The CadQuery Ecosystem

Jeremy Wright

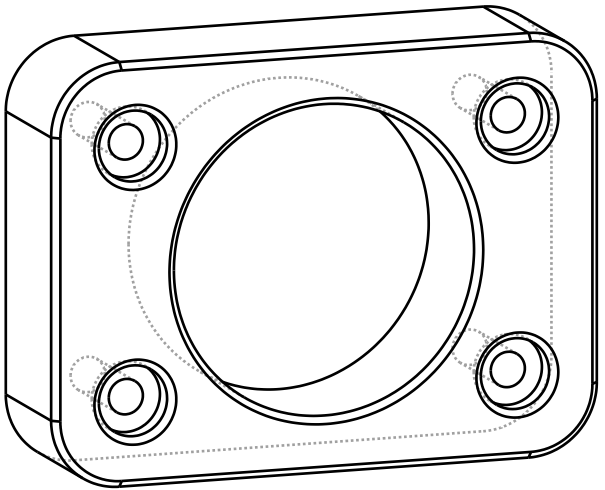
Who We Are

- Dave Cowden (CadQuery's creator)
- Adam Urbańczyk
- Marcus Boyd
- Jeremy Wright (that's me)
- A great community of creators and contributors

Introduction

CadQuery is a Python module for building parametric 3D CAD models in boundary representation (B-rep)

```
import cadquery as cq

height = 40.0
width = 30.0
thickness = 10.0
radius = 11.0
padding = 12.0
rf = 5
cbore_r1,cbore_r2, cbore_d = 2.5, 5, 2
ch = .5

Result = (
    cq.Workplane()
    .rect(height, width).circle(radius).extrude(thickness)
    .faces(">Z").workplane()
    .rect(height - padding, width - padding, forConstruction=True)
    .vertices().cboreHole(cbore_r1,cbore_r2, cbore_d)
    .edges("|Z").fillet(rf)
    .faces('>Z').chamfer(ch)
)
```

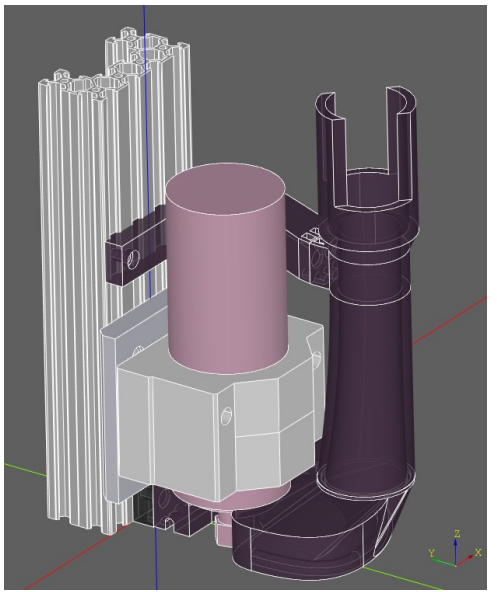


Capabilities

- 2D primitives
  - Rectangle, circle, ellipse, arc, polyline, slot
  - Spline
  - Parametric curves
  - Offset
- 3D primitives
  - Box, sphere
- CSG operations
  - Cut
  - Intersect
  - Union
- Selectors DSL
  - Choose vertices, edges, faces, solids
  - Combine selectors logically or chain them
  - Support tagging of elements

- 3D operations
  - Extrude (tapered, twisted)
  - Revolve
  - Loft
  - Shell
  - Fillet, chamfer
  - Sweep / multi-section sweep
  - 3D text
  - Fill
- Supported formats
  - STEP (R/W)
  - DXF (R/W)
  - BREP (R/W)
  - STL (W)
  - AMF (W)
  - SVG (W)
  - VRML (W)

- Assembly
  - Constraints
  - Multi-color STEP export
- Sketching (2D)
  - Constraints
  - DXF export



Assembly Example
Credit: @marcus7070

Current State

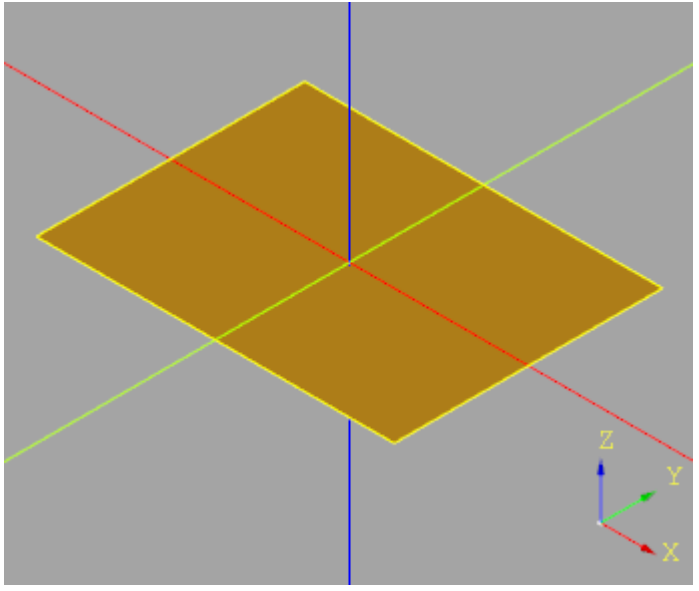
- Based on the OpenCASCADE (OCCT) CAD kernel, version 7.5, with custom Python bindings (OCP) using pybind11
- Update to OCCT 7.6 is in-progress
- Assembly feature is maturing
- Sketch feature is still being explored by the community

Sketch – Starting Out

```
1  import cadquery as cq
2
3  result = (
4      cq.Sketch()
5  )
6
7  show_object(result)
```

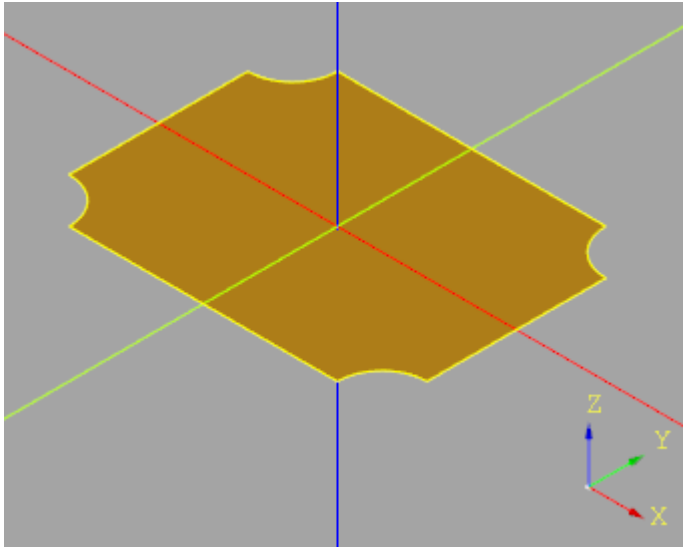
Sketch – General Shape

```
1  import cadquery as cq
2
3  result = (
4      cq.Sketch()
5      .trapezoid(4,3,90)
6  )
7
8  show_object(result)
```



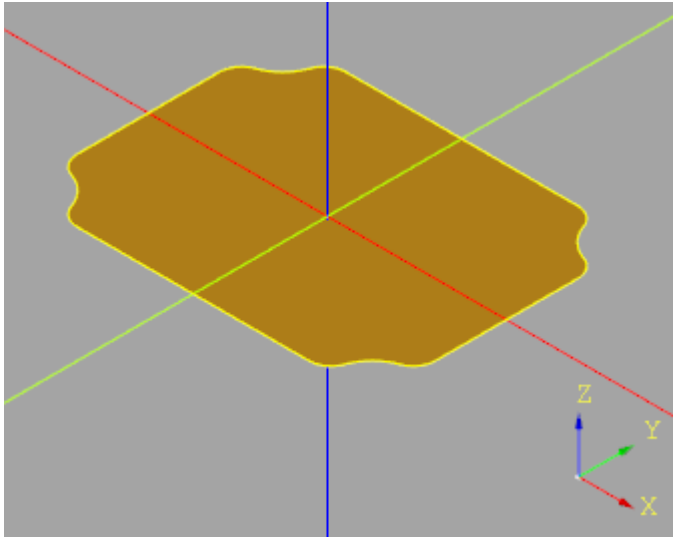
Sketch – Corner Cutouts

```
1   import cadquery as cq
2
3   result = (
4       cq.Sketch()
5       .trapezoid(4,3,90)
6       .vertices()
7       .circle(.5, mode='s')
8   )
9
10  show_object(result)
```



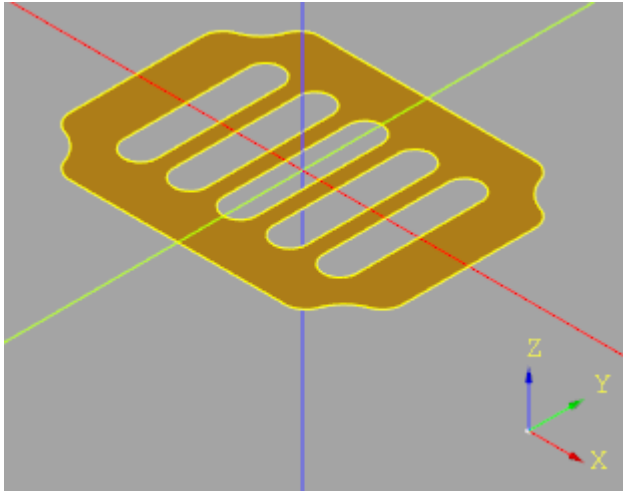
Sketch - Fillets

```python
import cadquery as cq

result = (
    cq.Sketch()
    .trapezoid(4,3,90)
    .vertices()
    .circle(.5, mode='s')
    .reset()
    .vertices()
    .fillet(.25)
)

show_object(result)
```



Sketch - Slots

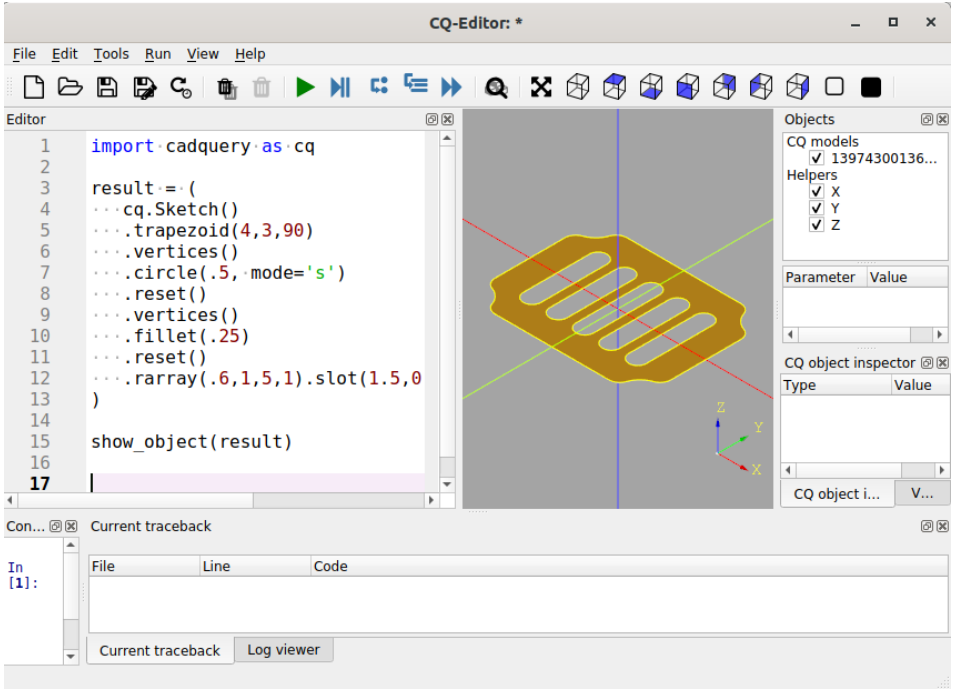
```python
import cadquery as cq

result = (
    cq.Sketch()
    .trapezoid(4,3,90)
    .vertices()
    .circle(.5, mode='s')
    .reset()
    .vertices()
    .fillet(.25)
    .reset()
    .rarray(.6,1,5,1).slot(1.5,0.4, mode='s', angle=90)
)

show_object(result)
```



Learn More

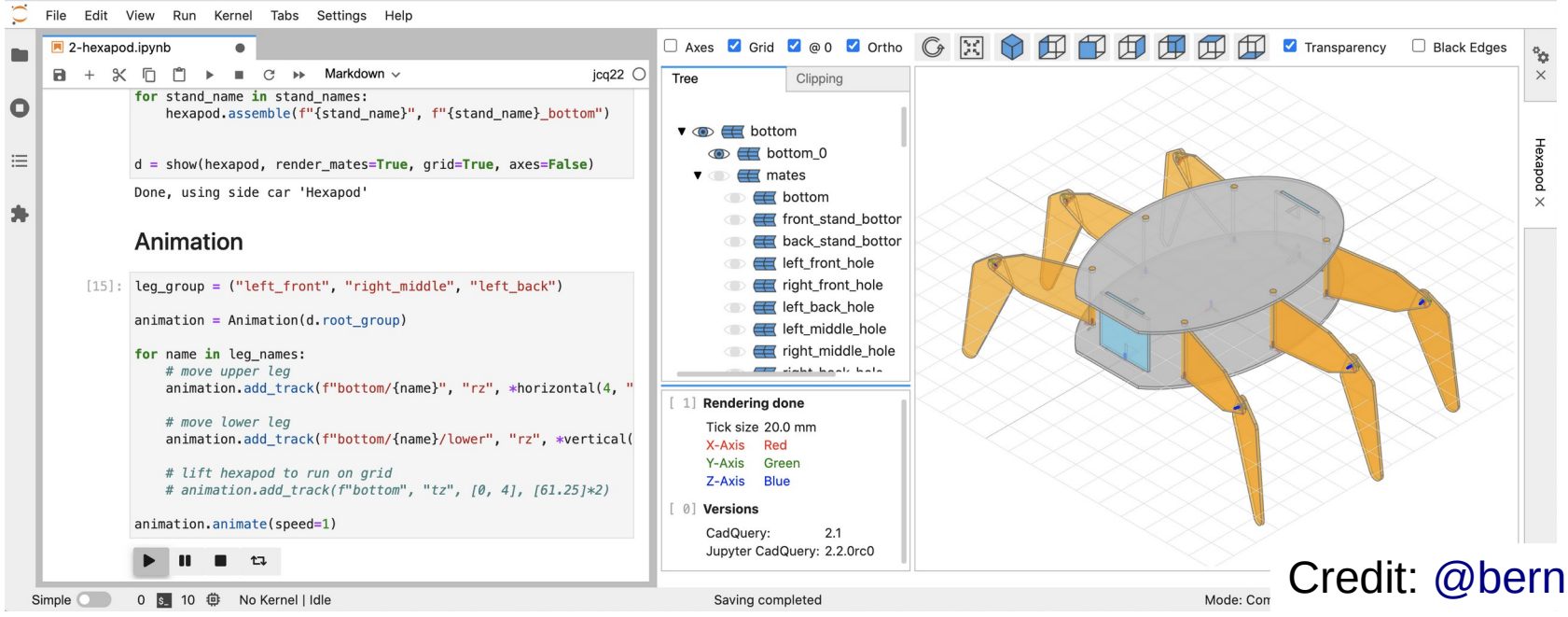
Editors: CQ-editor

- Desktop application IDE (Qt based)
- Maintained by core team
- Debugger features: breakpoints, step into, step over
- Object inspection
- Automatic reload of imported modules



Editors: jupyter-cadquery

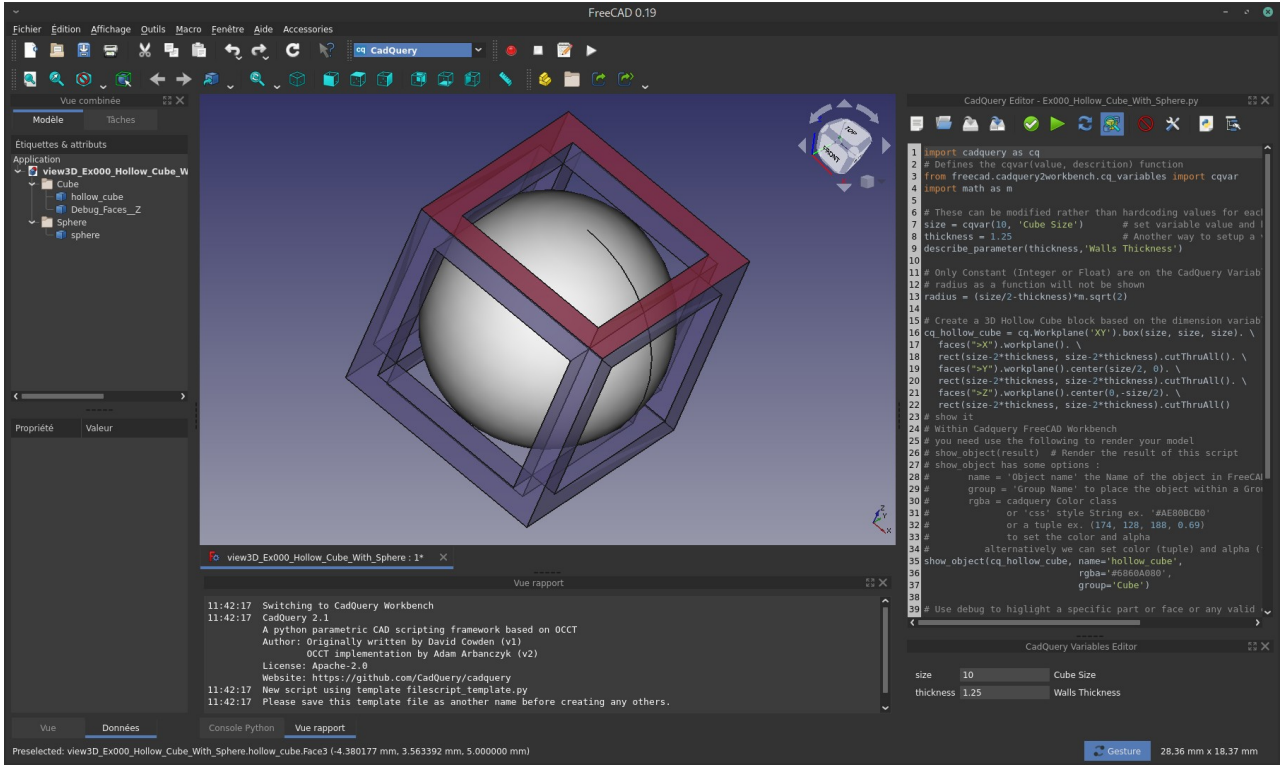
- Web based IDE built on Jupyter
- Allows integration of engineering and scientific notebooks with CadQuery
- Includes a Visual Studio Code integration



Credit: @bernhard-42

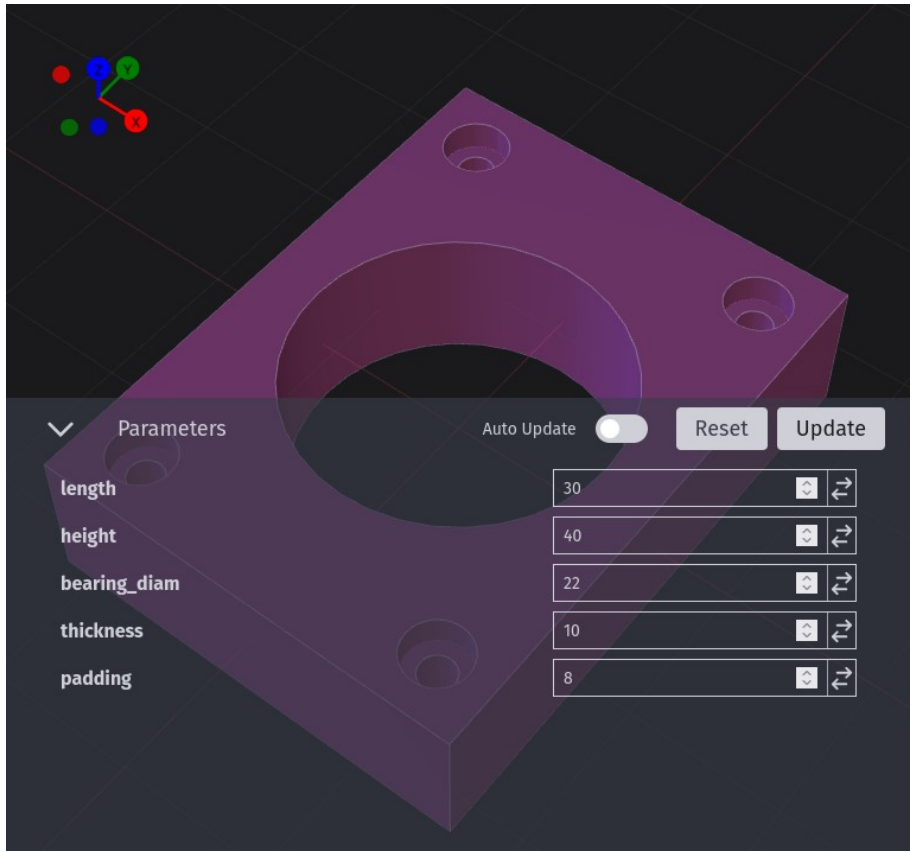
Editors: freecad-cadquery2-workbench

- Updated FreeCAD Workbench
- Supports CadQuery 2.x (installed separately)
- Great for using CadQuery with other FreeCAD workbenches



Credit: @jpmlt

Sharing: CadHub

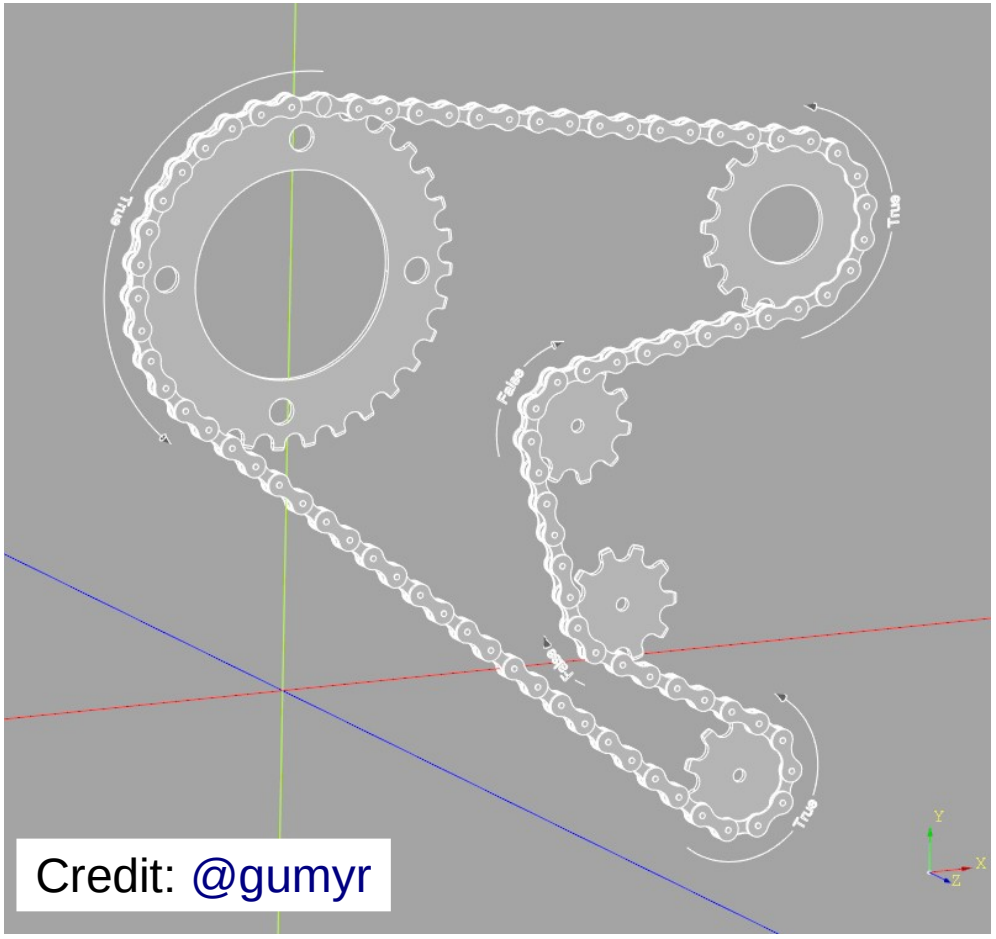


Credit: @Irev-Dev

- “CodeCAD” sharing website
- Parameter customizer, simple editor and STL export
- Multiple integrations in one service (CadQuery, OpenSCAD, Open JSCAD, Curv3D)
- Discord with a nice intersection of developers and users from all those projects

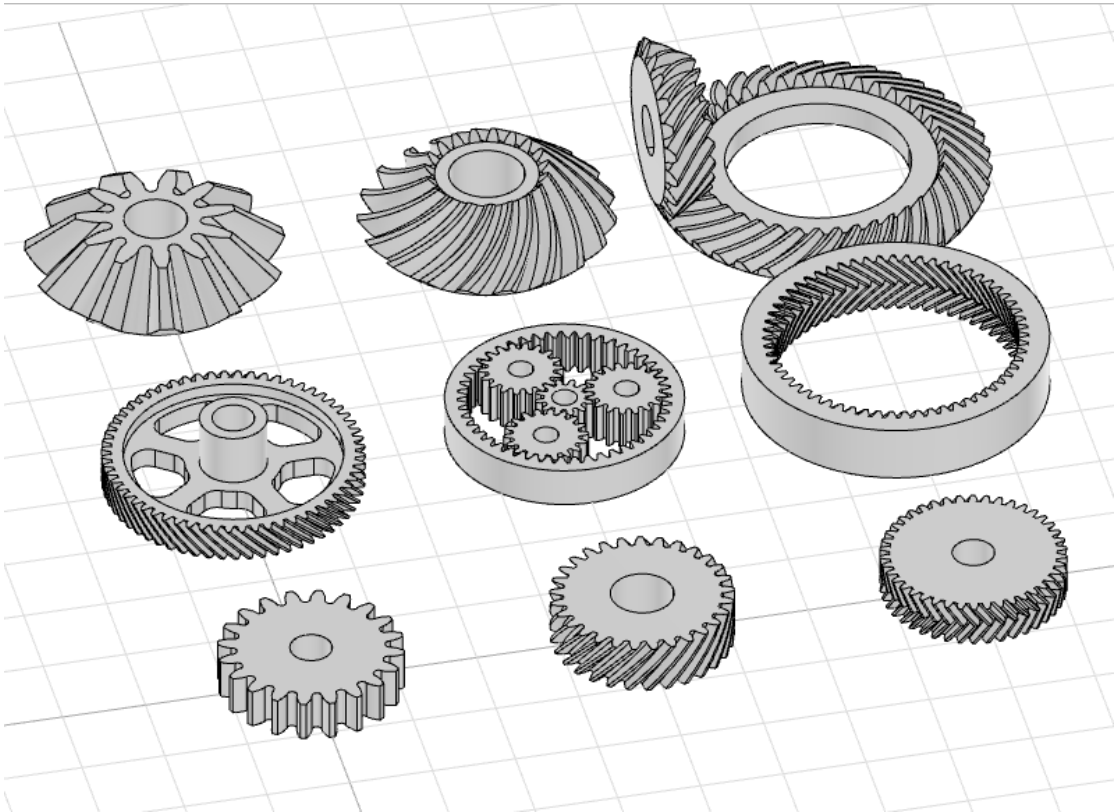
Extensions: cq_warehouse

- Sprockets and chains
- Threads, holes and fasteners
- Fastener locations for holes in mating parts
- Drafting (model-based definition)
- Fastener bill of materials
- CadQuery Workplane extensions (Assembly, Vector, Vertex)



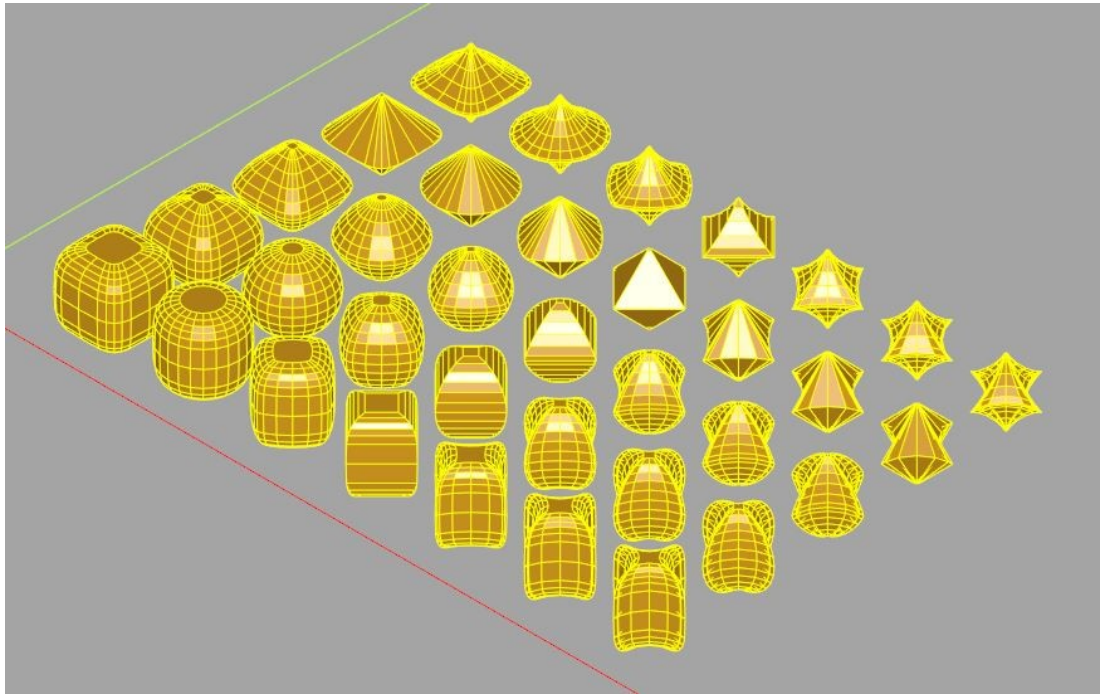
Extensions: cq_gears

- Spur gear
- Helical gear
- Herringbone gear
- Ring gear
- Planetary gearsets
- Straight and helical bevel gears
- Gear rack



Credit: @meadiode

Extensions: cqMore

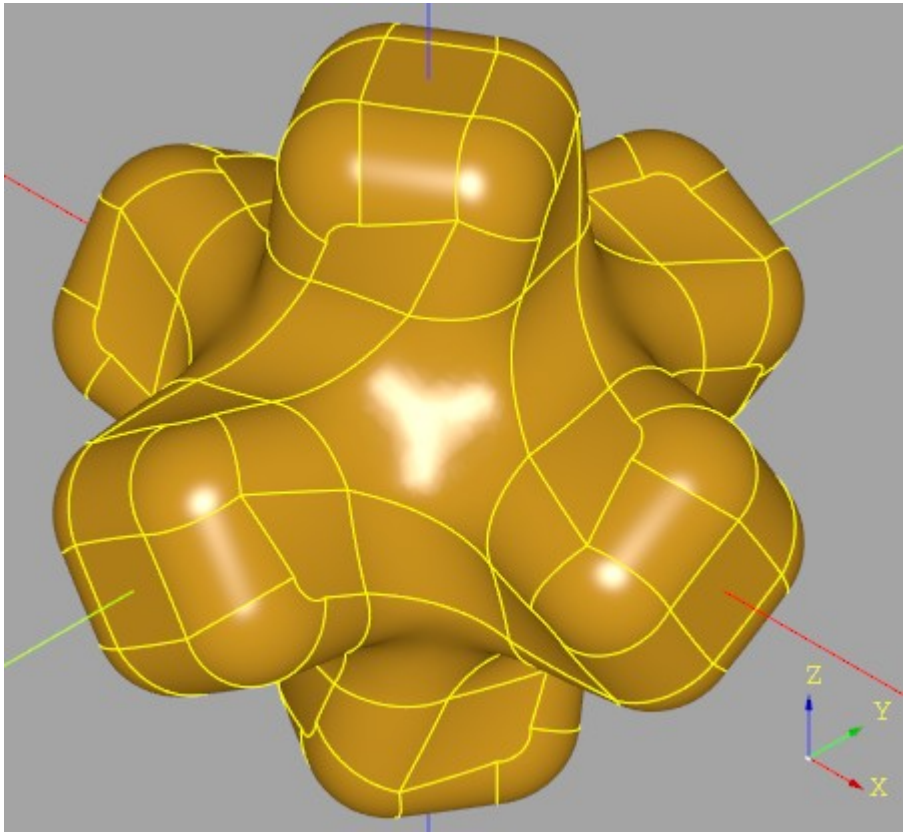


Credit: @JustinSDK

- Extensions to CadQuery’s core Workplane class
- Polyhedron generation
- Matrix operations
- Curves

Extensions: cadquery-plugins

- Community contributed extensions
- Caching
- Expanded selectors
- Heatset inserts
- Apply operations to non-planar faces



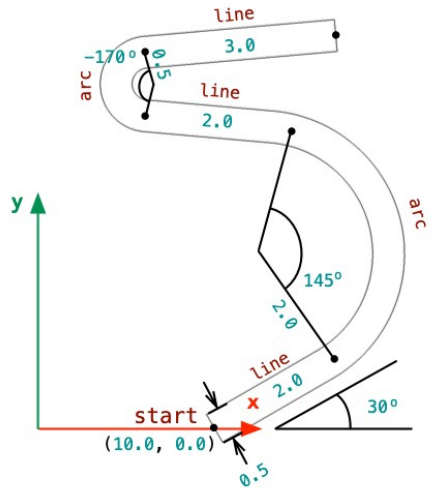
Credit @fedorkotov

Extensions: cq-kit

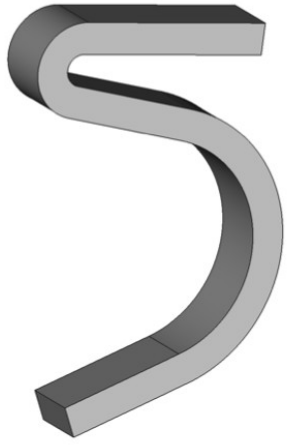
- Additional selectors
- File formats (IGES import and export)
- Pretty-printing of objects
- Solid and edge discretization
- Ribbon class (see screenshot)

```
path = [
    ("start", {"position": (10.0, 0.0), "direction": 30.0, "width": 0.5}),
    ("line", {"length": 2.0}),
    ("arc", {"radius": 2.0, "angle": 145.0}),
    ("line", {"length": 2}),
    ("arc", {"radius": 0.5, "angle": -170}),
    ("line", {"length": 3}),
]
```

```
rb = Ribbon("XY", path)
r = rb.render()
```

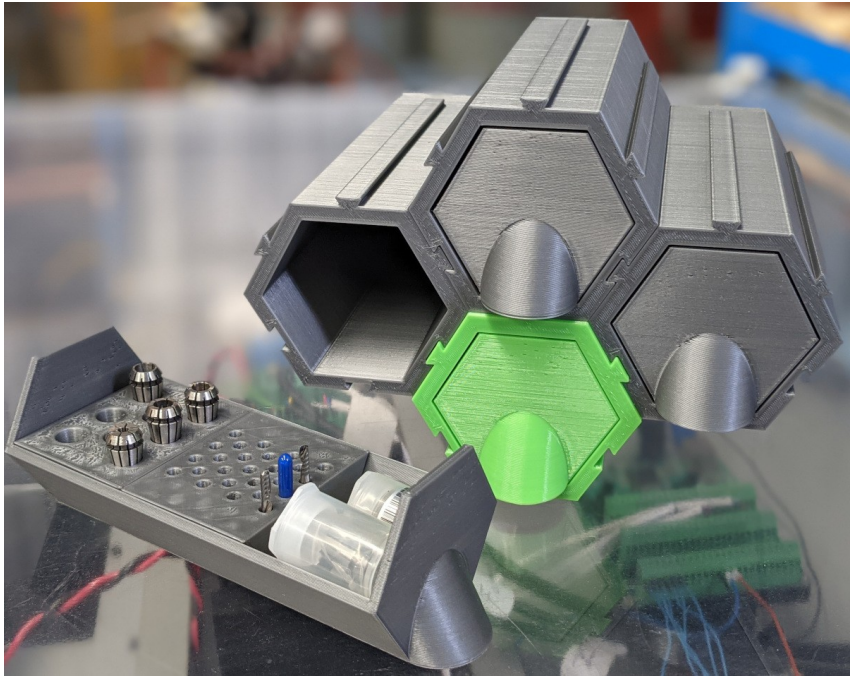



```
rb = Ribbon("XY", path)
r = rb.render().extrude(1)
```



Credit: @michaelgale

Examples: cadquery-contrib

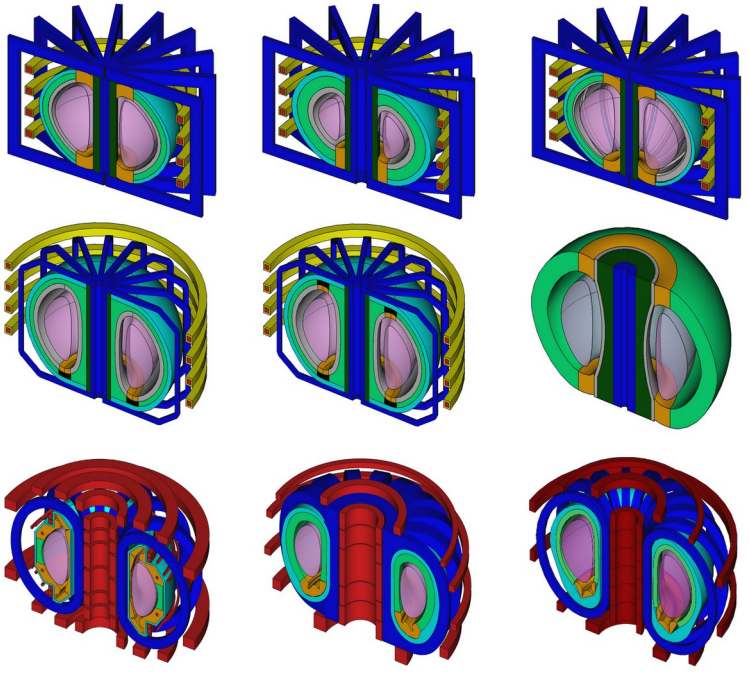


Credit: @marcus7070

- Collection of advanced examples
- Kelvin Cell
- Involute gear
- Thread generator
- Modular tooling drawers

Using CQ: Paramak

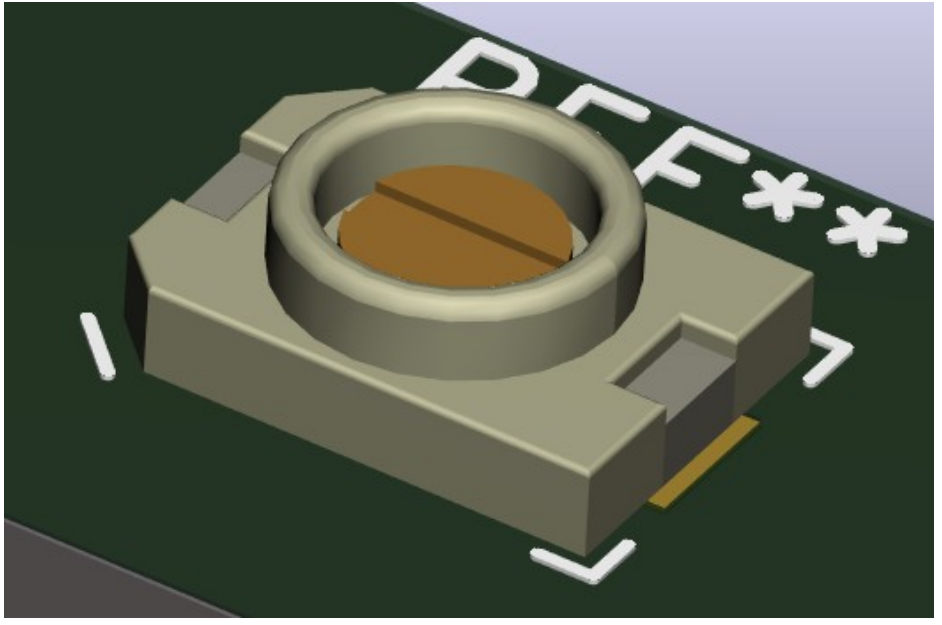
- Generator package allows rapid production of 3D CAD models of fusion reactors
- On-demand fusion reactor models and neutronics simulations on the Web



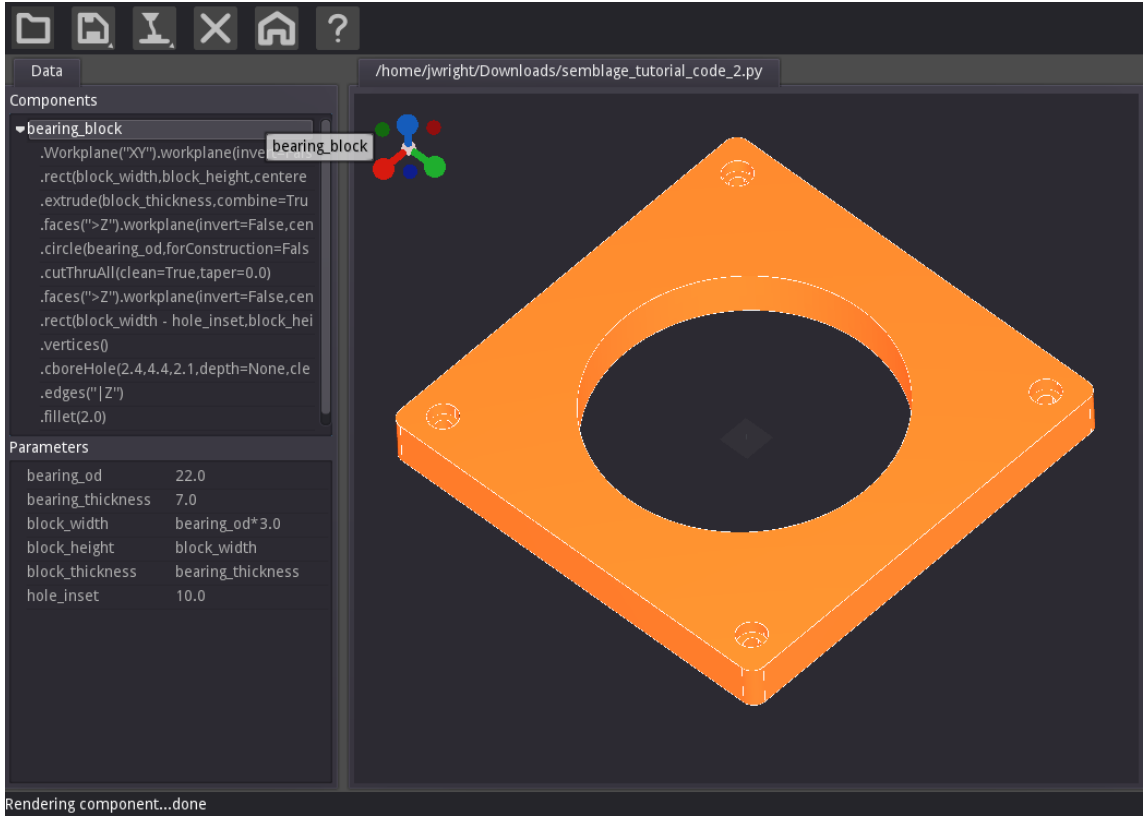
Credit: Paramak and @shimwell

Using CQ - KiCAD

- Python scripts for generating 3D models of electronic components
- Exports STEP and VRML format
- Enables 3D preview of designs
- Being updated for CadQuery 2.x



Using CQ - Semblage



- CAD GUI that generates CadQuery code based on user mouse interactions
- Generate selectors from mouse selections
- Currently in alpha (users still need CadQuery knowledge)
- Being used for light project work

Community

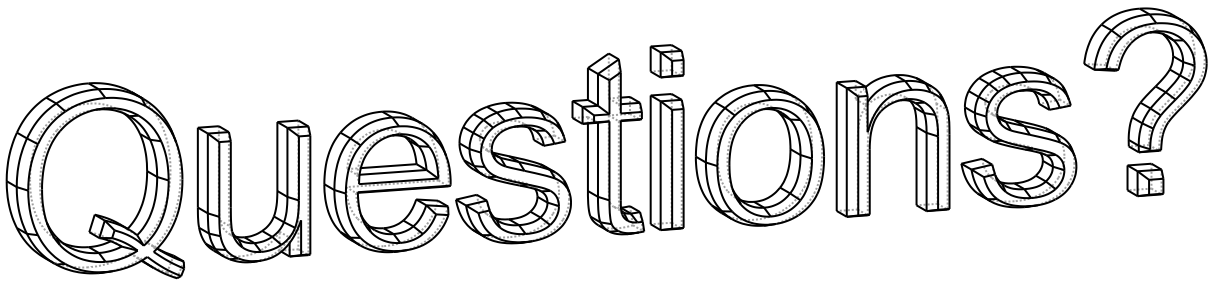
- Discord
- Matrix (bridged to Discord server)
- Google Group
- GitHub

On the Shoulders of Giants

CadQuery and CQ-editor would not be possible without the following open source projects

- Python
- OpenCascade
- FreeCAD
- PythonOCC
- PyParsing
- Conda
- PyOCCT
- Qt/PyQt
- Spyder
- PyQtGraph
- PyInstaller
- EzDXF
- Pybind11
- nlopt


```
import cadquery as cq

res = (
    cq.Workplane()
    .text('Questions?',10,2)
    .faces('>Z')
    .chamfer(.8,.2)
)
```



Contact

- @jmwright on GitHub, GitLab, Patreon, Discord
- @wrightjmf on Twitter