Towards CadQuery 2.0

Adam Urbańczyk
Introduction

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

```python
import cadquery as cq

dimension = 40.0
width = 30.0
thickness = 10.0
diameter = 22.0
padding = 12.0
rf = 5
cbore_r1, cbore_r2, cbore_d = 2.5, 5, 2
ch = .5

result = (cq.Workplane("XY").box(dimension, width, thickness)
    .faces(">Z").workplane().hole(diameter)
    .faces(">Z").workplane()
    .rect(dimension - padding, width - padding, forConstruction=True)
    .vertices().cboreHole(cbore_r1, cbore_r2, cbore_d)
    .edges("|Z").fillet(rf).faces('>Z').chamfer(ch))
```
List of contributors

• Core team
  - Dave Cowden, Jeremy Wright and Adam Urbańczyk

• Contributors CadQuery
  - hyOzd, bragostin, mgreminger, justbuchanan, huskier, fragmuffin, Peque, bweissinger, osterwood, moeb, asukiaaa, HLevering, westurner, Renha, armyofevilrobots, gebner, krasin, Grawp, phillipthelen, bsilvereagle, xix-xeaon

• Contributors CQ-editor
  - gebner, justbuchanan, jmwright
History

- Project started by Dave
  - Used in the backend of a parametric modeling website
- Jeremy wrote a FreeCAD workbench
  - https://github.com/jmwright/cadquery-freecad-module
- I joined the team last
  - Started the PythonOCC transition and CQ-editor
Project goals and motivation

- Use standard and user-friendly programming language (Python)
- B-rep CAD kernel (OpenCascade)
- Fluent API
- Advanced modeling capabilities
- Ability to import and export STEP models
- High performance
Capabilities

- **2D primitives**
  - Rectangle, circle, polygon/polyline
  - Spline
  - Parametric curves

- **3D primitives**
  - Box, sphere

- **CSG operations**
  - Cut
  - Intersect
  - Union

- **Selectors DSL**
  - Choose vertices, edges, faces, solids
  - Combine selectors logically or chain them

- **3D operations**
  - Extrude (tapered, twisted)
  - Revolve
  - Loft
  - Shell
  - Fillet, chamfer
  - Sweep / multi-section sweep
  - 3D text

- **Supported formats**
  - STEP (R/W)
  - STL (W)
  - AMF (W)
  - SVG (W)
import cadquery as cq

w, d, h = 100, 60, 4
pitch = 20
nx, ny = 5, 3
r_hole = 7
ch = .5

Plate = (cq.Workplane('XY')
    .box(w, d, h)
    .edges('|Z')
    .fillet(5)
    .faces('>|Z').workplane()
    .rarray(pitch, pitch, nx, ny, True)
    .hole(r_hole)
    .rarray(pitch, pitch, nx-1, 1, True)
    .slot2D(d*0.8, r_hole, 90).cutThruAll()
    .faces('>|Z').edges()
    .chamfer(ch))
import cadquery as cq
from math import sin, cos, pi, floor

def hypocycloid(t, r1, r2):
    return ((r1-r2)*cos(t)+r2*cos(r1/r2*t-t), (r1-r2)*sin(t) + r2*sin(-(r1/r2*t-t)))

def epicycloid(t, r1, r2):
    return ((r1+r2)*cos(t)-r2*cos(r1/r2*t+t), (r1+r2)*sin(t) - r2*sin(r1/r2*t+t))

def gear(t, r1=4, r2=1):
    if (-1)**(1+floor(t/2/pi*(r1/r2))) < 0:
        return epicycloid(t, r1, r2)
    else:
        return hypocycloid(t, r1, r2)

h = 20; twist = 180; d = 2

res = (cq.Workplane('XY')
    .parametricCurve(lambda t: gear(t*2*pi, 6, 1))
    .twistExtrude(20, 180)
    .faces('>Z').workplane()
    .hole(d))
User showcase

@eddieliberato - eddieliberato.github.io

@bragostin

@michaelgale - www.fxbricks.com

@Peque - bulebule.readthedocs.io

@hyOzd

NB: GitHub @usernames
CQ-editor

- Lightweight GUI for CQ
- Based on PyQt and Spyder
- CQ specific goodies
  - Graphical debugging
  - CQ stack inspection
Future plans

- Release CQ 2.0
- Release CQ-editor 0.1
- Move to OpenCascade 7.4
- Boolean operations speed improvements
- Additional surface modeling capabilities
- DXF import
- glTF export
Standing on the shoulders of giants

CQ and CQ-editor wouldn’t be possible without the following open source projects

- Python
- OpenCascade
- FreeCAD
- PythonOCC
- PyParsing
- Conda
- Qt
- PyQt
- Spyder
- PyQtGraph
- PyInstaller