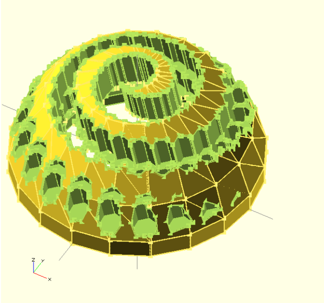
Using OpenSCAD to Design RepRap Printable Parts

Ching Luan Chung
r2dii@yahoo.com

October 20, 2014

What is OpenSCAD?

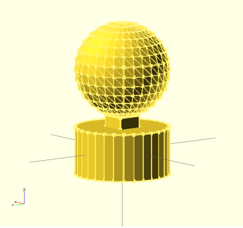


- open source programmer's solid 3D CAD modeller
- runs on Linux, Windows and Mac OS X
- To install: %sudo apt-get install openscad
- website: openscad.org

Advantages of OpenSCAD

- It is designed for programmer. Feels like a programming language instead of CAD software
- easy to scale by changing the input arguments to module
- simple user-interface
- Very easy to use (for simple designs anyway)
- Generate STL file for 3D print

```
cylinder(r=20,h=20,center=true);
translate([0,0,10]) cube([10,10,10],center=true);
translate([0,0,35]) sphere(r=20,center=true);
```



How does it fit in the reprap tool-chain?

- design your part in OpenSCAD and export the design to STL file.
- import STL file in Reptier-Host
- slice it using cura or slic3r to generate gcode
- send g-code to marlin firmware

Basic OpenSCAD Workflow

- (Step 1) Open OpenSCAD from command line.
- (Step 2) Start programming on the left panel
- (Step 3) F5 or F6 to visualize the model on the right panel.
- (Step 4) Correct syntax error if needed.
- (Step 5) If the design needs changes, go to step 2.
- (Step 6) Export STL file

Primitives

3D:

- cylinder
- cube
- sphere
- polyhedron

2D:

- circle
- square
- polygon
- import dxf

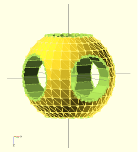
Basic CSG (Constructive Solid Geometry) Operations

Create complex objects using Boolean Operators

- union() {cube([1,2,3]); cube([2,3,4]);};
- difference{cylinder(r=10,h=1); cylinder(r=8,h=1);};
- intersection{cube([1,2,3]); cube([2,3,4]); };

Other Related Operations:

- resize([1,1.5,1.5])
- translate([0,0,1])
- rotate([90,0,90])
- multmatrix(m=[[1,0,0,1],[0,1,0,2],[0,0,1,3],[0,0,0,1]]);



Some Special Variables

- \$fn: number of fragments
- \$fa: minimum angle for a fragment
- \$fs: minimum size of fragment
- \$t: used for animation

Source: openscad wiki book

Extrusion Operations

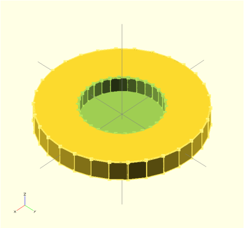
- linear_extrude(): create object based on polygon
- rotate_extrude(): create object by rotating an object

Examples from openscad doc:

```
linear_extrude(height=10,center=true,twist=0)
  translate([2,0,0])
    circle(r=1);

$fn=20;
translate([0,0,10])
rotate_extrude()
translate([2,0,0]) circle(r=3);
```

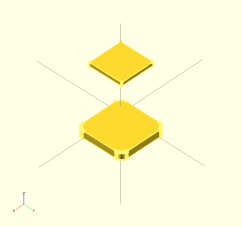
A Simple Example: Cylinder With a Hole

```
difference()
{
  cylinder(r=20,h=5,center=true);
  translate([0,0,1])cylinder(r=10,h=5,center=true);
}
```



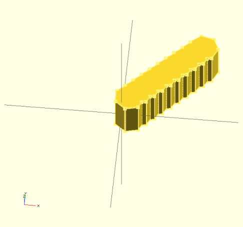
Minkowski Sum

```
minkowski()
{
  cube([5,5,5]);
  cylinder(r=1,h=1);
  // sphere(r=2); // for rounded cube
}
```



Reuse Code

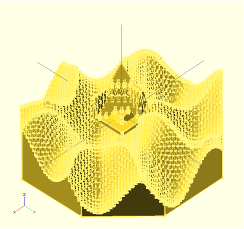
```
module hole(r,h)
{
  cylinder(r=r,h=h,center=true);
}

hole(2,5);
for(i=[1:1:10])
translate([i,i])hole(2,5);
```



Use Octave to Specify 3D Surface

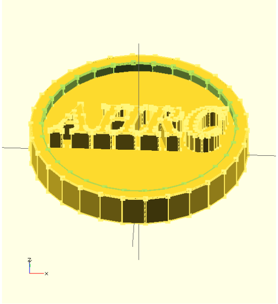
Examples from openscad doc:

```
surface(file = "surface.dat", center = true, convexity = 5)
translate([0,0,5])cube([10,10,10], center =true);

//surface.scad
surface(file = "surface.dat", center = true,
convexity = 5)
translate([0,0,5])cube([10,10,10], center =true);
```



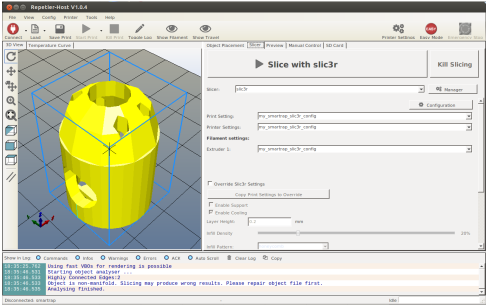
Example of Importing DXF File

```
translate([-32,-15,0]) linear_extrude(height=8, convexity=3

cylinder(h=4,r=20);
difference() {
  cylinder(h=6,r=20);
  cylinder(h=6,r=18);
}
```



Demo Workflow in Repetier-Host

- Open repetier-host
- import object via stl
- slice the object



Considerations for 3D Printing

- Need to decide what orientation to print (see arm1.scad)
- holes are sliced too small. Introduce fudge factor in OpenSCAD to compensate.
- 45 degree overhang rule
- break complex part into small interlocking parts (see finger.scad).
- z axis does not have tensile strength. Avoid small extruding pin in parts. (see anitwobble.scad)
- take advantage of the parametric feature of openscad to quickly change part dimensions.