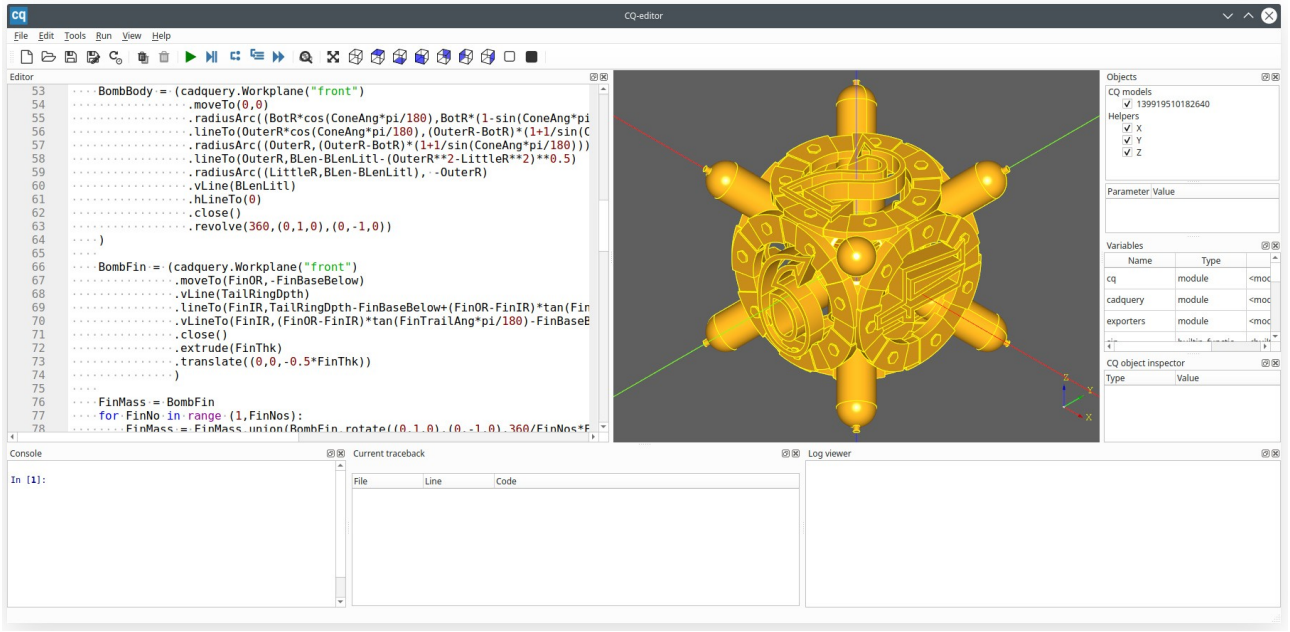
Parametric Bomb Dice Using CADQuery

I have been keen to try my hand at parametric models using CADQuery. I felt a good test would be to translate the OpenSCAD Bomb Dice into CADQuery.



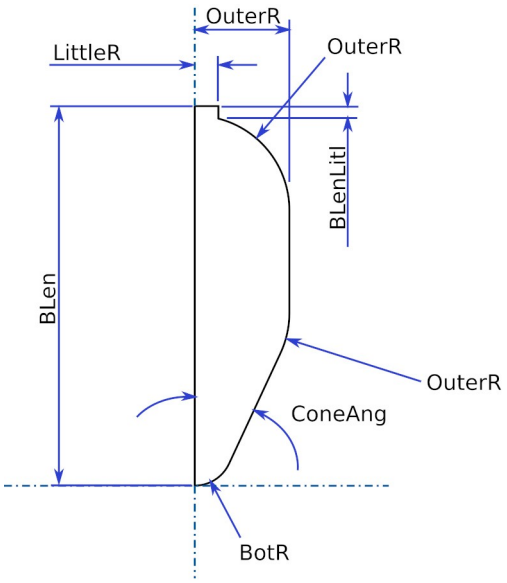
What surprised me was how much more tricky it would be. OpenSCAD has some rather nice features which I make heavy use of in my designs, namely **Convex Hulls**. At the time of this project the latest edition of CADQuery had introduced convex hulls as a function for 2D sketches. Unfortunately there were problems trying to use it and so I had to fall back on an older edition bundled with the CQ-Editor and create the shape outlines in a more manual fashion. So no nice easy shortcuts for me.

Once I got into the swing of creating the 2D shapes it was a powerful technique and yielded excellent outputs. Because 2D shapes formed the basis of most of the model, it became a very different beast from the OpenSCAD model where I could work directly with 3D shapes.

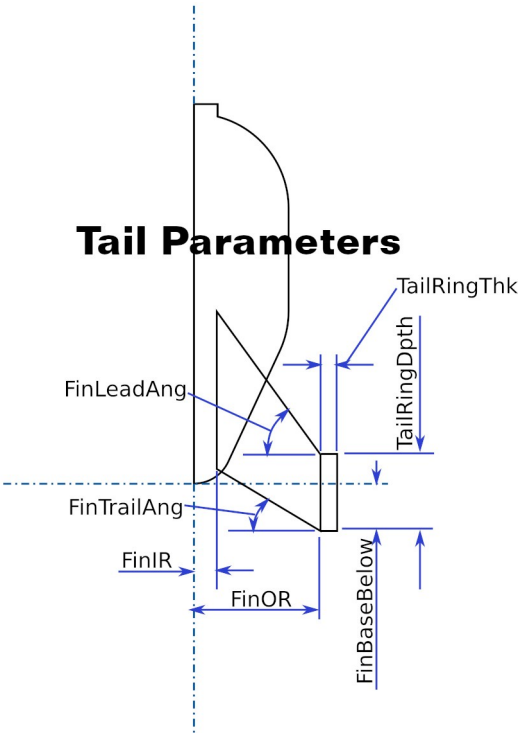
The big advantage of CADQuery over OpenSCAD was its ability to output STEP format models. This opened it to a greater number of Engineering Application, especially here in New Zealand where CNC Machinist insist on working with STEP files. Of course the other major advantage is having the scripting might of python behind it.

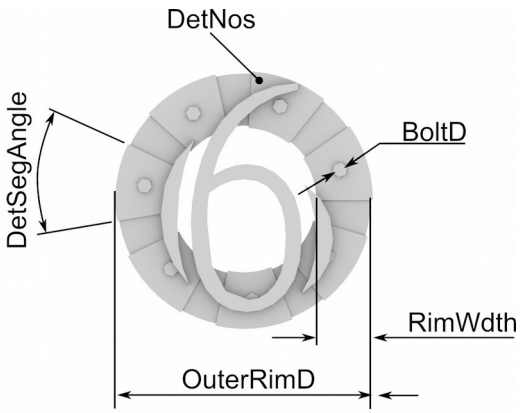
Bomb Dice Parameters

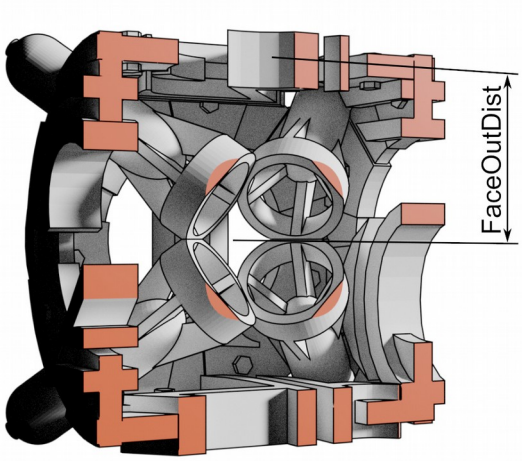
`RenderYN`	Setting this to “yes” will output the two model files ***BombDiceD6v0.step*** and ***BombDiceD6v0.stl*** into the directory where you‘ve placed the ***BombDiceD6v0.py*** script.
Bomb shape Parameters (refer to the diagrams below)	
`LittleR`	The radius of the little tip bit of the bomb
`OuterR`	The overall radius of the bomb’s main shell. This is also used for the radius of the other main hemispherical surfaces
`BLen`	The overall length of the bomb’s main body
`BotR`	The radius of the rounded hemisphere on the tail end of the main bomb body
`BLenLitl`	The length of the little tip bit at the top of the bomb body
`ConeAng`	The angle of the bomb body bottom end cone
`FinNos`	The number of fins
`FinThk`	The thickness of the fins
`FinOR`	Outer edge of the fins. This is also used to set the radius of the tail ring
`FinIR`	Inner edge of the fins
`FinBaseBelow`	Distance between the bottom-most point of the fins and the bottom-most end of the main bomb body. The bottom edge of the tail ring is in line with this too.
`FinLeadAng`	Angle of the fin leading edge
`FinTrailAng`	Angle of the fin trailing edge
`TailRingDpth`	The length of the tail ring
`TailRingThk`	The section thickness of the tail ring
`OutwardAdj`	Radial distance to shift the bomb units outwards
Variables for Face Ring (see diagrams below)	
`OuterRimD`	Outer rim diameter of the main face ring
`RimWdth`	Width of the main face ring band
`FaceThk`	Thickness of the main face ring
`DetailHt`	Height that the details segments project above the surface of the main ring
`DetSegAngle`	Angle covered by detail segment
`DetNos`	Number of detail segments
`BoltD`	Diameter of the bolt head on the detail segment
`FaceOutDist`	Distance of the face mid-plane out from the centre of the cube.
Text Parameters	
`Bombfont`	The name of any font you have on your system that you want to use.
`LettrSiz`	The size of the letter
`LetHt`	The amount the letter is extruded by
`LetVPozAdj`	A factor to adjust the position of the letter vertically
`LetHPozAdj`	A factor to adjust the position of the letter horizontally
`FaceTXTList`	A list of the six text items you want to see on the dice faces. You can put anything in here. The default is the six numbers of a balanced dice ["1","6","3","4","2","5"]

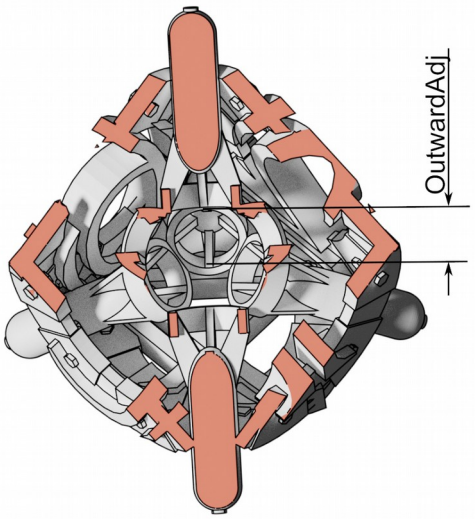


Main Body Parameters









The Code

In a shameless attempt to increase the page count for this article and potentially absorb a few more google hits, here is the listing (just mind the word wrap if copying this from the pdf).

You can download the code from here: ***BombDiceD6v0.py***

```
#BombDiceD6v0.py

import cadquery
from cadquery import exporters
from math import sin, cos, tan, pi, acos, atan2

RenderYN = "no"  #Render to file (yes/no)?

#Bomb shape Parameters
LittleR = 1
OuterR = 6
BLen = 40
BotR = 2
BLenLitl = 1.6
ConeAng = 30

FinNos = 4
FinThk = 1
FinOR = 8
FinIR = 1
FinBaseBelow = 4
FinLeadAng = 60
FinTrailAng = 45
TailRingDpth = 8
TailRingThk = 1

OutwardAdj = 15


#Variables for Face Ring
OuterRimD = 50
RimWdth = 8
FaceThk = 2.5
DetailHt = 0.5
DetSegAngle = 35    #Angle covered by detail segment
DetNos = 7      #Number of detail segments
BoltD = 4

FaceOutDist = 25

Bombfont = "Walshes Outline"#["MetropolitainesD", "ZapfChan Bd BT","Xtraflexidisc", "Liberation
Sans:style=Italic", "Liberation Mono", "Liberation Serif"]
#"WeddingText BT","Walshes Outline","Vertigo Upright BRK","Vanilla Whale","Ubiquity BRK","Top
Bond","TimeScrDBol","Stencil","StayPuft",
#"Snap ITC","Showcard Gothic","Colonna MT"

LettrSiz = 52
LetHt = 6
LetVPozAdj = 0.00  # A factor to adjust the position of the letter vertically
LetHPozAdj = -0.04

FaceTXTList = ["1","6","3","4","2","5"]

def BombSpikes():
    BombBody = (cadquery.Workplane("front")
                .moveTo(0,0)
                .radiusArc((BotR*cos(ConeAng*pi/180),BotR*(1-sin(ConeAng*pi/180))), -BotR)
                .lineTo(OuterR*cos(ConeAng*pi/180),(OuterR-BotR)*(1+1/sin(ConeAng*pi/180))-
OuterR*sin(ConeAng*pi/180))
                .radiusArc((OuterR,(OuterR-BotR)*(1+1/sin(ConeAng*pi/180))), -OuterR)
                .lineTo(OuterR,BLen-BLenLitl-(OuterR**2-LittleR**2)**0.5)
                .radiusArc((LittleR,BLen-BLenLitl), -OuterR)
                .vLine(BLenLitl)
                .hLineTo(0)
                .close()
                .revolve(360,(0,1,0),(0,-1,0))
    )
```

```
    BombFin = (cadquery.Workplane("front")
               .moveTo(FinOR,-FinBaseBelow)
               .vLine(TailRingDpth)
               .lineTo(FinIR,TailRingDpth-FinBaseBelow+(FinOR-FinIR)*tan(FinLeadAng*pi/180))
               .vLineTo(FinIR,(FinOR-FinIR)*tan(FinTrailAng*pi/180)-FinBaseBelow)
               .close()
               .extrude(FinThk)
               .translate((0,0,-0.5*FinThk))
               )

    FinMass = BombFin
    for FinNo in range (1,FinNos):
        FinMass = FinMass.union(BombFin.rotate((0,1,0),(0,-1,0),360/FinNos*FinNo))

    BombTail = (cadquery.Workplane("front")
                .moveTo(FinOR,-FinBaseBelow)
                .rect(TailRingThk,TailRingDpth)
                .revolve(360,(0,1,0),(0,-1,0))
                .translate((0,0.5*TailRingDpth))
                )

    BombUnit = BombBody.union(FinMass).union(BombTail)

    # Eight Diagonals
    b = acos(1/3**0.5)
    c = atan2(1,1)
    Diag1 = BombUnit.translate((0,OutwardAdj,0)).rotate((1,0,0),(-1,0,0),b*180/pi).rotate((0,1,0),
(0,-1,0),c*180/pi)
    Diag2 = BombUnit.translate((0,OutwardAdj,0)).rotate((1,0,0),(-1,0,0),-b*180/pi).rotate((0,1,0),
(0,-1,0),c*180/pi)
    Diag3 = BombUnit.translate((0,OutwardAdj,0)).rotate((1,0,0),(-1,0,0),-b*180/pi).rotate((0,1,0),
(0,-1,0),-c*180/pi)
    Diag4 = BombUnit.translate((0,OutwardAdj,0)).rotate((1,0,0),(-1,0,0),b*180/pi).rotate((0,1,0),
(0,-1,0),-c*180/pi)

    DiagSide1 = Diag1.union(Diag2).union(Diag3).union(Diag4)
    DiagSide2 = DiagSide1.rotate((0,0,1),(0,0,-1),180)

    Diags = DiagSide1.union(DiagSide2)

    return(Diags)

def FacePlate():

    BombFacePlate = (cadquery.Workplane("top")
                     .moveTo(0.5*(OuterRimD-RimWdth),0)
                     .rect(RimWdth,FaceThk,centered=True)
                     .revolve(360,(0,1,0),(0,-1,0))
    )

    BombFaceDetail = (cadquery.Workplane("top")
                      .moveTo(0.5*(OuterRimD-RimWdth),0)
                      .rect(RimWdth+2*DetailHt,FaceThk+2*DetailHt,centered=True)
                      .revolve(DetSegAngle,(0,1,0),(0,-1,0))
    )

    FaceBoltDetail = (cadquery.Workplane("top")
                       .moveTo(0.5*(OuterRimD-RimWdth)*cos(pi*DetSegAngle/360),0.5*(OuterRimD-
RimWdth)*sin(pi*DetSegAngle/360))
                       .polygon(6,BoltD)
                       .extrude(FaceThk+4*DetailHt)
                       .rotate((-1,0,0),(1,0,0),90)
                       .translate((0,0,-0.5*FaceThk-2*DetailHt))
    )
    #Build the FacePlate up from the parts
    BombFaceDetFul = BombFaceDetail.union(FaceBoltDetail)
    BombFaceDetFulMass = BombFaceDetFul
    for DetNo in range (1,DetNos):
        BombFaceDetFulMass = BombFaceDetFulMass.union(BombFaceDetFul.rotate((0,0,1),(0,0,-1),360/
DetNos*DetNo))
    FaceSingle = BombFacePlate.union(BombFaceDetFulMass)
    return(FaceSingle)

def FaceTXT(TNum):
    BombFaceTXT = (cadquery.Workplane("front")
                   .text(TNum,LettrSiz,LetHt,combine = True,font = Bombfont,kind = "regular",halign
= 'center',valign = 'center')
```

```
                    .translate((LetHPozAdj*LettrSiz,LetVPozAdj*LettrSiz,-0.5*LetHt))
                    )
    return(BombFaceTXT)

def ComboFace():
    Face1 = FacePlate().union(FaceTXT(FaceTXTList[0])).translate((0,0,FaceOutDist)).rotate((1,0,0),
(-1,0,0),90)
    Face2 = FacePlate().union(FaceTXT(FaceTXTList[1])).translate((0,0,FaceOutDist)).rotate((-1,0,0),
(1,0,0),90)
    Face3 = FacePlate().union(FaceTXT(FaceTXTList[2])).translate((0,0,FaceOutDist)).rotate((0,1,0),
(0,-1,0),90)
    Face4 = FacePlate().union(FaceTXT(FaceTXTList[3])).translate((0,0,FaceOutDist)).rotate((0,-1,0),
(0,1,0),90)
    Face5 = FacePlate().union(FaceTXT(FaceTXTList[4])).translate((0,0,FaceOutDist))
    Face6 = FacePlate().union(FaceTXT(FaceTXTList[5])).translate((0,0,FaceOutDist)).rotate((0,-1,0),
(0,1,0),180)

    FacesCombo = Face1.union(Face2).union(Face3).union(Face4).union(Face5).union(Face6)
    return(FacesCombo)

############################

D6BombDice = BombSpikes().union(ComboFace())

show_object(D6BombDice)

if RenderYN.lower() == "yes":
    exporters.export(D6BombDice, 'BombDiceD6v0.step')
    exporters.export(D6BombDice, 'BombDiceD6v0.stl')
```

Downloads

And, here's where you can download the CADQuery file. ***BombDiceD6v0.py***

and the bomb dice model in an STL or STEP file formats here;

- **The Bomb Dice - STEP**
- **The Bomb Dice - STL**.

www.techmonkeybusiness.com