=20;
difference()
{
  sphere(r=r);
}
```



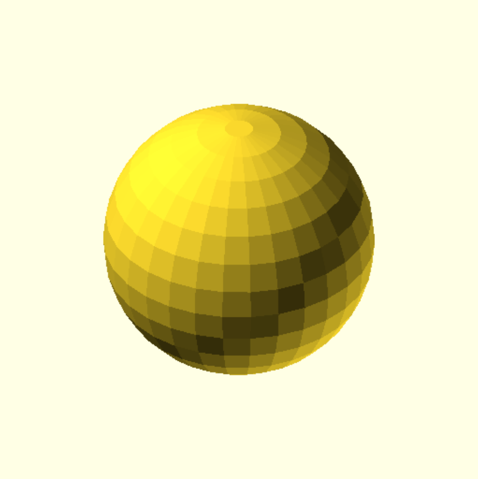
Starting point

```
r=20;
difference()
{
  sphere(r=r);
}
```



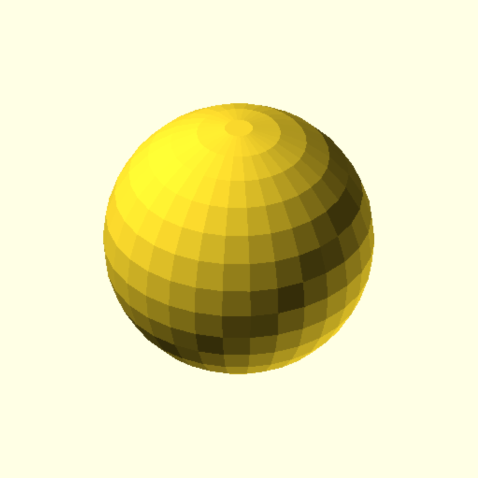
Starting point

```
r=20;
difference()
{
  sphere(r=r);
}
```



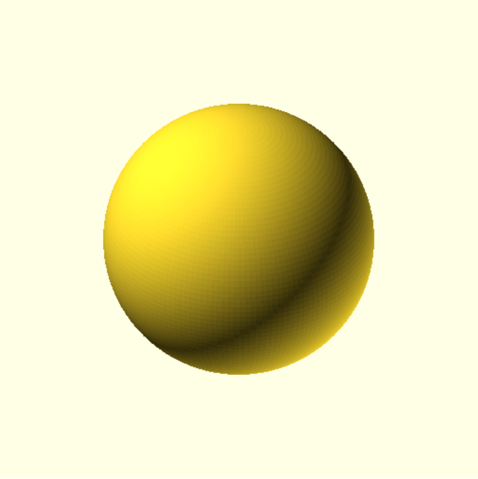
Starting point

```
r=20;
difference()
{
  sphere(r=r);
}
```



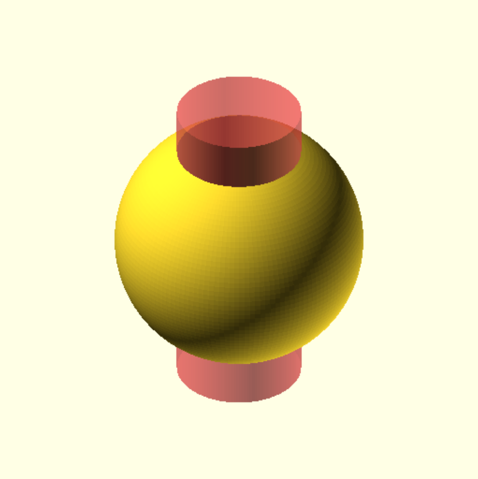
A better starting point

```
$fn=150;
r=20;
difference()
{
  sphere(r=r);
}
```



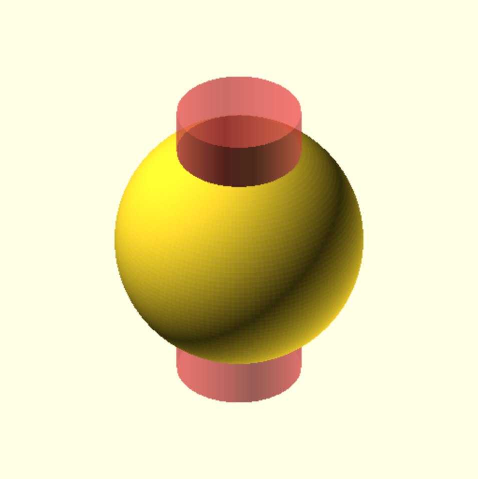
OZ

```
$fn=150;
r=20;
h=2.5*r;
difference()
{
  sphere(r=r);
  #translate([0, 0, -h/2])
    cylinder(r=r/2, h=h);
}
```



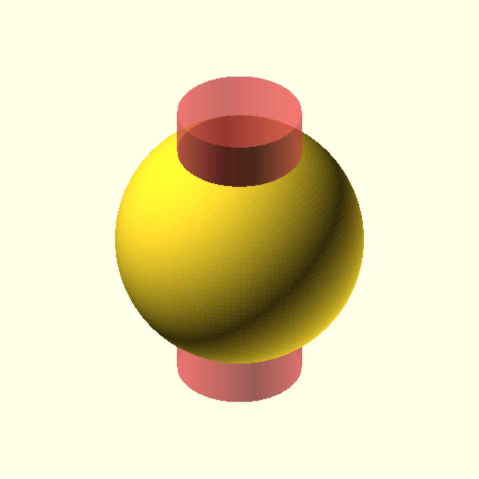
OZ

```
$fn=150;
r=20;
h=2.5*r;
difference()
{
  sphere(r=r);
  #translate([0, 0, -h/2])
    cylinder(r=r/2, h=h);
}
```



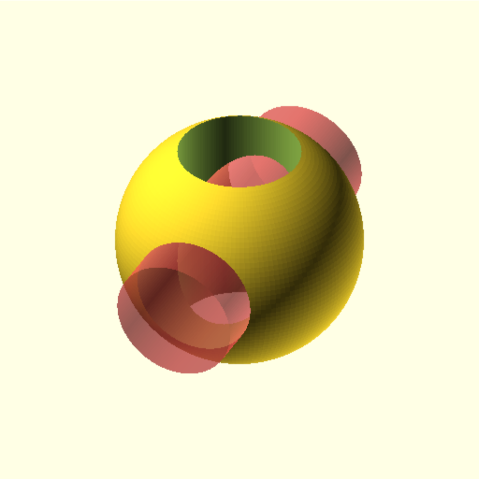
OZ

```
$fn=150;
r=20;
h=2.5*r;
difference()
{
  sphere(r=r);
  #translate([0, 0, -h/2])
    cylinder(r=r/2, h=h);
}
```



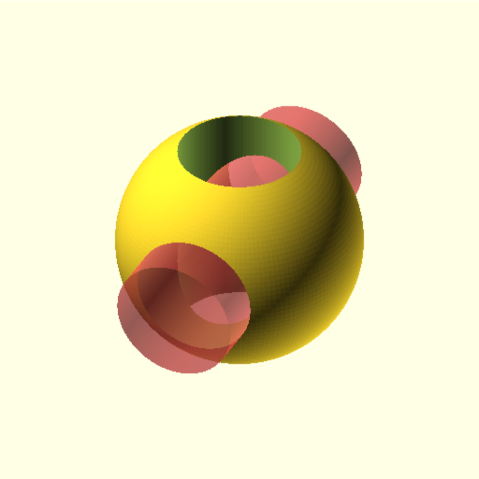
OY

```
$fn=150;
r=20;
h=2.5*r;
difference()
{
  sphere(r=r);
  translate([0, 0, -h/2])
    cylinder(r=r/2, h=h);
  #rotate([90,0,0])
    translate([0, 0, -h/2])
      cylinder(r=r/2, h=h);
}
```



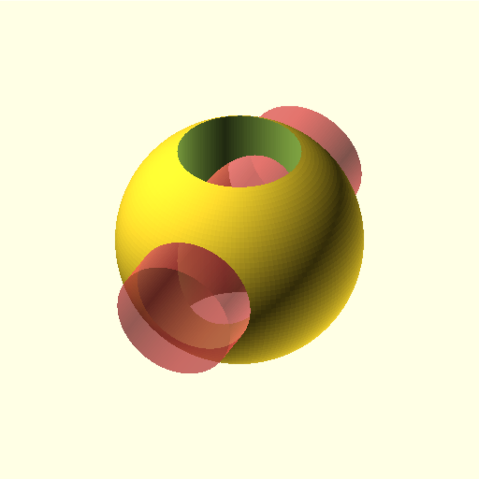
OY

```
$fn=150;
r=20;
h=2.5*r;
difference()
{
  sphere(r=r);
  translate([0, 0, -h/2])
    cylinder(r=r/2, h=h);
  #rotate([90,0,0])
    translate([0, 0, -h/2])
      cylinder(r=r/2, h=h);
}
```



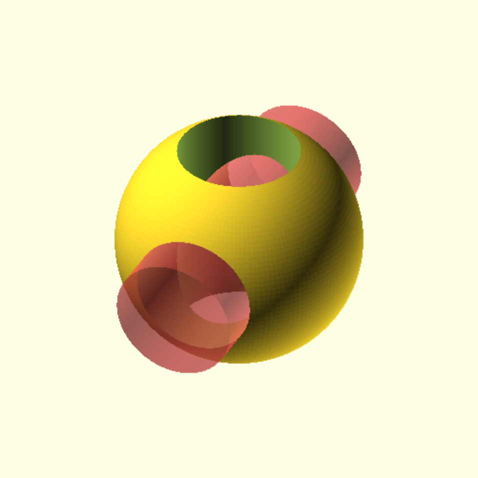
OY

```
$fn=150;
r=20;
h=2.5*r;
difference()
{
  sphere(r=r);
  translate([0, 0, -h/2])
    cylinder(r=r/2, h=h);
  #rotate([90,0,0])
    translate([0, 0, -h/2])
      cylinder(r=r/2, h=h);
}
```



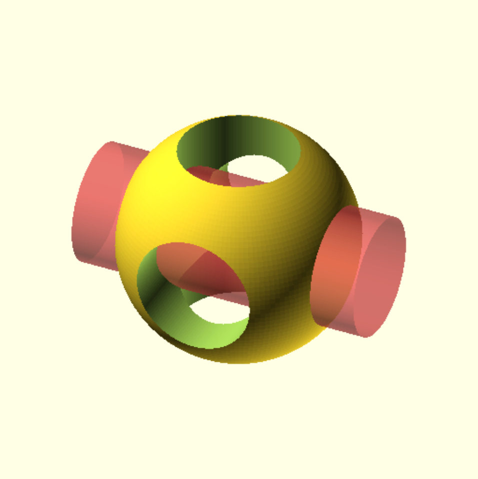
OY

```
$fn=150;
r=20;
h=2.5*r;
difference()
{
  sphere(r=r);
  translate([0, 0, -h/2])
    cylinder(r=r/2, h=h);
  #rotate([90,0,0])
    translate([0, 0, -h/2])
      cylinder(r=r/2, h=h);
}
```



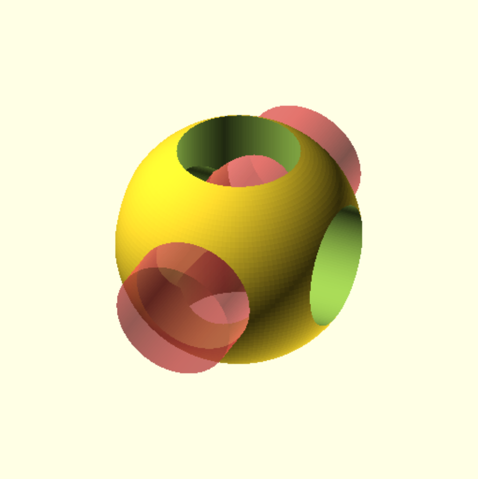
OX

```
$fn=150;   r=20;   h=2.5*r;
difference()
{
  sphere(r=r);
  translate([0, 0, -h/2])
    cylinder(r=r/2, h=h);
  rotate([90,0,0])
    translate([0, 0, -h/2])
      cylinder(r=r/2, h=h);
  #rotate([0,90,0])
    translate([0, 0, -h/2])
      cylinder(r=r/2, h=h);
}
```



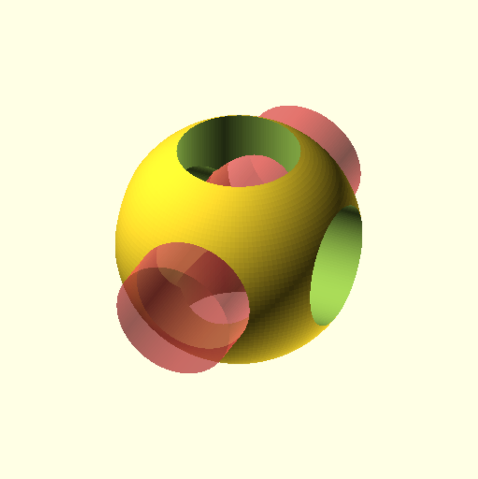
Good as new!

```
$fn=150;   r=20;   h=2.5*r;
difference()
{
  sphere(r=r);
  translate([0, 0, -h/2])
    cylinder(r=r/2, h=h);
  #rotate([90,0,0])
    translate([0, 0, -h/2])
      cylinder(r=r/2, h=h);
  rotate([0,90,0])
    translate([0, 0, -h/2])
      cylinder(r=r/2, h=h);
}
```



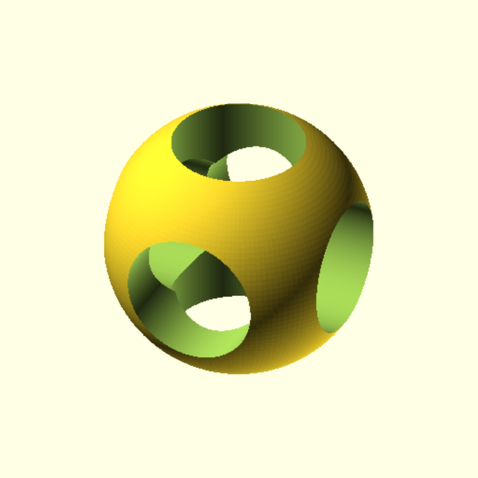
Good as new!

```
$fn=150;   r=20;   h=2.5*r;
difference()
{
  sphere(r=r);
  translate([0, 0, -h/2])
    cylinder(r=r/2, h=h);
  #rotate([90,0,0])
    translate([0, 0, -h/2])
      cylinder(r=r/2, h=h);
  rotate([0,90,0])
    translate([0, 0, -h/2])
      cylinder(r=r/2, h=h);
}
```



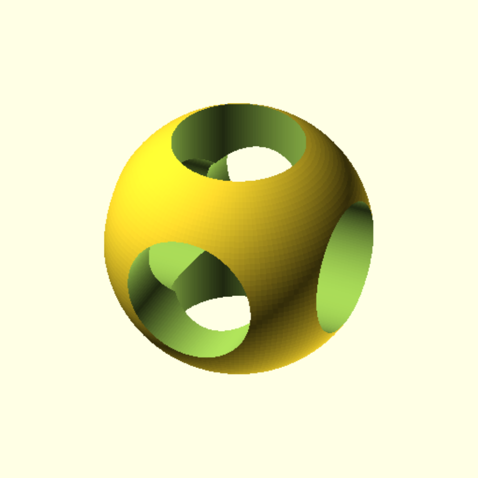
Oh! BTW!

```
$fn=150;  r=20;  h=2.5*r;
difference()
{
  sphere(r=r);
  for(rot=[ [0,  0, 0],
            [90, 0, 0],
            [0, 90, 0] ])
    rotate(rot)
      translate([0,0,-h/2])
        cylinder(r=r/2,h=h);
}
```



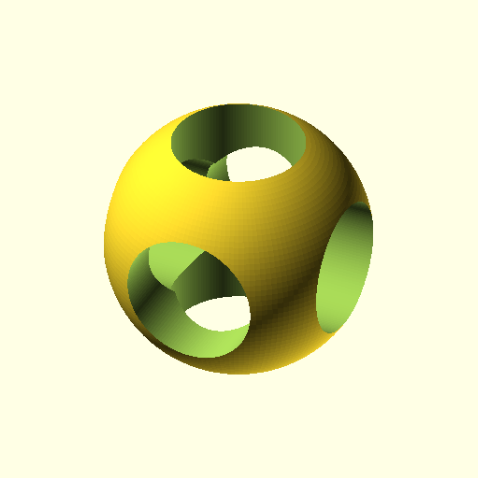
Oh! BTW!

```
$fn=150;  r=20;  h=2.5*r;
difference()
{
  sphere(r=r);
  for(rot=[ [0,  0, 0],
            [90, 0, 0],
            [0, 90, 0] ])
    rotate(rot)
      translate([0,0,-h/2])
        cylinder(r=r/2,h=h);
}
```



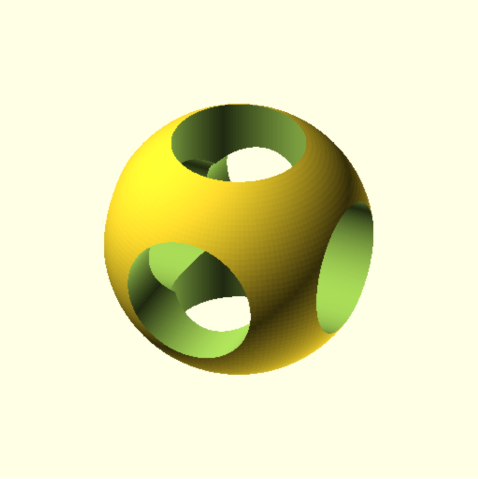
Oh! BTW!

```
$fn=150;  r=20;  h=2.5*r;
difference()
{
  sphere(r=r);
  for(rot=[ [0,  0, 0],
            [90, 0, 0],
            [0, 90, 0] ])
    rotate(rot)
      translate([0,0,-h/2])
        cylinder(r=r/2,h=h);
}
```



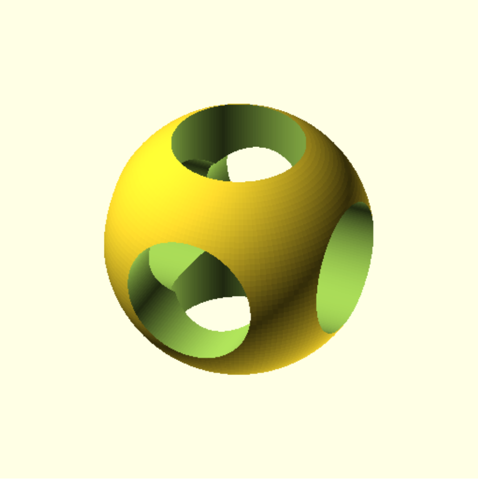
Oh! BTW!

```
$fn=150;  r=20;  h=2.5*r;
difference()
{
  sphere(r=r);
  for(rot=[ [0,  0, 0],
            [90, 0, 0],
            [0, 90, 0] ])
    rotate(rot)
      translate([0,0,-h/2])
        cylinder(r=r/2,h=h);
}
```



Oh! BTW!

```
$fn=150;  r=20;  h=2.5*r;
difference()
{
  sphere(r=r);
  for(rot=[ [0,  0, 0],
            [90, 0, 0],
            [0, 90, 0] ])
    rotate(rot)
      translate([0,0,-h/2])
        cylinder(r=r/2,h=h);
}
```

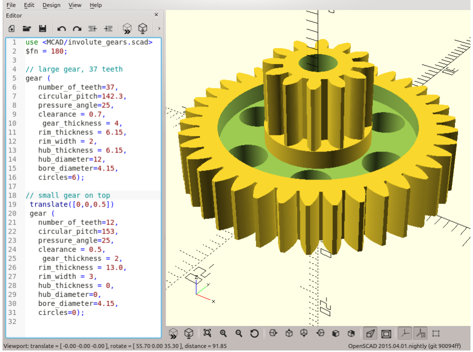


What’s all?

https://betanews.com/wp-content/uploads/2014/07/Im-an-expert-600x412.jpg

Gains

- OpenSCAD **source code**
- VCS! :)



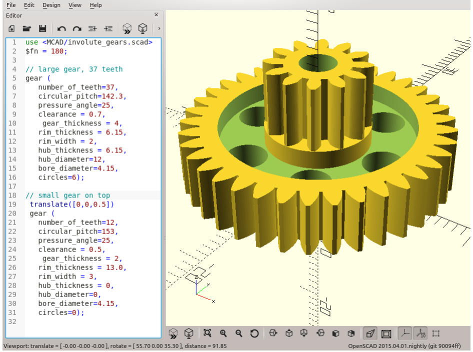
```
use <MCAD/involute_gears.scad>
$fn = 180;

// large gear, 37 teeth
gear (
   number_of_teeth=37,
   circular_pitch=142.3,
   pressure_angle=25,
   clearance = 0.7,
    gear_thickness = 4,
   rim_thickness = 6.15,
   rim_width = 2,
   hub_thickness = 6.15,
   hub_diameter=12,
   bore_diameter=4.15,
   circles=6);

// small gear on top
 translate([0,0,0.5])
 gear (
   number_of_teeth=12,
   circular_pitch=153,
   pressure_angle=25,
   clearance = 0.5,
    gear_thickness = 2,
   rim_thickness = 13.0,
   rim_width = 3,
   hub_thickness = 0,
   hub_diameter=0,
   bore_diameter=4.15,
   circles=0);

```

Viewport: translate = [-0.00 -0.00 -0.00], rotate = [55.70 0.00 35.30], distance = 91.85

OpenSCAD 2015.04.01.nightly (git 90094ff)

Gains

- OpenSCAD **source code**
- VCS! :)
- Standard workflow:
  - Code diff
  - Pull/merge requests
  - Code review



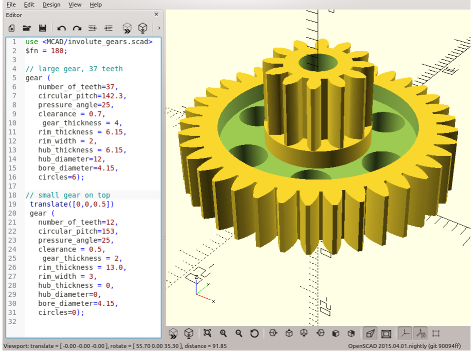
```
use <MCAD/involute_gears.scad>
$fn = 180;

// large gear, 37 teeth
gear (
   number_of_teeth=37,
   circular_pitch=142.3,
   pressure_angle=25,
   clearance = 0.7,
    gear_thickness = 4,
   rim_thickness = 6.15,
   rim_width = 2,
   hub_thickness = 6.15,
   hub_diameter=12,
   bore_diameter=4.15,
   circles=6);

// small gear on top
 translate([0,0,0.5])
 gear (
   number_of_teeth=12,
   circular_pitch=153,
   pressure_angle=25,
   clearance = 0.5,
    gear_thickness = 2,
   rim_thickness = 13.0,
   rim_width = 3,
   hub_thickness = 0,
   hub_diameter=0,
   bore_diameter=4.15,
   circles=0);

```

Viewport: translate = [-0.00 -0.00 -0.00], rotate = [55.70 0.00 35.30], distance = 91.85

OpenSCAD 2015.04.01.nightly (git 90094ff)

Gains

- OpenSCAD **source code**
- VCS! :)
- Standard workflow:
  - Code diff
  - Pull/merge requests
  - Code review
  - Branching and merging
  - CI-based build
  - Preview generation



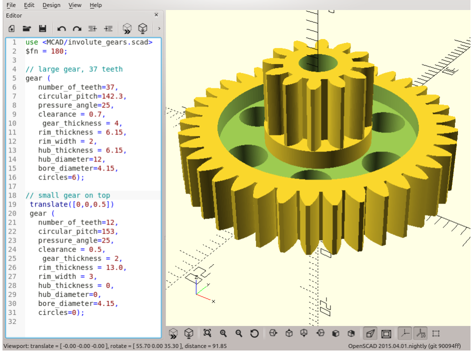
```
use <MCAD/involute_gears.scad>
$fn = 180;

// large gear, 37 teeth
gear (
   number_of_teeth=37,
   circular_pitch=142.3,
   pressure_angle=25,
   clearance = 0.7,
    gear_thickness = 4,
   rim_thickness = 6.15,
   rim_width = 2,
   hub_thickness = 6.15,
   hub_diameter=12,
   bore_diameter=4.15,
   circles=6);

// small gear on top
 translate([0,0,0.5])
 gear (
   number_of_teeth=12,
   circular_pitch=153,
   pressure_angle=25,
   clearance = 0.5,
    gear_thickness = 2,
   rim_thickness = 13.0,
   rim_width = 3,
   hub_thickness = 0,
   hub_diameter=0,
   bore_diameter=4.15,
   circles=0);

```

Viewport: translate = [-0.00 -0.00 -0.00], rotate = [55.70 0.00 35.30], distance = 91.85

OpenSCAD 2015.04.01.nightly (git 90094ff)

Gains

- OpenSCAD **source code**
- VCS! :)
- Standard workflow:
  - Code diff
  - Pull/merge requests
  - Code review
  - Branching and merging
  - CI-based build
  - Preview generation
- Techniques:
  - Code-reuse
  - Avoid hardcodes
  - Modularization
  - Encapsulation



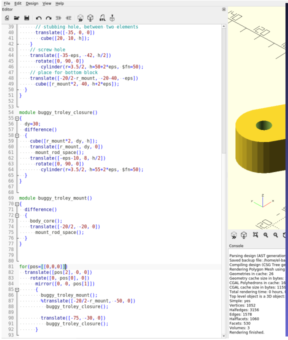
```
use <MCAD/involute_gears.scad>
$fn = 180;

// large gear, 37 teeth
gear (
   number_of_teeth=37,
   circular_pitch=142.3,
   pressure_angle=25,
   clearance = 0.7,
    gear_thickness = 4,
   rim_thickness = 6.15,
   rim_width = 2,
   hub_thickness = 6.15,
   hub_diameter=12,
   bore_diameter=4.15,
   circles=6);

// small gear on top
 translate([0,0,0.5])
 gear (
   number_of_teeth=12,
   circular_pitch=153,
   pressure_angle=25,
   clearance = 0.5,
    gear_thickness = 2,
   rim_thickness = 13.0,
   rim_width = 3,
   hub_thickness = 0,
   hub_diameter=0,
   bore_diameter=4.15,
   circles=0);

```

Viewport: translate = [-0.00 -0.00 -0.00], rotate = [55.70 0.00 35.30], distance = 91.85 OpenSCAD 2015.04.01.nightly (git 90094ff)

http://arnholm.org/wp-content/uploads/2015/04/image51.png

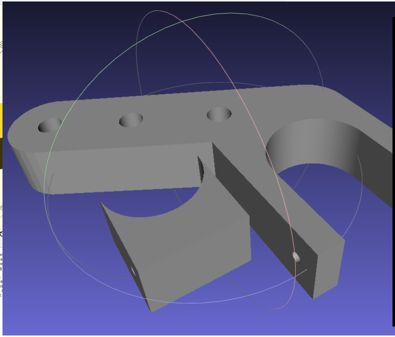
Where we are?

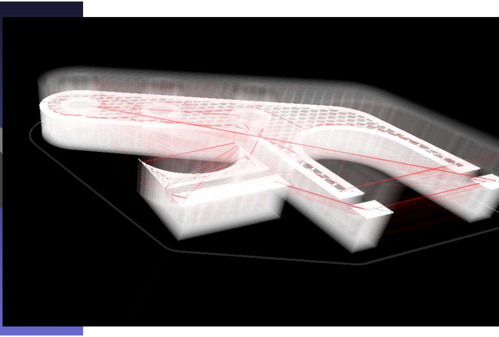
- **CAD model**
- **STL**
- **G-code**



```
      // stubbing hole, between two elements
      translate([-35, 0, 0])
        cube([20, 10, h]);
    }
    // screw hole
    translate([-35-eps, -42, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=50+2*eps, $fn=50);
    // place for bottom block
    translate([-20/2-r_mount, -20-40, -eps])
      cube([r_mount*2, 40, h+2*eps]);
  }
}


module buggy_troley_closure()
{
  dy=30;
  difference()
  {
    cube([r_mount*2, dy, h]);
    translate([r_mount, dy, 0])
      mount_rod_space();
    translate([-eps-10, 8, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=55+2*eps, $fn=50);
  }
}


module buggy_troley_mount()
{
  difference()
  {
    body_core();
    translate([-20/2, -20, 0])
      mount_rod_space();
  }
}



for(pos=[[0,0,0]])
  translate([pos[2], 0, 0])
    rotate([0, pos[0], 0])
      mirror([0, 0, pos[1]])
      {
        buggy_troley_mount();
        %translate([-20/2-r_mount, -50, 0])
          buggy_troley_closure();

        translate([-75, -30, 0])
          buggy_troley_closure();
      }
```

```
Parsing design (AST generation
Saved backup file: /home/el-ba
Compiling design (CSG Tree ge
Rendering Polygon Mesh using
Geometries in cache: 26
Geometry cache size in bytes:
CGAL Polyhedrons in cache: 16
CGAL cache size in bytes: 115
Total rendering time: 0 hours,
Top level object is a 3D object:
Simple: yes
Vertices: 1052
Halfedges: 3156
Edges: 1578
Halffacets: 1060
Facets: 530
Volumes: 3
Rendering finished.
```





"Java source"

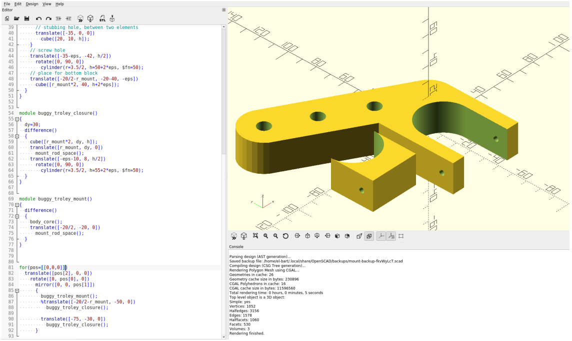
"Java IR"

"Machine code"

https://github.com/el-bart/mini/tree/master/buggy_troley_mount

Where we are?

- **CAD model**



```
      // stubbing hole, between two elements
      translate([-35, 0, 0])
        cube([20, 10, h]);
    }
    // screw hole
    translate([-35-eps, -42, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=50+2*eps, $fn=50);
    // place for bottom block
    translate([-20/2-r_mount, -20-40, -eps])
      cube([r_mount*2, 40, h+2*eps]);
  }
}


module buggy_troley_closure()
{
  dy=30;
  difference()
  {
    cube([r_mount*2, dy, h]);
    translate([r_mount, dy, 0])
      mount_rod_space();
    translate([-eps-10, 8, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=55+2*eps, $fn=50);
  }
}


module buggy_troley_mount()
{
  difference()
  {
    body_core();
    translate([-20/2, -20, 0])
      mount_rod_space();
  }
}



for(pos=[[0,0,0]])
  translate([pos[2], 0, 0])
    rotate([0, pos[0], 0])
      mirror([0, 0, pos[1]])
      {
        buggy_troley_mount();
        %translate([-20/2-r_mount, -50, 0])
          buggy_troley_closure();

        translate([-75, -30, 0])
          buggy_troley_closure();
      }
```

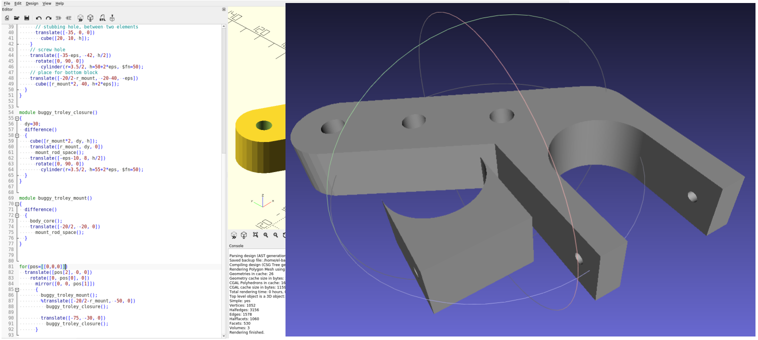
```
Parsing design (AST generation)...
Saved backup file: /home/el-bart/.local/share/OpenSCAD/backups/mount-backup-fkvWyLcT.scad
Compiling design (CSG Tree generation)...
Rendering Polygon Mesh using CGAL...
Geometries in cache: 26
Geometry cache size in bytes: 230896
CGAL Polyhedrons in cache: 16
CGAL cache size in bytes: 11596560
Total rendering time: 0 hours, 0 minutes, 5 seconds
Top level object is a 3D object:
Simple: yes
Vertices: 1052
Halfedges: 3156
Edges: 1578
Halffacets: 1060
Facets: 530
Volumes: 3
Rendering finished.
```

"Java source"

https://github.com/el-bart/mini/tree/master/buggy_troley_mount

Where we are?

- **CAD model**
- **STL**



```
      // stubbing hole, between two elements
      translate([-35, 0, 0])
        cube([20, 10, h]);
    }
    // screw hole
    translate([-35-eps, -42, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=50+2*eps, $fn=50);
    // place for bottom block
    translate([-20/2-r_mount, -20-40, -eps])
      cube([r_mount*2, 40, h+2*eps]);
  }
}


module buggy_troley_closure()
{
  dy=30;
  difference()
  {
    cube([r_mount*2, dy, h]);
    translate([r_mount, dy, 0])
      mount_rod_space();
    translate([-eps-10, 8, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=55+2*eps, $fn=50);
  }
}


module buggy_troley_mount()
{
  difference()
  {
    body_core();
    translate([-20/2, -20, 0])
      mount_rod_space();
  }
}



for(pos=[[0,0,0]])
  translate([pos[2], 0, 0])
    rotate([0, pos[0], 0])
      mirror([0, 0, pos[1]])
      {
        buggy_troley_mount();
        %translate([-20/2-r_mount, -50, 0])
          buggy_troley_closure();

        translate([-75, -30, 0])
          buggy_troley_closure();
      }
```

```
Parsing design (AST generation
Saved backup file: /home/el-ba
Compiling design (CSG Tree ge
Rendering Polygon Mesh using
Geometries in cache: 26
Geometry cache size in bytes:
CGAL Polyhedrons in cache: 16
CGAL cache size in bytes: 1159
Total rendering time: 0 hours, 0
Top level object is a 3D object:
Simple: yes
Vertices: 1052
Halfedges: 3156
Edges: 1578
Halffacets: 1060
Facets: 530
Volumes: 3
Rendering finished.
```

"Java source"

"Java IR"

https://github.com/el-bart/mini/tree/master/buggy_troley_mount

STL generation

```
openscad \
  -o "stuff.stl" \
  "stuff.scad"
```

https://pleated-jeans.com/wp-content/uploads/2015/09/trees4-1.jpg

STL generation

```
openscad \
  -o "stuff.stl" \
  "stuff.scad"
```

https://pleated-jeans.com/wp-content/uploads/2015/09/trees4-1.jpg

STL generation

```
openscad \
  -o "stuff.stl" \
  "stuff.scad"
```

https://pleated-jeans.com/wp-content/uploads/2015/09/trees4-1.jpg

STL generation

```
openscad \
  -o "stuff.stl" \
  "stuff.scad"
```

https://pleated-jeans.com/wp-content/uploads/2015/09/trees4-1.jpg

STL generation

```
openscad \
   -o "stuff.stl" \
   "stuff.scad"
```

https://pleated-jeans.com/wp-content/uploads/2015/09/trees4-1.jpg

...or even better!

```
openscad \
  --hardwarnings \
  --check-parameters true \
  --check-parameter-ranges true \
  -o "stuff.stl" \
  "stuff.scad"
```

. . . or even better!

```
openscad \
   --hardwarnings \
   --check-parameters true \
   --check-parameter-ranges true \
   -o "stuff.stl" \
   "stuff.scad"
```

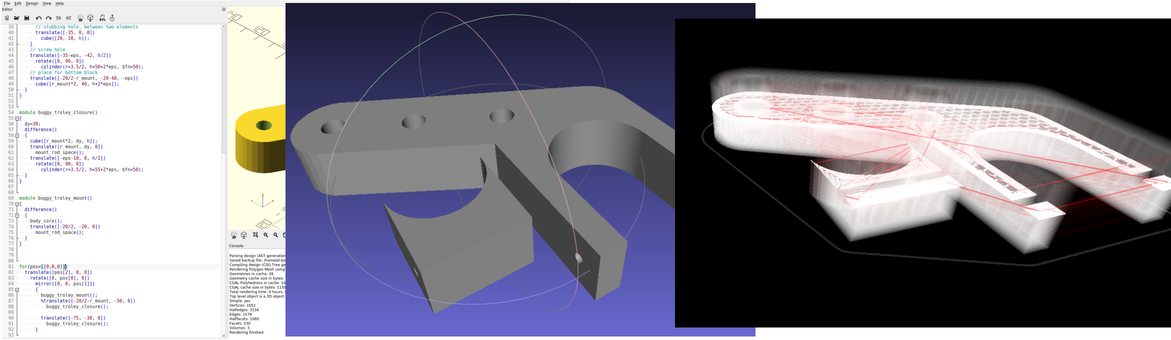
. . . or even better!

Time for. . .

1. 3D printing 101
2. Modeling in OpenSCAD
3. Slicing
4. ETA
5. Pipeline
6. Ending

Remeber pipeline?

- **CAD model**
- **STL**
- **G-code**



```
      // stubbing hole, between two elements
      translate([-35, 0, 0])
        cube([20, 10, h]);
    }
    // screw hole
    translate([-35-eps, -42, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=50+2*eps, $fn=50);
    // place for bottom block
    translate([-20/2-r_mount, -20-40, -eps])
      cube([r_mount*2, 40, h+2*eps]);
  }
}


module buggy_troley_closure()
{
  dy=30;
  difference()
  {
    cube([r_mount*2, dy, h]);
    translate([r_mount, dy, 0])
      mount_rod_space();
    translate([-eps-10, 8, h/2])
      rotate([0, 90, 0])
        cylinder(r=3.5/2, h=55+2*eps, $fn=50);
  }
}


module buggy_troley_mount()
{
  difference()
  {
    body_core();
    translate([-20/2, -20, 0])
      mount_rod_space();
  }
}



for(pos=[[0,0,0]])
  translate([pos[2], 0, 0])
    rotate([0, pos[0], 0])
      mirror([0, 0, pos[1]])
      {
        buggy_troley_mount();
        %translate([-20/2-r_mount, -50, 0])
          buggy_troley_closure();

        translate([-75, -30, 0])
          buggy_troley_closure();
      }
```

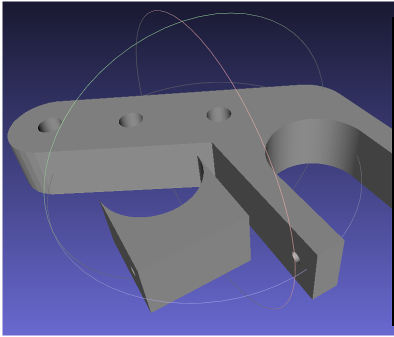
```
Parsing design (AST generation
Saved backup file: /home/el-ba
Compiling design (CSG Tree ge
Rendering Polygon Mesh using
Geometries in cache: 26
Geometry cache size in bytes:
CGAL Polyhedrons in cache: 16
CGAL cache size in bytes: 1159
Total rendering time: 0 hours,
Top level object is a 3D object:
Simple: yes
Vertices: 1052
Halfedges: 3156
Edges: 1578
Halffacets: 1060
Facets: 530
Volumes: 3
Rendering finished.
```

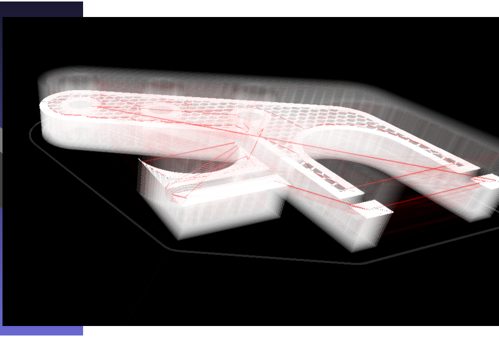
"Java source" "Java IR" "Machine code"

https://github.com/el-bart/mini/tree/master/buggy_troley_mount

Remeber pipeline?

- **STL**
- **G-code**





"Java IR"

"Machine code"

https://github.com/el-bart/mini/tree/master/buggy_troley_mount

Tools

Tools

Slic3r

PrusaSlicer
THE EASIEST WAY TO GET PERFECT PRINTS
ON THE ORIGINAL PRUSA 3D PRINTERS!

Tools

Slic3r

PrusaSlicer
THE EASIEST WAY TO GET PERFECT PRINTS
ON THE ORIGINAL PRUSA 3D PRINTERS!

cura.

Tools

Slic3r

PrusaSlicer

THE EASIEST WAY TO GET PERFECT PRINTS
ON THE ORIGINAL PRUSA 3D PRINTERS!

cura.

SIMPLIFY3D™

Slic3r from now on

Command line

```
slic3r \
    --load "config.ini" \
    --output "model.gcode" \
    "model.stl"
```

Command line

```
slic3r \
  --load "config.ini" \
  --output "model.gcode" \
  "model.stl"
```

Command line

```
slic3r \
  --load "config.ini" \
  --output "model.gcode" \
  "model.stl"
```

Command line

```
slic3r \
  --load "config.ini" \
  --output "model.gcode" \
  "model.stl"
```

Command line

```
slic3r \
    --load "config.ini" \
    --output "model.gcode" \
    "model.stl"
```

Command line

```
slic3r \
  --load "config.ini" \
  --output "model.gcode" \
  "model.stl"
```

Obtaining config

```
slic3r --save "config.ini"
vim "config.ini"
```

Obtaining config

```
slic3r --save "config.ini"
vim "config.ini"
```

Obtaining config

```
slic3r --save "config.ini"
vim "config.ini"
```

Obtaining config

```
slic3r --save "config.ini"
vim "config.ini"
```

config.ini

```
# generated by Slic3r 1.3.0 on 2021-10-25 13:04:28
adaptive_slicing = 0
adaptive_slicing_quality = 75%
avoid_crossing_perimeters = 0
bed_shape = 0x0,200x0,200x200,0x200
bed_temperature = 0
before_layer_gcode =
between_objects_gcode =
bottom_infill_pattern = rectilinear
bottom_solid_layers = 3
bridge_acceleration = 0
bridge_fan_speed = 100
bridge_flow_ratio = 1
bridge_speed = 60
brim_connections_width = 0
brim_width = 0
complete_objects = 0
cooling = 1
default_acceleration = 0
disable_fan_first_layers = 3
dont_support_bridges = 1
duplicate_distance = 6
end_filament_gcode = "; Filament-specific end gcode \n;END gcode for filament\n"
end_gcode = M104 S0 ;\nG28 X0  ;\nM84     ;\n
external_perimeter_extrusion_width = 0
external_perimeter_speed = 50%
external_perimeters_first = 0
extra_perimeters = 1
extruder_clearance_height = 20
extruder_clearance_radius = 20
extruder_offset = 0x0
extrusion_axis = E
extrusion_multiplier = 1
extrusion_width = 0
fan_always_on = 0
fan_below_layer_time = 60
filament_colour = #FFFFFF
filament_cost = 0
filament_density = 0
filament_diameter = 3
filament_max_volumetric_speed = 0
filament_notes = ""
fill_angle = 45
fill_density = 20%
fill_gaps = 1
fill_pattern = stars
first_layer_acceleration = 0
first_layer_bed_temperature = 0
first_layer_extrusion_width = 200%
first_layer_height = 0.35
first_layer_speed = 30
first_layer_temperature = 200
gap_fill_speed = 20
gcode_arcs = 0
gcode_comments = 0
gcode_flavor = reprap
has_heatbed = 1
host_type = octoprint
infill_acceleration = 0
infill_every_layers = 1
infill_extruder = 1
infill_extrusion_width = 0
infill_first = 0
infill_only_where_needed = 0
infill_overlap = 55%
infill_speed = 80
interface_shells = 0
interior_brim_width = 0
layer_gcode =
layer_height = 0.3
match_horizontal_surfaces = 0
max_fan_speed = 100
max_layer_height = 0.3
max_print_speed = 80
max_volumetric_speed = 0
min_fan_speed = 35
min_layer_height = 0.15
min_print_speed = 10
min_skirt_length = 0
notes =
nozzle_diameter = 0.5
octoprint_apikey =
only_retract_when_crossing_perimeters = 1
ooze_prevention = 0
output_filename_format = [input_filename_base].gcode
overhangs = 1
perimeter_acceleration = 0
perimeter_extruder = 1
perimeter_extrusion_width = 0
perimeter_speed = 60
perimeters = 3
post_process =
pressure_advance = 0
print_host =
printer_notes =
raft_layers = 0
regions_overlap = 0
resolution = 0
retract_before_travel = 2
retract_layer_change = 0
retract_length = 2
retract_length_toolchange = 10
retract_lift = 0
retract_lift_above = 0
retract_lift_below = 0
retract_restart_extra = 0
retract_restart_extra_toolchange = 0
retract_speed = 40
seam_position = aligned
sequential_print_priority = 0
serial_port =
serial_speed = 250000
skirt_distance = 6
skirt_height = 1
skirts = 1
slowdown_below_layer_time = 5
small_perimeter_speed = 15
solid_infill_below_area = 70
solid_infill_every_layers = 0
solid_infill_extruder = 1
solid_infill_extrusion_width = 0
solid_infill_speed = 20
spiral_vase = 0
standby_temperature_delta = -5
start_filament_gcode = "; Filament gcode\n"
start_gcode = G28 ; home all axes\nG1 Z5 F5000 ; lift nozzle\n
support_material = 0
support_material_angle = 0
support_material_buildplate_only = 0
support_material_contact_distance = 0.2
support_material_enforce_layers = 0
support_material_extruder = 1
support_material_extrusion_width = 0
support_material_interface_extruder = 1
support_material_interface_extrusion_width = 0
support_material_interface_layers = 3
support_material_interface_spacing = 0
support_material_interface_speed = 100%
support_material_max_layers = 0
support_material_pattern = pillars
support_material_spacing = 2.5
support_material_speed = 60
support_material_threshold = 60%
temperature = 200
thin_walls = 1
threads = 8
toolchange_gcode =
top_infill_extrusion_width = 0
top_infill_pattern = rectilinear
top_solid_infill_speed = 15
top_solid_layers = 3
travel_speed = 130
use_firmware_retraction = 0
use_relative_e_distances = 0
use_set_and_wait_bed = 0
use_set_and_wait_extruder = 0
use_volumetric_e = 0
vibration_limit = 0
wipe = 0
xy_size_compensation = 0
z_offset = 0
z_steps_per_mm = 0
```

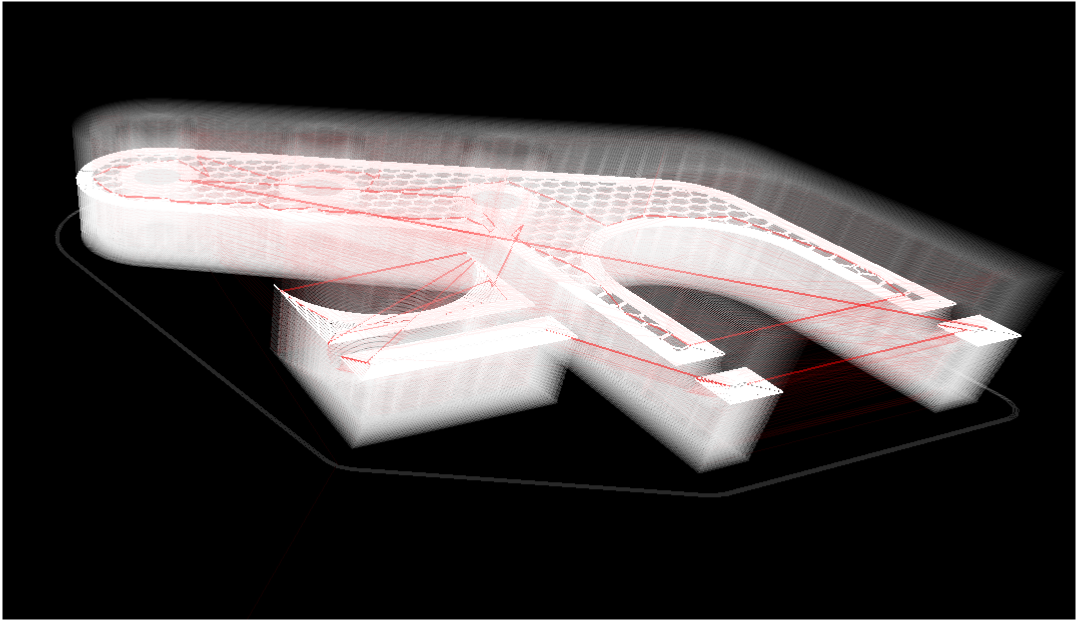
https://miro.medium.com/max/1024/1*uiDzJzN3Oo8lHTh_RfCMXQ.jpeg

Example G-code

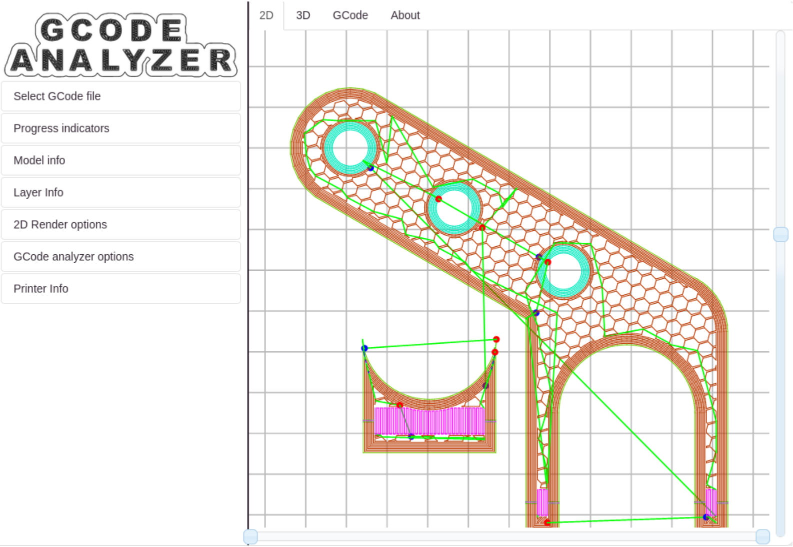
```
M190 S70 ; bed temp.
M104 S210 ; nozzle temp.
G28 ; home all axes
G1 Z5 F5000 ; lift nozzle
; Filament gcode
M109 S210 ; nozzle temp.
G21 ; set units to mm
G90 ; use abs. coordinates
M82 ; use abs. extrusion
G92 E0
G1 E-3.00000 F6000.00000
G92 E0
G1 Z0.200 F3000.000
G1 X177.664 Y152.561 F3000.000
G1 E3.10000 F6000.00000
G1 F1800
G1 X178.919 Y151.238 E3.12621
G1 X180.062 Y150.163 E3.14876
```

```
G1 X185.141 Y146.506 E3.23882
G1 X186.292 Y145.881 E3.25763
G1 X187.704 Y145.200 E3.28016
G1 X189.149 Y144.594 E3.30268
G1 X190.869 Y143.985 E3.32889
G1 X192.373 Y143.547 E3.35139
G1 X193.648 Y143.240 E3.37024
G1 X195.189 Y142.948 E3.39276
G1 X196.995 Y142.710 E3.41894
G1 X198.562 Y142.594 E3.44151
G1 X200.131 Y142.559 E3.46407
G1 X201.696 Y142.607 E3.48656
G1 X203.001 Y142.710 E3.50537
G1 X204.809 Y142.948 E3.53157
G1 X206.352 Y143.240 E3.55413
G1 X207.873 Y143.613 E3.57663
G1 X209.129 Y143.985 E3.59545
G1 X210.609 Y144.502 E3.61797
```

YAGV: Yet Another G-code Viewer



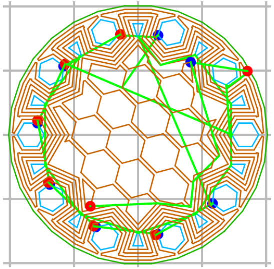
Online tools



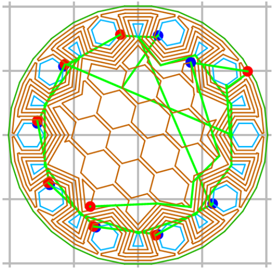
https://gcode.ws/

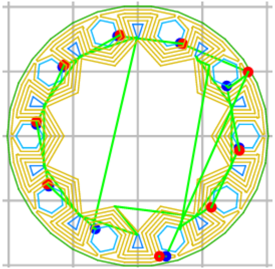
Sharp eye time!

Sharp eye time!



Sharp eye time!





https://external-preview.redd.it/vMG3DNe--UssYbp5FOSO9CUADsLVfqPxBiYulgN7kjc.png?auto=webp&s=dd126fd25766bc950fc394dcc3b66345a2f2ddfe

Time for. . .

1. 3D printing 101
2. Modeling in OpenSCAD
3. Slicing
4. ETA
5. Pipeline
6. Ending

Note from the author

(FDM)

3D printing is

SLOW

How long?

https://i.ytimg.com/vi/h0o51Anhz78/maxresdefault.jpg

gcoder

gcoder

```
cd /usr/lib/python3/dist-packages/printrun
python3 gcoder.py "/path/to/model/stuff.gcode"
```

gcoder

```
cd /usr/lib/python3/dist-packages/printrun
python3 gcoder.py "/path/to/model/stuff.gcode"
```

```
Line object size: 176
Light line object size: 48
Dimensions:
X: 141.28 - 258.71 (117.43)
Y: 140.22 - 259.76 (119.54)
Z: 0.00 - 20.00 (20.00)
Filament used: 25225.73mm
E0 25225.73mm
Number of layers: 100
Estimated duration: 5:52:38
```

https://bleedingcool.com/wp-content/uploads/2020/08/Here-Be-Dragons-Main-Art-1200x628.jpg

Time for. . .

1. 3D printing 101
2. Modeling in OpenSCAD
3. Slicing
4. ETA
5. Pipeline
6. Ending

http://hitechbusinessnetworks.com/wp-content/uploads/2019/04/Car-Factory-Automation.jpg

Repeatability

Repeatability

Reproducible builds

Reproducible builds, also known as deterministic compilation, is a process of compiling software which ensures the resulting binary code can be reproduced.
Source code compiled using deterministic compilation will always output the same binary.

https://en.wikipedia.org/wiki/Reproducible_builds

https://media.smallbiztrends.com/2016/02/blocks.jpg

GNU/Make it happen!

GNU/Make it happen!

```
SCADS :=$(wildcard *.scad)
GCODES:=$(SCADS:.scad=.gcode)
all: $(GCODES)
%.stl: %.scad
        openscad -o "$@" -d "$<.d" "$<"
%.gcode: %.stl
        slic3r --load "config.ini" -o "$@" "$<"
time: $(GCODES)
        for f in $(GCODES) ; do gcoder "$$f" ; done
clean:
        rm -fv *.d *.stl *.gcode
-include $(SCADS:.scad=.scad.d)
```

GNU/Make it happen!

```
SCADS :=$(wildcard *.scad)
GCODES:=$(SCADS:.scad=.gcode)
all: $(GCODES)
%.stl: %.scad
        openscad -o "$@" -d "$<.d" "$<"
%.gcode: %.stl
        slic3r --load "config.ini" -o "$@" "$<"
time: $(GCODES)
        for f in $(GCODES) ; do gcoder "$$f" ; done
clean:
        rm -fv *.d *.stl *.gcode
-include $(SCADS:.scad=.scad.d)
```

GNU/Make it happen!

```makefile
SCADS :=$(wildcard *.scad)
GCODES:=$(SCADS:.scad=.gcode)
all: $(GCODES)
%.stl: %.scad
	openscad -o "$@" -d "$<.d" "$<"
%.gcode: %.stl
	slic3r --load "config.ini" -o "$@" "$<"
time: $(GCODES)
	for f in $(GCODES) ; do gcoder "$$f" ; done
clean:
	rm -fv *.d *.stl *.gcode
-include $(SCADS:.scad=.scad.d)
```

GNU/Make it happen!

```
SCADS :=$(wildcard *.scad)
GCODES:=$(SCADS:.scad=.gcode)
all: $(GCODES)
%.stl: %.scad
	openscad -o "$@" -d "$<.d" "$<"
%.gcode: %.stl
	slic3r --load "config.ini" -o "$@" "$<"
time: $(GCODES)
	for f in $(GCODES) ; do gcoder "$$f" ; done
clean:
	rm -fv *.d *.stl *.gcode
-include $(SCADS:.scad=.scad.d)
```

GNU/Make it happen!

```
SCADS :=$(wildcard *.scad)
GCODES:=$(SCADS:.scad=.gcode)
all: $(GCODES)
%.stl: %.scad
        openscad -o "$@" -d "$<.d" "$<"
%.gcode: %.stl
        slic3r --load "config.ini" -o "$@" "$<"
time: $(GCODES)
        for f in $(GCODES) ; do gcoder "$$f" ; done
clean:
        rm -fv *.d *.stl *.gcode
-include $(SCADS:.scad=.scad.d)
```

GNU/Make it happen!

```
SCADS :=$(wildcard *.scad)
GCODES:=$(SCADS:.scad=.gcode)
all: $(GCODES)
%.stl: %.scad
	openscad -o "$@" -d "$<.d" "$<"
%.gcode: %.stl
	slic3r --load "config.ini" -o "$@" "$<"
time: $(GCODES)
	for f in $(GCODES) ; do gcoder "$$f" ; done
clean:
	rm -fv *.d *.stl *.gcode
-include $(SCADS:.scad=.scad.d)
```

GNU/Make it happen!

```
SCADS :=$(wildcard *.scad)
GCODES:=$(SCADS:.scad=.gcode)
all: $(GCODES)
%.stl: %.scad
        openscad -o "$@" -d "$<.d" "$<"
%.gcode: %.stl
        slic3r --load "config.ini" -o "$@" "$<"
time: $(GCODES)
        for f in $(GCODES) ; do gcoder "$$f" ; done
clean:
        rm -fv *.d *.stl *.gcode
-include $(SCADS:.scad=.scad.d)
```

GNU/Make it happen!

```
SCADS :=$(wildcard *.scad)
GCODES:=$(SCADS:.scad=.gcode)
all: $(GCODES)
%.stl: %.scad
        openscad -o "$@" -d "$<.d" "$<"
%.gcode: %.stl
        slic3r --load "config.ini" -o "$@" "$<"
time: $(GCODES)
        for f in $(GCODES) ; do gcoder "$$f" ; done
clean:
        rm -fv *.d *.stl *.gcode
-include $(SCADS:.scad=.scad.d)
```

GNU/Make it happen!

```
SCADS :=$(wildcard *.scad)
GCODES:=$(SCADS:.scad=.gcode)
all: $(GCODES)
%.stl: %.scad
        openscad -o "$@" -d "$<.d" "$<"
%.gcode: %.stl
        slic3r --load "config.ini" -o "$@" "$<"
time: $(GCODES)
        for f in $(GCODES) ; do gcoder "$$f" ; done
clean:
        rm -fv *.d *.stl *.gcode
-include $(SCADS:.scad=.scad.d)
```

GNU/Make it happen!

```
SCADS :=$(wildcard *.scad)
GCODES:=$(SCADS:.scad=.gcode)
all: $(GCODES)
%.stl: %.scad
	openscad -o "$@" -d "$<.d" "$<"
%.gcode: %.stl
	slic3r --load "config.ini" -o "$@" "$<"
time: $(GCODES)
	for f in $(GCODES) ; do gcoder "$$f" ; done
clean:
	rm -fv *.d *.stl *.gcode
-include $(SCADS:.scad=.scad.d)
```

GNU/Make it happen!

```
SCADS :=$(wildcard *.scad)
GCODES:=$(SCADS:.scad=.gcode)
all: $(GCODES)
%.stl: %.scad
	openscad -o "$@" -d "$<.d" "$<"
%.gcode: %.stl
	slic3r --load "config.ini" -o "$@" "$<"
time: $(GCODES)
	for f in $(GCODES) ; do gcoder "$$f" ; done
clean:
	rm -fv *.d *.stl *.gcode
-include $(SCADS:.scad=.scad.d)
```

GNU/Make it happen!

```
SCADS :=$(wildcard *.scad)
GCODES:=$(SCADS:.scad=.gcode)
all: $(GCODES)
%.stl: %.scad
	openscad -o "$@" -d "$<.d" "$<"
%.gcode: %.stl
	slic3r --load "config.ini" -o "$@" "$<"
time: $(GCODES)
	for f in $(GCODES) ; do gcoder "$$f" ; done
clean:
	rm -fv *.d *.stl *.gcode
-include $(SCADS:.scad=.scad.d)
```

GNU/Make it happen!

```
SCADS :=$(wildcard *.scad)
GCODES:=$(SCADS:.scad=.gcode)
all: $(GCODES)
%.stl: %.scad
        openscad -o "$@" -d "$<.d" "$<"
%.gcode: %.stl
        slic3r --load "config.ini" -o "$@" "$<"
time: $(GCODES)
        for f in $(GCODES) ; do gcoder "$$f" ; done
clean:
        rm -fv *.d *.stl *.gcode
-include $(SCADS:.scad=.scad.d)
```

Usage

```
echo 'cube(10*[1,1,1]);' > foo.scad
echo 'cube(13*[1,1,1]);' > bar.scad
make # all
make time
```

Usage

```
echo 'cube(10*[1,1,1]);' > foo.scad
echo 'cube(13*[1,1,1]);' > bar.scad
make # all
make time
```

Usage

```
echo 'cube(10*[1,1,1]);' > foo.scad
echo 'cube(13*[1,1,1]);' > bar.scad
make # all
make time
```

Usage

```
echo 'cube(10*[1,1,1]);' > foo.scad
echo 'cube(13*[1,1,1]);' > bar.scad
make # all
make time
```

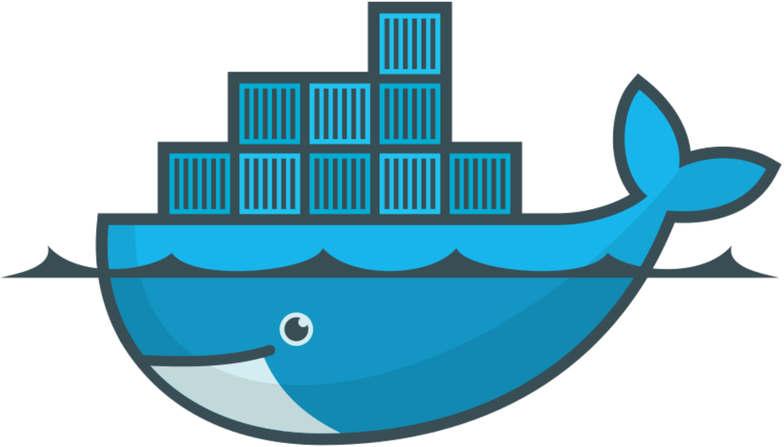
Usage

```
echo 'cube(10*[1,1,1]);' > foo.scad
echo 'cube(13*[1,1,1]);' > bar.scad
make # all
make time
```

https://amorphia-apparel.com/image/rex.1200.png

Stable SDK

Stable SDK



Dockerfile

```
FROM debian:11
RUN apt-get update && apt-get install -y \
        coreutils \
        make \
        openscad \
        printrun-common \
        slic3r \
        yagv
```

Dockerfile

```
FROM debian:11
RUN apt-get update && apt-get install -y \
    coreutils \
    make \
    openscad \
    printrun-common \
    slic3r \
    yagv
```

Dockerfile

```
FROM debian:11
RUN apt-get update && apt-get install -y \
    coreutils \
    make \
    openscad \
    printrun-common \
    slic3r \
    yagv
```

Dockerfile

```
FROM debian:11
RUN apt-get update && apt-get install -y \
    coreutils \
    make \
    openscad \
    printrun-common \
    slic3r \
    yagv
```

Dockerfile

```
FROM debian:11
RUN apt-get update && apt-get install -y \
    coreutils \
    make \
    openscad \
    printrun-common \
    slic3r \
    yagv
```

Running with SDK

```
docker build -t "sdk" .

docker run \
    --interactive \
    --tty \
    --rm \
    --user "$(id -u):$(id -g)" \
    --volume "$PWD:/mnt" \
    --workdir "/mnt" \
    "sdk"

make # {all,time,clean}
```

Running with SDK

```
docker build -t "sdk" .

docker run \
    --interactive \
    --tty \
    --rm \
    --user "$(id -u):$(id -g)" \
    --volume "$PWD:/mnt" \
    --workdir "/mnt" \
    "sdk"

make # {all,time,clean}
```

Running with SDK

```
docker build -t "sdk" .

docker run \
  --interactive \
  --tty \
  --rm \
  --user "$(id -u):$(id -g)" \
  --volume "$PWD:/mnt" \
  --workdir "/mnt" \
  "sdk"

make # {all,time,clean}
```

Running with SDK

```
docker build -t "sdk" .

docker run \
   --interactive \
   --tty \
   --rm \
   --user "$(id -u):$(id -g)" \
   --volume "$PWD:/mnt" \
   --workdir "/mnt" \
   "sdk"

make # {all,time,clean}
```

Running with SDK

```
docker build -t "sdk" .

docker run \
   --interactive \
   --tty \
   --rm \
   --user "$(id -u):$(id -g)" \
   --volume "$PWD:/mnt" \
   --workdir "/mnt" \
   "sdk"

make # {all,time,clean}
```

Running with SDK

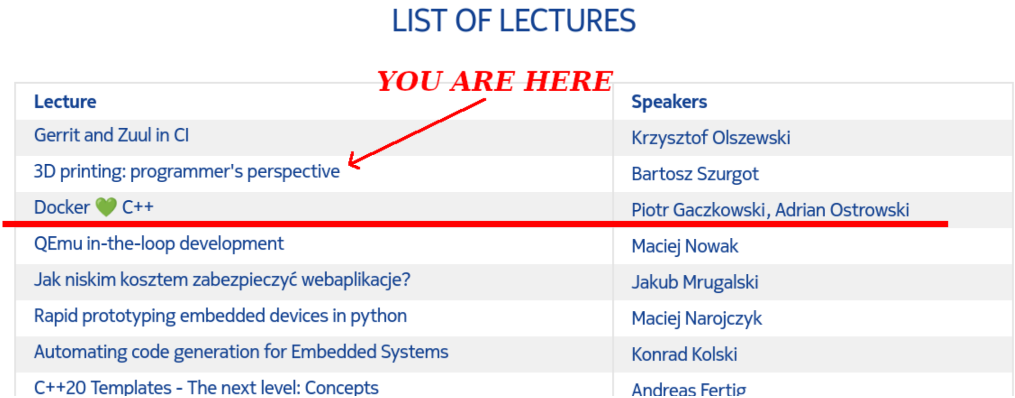
```
docker build -t "sdk" .

docker run \
    --interactive \
    --tty \
    --rm \
    --user "$(id -u):$(id -g)" \
    --volume "$PWD:/mnt" \
    --workdir "/mnt" \
    "sdk"

make # {all,time,clean}
```

Running with SDK

```
docker build -t "sdk" .

docker run \
  --interactive \
  --tty \
  --rm \
  --user "$(id -u):$(id -g)" \
  --volume "$PWD:/mnt" \
  --workdir "/mnt" \
  "sdk"

make # {all,time,clean}
```

Never used Docker?

Never used Docker?



LIST OF LECTURES

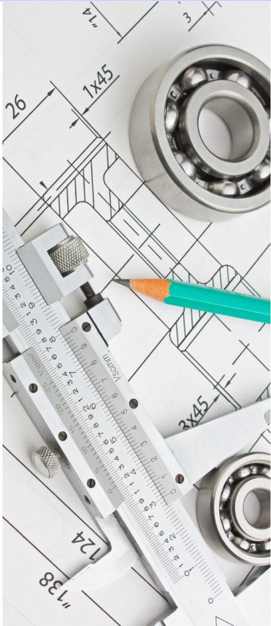
YOU ARE HERE

Lecture	Speakers
Gerrit and Zuul in CI	Krzysztof Olszewski
3D printing: programmer's perspective	Bartosz Szurgot
Docker 💚 C++	Piotr Gaczkowski, Adrian Ostrowski
QEmu in-the-loop development	Maciej Nowak
Jak niskim kosztem zabezpieczyć webaplikacje?	Jakub Mrugalski
Rapid prototyping embedded devices in python	Maciej Narojczyk
Automating code generation for Embedded Systems	Konrad Kolski
C++20 Templates - The next level: Concepts	Andreas Fertig

`https://codedive.pl/2021/docker-c`

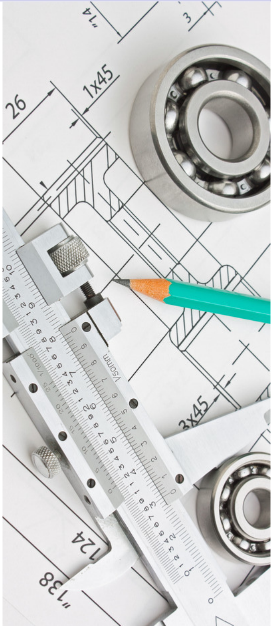
All done?

All done?



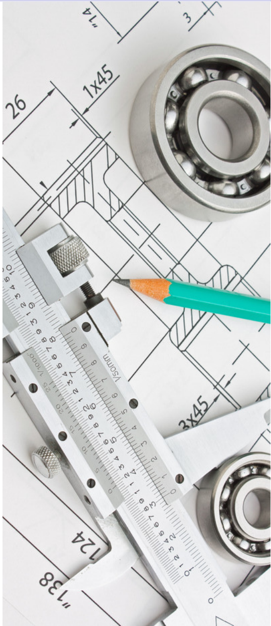
All done?

- Print materials:
  - PLA?
  - PET-G?
  - TPU?
  - ...



All done?

- Print materials:
  - PLA?
  - PET-G?
  - TPU?
  - . . .
- Print properties:
  - Draft print (fast)
  - Pretty-print
  - Print-for-strength
  - Waterproof print
  - . . .
- And more. . .



3D printing SDK project

- Dockerized SDK
- OpenSCAD support
- STL support
- Multiple filaments
- Different build modes
- Extensible interface

https://github.com/el-bart/3d_printing_sdk

Time for. . .

1. 3D printing 101
2. Modeling in OpenSCAD
3. Slicing
4. ETA
5. Pipeline
6. Ending

What we’ve learned?

- Introduction to 3D printing

What we’ve learned?

- Introduction to 3D printing
- OpenSCAD basics

What we’ve learned?

- Introduction to 3D printing
- OpenSCAD basics
- Slicing concept

What we've learned?

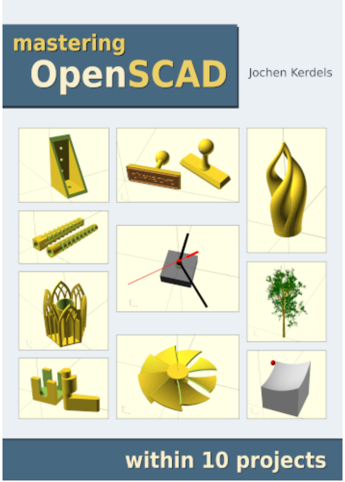
- Introduction to 3D printing
- OpenSCAD basics
- Slicing concept
- Estimating print time

What we’ve learned?

- Introduction to 3D printing
- OpenSCAD basics
- Slicing concept
- Estimating print time
- Automating whole pipeline

https://upload.wikimedia.org/wikipedia/commons/7/7f/3DBenchy.png

Next step



mastering OpenSCAD

Jochen Kerdels

within 10 projects

https://mastering-openscad.eu

Knowledge bootstrap

- https://youtube.com/c/CNCKitchen
- https://help.prusa3d.com/en/
- https://3dprinting.stackexchange.com
- https://thingiverse.com
- https://baszerr.eu/doku.php?id=docs:3d_printing_101:3d_printing_101
- https://youtube.com/c/TheMeltzonePodcast/featured
- https://www.youtube.com/watch?v=fzBRYsiyxjI
- https://gcode.ws
- https://mastering-openscad.eu
- https://en.wikibooks.org/wiki/OpenSCAD_User_Manual/The_OpenSCAD_Language
- https://openscad.org/cheatsheet/index.html
- https://manual.slic3r.org
- https://reprap.org/wiki/G-code
- https://openscad.org/libraries.html
- . . . or just use your favourite search engine :D

https://3dprint.com/wp-content/uploads/2017/11/33630354656_e8ba8b70f7_o-300x300.jpg *https://images-cdn.9gag.com/photo/118_700b.jpg*

https://images-cdn.9gag.com/photo/118_700b.jpg