Computational Design + Fabrication: SolidPython

Jonathan Bachrach
EECS UC Berkeley
September 4, 2015
- python layer to openscad
- allows full power of python to script solid creation
- better syntax
- adds few utilities
import solid, numpy
Primitives

- circle, square
- sphere, cube, cylinder

circle(10)
circle(d=10)
square(10)
Compound Functions

- union, intersection, difference
- linear_extrude, rotate_extrude

union() ( circle(d=10), square(size=9) )

versus

union() { circle(d=10); square(size=9) }
+ , * , -

union, intersection, difference

circle(d=10) + square(size=9)

versus

union() ( circle(d=10), square(size=9) )
Translations Shortcuts

- up, down, left, right, forward, back

\[
\text{up}(10)(\ \text{circle}(10) )
\]

versus

\[
\text{translate}([0,0,10])(\ \text{circle}(10) )
\]
circle(d=10) + square(size=9)
left(3)( circle(d=10) ) + square(size=9)
\[
c = \text{circle}(d=10) \\
\text{left}(3)(c) + c
\]
- keeping holes holes
- move hole to end
- allow screw in hole with parts

\[
cylinder(r=5,h=20) + \text{hole}()\left( \text{cylinder}(r=3,h=21) \right)
\]

\[
\text{part}()\left( \text{cylinder}(r=5,h=20) + \text{hole}()\left( \text{cylinder}(r=3,h=21) \right) \right) + \text{cylinder}(r=1,h=20)
\]
Arcs

- arc
- for fillets and rounds
- rad, start_degrees, end_degrees
- arc_inverted – not in the square

arc(rad=10, start_degrees=90, end_degrees=210)
Extrude Along Path

```python
utils.extrude_along_path(shape_pts, path_pts, scale_factors=None)
```

```python
sqr = [[-4,-4],[-4,4],[4,4],[4,-4],[-4,-4]]
pth = [[-10,-10],[-10,10],[10,10],[10,-10]]
extrude_along_path(sqr, pth)
```
Offset

- not yet able to access openscad's offset

```python
offset_points(point_arr, offset, inside=True)
```
- color
- built in colors

```
color([0.0,1.0,0.5])( circle(5) ) +
left(15)( color(Red)( circle(10) ) )
```
Attributes

- background, debug, root, disable

\[
\text{background( sphere(10) ) + cube(2)}
\]
sphere(10) - debug( cylinder(r=2,h=15) )
Open SCAD – http://www.openscad.org