An Enhanced Programming Environment for

Generative Design

Guilherme Ferreira
Instituto Superior Técnico

1. Problem
2. Solution
3. Conclusions

1

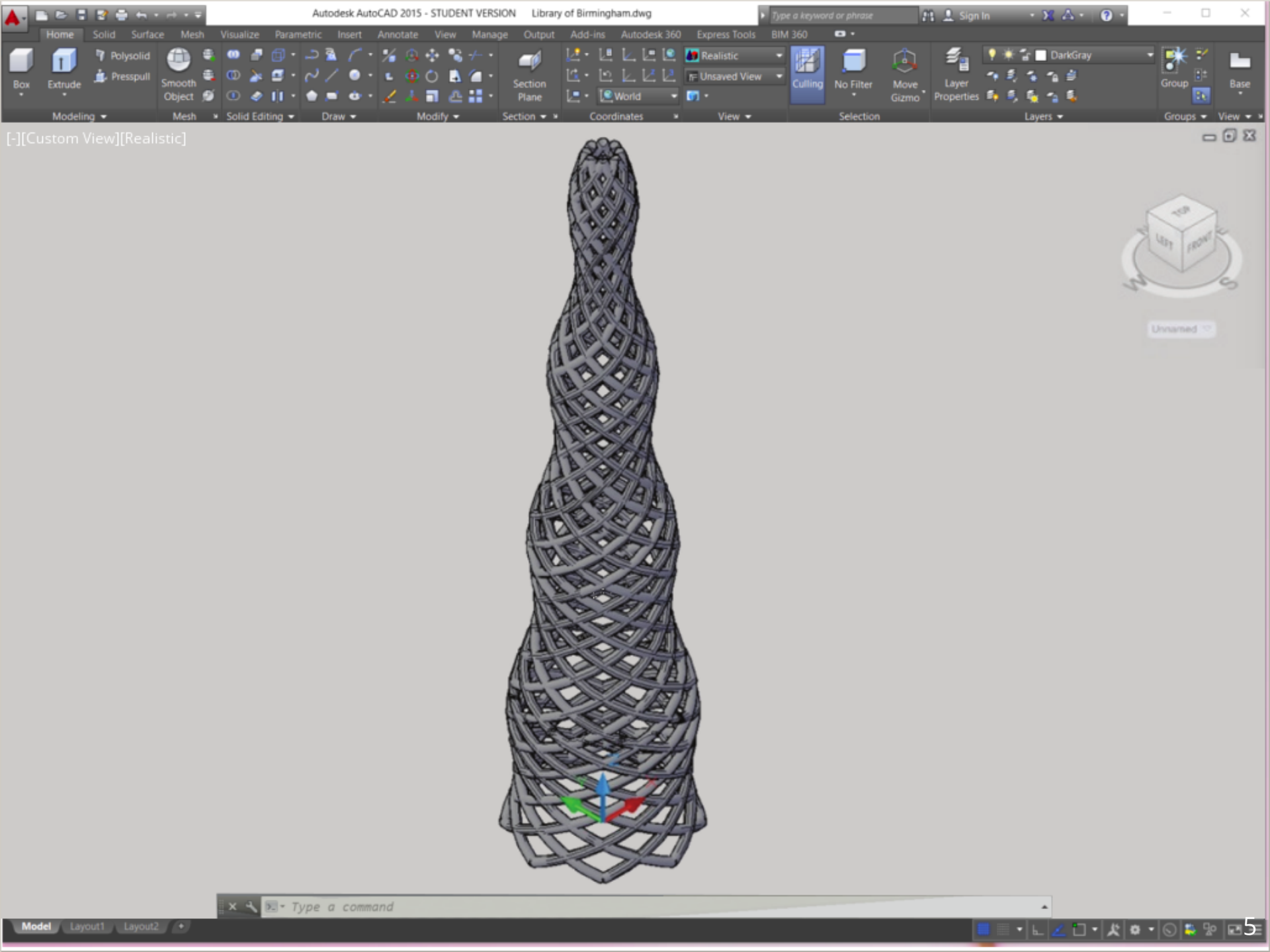
Problem

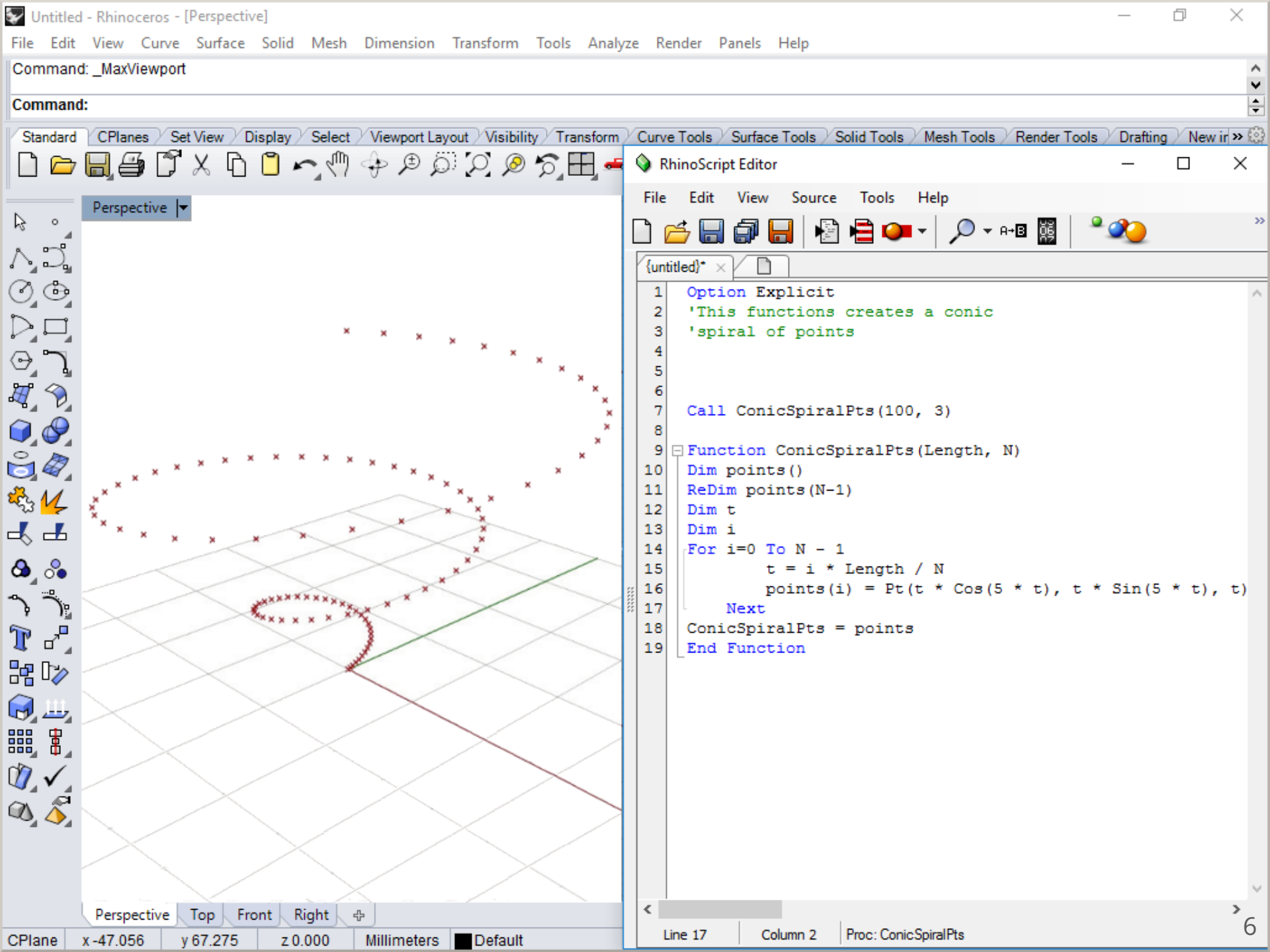
Museum of Tomorrow
Rio de Janeiro, Brazil

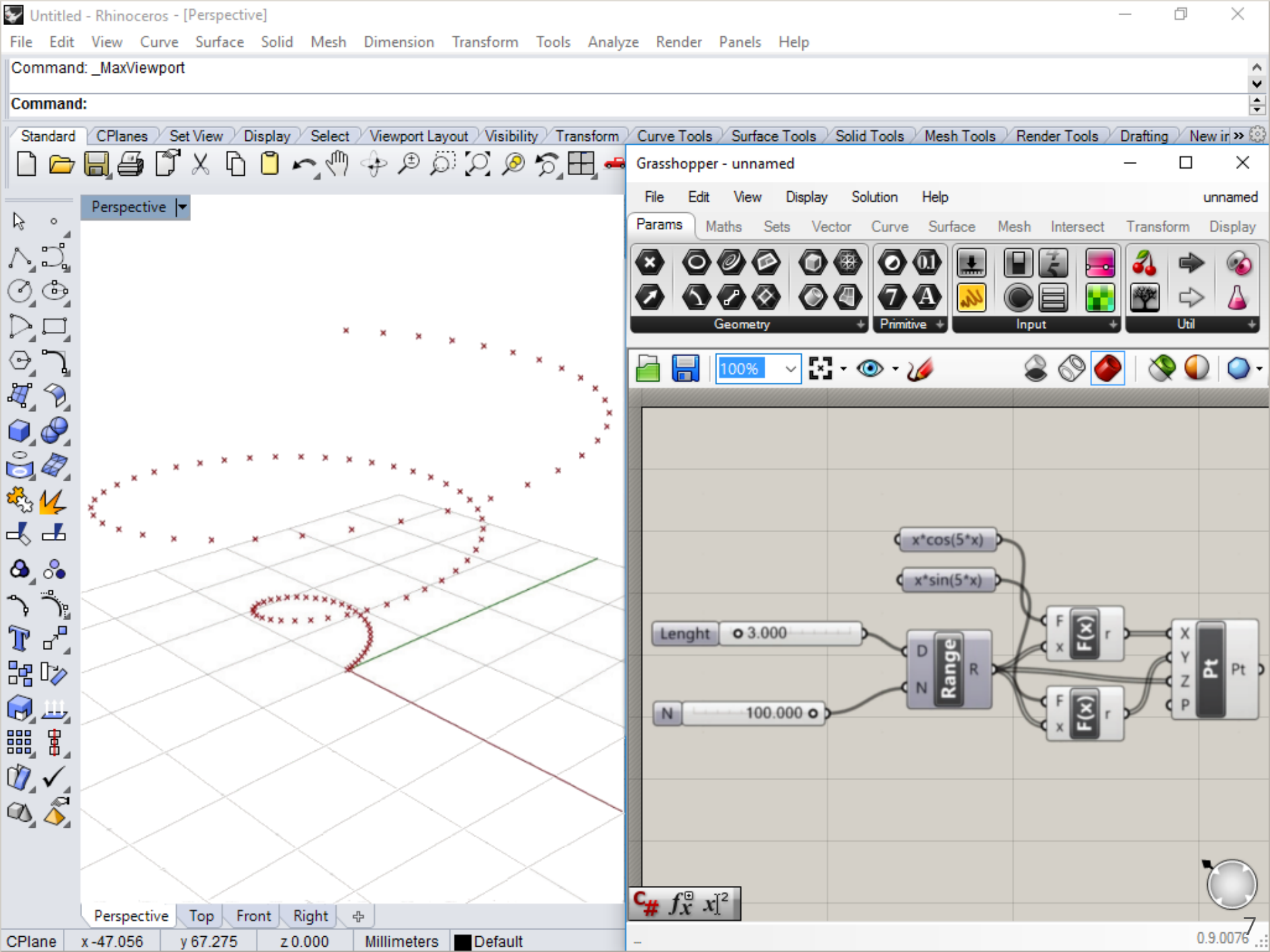
Turning Torso
Malmö, Sweden
(2005)

World Trade Center Transit Hub
New York, USA (2016)

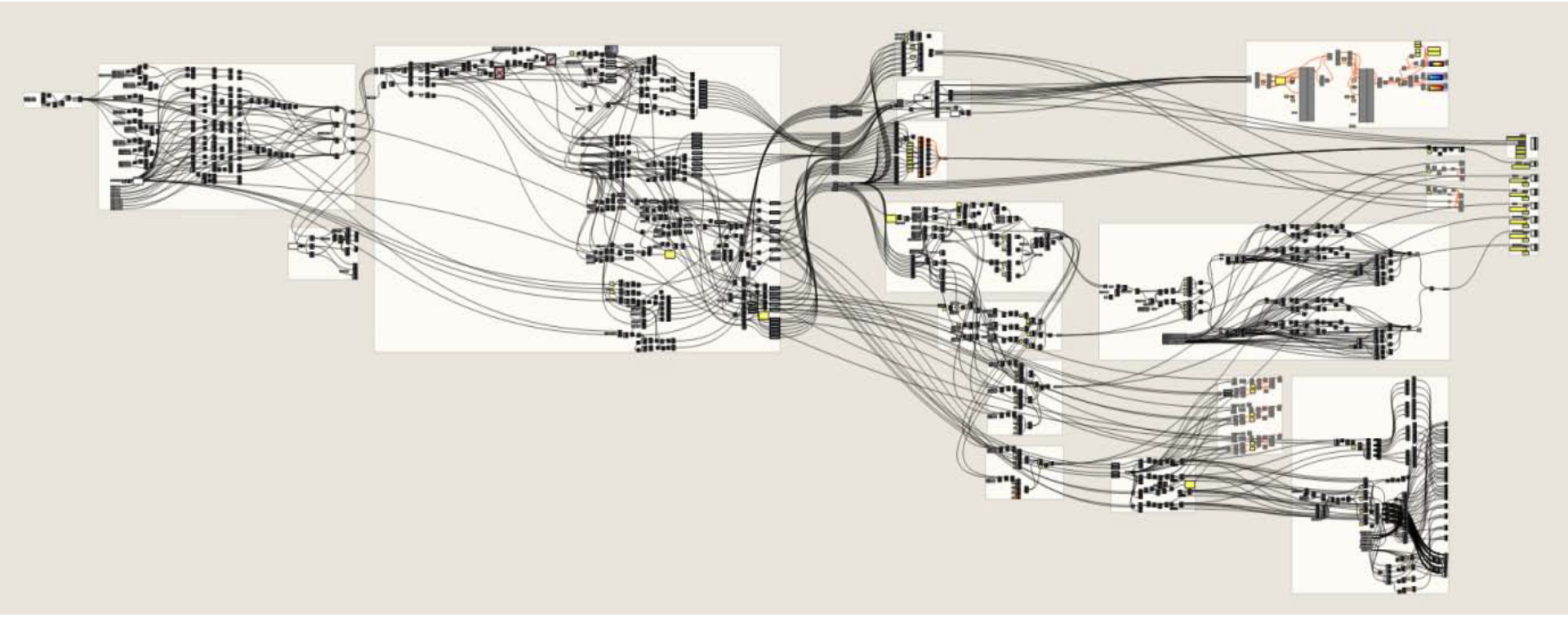
Queen Sofia Palace of the Arts
Valencia, Spain (2005)

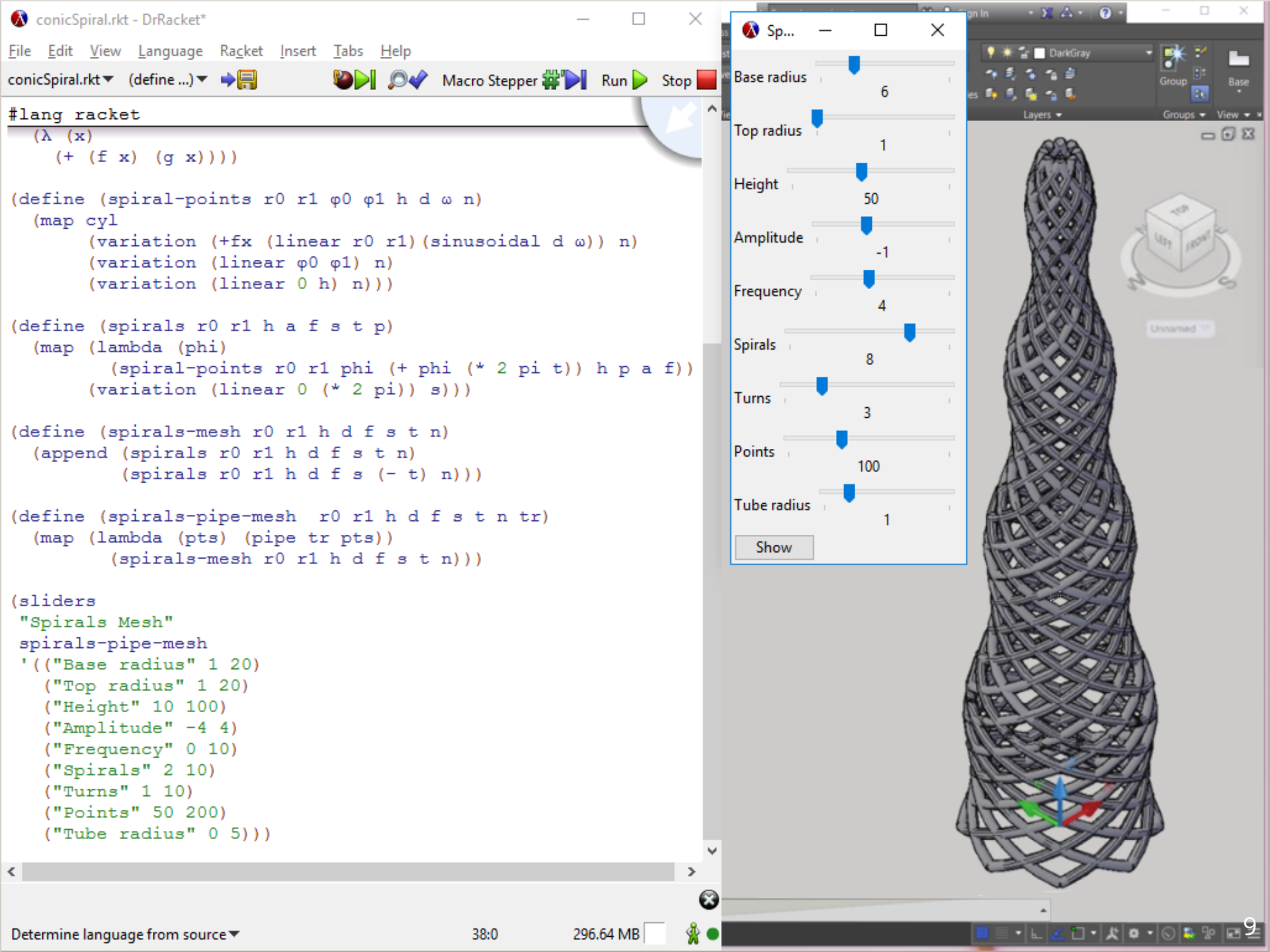

Autodesk AutoCAD 2015 - STUDENT VERSION
Library of Birmingham.dwg
Home
Solid
Surface
Mesh
Visualize
Parametric
Insert
Annotate
View
Manage
Output
Add-ins
Autodesk 360
Express Tools
BIM 360
Box
Extrude
Polysolid
Presspull
Smooth Object
Section Plane
Realistic
Unsaved View
World
Culling
No Filter
Move Gizmo
Layer Properties
DarkGray
Group
Base
Modeling
Mesh
Solid Editing
Draw
Modify
Section
Coordinates
View
Selection
Layers
Groups
[-][Custom View][Realistic]
Type a command
Model
Layout1
Layout2


Untitled - Rhinoceros - [Perspective]
File Edit View Curve Surface Solid Mesh Dimension Transform Tools Analyze Render Panels Help
Command: _MaxViewport
Command:
Standard CPlanes Set View Display Select Viewport Layout Visibility Transform Curve Tools Surface Tools Solid Tools Mesh Tools Render Tools Drafting New in
Perspective
Perspective Top Front Right
CPlane x -47.056 y 67.275 z 0.000 Millimeters Default
RhinoScript Editor
File Edit View Source Tools Help
{untitled}*
1 Option Explicit
2 'This functions creates a conic
3 'spiral of points
4
5
6
7 Call ConicSpiralPts(100, 3)
8
9 Function ConicSpiralPts(Length, N)
10 Dim points()
11 ReDim points(N-1)
12 Dim t
13 Dim i
14 For i=0 To N - 1
15 t = i * Length / N
16 points(i) = Pt(t * Cos(5 * t), t * Sin(5 * t), t)
17 Next
18 ConicSpiralPts = points
19 End Function
Line 17 Column 2 Proc: ConicSpiralPts


Untitled - Rhinoceros - [Perspective]
File Edit View Curve Surface Solid Mesh Dimension Transform Tools Analyze Render Panels Help
Command: _MaxViewport
Command:
Standard CPlanes Set View Display Select Viewport Layout Visibility Transform Curve Tools Surface Tools Solid Tools Mesh Tools Render Tools Drafting
Perspective
Perspective Top Front Right
CPlane x -47.056 y 67.275 z 0.000 Millimeters Default
Grasshopper - unnamed
File Edit View Display Solution Help unnamed
Params Maths Sets Vector Curve Surface Mesh Intersect Transform Display
Geometry Primitive Input Util
100%
x*cos(5*x)
x*sin(5*x)
Lenght 3.000
N 100.000
D N Range R
F x F(x) r
X Y Z P Pt Pt
0.9.0076

How can I change this program?




conicSpiral.rkt - DrRacket*
File Edit View Language Racket Insert Tabs Help
conicSpiral.rkt (define ...) Macro Stepper Run Stop
#lang racket
(λ (x)
(+ (f x) (g x))))
(define (spiral-points r0 r1 φ0 φ1 h d ω n)
(map cyl
(variation (+fx (linear r0 r1)(sinusoidal d ω)) n)
(variation (linear φ0 φ1) n)
(variation (linear 0 h) n)))
(define (spirals r0 r1 h a f s t p)
(map (lambda (phi)
(spiral-points r0 r1 phi (+ phi (* 2 pi t)) h p a f))
(variation (linear 0 (* 2 pi)) s)))
(define (spirals-mesh r0 r1 h d f s t n)
(append (spirals r0 r1 h d f s t n)
(spirals r0 r1 h d f s (- t) n)))
(define (spirals-pipe-mesh r0 r1 h d f s t n tr)
(map (lambda (pts) (pipe tr pts))
(spirals-mesh r0 r1 h d f s t n)))
(sliders
"Spirals Mesh"
spirals-pipe-mesh
'(("Base radius" 1 20)
("Top radius" 1 20)
("Height" 10 100)
("Amplitude" -4 4)
("Frequency" 0 10)
("Spirals" 2 10)
("Turns" 1 10)
("Points" 50 200)
("Tube radius" 0 5)))
Determine language from source
38:0
296.64 MB
Sp...
Base radius
6
Top radius
1
Height
50
Amplitude
-1
Frequency
4
Spirals
8
Turns
3
Points
100
Tube radius
1
Show

What is this program doing?

```
#lang python
from "racket" import sqrt, min, cos, sin, pi
from "rosetta" import *
from xyz import XYZ

trussKnotRadius = 0.2
trussKnot = sphere

def trussKnots(cs, radius):
  return [trussKnot(c, radius) for c in cs]

trussBarRadius = 0.07
trussBar = cylinder

def trussBars(cs1, cs2, radius):
  return map (lambda c1, c2: trussBar(c1, radius, c2),
              cs1,
              cs2)

def spatialTruss(curves, knotRadius, barRadius):
  a = curves[0]
  b = curves[1]
  c = curves[2]
  trussKnots(a, knotRadius)
  trussKnots(b, knotRadius)
  trussBars(a, c, barRadius)
  trussBars(b, a[:-1], barRadius)
  trussBars(b, c[:-1], barRadius)
  trussBars(b, a[1:], barRadius)
  trussBars(b, c[1:], barRadius)
  trussBars(a[1:], a[:-1], barRadius)
  trussBars(b[1:], b[:-1], barRadius)
  if len(curves)==3:
    trussKnots(c, knotRadius)
    trussBars(c[1:], c[:-1], barRadius)
  else:
    trussBars(b, curves[3], barRadius)
    spatialTruss(curves[2:], knotRadius, barRadius)

# utils

def crossProduct(c1, c2):
  return xyz((c1.y-c2.y)*(c1.z+c2.z),
             (c1.z-c2.z)*(c1.x+c2.x),
             (c1.x-c2.x)*(c1.y+c2.y))

def crossProducts(cs):
  if len(cs) == 1:
    return u0()
  else:
    return crossProduct(cs[0], cs[1])+crossProducts(cs[1:])

def polygonNormal(cs):
  return norm_c(crossProducts(cs + [cs[0]]) * -1)

def midcoord(p0, p1):
  return (p0+p1)/2
```

```
def quadCenter(c1, c2, c3, c4):
  return midcoord(midcoord(c1, c3), midcoord(c2, c4))

def quadNormal(c1, c2, c3, c4):
  return polygonNormal([c1, c2, c3, c4])

def quadPyramidApex(c1, c2, c3, c4):
  h = (distance(c1, c2)+distance(c2, c3)+distance(c3, c4)+distance(c4, c1))/4/sqrt(2)
  return quadCenter(c1, c2, c3, c4)+quadNormal(c1, c2, c3, c4)*h

def insertPyramidApex2Curves(cs1, cs2):
  l = [quadPyramidApex(cs1[0], cs2[0], cs2[1], cs1[1])]
  if len(cs1)==2:
    return l
  else:
    return l+insertPyramidApex2Curves(cs1[1:], cs2[1:])

def insertPyramidApexCurves(curves):
  if len(curves) == 1:
    return curves
  else:
    return curves[:1]+\
           [insertPyramidApex2Curves(curves[0], curves[1])]+\
           insertPyramidApexCurves(curves[1:])

def spatialTrussInsertApex(cs):
  c1 = cs[0][0]
  c2 = cs[1][0]
  c4 = cs[0][1]
  d = min(distance(c1, c2), distance(c1, c4))
  knotRadius = d/5
  barRadius = d/15
  return spatialTruss(insertPyramidApexCurves(cs), knotRadius, barRadius)

# moebius truss

def linspace(start, stop, n):
    n = n + 1
    L = [0.0] * n
    nm1inv = 1.0 / (n-1)
    for i in range(n):
        L[i] = nm1inv * (start*(n - i) + stop*i)
    return L


def moebiusCs(r, u1, u2, m, v1, v2, n):
  return [[cyl(r*(1+v*cos(u/2)), u, r*(1+v*sin(u/2)))
           for v in linspace(v1,v2,n)]
          for u in linspace(u1,u2,m)]

def moebiusTruss(r, u1, u2, m, v1, v2, n):
  return spatialTrussInsertApex(moebiusCs(r, u1, u2, m, v1, v2, n))


def moebiusTrussExample():
   return moebiusTruss(1, 0, pi*4, 80, 0, 0.3, 1)
```

2

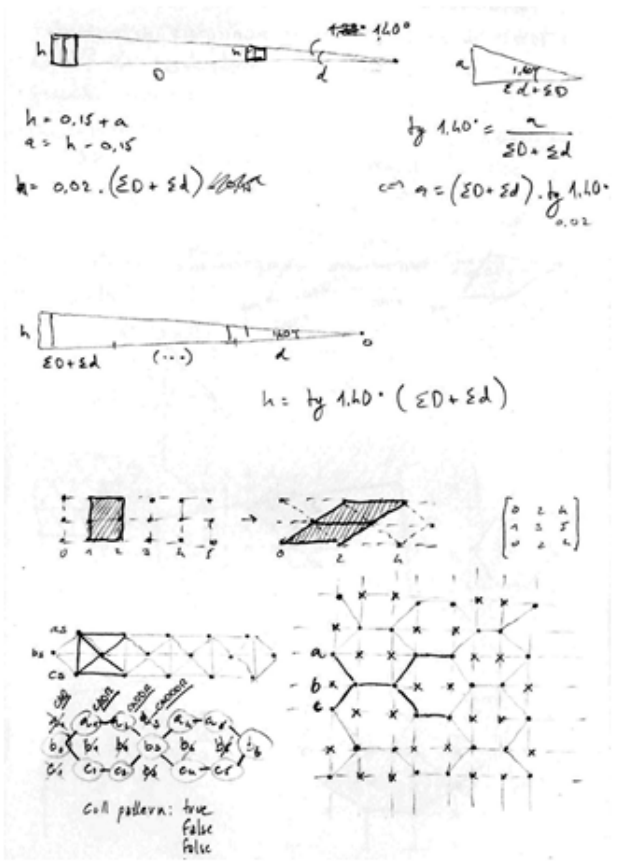
Solution

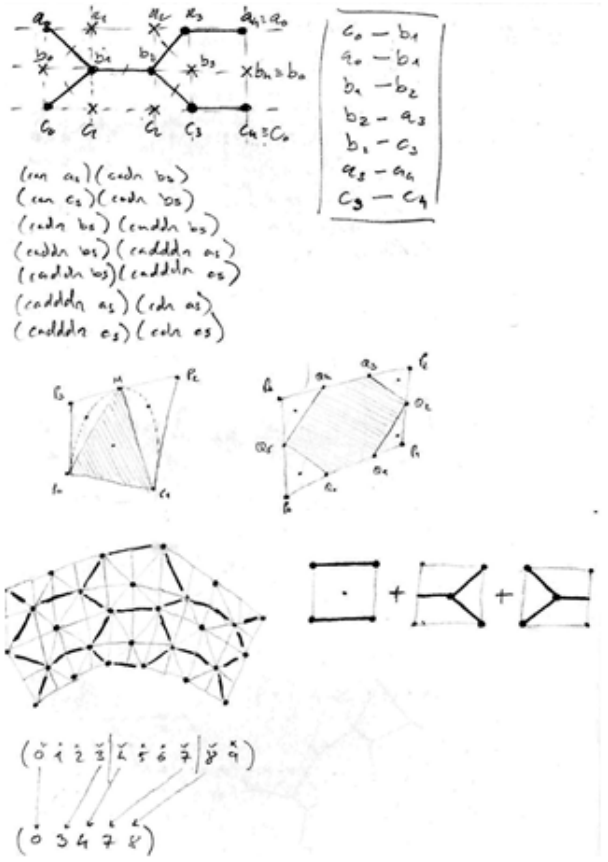
CONCEPT

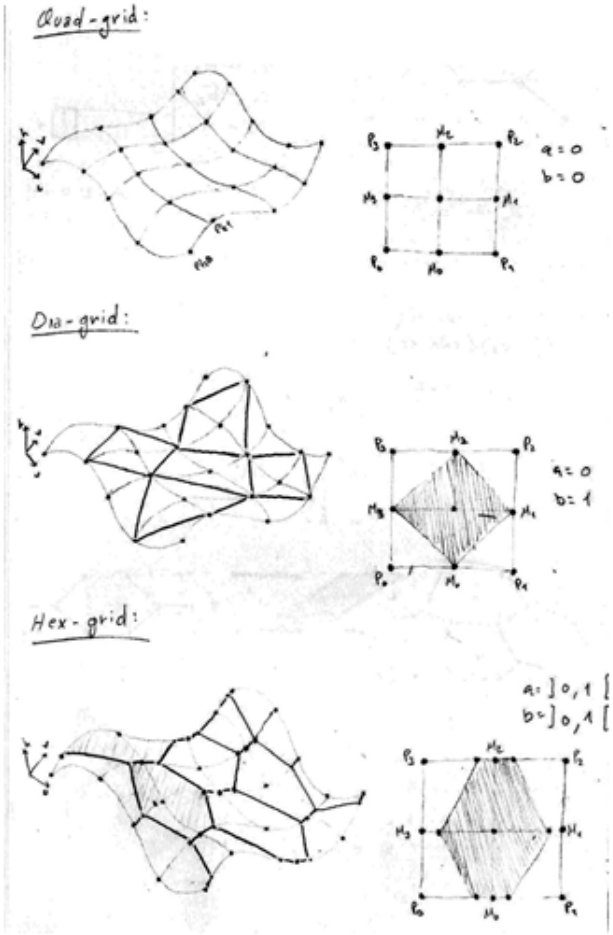
Sketches
Diagrams

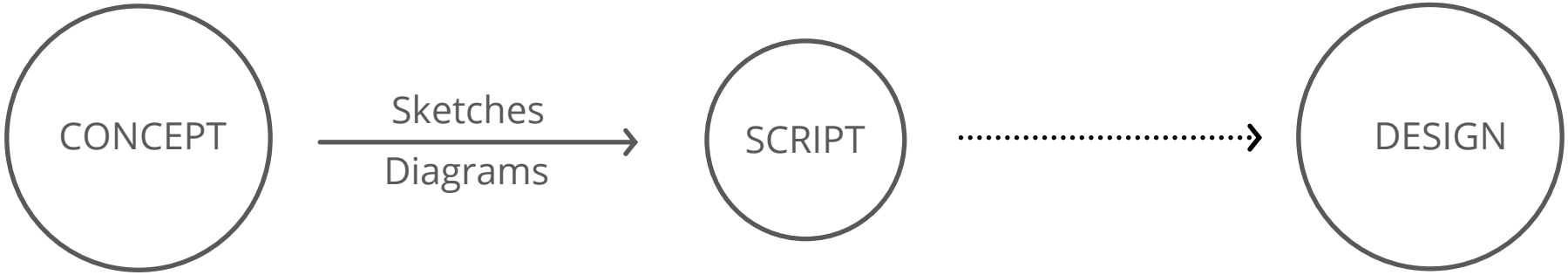
SCRIPT

DESIGN









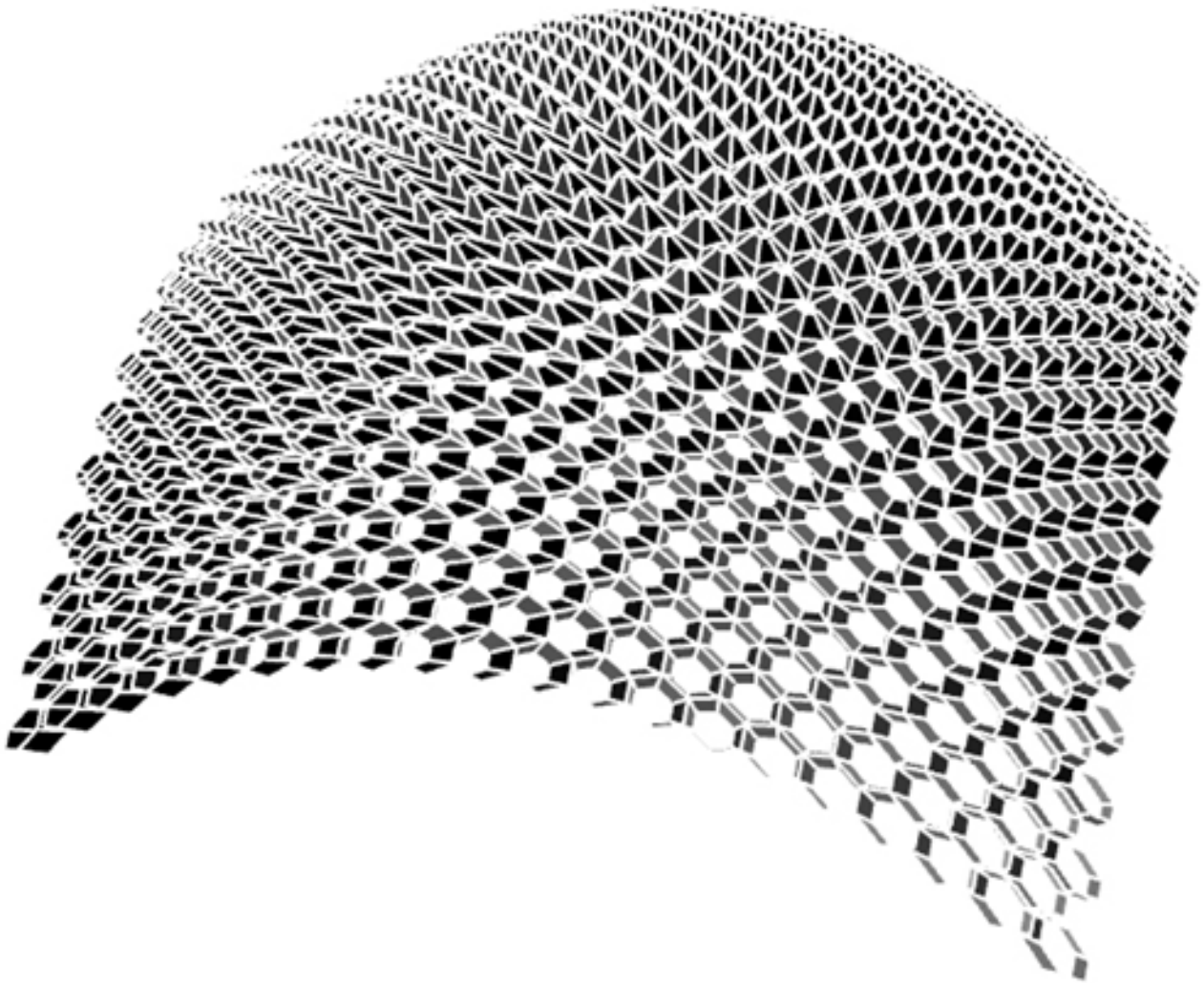

```
(define (skin-open-srf srf panel panel-f panel-h att)
  (sphere att 0.03)
  (iterate-quads
    (lambda (h0 h1 h2 h3)
      (panel (list-ref h0 4)
             (list-ref h0 3)
             (list-ref h1 4)
             (list-ref h2 1)
             (list-ref h2 0)
             (list-ref h3 1)
            panel-f panel-h att))
    (iterate-quads
      (lambda (p0 p1 p2 p3)
        (let* ((a (ptf p0 p3 0.17))
               (b (ptf p0 p1 1/2))
               (f (ptf p0 p3 1/2))
               (j (ptf p3 p2 1/2))
               (e (ptf b j 2/3))
               (k p1)
               (l p2)
               (c (ptf k l 0.17))
               (d (ptf k l 1/2))
               (w (ptf k l 2/3))
               (g (ptf (ptf p0 p3 2/3) (ptf p1 p2 2/3) 3/2))
               (h (ptf p3 p2 3/2))
               (i (ptf k l 1.17)))
          (panel a b c d e f panel-f panel-h att)
          (list a b c d e f)))
      (transpose-matrix srf))))

(define (hex-panel-1 p0 p1 p2 p3 p4 p5 f h att)
  (let* ((c (quad-center p0 p1 p2 p3))
         (n (quad-normal p0 p1 p2 p3))
         (d (* f (distance att c))))
    (loft (list
     (closed-line p0 p1 p2 p3 p4 p5)
     (move
      (scale
       (closed-line p0 p1 p2 p3 p4 p5) d c)
       (*c n (- h)))))))

(define (hex-panel-2 p0 p1 p2 p3 p4 p5 f h att)
  (let* ((c (quad-center p0 p1 p2 p3))
         (n (quad-normal p0 p1 p2 p3))
         (d (* f (distance att c))))
    (loft (list
     (closed-line p0 p1 p2 p3 p4 p5)
     (move
      (scale
       (closed-line p0 p1 p2 p3 p4 p5)
       (/ d 1.3) c)
      (*c n (/ (- h) 2.0)))
     (move
      (scale
       (closed-line p0 p1 p2 p3 p4 p5) d c)
       (*c n (- h)))))))
```

CONCEPT

Sketches
Diagrams

SCRIPT

DESIGN



Literate Programming

To combine evidences, we cannot simply ``add'' them. We must use a formula that maintains the combined evidence within the interval $[-1, 1]$. Again, we refer to Shortliffe to justify the following combination function C:

$$C(x, y) = \begin{cases} x + y - xy & \text{if } x > 0 \text{ and } y > 0, \\ x + y + xy & \text{if } x < 0 \text{ and } y < 0, \\ \dfrac{x+y}{1-\min(|x|,|y|)} & \text{otherwise.} \end{cases}$$

The parameters of the function are two certainty factors that we want to combine. The function combines evidences in such a way that combining a true evidence with something else except false is true, combining a false evidence with something else except true is false, combining unknown with something else won't change the result and combining a true evidence with a false evidence is an error as it would be equivalent to a contradiction. When two evidences are *for* something, that is, they are both positive, its combination is an even stronger evidence for that thing. When two evidences are *against* something, that is, they are both negative, its combination is an even stronger evidence against that thing. When one of them is positive and the other is negative, its combination will be something in between.

Documentation

```
(defun combine-evidence (cf1 cf2)
  (cond ((and (> cf1 0) (> cf2 0))
         (+ cf1 cf2 (* -1 cf1 cf2)))
        ((and (< cf1 0) (< cf2 0))
         (+ cf1 cf2 (* cf1 cf2)))
        (t
         (/ (+ cf1 cf2) (- 1 (min (abs cf1) (abs cf2)))))))
```

Program

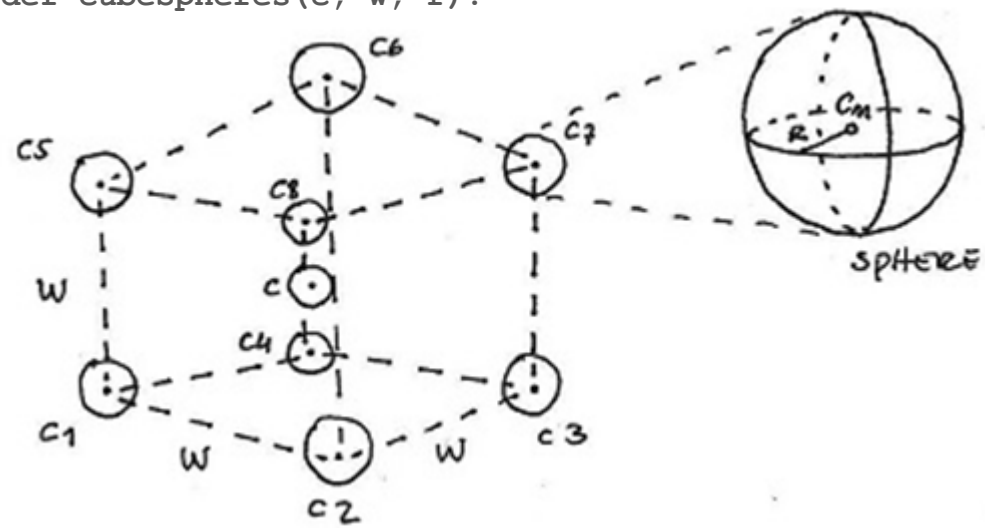
While comparing projects, this function is called *zillions* of times. A simple time and space profiler of the project revealed that this function is absolutely critical, which suggested us that it should be improved. The problem is essentially due to excessive boxing and unboxing of floating-point values. To solve this, we must advise the compiler to use floating-point arithmetic and to avoid some repeated tests about the signal of the arguments.

Documentation

Program-sketch Correlation

Program

```
def cubeSpheres(c, w, r):
```



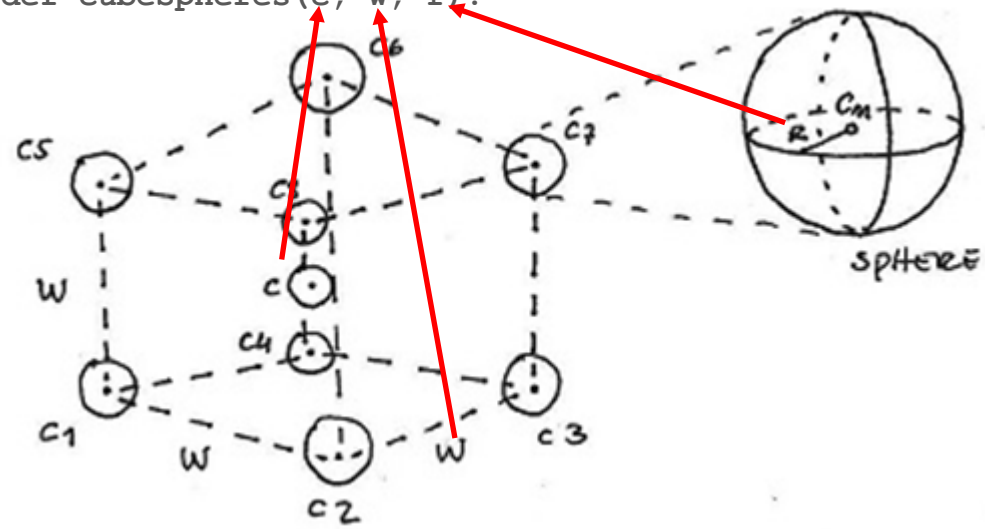

```
  c1 = c+xyz(-w,-w,-w)
  c2 = c+xyz(+w,-w,-w)
  c3 = c+xyz(+w,+w,-w)
  c4 = c+xyz(-w,+w,-w)
  c5 = c+xyz(-w,-w,+w)
  c6 = c+xyz(+w,-w,+w)
  c7 = c+xyz(+w,+w,+w)
  c8 = c+xyz(-w,+w,+w)
  for p in [c1,c2,c3,c4,c5,c6,c7,c8]:
    sphere(p, r)
```

Documentation

Program-sketch Correlation

Program

```
def cubeSpheres(c, w, r):
```

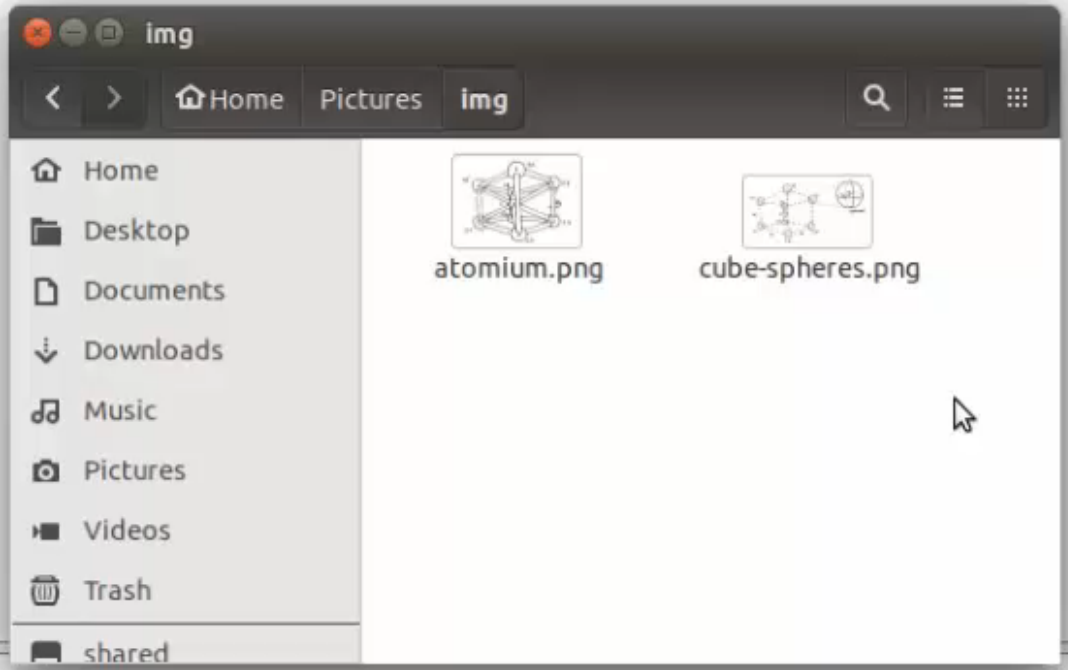



```
  c1 = c+xyz(-w,-w,-w)
  c2 = c+xyz(+w,-w,-w)
  c3 = c+xyz(+w,+w,-w)
  c4 = c+xyz(-w,+w,-w)
  c5 = c+xyz(-w,-w,+w)
  c6 = c+xyz(+w,-w,+w)
  c7 = c+xyz(+w,+w,+w)
  c8 = c+xyz(-w,+w,+w)
  for p in [c1,c2,c3,c4,c5,c6,c7,c8]:
    sphere(p, r)
```

Documentation

Untitled (define ...) Debug Auto Run Check Syntax Macro Stepper Run Stop

```
1 #lang racket
2
```



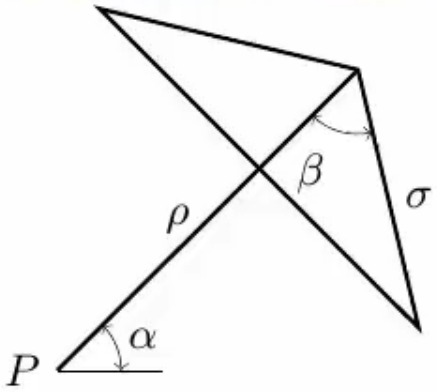
>

Determine language from source 3:2 325.03 MB

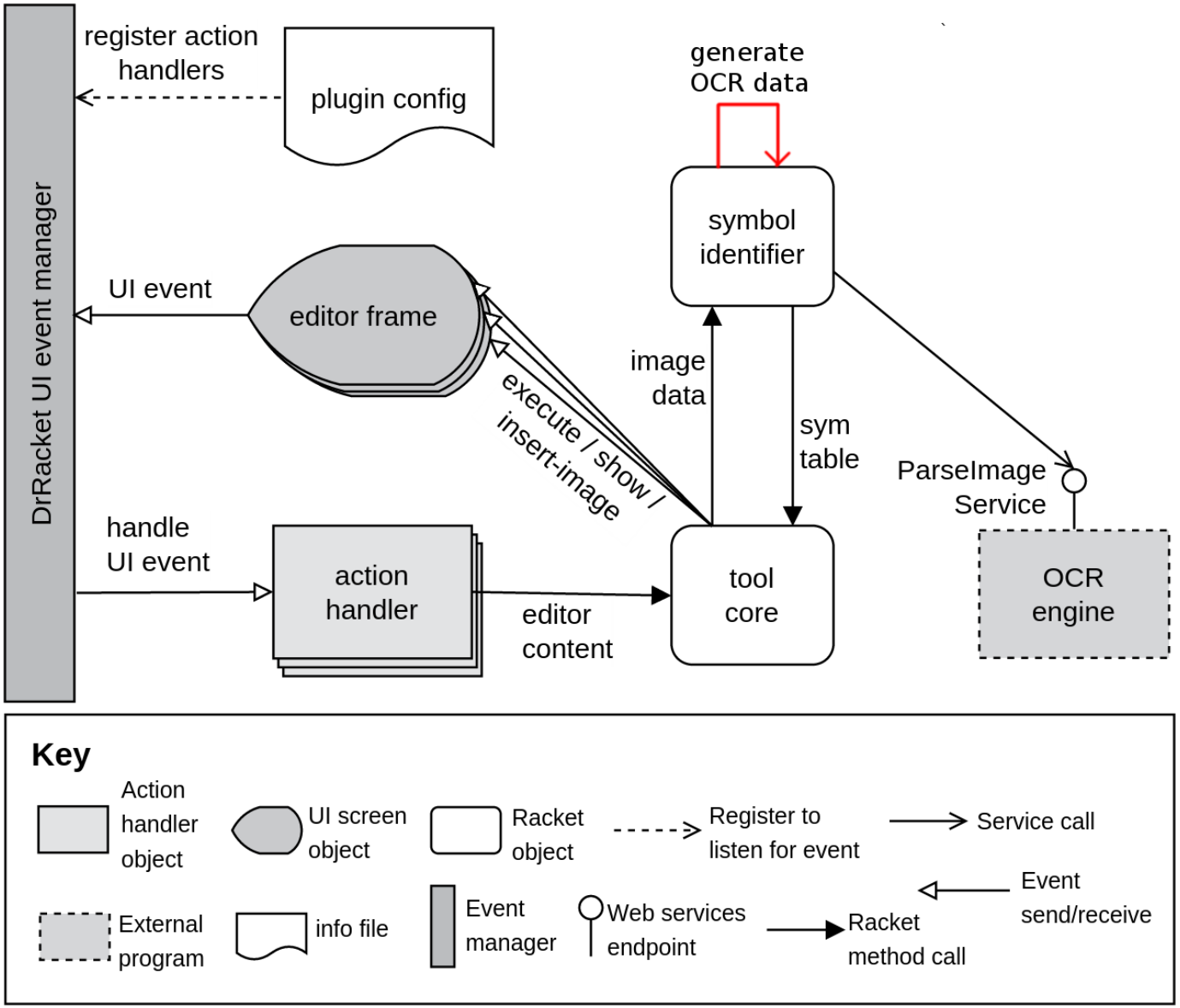
Immediate feedback

arrow.rkt (define ...) Run Stop

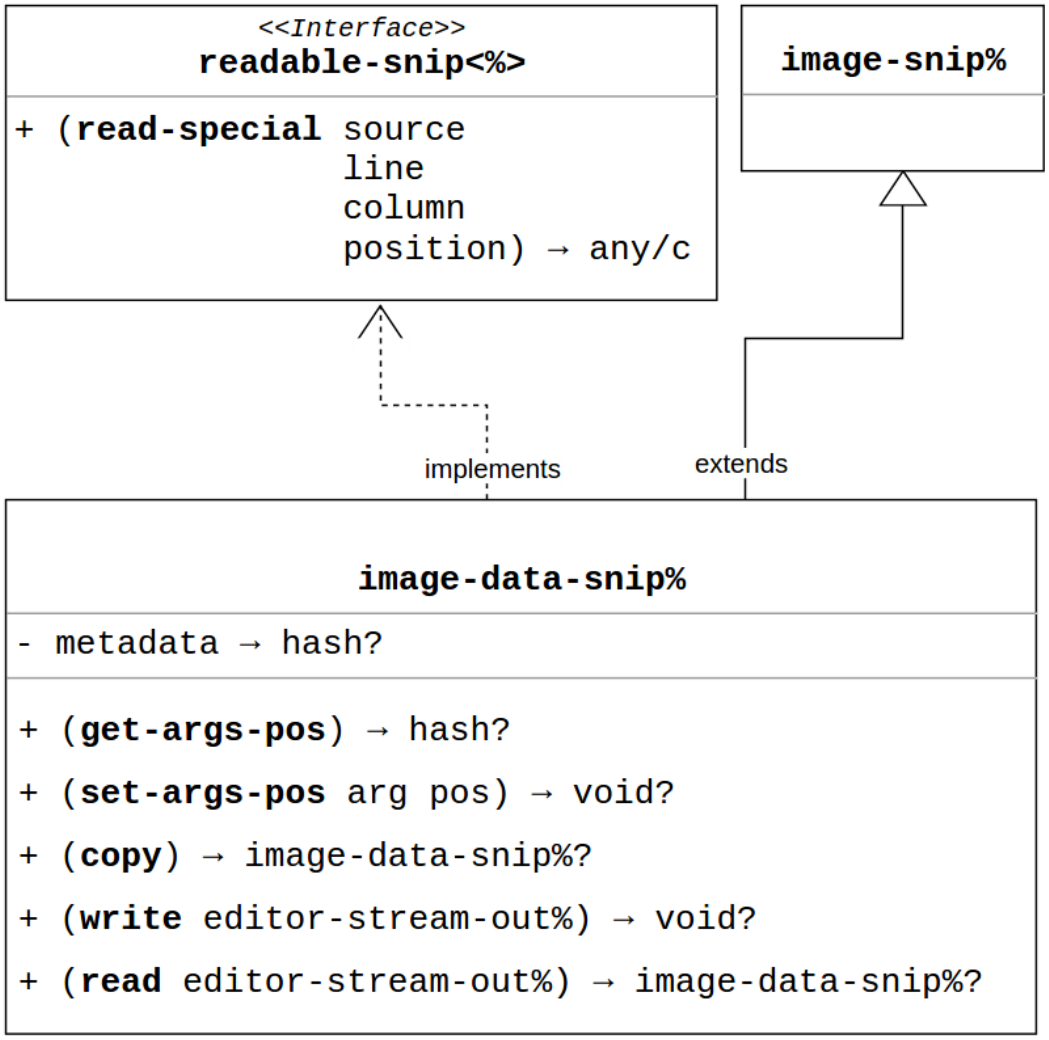
```
1  #lang racket
2  (require (planet aml/rosetta)
3           "tikz/tikz.rkt"
4           image-data-snip)
5
6  (define (arrow p alpha rho beta sigma)
7
8    (line p
9          (+pol p rho alpha)
10         (+pol (+pol p rho alpha) sigma (+ alp
11         (+pol (+pol p rho alpha) sigma (+ alp
12         (+pol p rho alpha)))
13
14 (arrow (xy 0 0) (* 1 pi/4) 3 (/ pi 3) 2)
15
16
17
```



How to implement these tools?



How to implement these tools?



Evaluation

Users feedback

- Awkward shortcuts
- Resize images
- Adaptive sliders

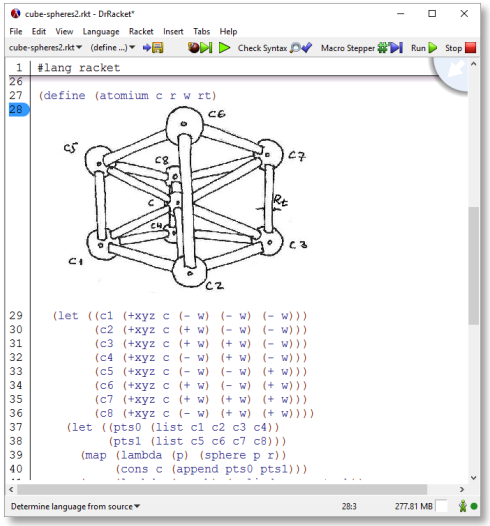
Limitations

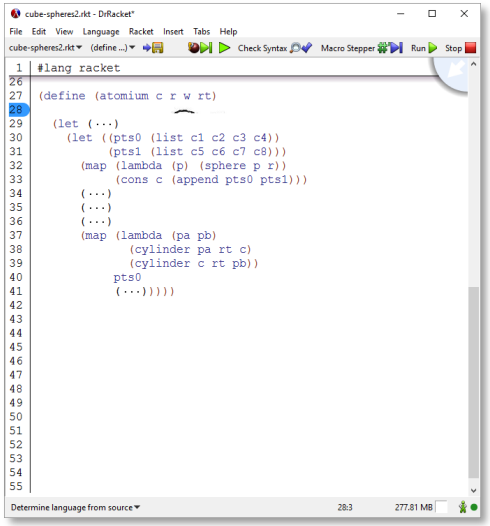
- No resize operation for bitmaps
- DrRacket slider GUI has fixed range
- DrRacket stores code and metadata as binary

3

Conclusions

Program-sketch Correlation Tool





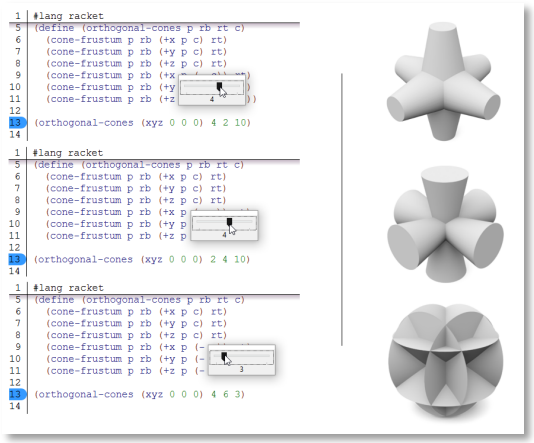
Current Work

- Correlates code with program artifacts
- Saves image and its metadata
- Provides useful operations over images

Future Work

- New correlation strategies
- OCR integration
- Support for image editing

Immediate Feedback Tool

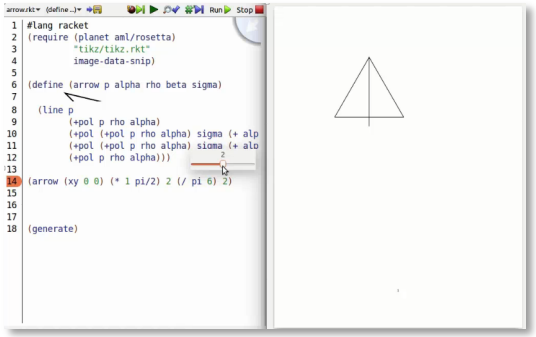


Current Work

- Processes immediately the code change
- Provides inline slider
- Supports change in any literal

Future Work

- Partial code execution
- Adaptive sliders
- Show model change



Atrium of Brookfield Place
Toronto, Canada (1992)

Ysios Winery
Laguardia, Spain (2001)

Thank you.
Questions?

Barajas Airport
Madrid, Spain