

openscad-test



Table of Contents

- 1. Document information 1
- 2. Objects 1
 - 2.1. Object - Library-container 1
 - 2.2. Object - case 6
 - 2.3. Object - fastener 9
 - 2.4. Object - geoTest 12
 - 2.5. Object - ikeabung 13
 - 2.6. Object - test 14
 - 2.7. Object - test2 15
- 3. To do 16

1. Document information

Links to Document



Document online



Document Source



Document PDF

test repo for building openscad files into different outputs

This is still work in progress but can already build a png and stl of each scad file in the opescad directory.

See the online or pdf versions for the images as the readme is really only the source and right now is not WYSIWYG!

2. Objects

2.1. Object - Library-container



Figure 1. image

Listing 1. Openscad source

```

module nibLeft(x,y,z,nibR) {
    translate([x,y/10,0]) cylinder(h=z,r=nibR);
    translate([x,9*(y/10),0]) cylinder(h=z,r=nibR);
}
module nibRight(x,y,z,nibR) {
    translate([0,y/10,0]) cylinder(h=z,r=nibR);
    translate([0,9*(y/10),0]) cylinder(h=z,r=nibR);
}
module nibBottom(x,y,z,nibR) {
    translate([x/10,0,0]) cylinder(h=z,r=nibR);
    translate([9*(x/10),0,0]) cylinder(h=z,r=nibR);
}
module nibTop(x,y,z,nibR) {
    translate([x/10,y,0]) cylinder(h=z,r=nibR);
    translate([9*(x/10),y,0]) cylinder(h=z,r=nibR);
}
module containerOpenLid(x,y,z,rimThick,bottomThick,nibYN,nibR) {
    rimR=rimThick/2;

```

```
//all 8 corners defined first
//corners should be AROUND the contained cube defined by x y z
corner000=[0,0,0];
corner0=[-rimR,-rimR,-(rimR+bottomThick)];
corner0x=[x+rimR,-rimR,-(rimR+bottomThick)];
corner0y=[-rimR,y+rimR,-(rimR+bottomThick)];
corner0xy=[x+rimR,y+rimR,-(rimR+bottomThick)];
corner0z=[-rimR,-rimR,z];
corner0xz=[x+rimR,-rimR,z];
corner0yz=[-rimR,y+rimR,z];
corner0xyz=[x+rimR,y+rimR,z]; //draw the debug contents
translate([0,0,-bottomThick]) cube([x,y,bottomThick]);
module corner0() {
    translate(corner0) sphere(r=rimR);
}
module corner0x() {
    translate(corner0x) sphere(r=rimR);
}
module corner0y() {
    translate(corner0y) sphere(r=rimR);
}
module corner0xy() {
    translate(corner0xy) sphere(r=rimR);
}
module corner0z() {
    translate(corner0z) sphere(r=rimR);
}
module corner0xz() {
    translate(corner0xz) sphere(r=rimR);
}
module corner0yz() {
    translate(corner0yz) sphere(r=rimR);
}
module corner0xyz() {
    translate(corner0xyz) sphere(r=rimR);
}
if (nibYN=="nibY") {
    nibBottom(x,y,z,nibR);
    nibLeft(x,y,z,nibR);
    nibRight(x,y,z,nibR);
    nibTop(x,y,z,nibR);
}
union(){
    //floor
    hull(){
        corner0();
        corner0x();
        corner0y();
        corner0xy();
    }
}
```

```

    //left
    hull(){
        corner0();
        corner0z();
        corner0y();
        corner0yz();
    }
    //right
    hull(){
        corner0x();
        corner0xy();
        corner0xyz();
        corner0xz();
    }
    //top
    hull(){
        corner0y();
        corner0yz();
        corner0xyz();
        corner0xy();
    }
    //bottom
    hull(){
        corner0();
        corner0z();
        corner0x();
        corner0xz();
    }
}

// example
//caseRim=1.5;
//$fn=100;
//odH=10;
//odW=156;
//odD=73;
//odJSH=6;
//#containerOpenLid(odW,odD,odH,caseRim,odJSH-caseRim,"nibY",.6);
//cube([odW,odD,odH]);

module containerVertSlot(x,y,z,rimThick,bottomThick,nibYN,nibR) {
    rimR=rimThick/2;
    //all 8 corners defined first
    //corners should be AROUND the contained cube defined by x y z
    corner000=[0,0,0];
    corner0=[-rimR,-rimR,-(rimR+bottomThick)];
    corner0x=[x+rimR,-rimR,-(rimR+bottomThick)];
    corner0y=[-rimR,y+rimR,-(rimR+bottomThick)];
    corner0xy=[x+rimR,y+rimR,-(rimR+bottomThick)];

```

```

corner0z=[-rimR,-rimR,z];
corner0xz=[x+rimR,-rimR,z];
corner0yz=[-rimR,y+rimR,z];
corner0xyz=[x+rimR,y+rimR,z];    //draw the debug contents
translate([0,0,-bottomThick]) cube([x,y,bottomThick]);
module corner0() {
    translate(corner0) sphere(r=rimR);
}
module corner0x() {
    translate(corner0x) sphere(r=rimR);
}
module corner0y() {
    translate(corner0y) sphere(r=rimR);
}
module corner0xy() {
    translate(corner0xy) sphere(r=rimR);
}
module corner0z() {
    translate(corner0z) sphere(r=rimR);
}
module corner0xz() {
    translate(corner0xz) sphere(r=rimR);
}
module corner0yz() {
    translate(corner0yz) sphere(r=rimR);
}
module corner0xyz() {
    translate(corner0xyz) sphere(r=rimR);
}
if (nibYN=="nibY") {
    nibLeft(x,y,z,nibR);
    nibRight(x,y,z,nibR);
}
union(){
    //floor
    hull(){
        corner0();
        corner0x();
        corner0y();
        corner0xy();
    }
    //left
    hull(){
        corner0();
        corner0z();
        corner0y();
        corner0yz();
    }
    //right
    hull(){

```

```
        corner0x();
        corner0xy();
        corner0xyz();
        corner0xz();
    }
}
}
// example
//caseRim=1.5;
//$fn=100;
//odH=10;
//odW=15;
//odD=73;
//odJSH=6;
//containerVertSlot(odW,odD,odH,caseRim,odJSH-caseRim,"nibY",.6);
//cube([odW,odD,odH]);

if (library) {} else {
    echo("trying to compile a library!");
    linear_extrude(height = 4) {
        text("trying to compile a library!");
    }
}
```

2.2. Object - case

A case for an odroid handgeld console and accessories.

The edges are rounded and there are cutouts for the parts that protrude from the console.

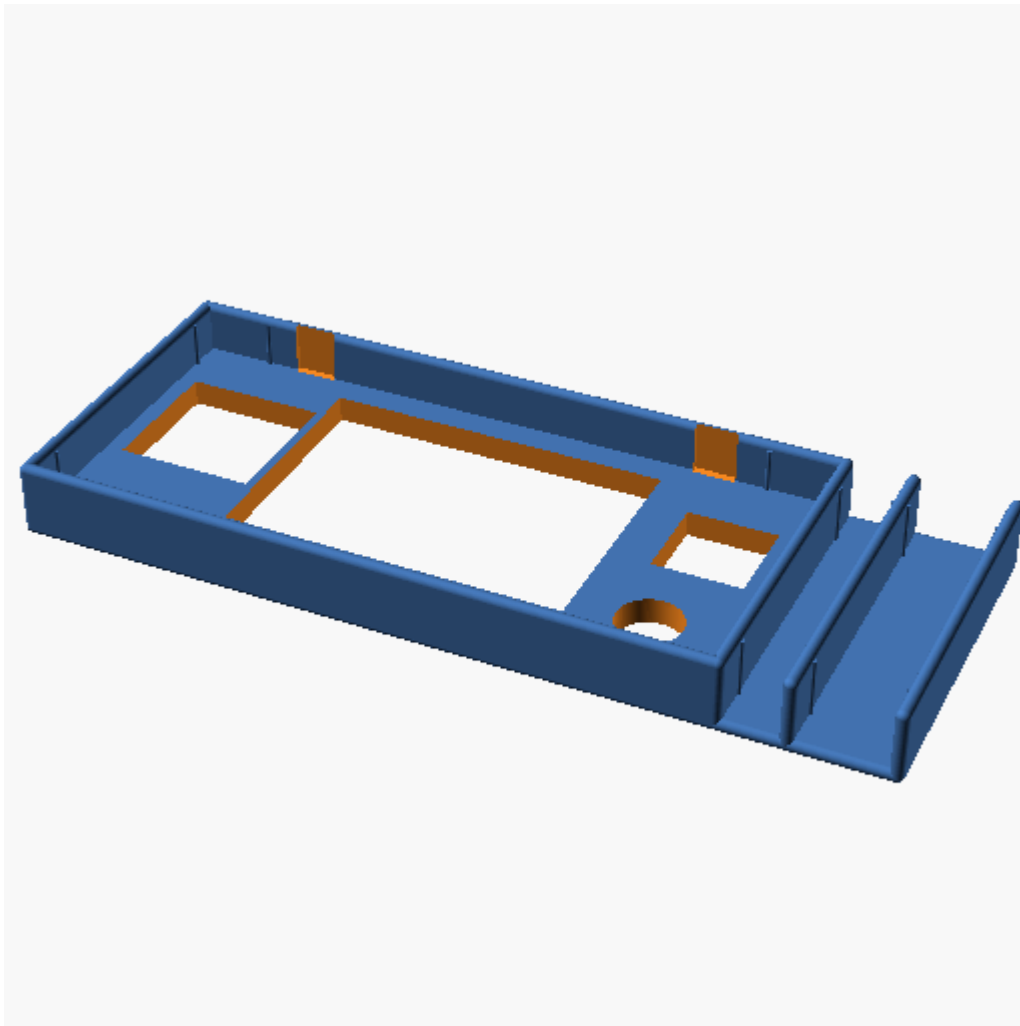


Figure 2. image

Listing 2. Openscad source

```
include <Library-container.scad>
caseRim=3;
holdersR=.7;
$fn=100;
kbD=82;
kbW=210;
kbH=7;
library=true;

odH=10;
odW=156;
odD=73;

odJSW=28-13;
odJSR=odJSW/2;
odJSoffX=13;
odJSoffY=11;
```

```

odJSH=6;
odBRW=118-38;
odBRD=12-5;
odBROffX=38;
odBROffY=5;
odCRW=28-7;
odCRD=58-37;
odCROffX=7;
odCROffY=37;
odTBW=152-120;
odTBD=61-27;
odTBOffX=120;
odTBOffY=27;
odTLOffX=23;
odTLOffY=odD;
odTLW=33-23;
odTLD=2;
odTH=20;
odTROffX=123;
odTROffY=odD;
odTRW=odTLW;
odTRD=odTLD;
odDPW=odBRW;
odDPD=67-14;
odDPOffY=14;
odDPOffX=odBROffX;

//the odroid travel case with cutouts for buttons etc
difference(){
    //the container itself
    translate([caseRim/2,caseRim/2,odJSH])
    containerOpenLid(odW,odD,odH,caseRim,odJSH-caseRim,"nibY",.6);
    offset=.01;
    //the cutouts
    translate([1.5,.75,-offset/2]) union() {
        translate([odW-odJSR*2-odJSoffX,odJSoffY,0]+[odJSR,odJSR,0])
    cylinder(h=odJSH+offset,r=odJSR);
        translate([odW-odBRW-odBROffX,odBROffY,0])
    cube([odBRW,odBRD,odJSH+offset]);
        translate([odW-odCRW-odCROffX,odCROffY,0])
    cube([odCRW,odCRW,odJSH+offset]);
        translate([odW-odTBW-odTBOffX,odTBOffY,0])
    cube([odTBW,odTBW,odJSH+offset]);
        translate([odW-odTLW-odTLOffX,odTLOffY,odJSH-.1])
    cube([odTLW,odTLD,odTH+offset]);
        translate([odW-odTRW-odTROffX,odTROffY,odJSH-.1])
    cube([odTRW,odTRD,odTH+offset]);
        translate([odW-odDPW-odDPOffX,odDPOffY,0])
    cube([odDPW,odDPD,odJSH+offset]);
    }
}

```

```
}  
//add on some slots for peripherals  
floorDepth=0;  
//microuter slot  
translate([caseRim/2+caseRim+odW,caseRim/2,floorDepth])  
    containerVertSlot(12,odD,odH+odJSH,caseRim,floorDepth-caseRim,"nibY",.6);  
//micro USB 3 Port Hub  
translate([caseRim/2+2*caseRim+odW+12,caseRim/2,floorDepth])  
    containerVertSlot(19.5,odD,odH+odJSH,caseRim,floorDepth-caseRim,"nibY",.6);
```

2.3. Object - fastener

This is a fastener for a writing Desk.

The idea is to add a magnet to hold it up and to print it so that it does not require a bearing.

- V1 is the first prototype for a first print test and fitting test
 - fits well and axle didn't print free so need update
- V2 added a better axle but didn't get printed
- V3 added a better cutout and is printed
 - The cutout is currently a dummy pending getting the axle to work to try it out with magnets taped into place
 - axle prints freely so moving on to screw holes, magnets, and covers
- V4 Added final OCD logo and screw caps etc.
 - Mounted and working.



Figure 3. image

Listing 3. Openscad source

```
$fn=100;
mainLength=50;
mainD=15;
mainH=10;
axleD=10;
axleDout=axleD+3;
ringH=2;

magnetX=17;
magnetY=5;
magnetZ=2;

module axle(xx1X,xx1Y) {
    translate([0,0,-xx1Y/2])cylinder(h=mainH+xx1Y,d=axleD+xx1X);
    translate([0,0,((mainH-ringH)/2)]) cylinder(h=ringH,d=axleDout+xx1X);
    translate([0,0,(mainH/2)-((axleDout-axleD)/2+ringH/2)])
cylinder(h=(axleDout-axleD)/2,d1=axleD+xx1X,d2=axleDout+xx1X);
    translate([0,0,(mainH/2)+(ringH/2)]) cylinder(h=((axleDout-
```

```

axleD)/2),d2=axleD+xxlX,d1=axleDout+xxlX);
}
module clip() {
  difference() {
    union(){
      hull(){
        cylinder(d=mainD,h=mainH);
        translate([mainLength,0,0]) cylinder(d=mainD,h=mainH);
      }
      translate([7,-3.5,mainH]) linear_extrude (height=1.5) {
        text("OCD",size=8);
      }
    }
  }
  //magnet
  translate([mainLength-magnetX,-magnetZ/2,(mainH-magnetY)/2+1])
cube([magnetX,magnetZ,magnetY+10]);
  //holder
  holderW=19;
  holderRin=33;
  holderRout=holderRin+holderW;
  difference(){
    translate([0,0,-.1]) cylinder(h=3+.1,r=holderRout);
    translate([0,0,-.11]) cylinder(h=3+.22,r=holderRin);
  }
}
}
module magnetCap(){
  //magnet cap
  difference(){
    cylinder(h=2.8,d=11);
    translate([0,0,-.1]) cylinder(h=2,d=10);
  }
}
module screwCap() {
  //screwcap axle
  cylinder(h=2,d=7.5);
  translate([0,0,2]) cylinder(h=1,d=axleD);
}

//add the clip
difference () {
  clip();
  axle(1,1);
}

//add the axle and drill a hole in it for a srew
difference(){
  axle(0,0);
  translate([0,0,-.05]) cylinder(h=mainH+.1,d=4);
  translate([0,0,mainH/2]) cylinder(h=(mainH/2)+.1,d=7.5);
}

```

```
//next two lines just a visual
//#translate([0,0,mainH+2]) screwCap();
//#translate([42,0,-.5]) magnetCap();
}
translate([0,-27,3]) rotate([0,180,0]) screwCap();
translate([0,-15,3]) rotate([0,180,0]) magnetCap();
```

2.4. Object - geoTest

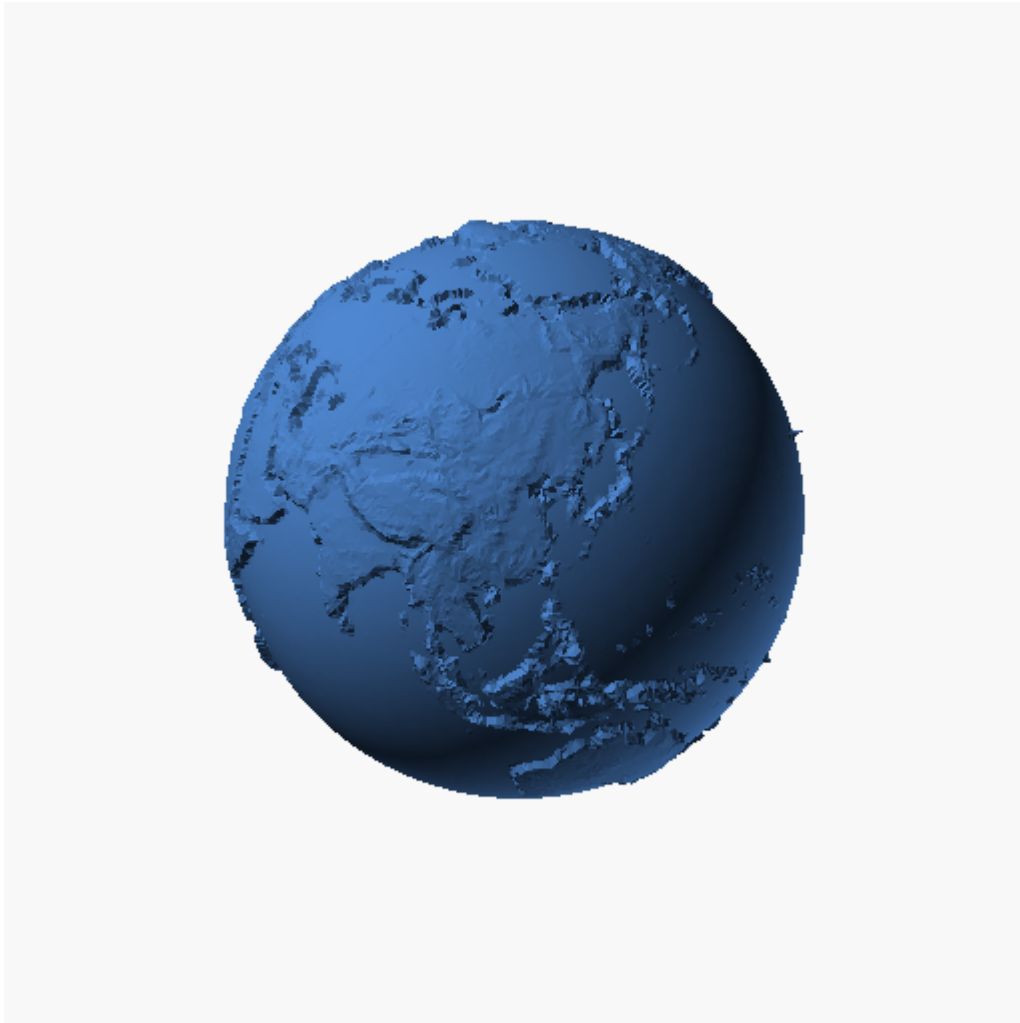


Figure 4. image

Listing 4. Openscad source

```
// Geody Planet 1 - SCAD
// Geody - https://www.geody.com/
// OpenSCAD - http://www.openscad.org/

wwrad=40; // Radius of the Planet
wrad=wwrad/20; // Radius of the Spot
wradp=wwrad-wrad/2; // Distance of the Spot from the center of the Planet
wres=50; // Resolution of the Spot
```

```
latx=48.782345; lonx=9.180819;  
  
rotate(a=[0,0,270]) { import("geody_earthmap.stl", convexity=4); } // download  
from https://www.geody.com/geody_earthmap.stl  
// sphere(r=wwrad, $fn=wres); // Test Planet  
  
translate([(-wradp)*cos(latx)*cos(lonx),(-  
wradp)*cos(latx)*sin(lonx),wradp*sin(latx)]) {sphere(r=wrad, $fn=wres,  
center=true);};
```

2.5. Object - ikeaabung

This was a replacement foot for an IKEA shelf.

The actual foot was screwed in with a bolt on the underside.

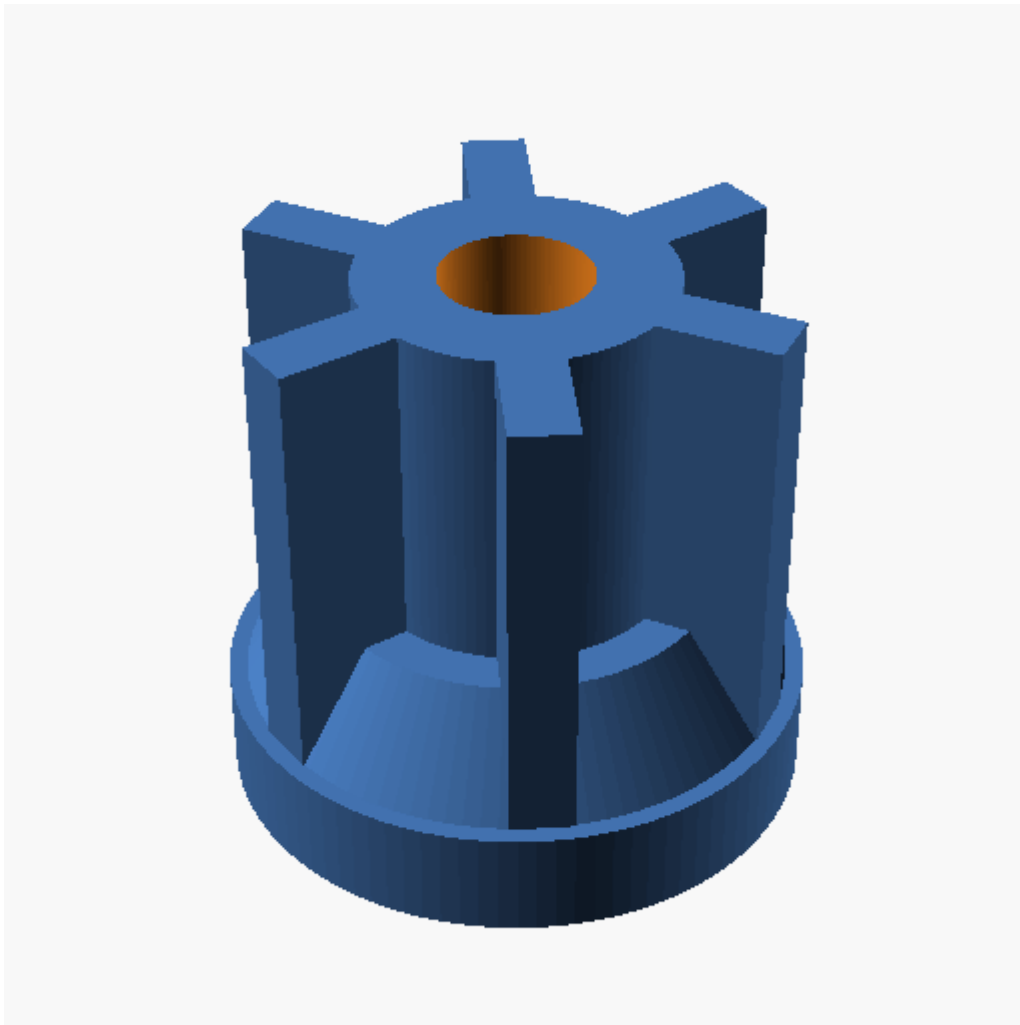


Figure 5. image

Listing 5. Openscad source

```
$fn=100;
totH=30;
baseH=6;
baseW=32;
wingW=3.5;
wingD=8;
centreD=17;

for (i = [0:360/6:360]) {
  rotate([0,0,i]) translate([(baseW-2)/2-wingD,-wingW/2,baseH])
  cube([wingD,wingW,totH-baseH]);
}

difference(){
  union(){
    cylinder(h=totH,d=centreD);
    cylinder(h=6,d=32);
    translate([0,0,baseH]) cylinder(h=6,d1=baseW-2,d2=22);
  }
  translate([0,0,-.1]) cylinder(h=totH+.2,d=8.2);

  translate([0,0,-.1]) cylinder(h=8.1,d=15,$fn=6);
}
```

2.6. Object - test

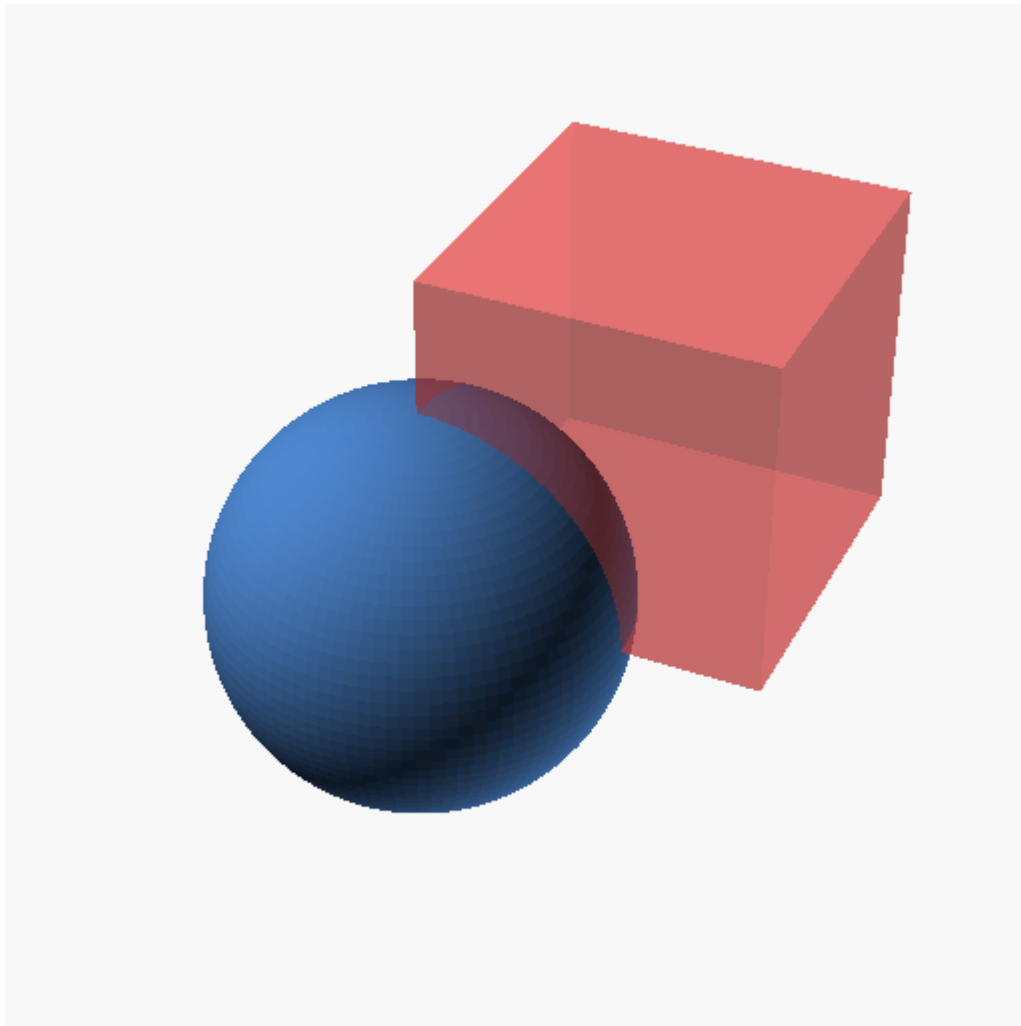


Figure 6. image

Listing 6. Openscad source

```
$fn=100;  
#cube([10,10,10]);  
sphere(d=12);
```

2.7. Object - test2

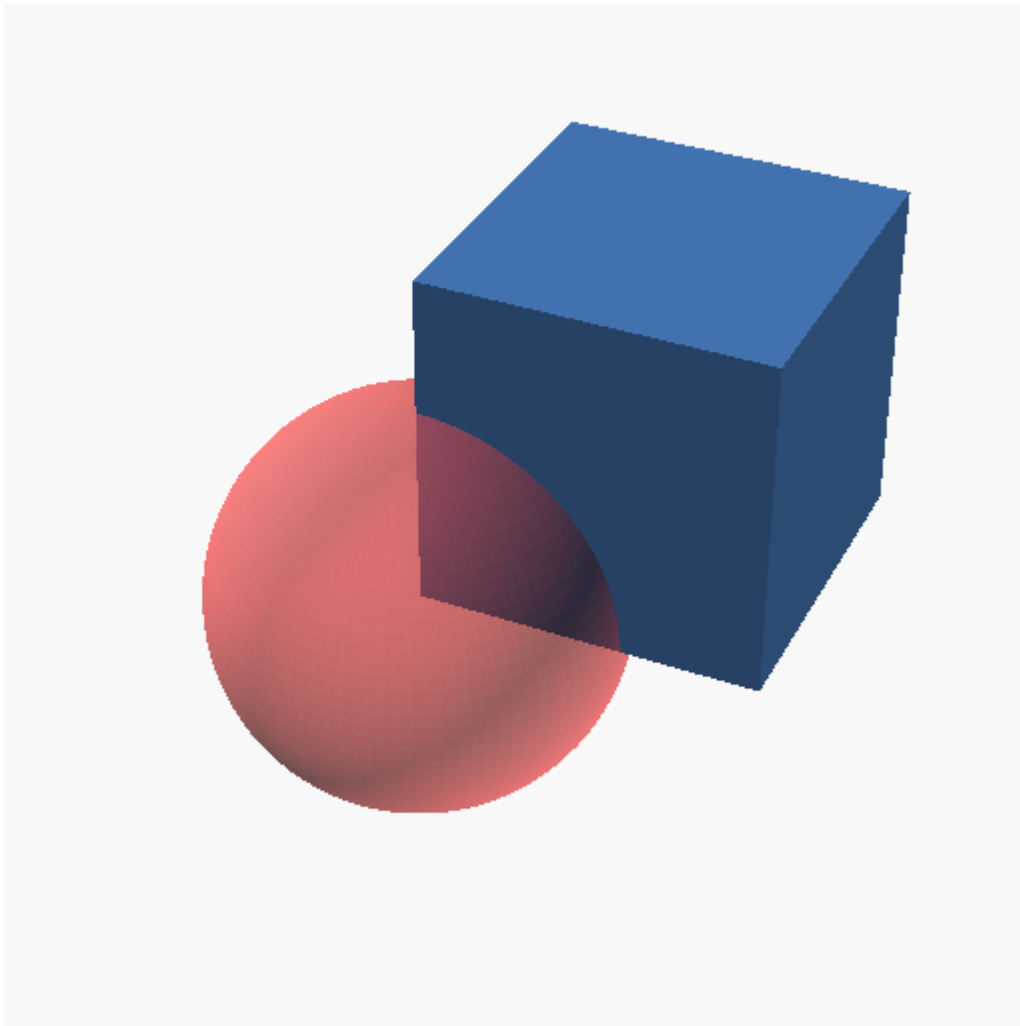


Figure 7. image

Listing 7. Openscad source

```
$fn=100;  
cube([10,10,10]);  
#sphere(d=12);
```

3. To do

Right now the github source is not perfect as the readme does not display the images when viewed in github.

Need to add further process steps for the images like meshlabserver to do further processing:

- glass rendering
- cleaning up the mesh
- Simplifying the mesh

- Stats

Need to add animation options.

- ☒ Need to add text display option for each item.

Need to add view parameters as options.