

OpenSCAD Exercises

Ashley Setter and Thorsten Wittemeier

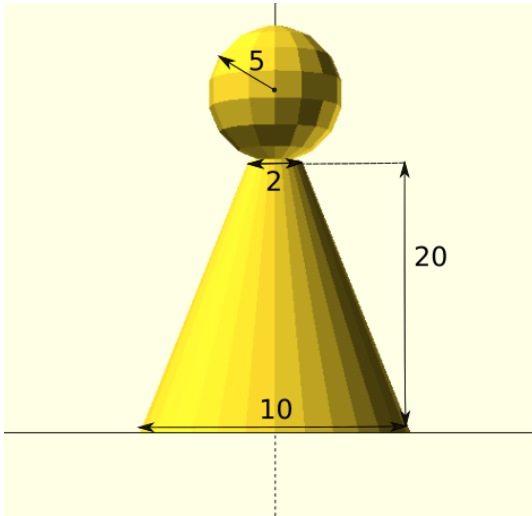
25/2/2016

The Virtual Machine

The virtual machine provides a very basic Debian installation with OpenSCAD 2014.3. Log in as the user **csg** with the password **csg**. There is no display manager installed, so you will need to start the X window environment using the *startx* command. You find OpenSCAD in the *Applications* menu under *Graphics*.

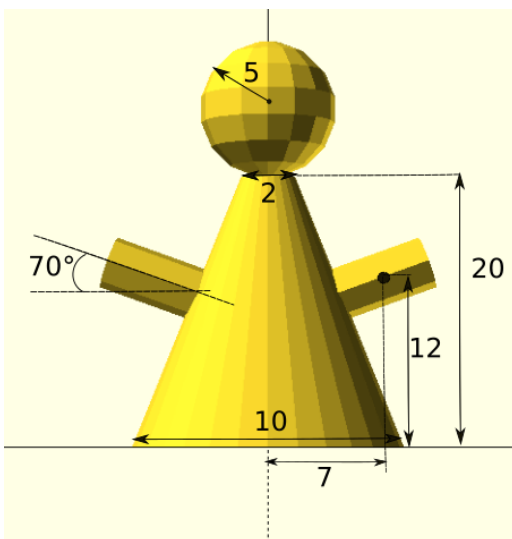
(Normally you will not need this, but if you want to modify this virtual machine, the **root** account has the password **feeg6003**. If you don't know what you are doing or if you wish so for your own peace of mind, you might consider running *apt - get install sudo* and then leaving the *root* account forever.)

Problem 1 - Basic Shapes and Translation



Using the primitive shapes sphere and cylinder attempt to construct the above object in OpenSCAD. Relative dimensions are given on the image such that if you use these relative dimensions the object you create looks like the image given.

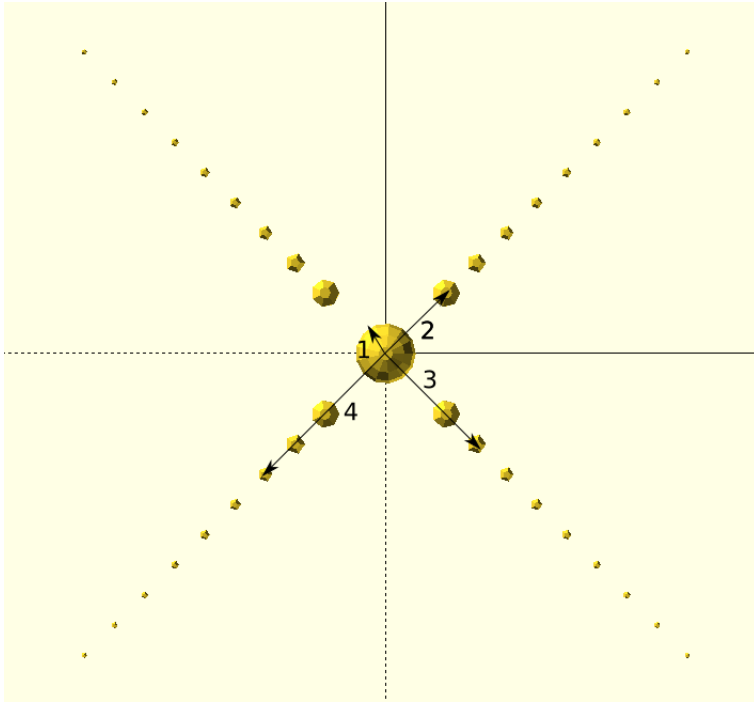
Problem 2 - Translation and Rotation



Building on the object created in Problem 1 attempt to add 2 cylinders at the

given position and rotation such that what you see rendered in OpenSCAD appears the same as the image on the sheet.

Problem 3 - For loops



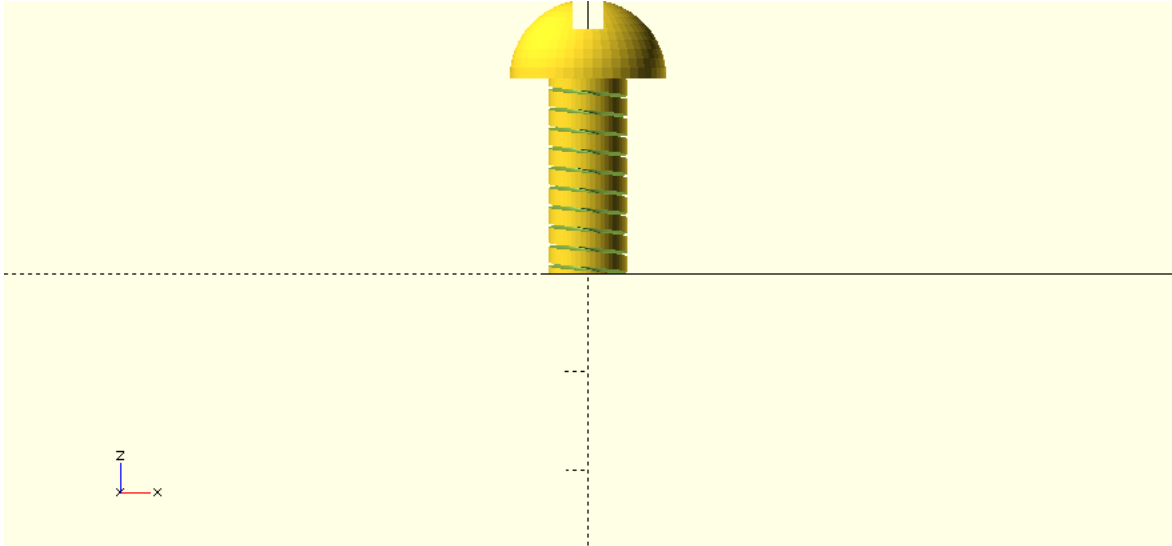
For this problem you will make use of for loops in OpenSCAD to create the more complex shape seen above. For loops are a powerful tool for creating complex shapes out of simple parts which is quite hard to do in typical CAD software.

The first step is to create a sphere with radius 1 at the centre. The next step is to create a for loop in which you place spheres with a radius of $1/i$ where i is the for loop index. The x and y position should also be a function of i such that the spheres spread out in position.

The Syntax for for loops in OpenSCAD can be seen below:

```
for (index = [start:step:end]) { ... }
```

Problem 4 - A simple screw

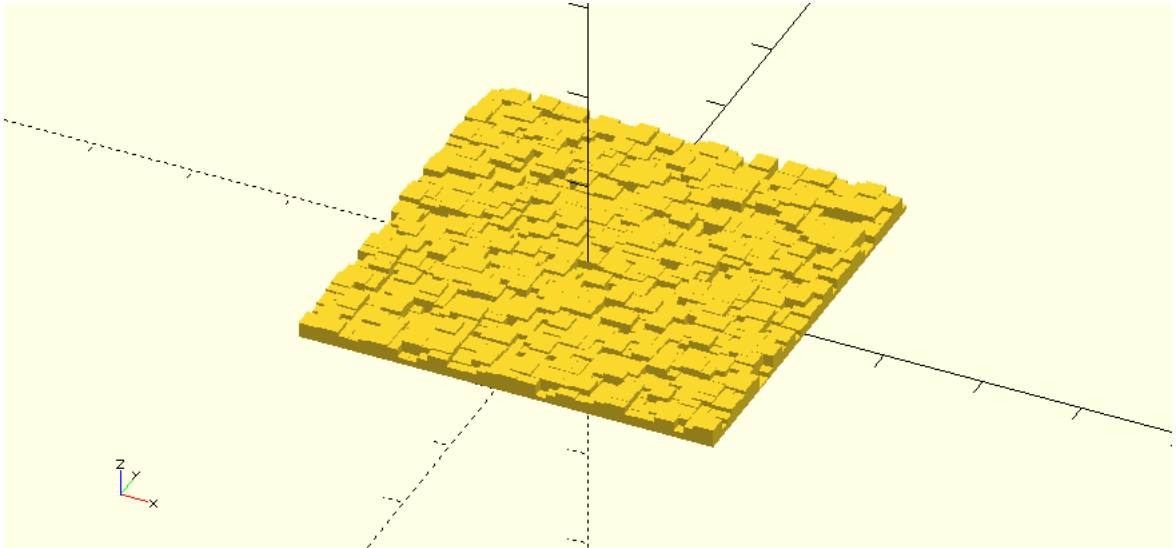


This is a slightly more complex application of the various concepts we have discussed before. In addition you will need linear extrusion to get the same result as in the picture. Try to understand what the various parameters in the example code do. Then figure out which combinations of simple geometries you need.

Listing 1: Usage example for the `linear_extrude` command

```
linear_extrude(height = 20, center = false, convexity = 10, twist = 2*360, $fn = 100)
translate([10, 15, 3])
circle(r = 2);
```

Problem 5 - A random surface



In this exercise, we try to create a random surface out of little rectangular cuboids (using the *cube* function). The first difficulty you will encounter is that the *rands* function can create a random vector, but not a random matrix.

To get an idea of the accuracy of our random distribution, the average should be computed. You will find that OpenSCAD does not provide functions for directly doing so, and you cannot use iterative definitions either.

There are two possible ways out of this dilemma (if you know another way, even better):

- Use a recursive function definitions.
- Use vector algebra.

(Remark: In OpenSCAD 2015.3 there are other, more elegant, options using list comprehension.)