

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 - ikeabung 9
 - 2.4. Object - test 10
 - 2.5. Object - test2 11

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 stl file in the opescad directory.

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

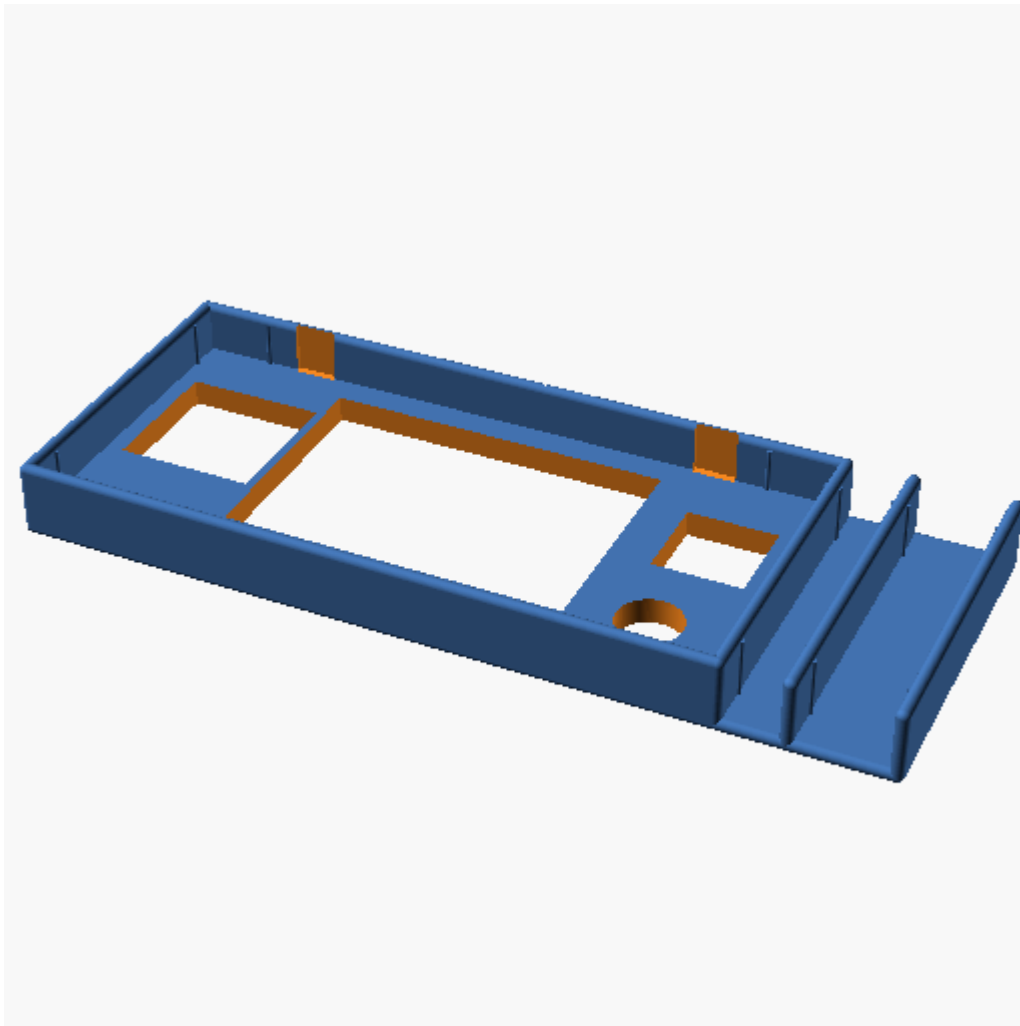


Figure 2. image

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 - ikeabung

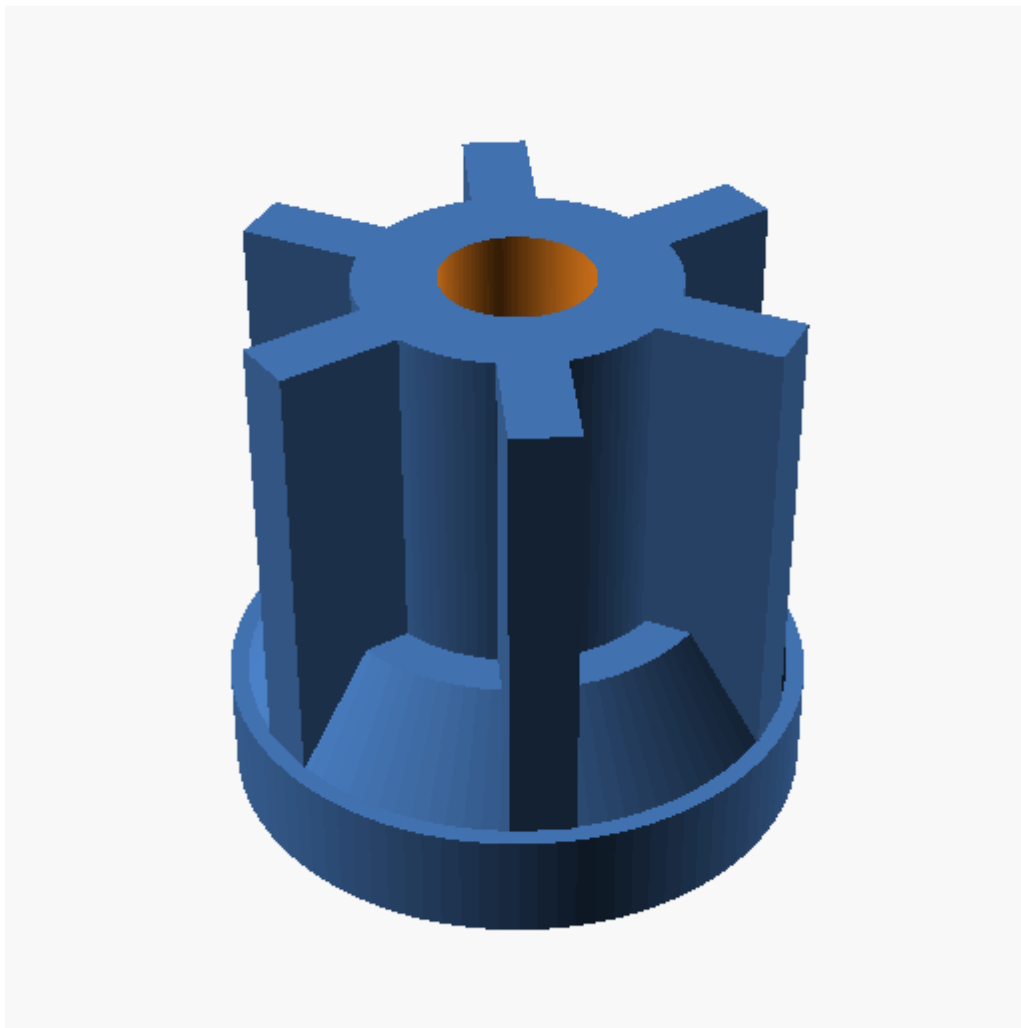


Figure 3. image

Listing 3. 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);
}
```

2.4. Object - test

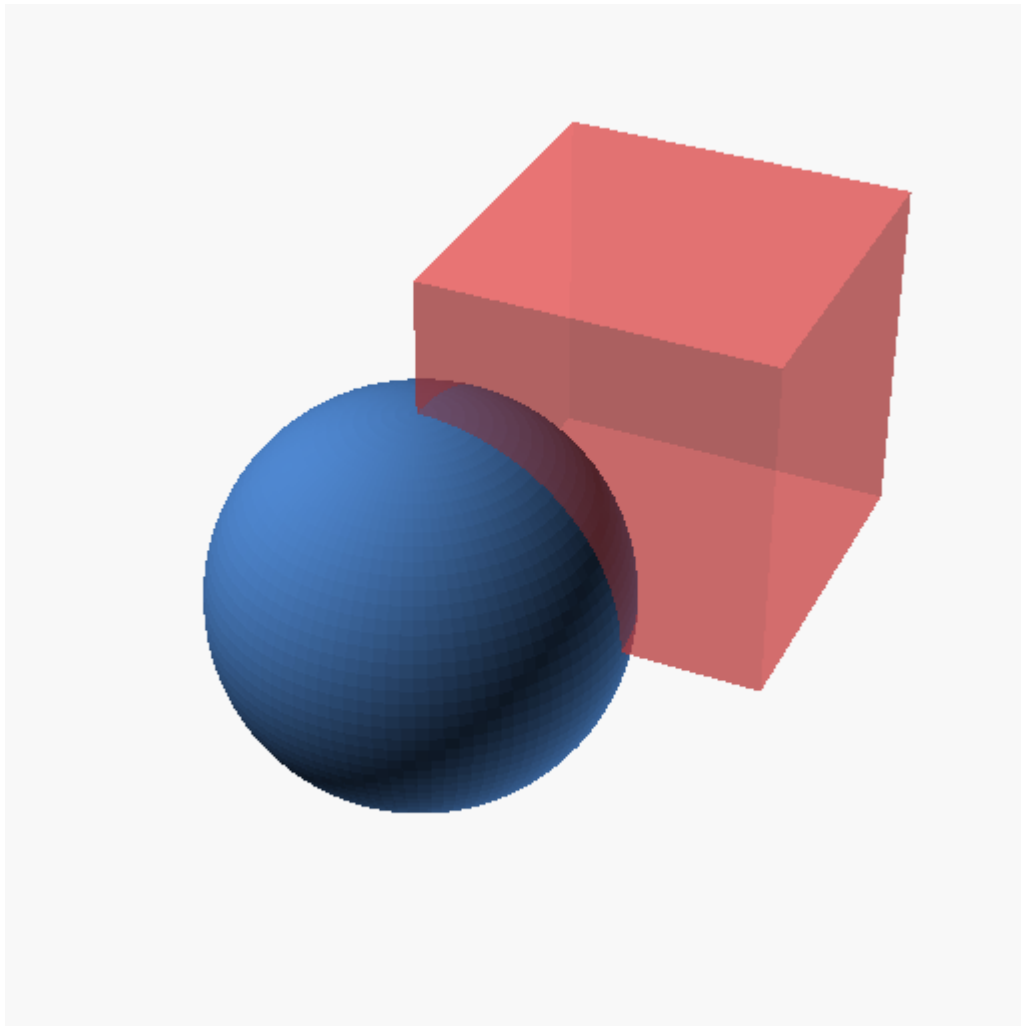


Figure 4. image

Listing 4. Openscad source

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

2.5. Object - test2

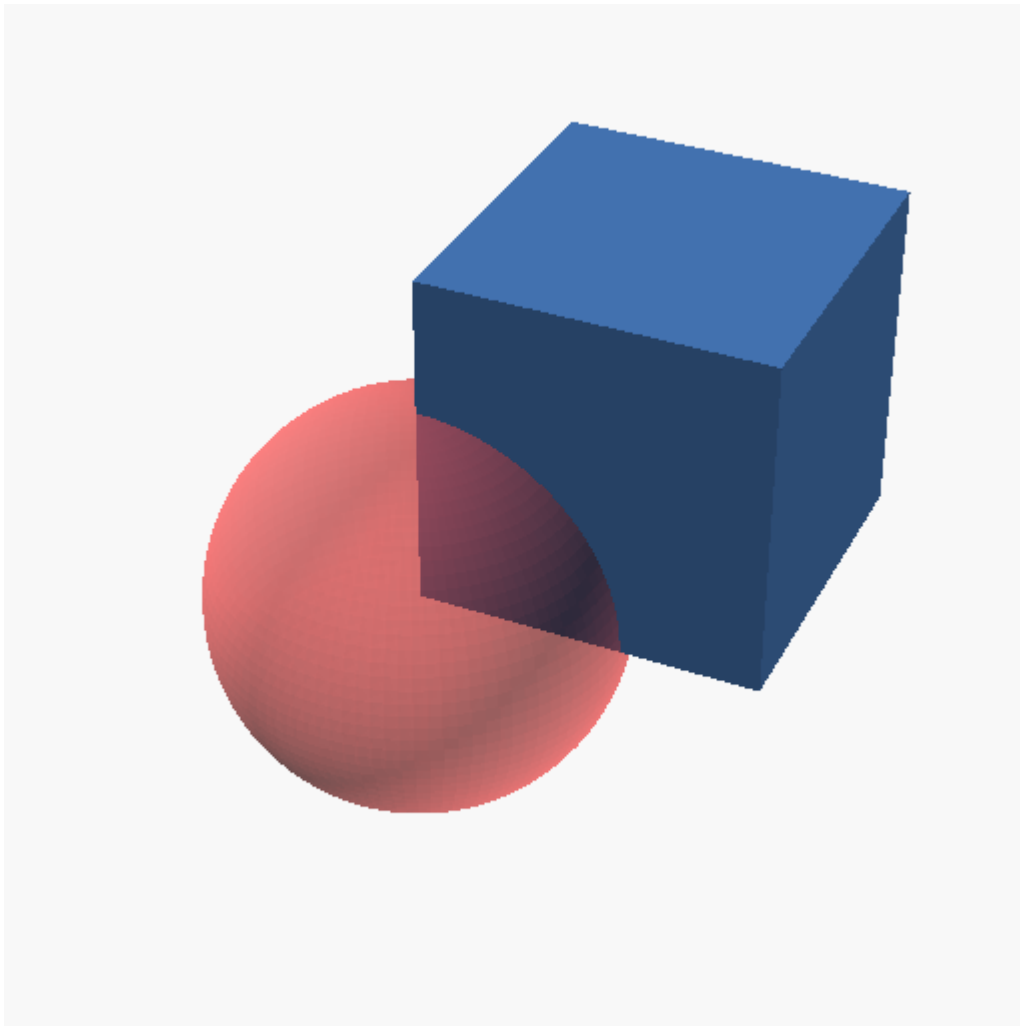


Figure 5. image

Listing 5. Openscad source

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