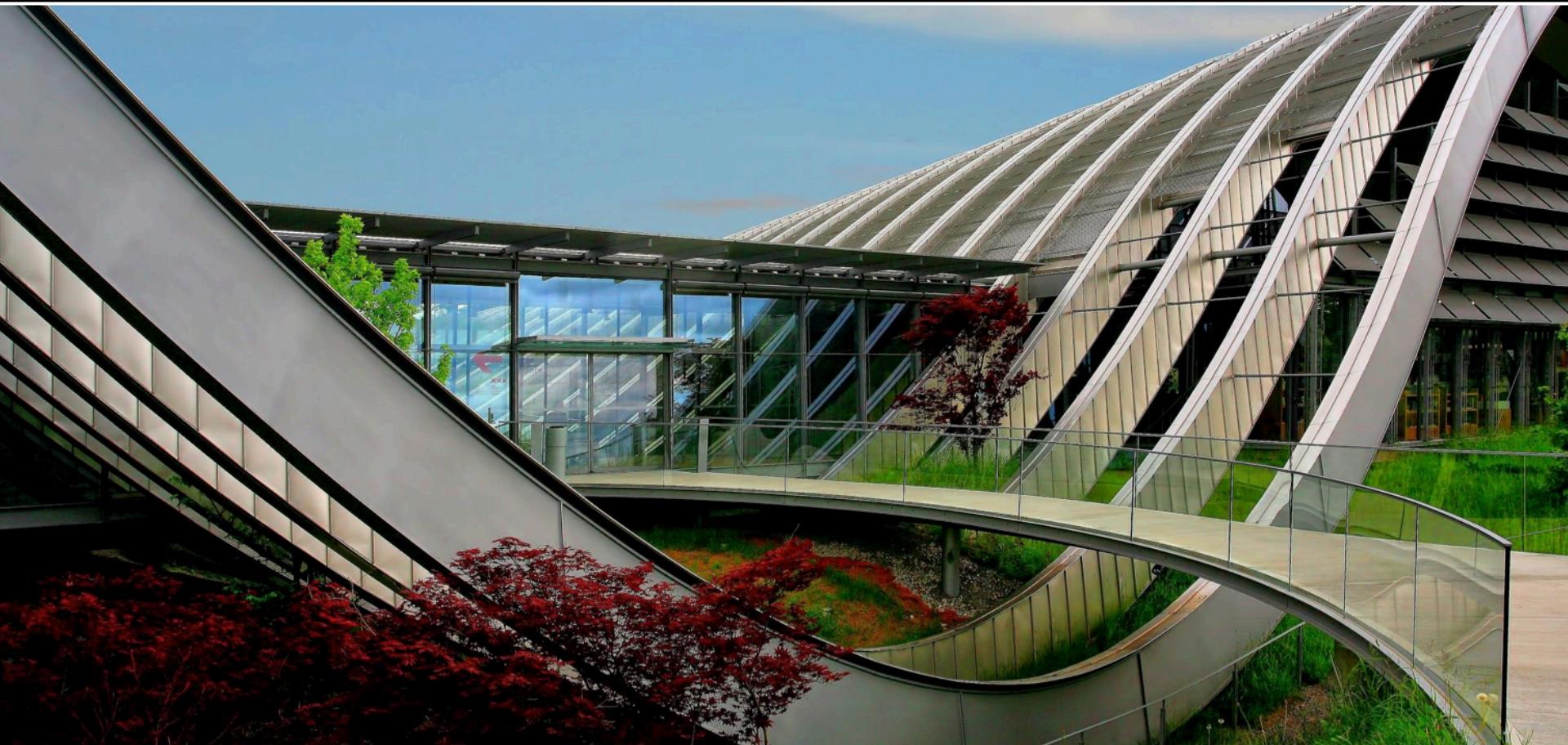




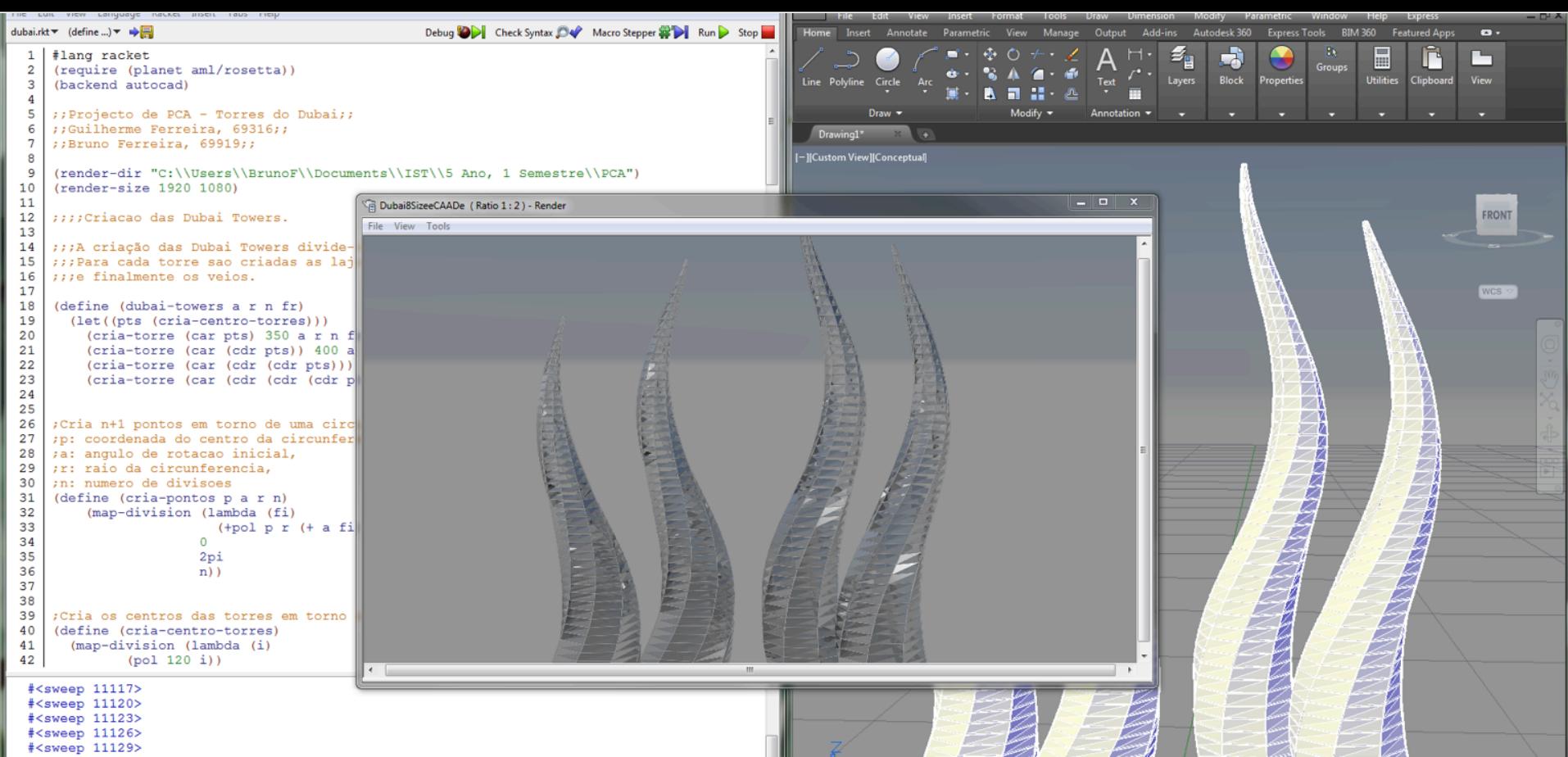
Context-Oriented Algorithmic Design

Bruno Ferreira
António Menezes Leitão

Algorithmic Design



CAD Tools



The image shows a CAD environment with two main windows. The left window is a text editor displaying Racket code for generating the Burj Khalifa. The right window is a 3D rendering window showing the Burj Khalifa's spires.

```
#lang racket
(require (planet aml/rosetta))
(backend autocad)

;; Projeto de PCA - Torres do Dubai;;
;; Guilherme Ferreira, 69316;;
;; Bruno Ferreira, 69919;;
;; (render-dir "C:\\Users\\BrunoF\\Documents\\IST\\5 Ano, 1 Semestre\\PCA")
;; (render-size 1920 1080)
;; ;;; Criação das Dubai Towers.

;; A criação das Dubai Towers divide-
;; Para cada torre são criadas as laj
;; e finalmente os veios.

(define (dubai-towers a r n f)
  (let((pts (cria-centro-torres)))
    (cria-torre (car pts) 350 a r n f)
    (cria-torre (car (cdr pts)) 400 a
    (cria-torre (car (cdr (cdr pts)))
    (cria-torre (car (cdr (cdr (cdr p
24
25
26 ;Cria n+1 pontos em torno de uma circ
;p: coordenada do centro da circunfer
;a: angulo de rotacao inicial,
;r: raio da circunferencia,
;n: numero de divisoes
(define (cria-pontos p a r n)
  (map-division (lambda (fi)
    (+pol p r (+ a fi
0
2pi
n)))
37
38
39 ;Cria os centros das torres em torno
40 (define (cria-centro-torres)
41   (map-division (lambda (i)
42     (pol 120 i))

#<sweep 11117>
#<sweep 11120>
#<sweep 11123>
#<sweep 11126>
#<sweep 11129>
```

BIM Tools

beamTest.rkt - DrRacket

File Edit View Language Racket Insert Tabs Help

beamTest.rkt (define ...) Debug Check Syntax Macro Stepper Run Stop

```

1 #lang racket
8 (define beam-family-1 (load-family "C:\\ProgramData\\Autodesk")
9
10 (define beam-family-2 (load-family "C:\\ProgramData\\Autodesk"
11
12 (define beam-family-3 (load-family "C:\\ProgramData\\Autodesk"
13
14
15 (define beam-family-element-1
16   (family-element beam-family-1))
17
18 (define beam-family-element-2
19   (family-element beam-family-2))
20
21 (define beam-family-element-3
22   (family-element beam-family-3))
23
24 (create-beam (xyz 0 0 0)
25               (xyz 0 10 0))
26
27 (create-beam (xyz 5 0 0)
28               (xyz 5 10 0)
29               #:family beam-family-element-1)

(idstrc '#hasheqv() """ 266712)
(idstrc '#hasheqv() """ 266719)
(idstrc '#hasheqv() """ 266806)
>

```

Determine language from source ▾

8:2 623.56 MB

Project1.0001 ... Type a keyword or phrase

Architecture Structure Systems Insert Annotate Analyze Massing & Site Collaborate View Manage Add-Ins Modify

Modify View Templates Render Render in Cloud Render Gallery Select Graphics Create Sheet Composition Windows Switch Windows Close Hidden User Interface

Properties

3D View: [3D] Edit Type

Graphics

View Scale: 1:100

Scale Value: 1:100

Detail Level: Medium

Parts Visibility: Show Original

Visibility/Graph... Edit...

Graphic Disp... Edit...

Discipline: Coordination

Show Hidden... By Discipline

Default Analysis: None

Sun Path

Identifier Tools Properties help Apply

Project Browser - Project1.0001

- [C:] Views (all)
 - Floor Plans
 - Level 1
 - Site
 - T.O. Fnd. Wall
 - T.O. Footing
 - T.O. Slab
 - Ceiling Plans
 - Level 1
 - 3D Views
 - Elevations (Building Elevation)
 - East
 - North
 - South
 - West
 - Legends
 - Schedules/Quantities

Click to select, TAB for alternates, CTRL adds, SHIFT unselects.

The image shows the AutoCAD software interface. On the left, there is a code editor window for DrRacket containing Racket code for creating beams. On the right, the main AutoCAD workspace displays three 3D models of structural beams: a simple I-beam, a channel-shaped beam, and a beam with circular holes. A 3D view ribbon is at the top, and a properties palette is open on the left side of the workspace. A project browser window is also visible.

BIM Tools

1_parts.rkt - DrRacket

File Edit View Language Racket Insert Tabs Help

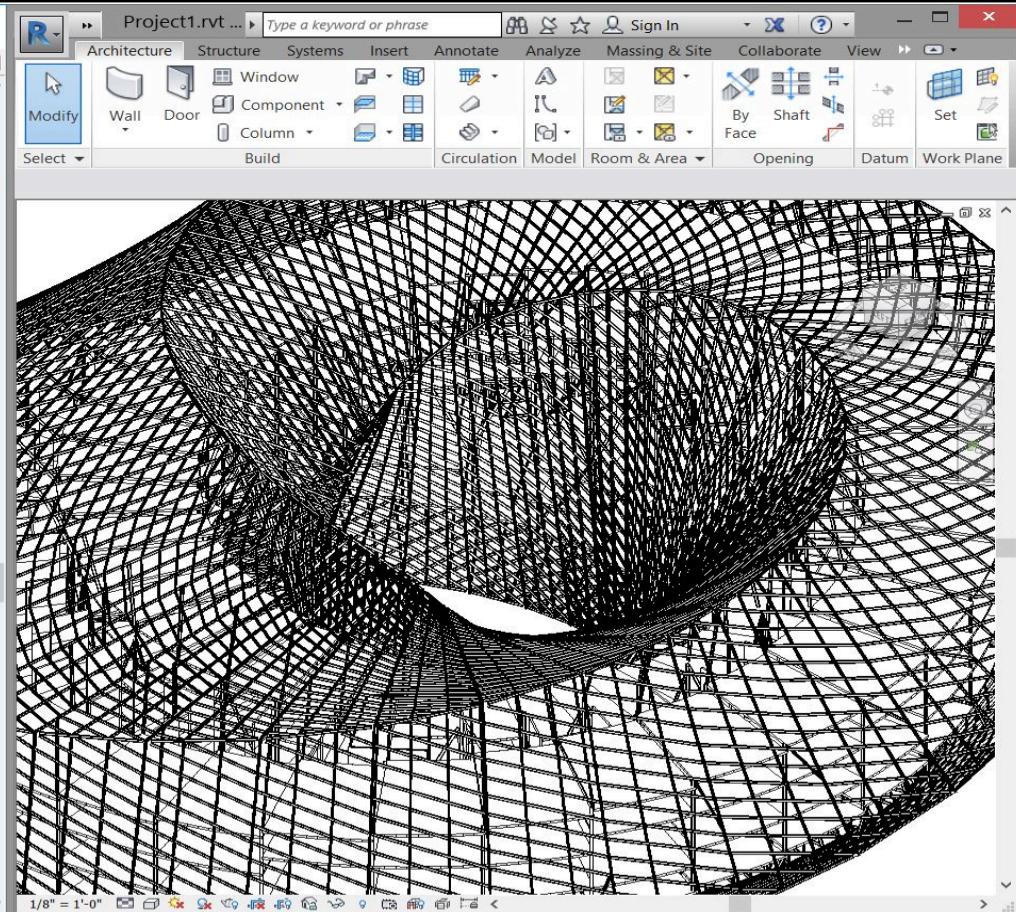
1_parts.rkt (define ...) ▾

1 #lang racket

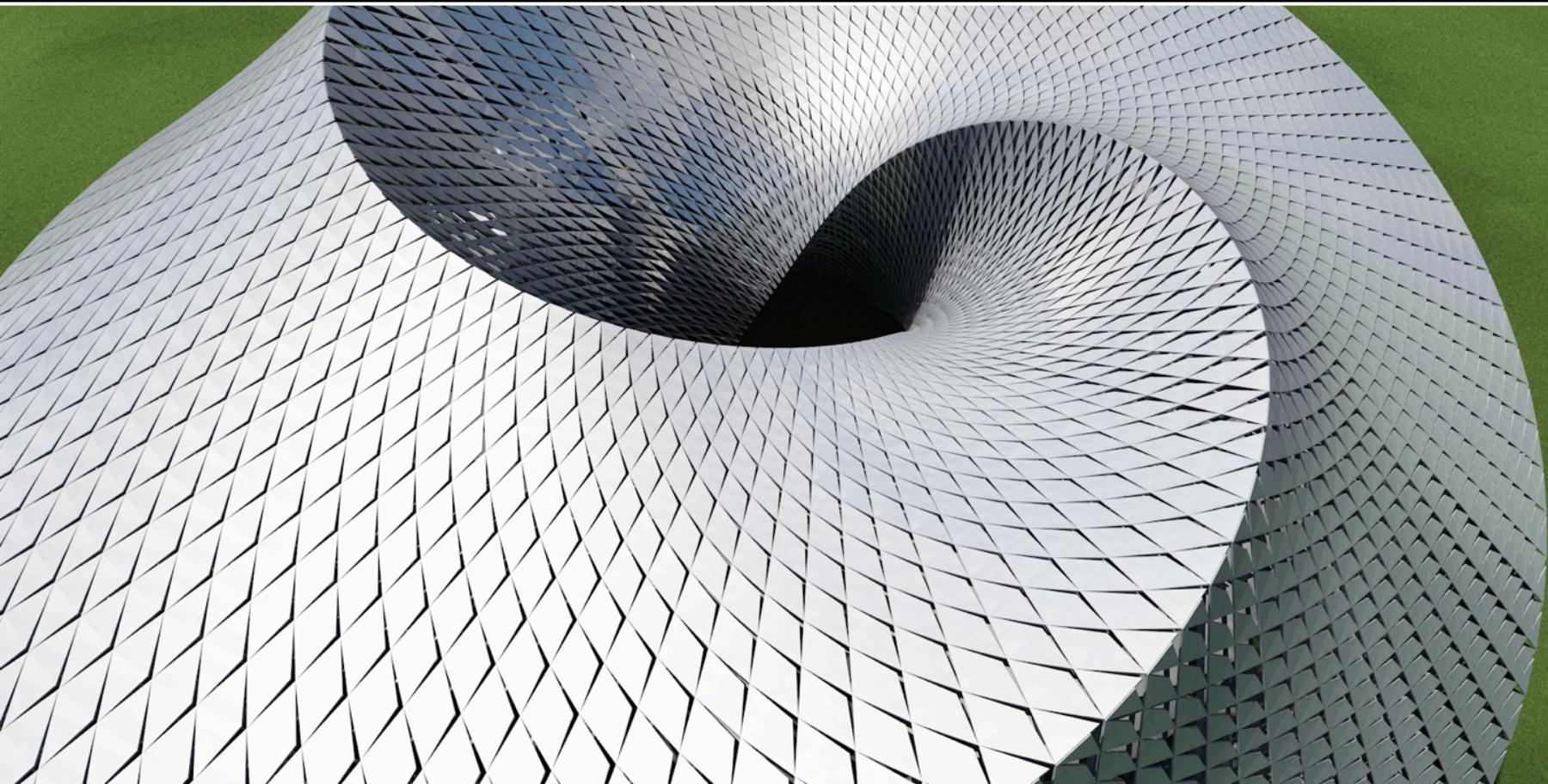
```

438
439
440 ;STEEL FRAMES
441
442 (define (bar pts)
443   (for/list ((p0 pts) (p1 (cdr pts)))
444     (beam p0 p1)))
445
446 (define (rotated-bar pts angles)
447   (for/list ((p0 pts) (p1 (cdr pts)) (a angles))
448     (beam p0 p1 a)))
449
450 (define (frames p)
451   (for/list ((fi (division 2pi 0 n-frames-for-frames #f))
452             (psi (division (/ pi -2) (+ (/ pi -2) skin-rotation) n-frames)))
453             (bar (list
454               (+spf (rot-base-point fi) ro-f fi (+ psi psiA))
455               (+spf (rot-base-point fi) ro-f fi (+ psi psiB))
456               (+spf (rot-base-point fi) ro-f fi (+ psi psiC))
457               (+spf (rot-base-point fi) ro-f fi (+ psi psiD))
458               (+spf (rot-base-point fi) ro-f fi (+ psi psiA)))))))
459
460 ;STEEL CONEXIONS
461
462 (define (conections p)
463   (define (ro-f x y) (sqrt (+ (sqr (/ x 2)) (sqr (/ y 2)))))
464   (define (psiA x y) (atan (/ y 2) (/ x 2)))
465   (define (psiB x y) (- pi (atan (/ y 2) (/ x 2))))
466   (define (psiC x y) (+ pi (atan (/ y 2) (/ x 2))))
467   (define (psiD x y) (- (atan (/ y 2) (/ x 2))))
468
469   (define (beam2column p1 p2)
470     (define (dif-almost? a b) (< -10 (- a b) 10))
471     (define (dif-almost0? a b) (< -1 (- a b) 1))
472     (if (string=? (current-backend-name) "ArchiCAD")
473         (if (and (dif-almost? (cx p1) (cx p2))
474                   (dif-almost? (cy p1) (cy p2)))

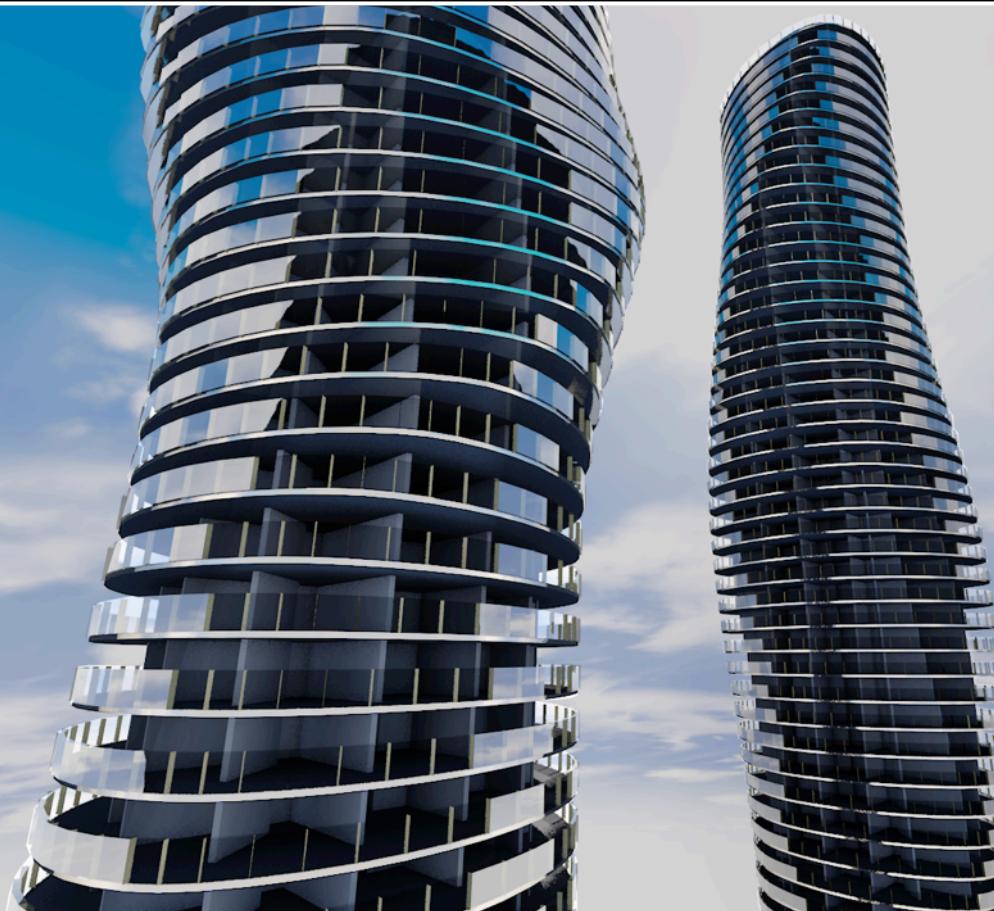
```



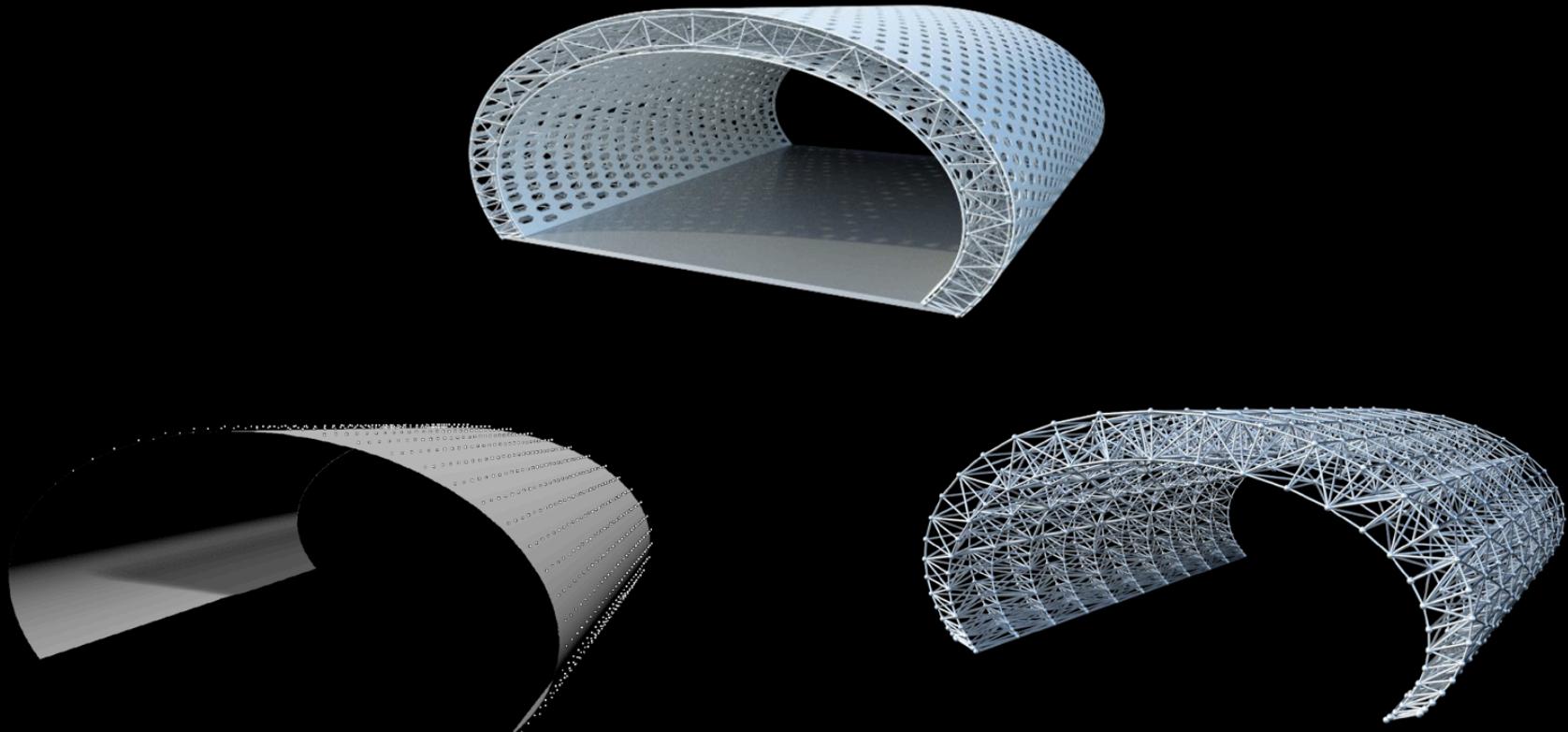
Complex Buildings



Complex Buildings



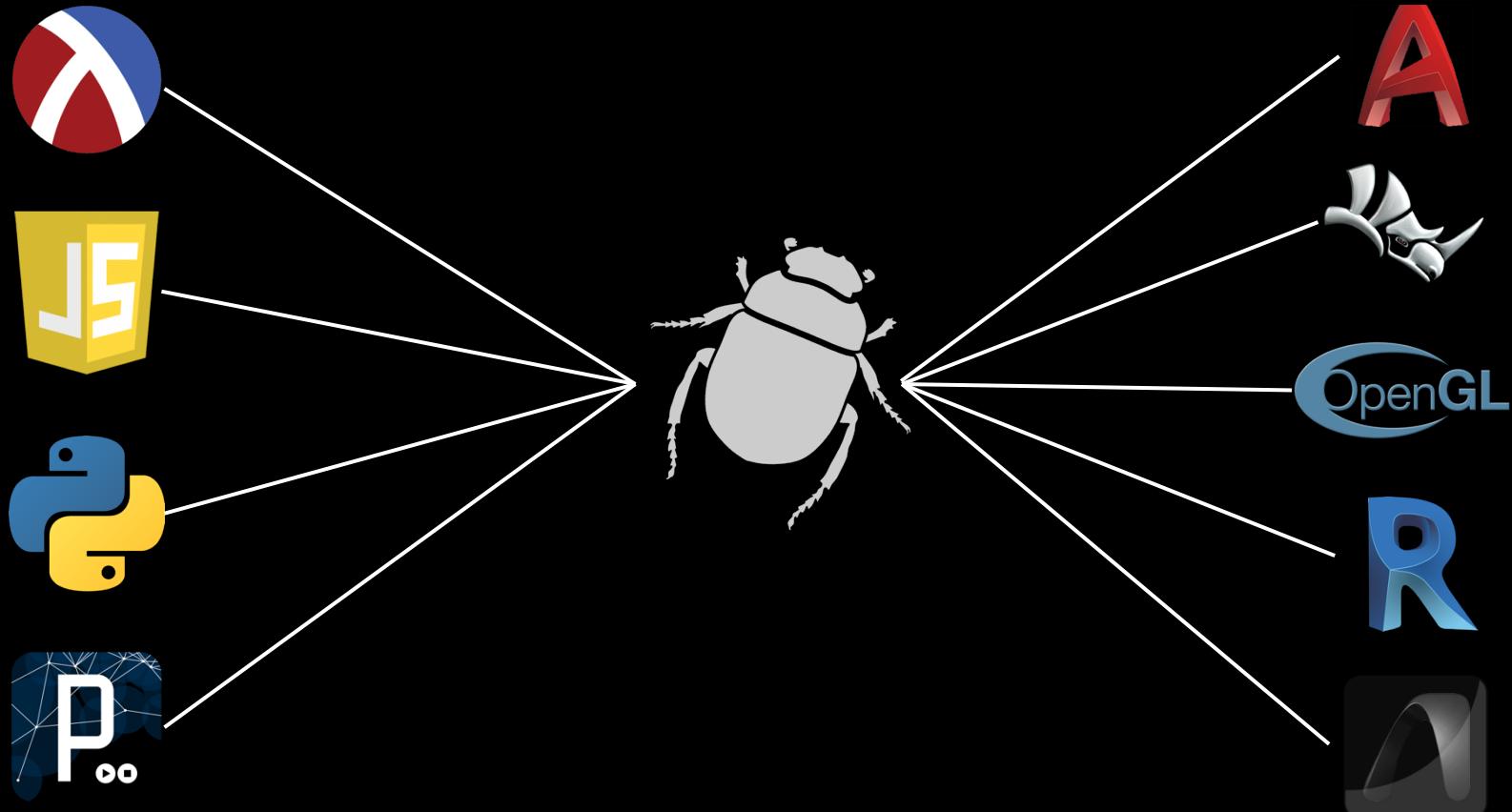
Need For Analysis



Several Tools



Khepri



Khepri

```
(provide spatial-truss
  spatial-truss-insert-apex)

;(backend rhino 'delete)

; spatial truss

(define truss-knot-radius 0.2)
(define truss-knot sphere)
(define (truss-knots cs radius) (map
  (lambda (c) (sphere (list (truss-knot-radius) (truss-knot radius)))) cs))

(define (spatial-truss curves (knots . rest))
  (let ((as (first curves))
        (bs (second curves))
        (cs (third curves)))
    (list
      (truss-knots as knot-radius)
      (truss-knots bs knot-radius)
      (truss-bars as cs bar-radius)
      (truss-bars bs (drop-right as))
      (truss-bars bs (drop-right cs))
      (truss-bars bs (rest as) bar-radius)
      (truss-bars bs (rest cs) bar-radius)
      (truss-bars (rest as) (drop-right as))
      (truss-bars (rest bs) (drop-right bs))
      (if (empty? (cdrrest curves))
          (list
            (truss-knots cs knot-radius)
            (truss-bars (rest cs) (drop-right cs)))
          (list
            (truss-bars bs (first (cdrrest curves)))
            (spatial-truss (drop-right curves)))))))

; spatial truss insert apex

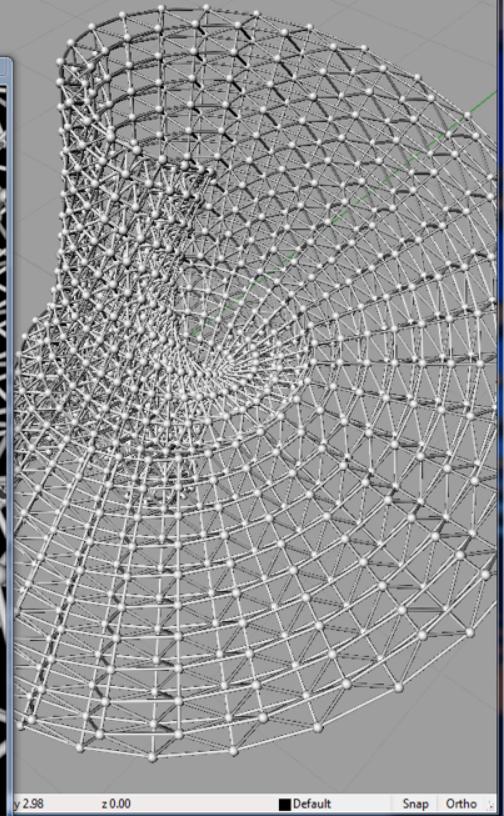
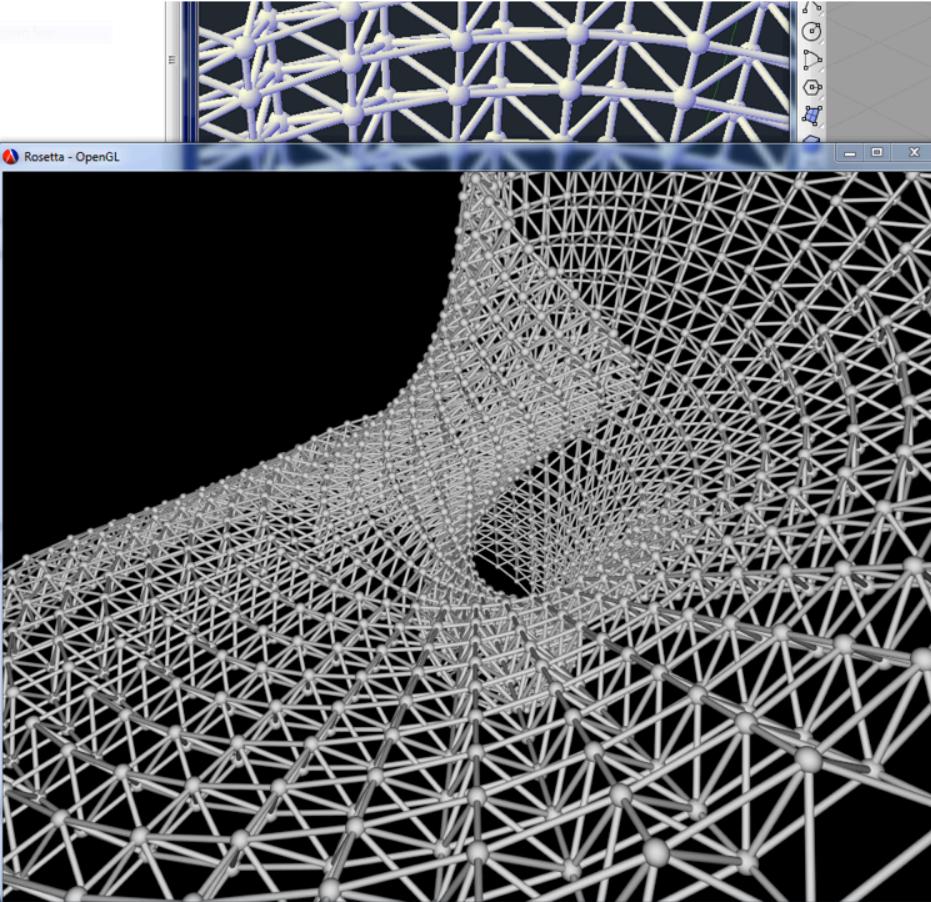
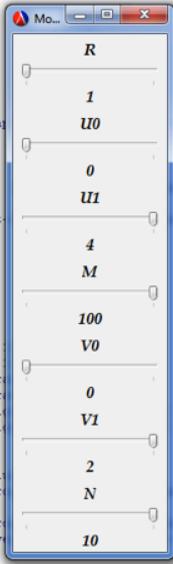
(provide spatial-truss-insert-apex)

; utils

(define (cross-product c1 c2)
  (xyz
    (* (- (xyz-y c1) (xyz-y c2)) (+ (xyz-z c1) (xyz-z c2)))
    (* (- (xyz-z c1) (xyz-z c2)) (+ (xyz-x c1) (xyz-x c2)))))

; on-size
on-size

Determine language from source▼
```



Khepri

Properties

3D View

3D View: (3D) Edit Type

Graphics

View Scale: 1:100

Scale Value: 1: 100

Detail Level: Medium

Parts Visibility: Show Original

Visibility/Graphics... Edit...

Graphic Display O... Edit...

Discipline: Coordination

Show Hidden Lines: By Discipline

Default Analysis D... None

Sun Path

Identity Data

View Template: <None>

View Name: (3D)

Dependency: Independent

Title on Sheet

Properties help Apply

Project Browser - Project1.0001

Views (all)

Floor Plans

- Level 1
- Level 2
- Level 3
- Level 4
- Level 5
- Level 6
- Level 7
- Level 8
- Level 9
- Level 10
- Level 11
- Level 12
- Level 13
- Level 14
- Level 15
- Level 16
- Level 17
- Level 18
- Level 19
- Level 20

1: 100

Main Model

Marquee

Design

- Wall
- Door
- Window
- Column
- Beam
- Slab
- Stair
- Roof
- Shell
- Skylight
- Curtain...
- Morph
- Object
- Zone
- Mesh

Dimension

- Level Di...
- Text
- All Label
- Fill
- Line
- Arc/Circle
- Polyline
- Drawing
- Section
- Elevation
- Interior...
- Worksheet...
- Detail
- Change

More

Properties

Generic Axonometry

Settings...

31. Story

30. Story

29. Story

28. Story

27. Story

26. Story

25. Story

24. Story

23. Story

22. Story

21. Story

20. Story

19. Story

18. Story

17. Story

16. Story

15. Story

14. Story

13. Story

12. Story

11. Story

10. Story

9. Story

8. Story

7. Story

6. Story

5. Story

4. Story

3. Story

2. Story

1. Story

0. Story

Sections

Elevations

Interior Elevation

Worksheets

Details

3D Documents

Generic Pers

Generic Ax

Schedules

Project Indexes

Lists

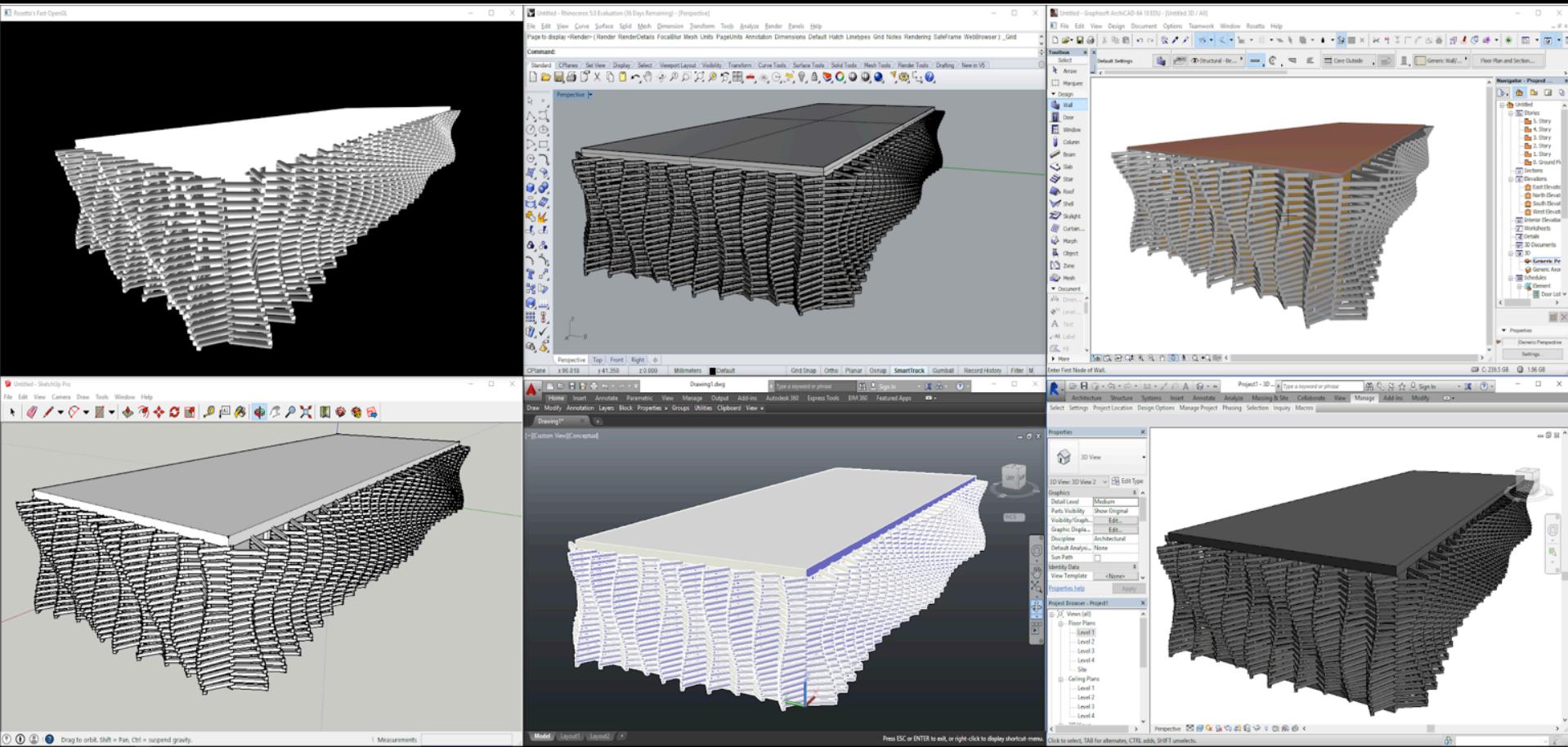
Info

Help

C: 86.6 GB 6.58 GB

Click to select, TAB for alternates, CTRL adds, SHIFT unselects.

Khepri



Code can become messy

This image shows two side-by-side screenshots of a browser's developer tools, specifically the Elements tab. Both screens are filled with extremely long, dense blocks of JavaScript code, demonstrating how unoptimized or poorly managed code can quickly become illegible.

The left screenshot shows a block of code starting with a multi-line comment and several nested functions. It includes various declarations like `var`, `function(e)`, and `if (e.type === 'submit')`. The code is heavily commented with explanatory text and includes logic for handling file inputs, event listeners, and server requests.

The right screenshot shows another equally long block of JavaScript, continuing from where the first one ended. This code also features multiple nested functions and deep object hierarchies. It contains numerous calls to `transform` and `translate` functions, which appear to be part of a custom animation library. The code is structured with many temporary variables and complex conditional statements.

Both snippets of code demonstrate the potential consequences of not using proper naming conventions, modularization, and tools like linters and formatters to maintain code readability.

One script for all models

Context-Oriented Programming

COP

- > Behavioral Variations
- > Layers
- > Activation mechanism
- > Context
- > Scoping

Objectives

- > Present and compare COP
- > Combine COP with AD

Related Work



Aspect-Oriented Programming

- > Cross-cutting concerns
- > Aspects
- > Pointcuts, Join Points, Advice

Subject-Oriented Programming

- > Subjects
- > Subject-activation
- > Subjective Dispatch

Context-Oriented Programming

- > Contexts
- > Contextual Dispatch

ContextL

- > Common Lisp
- > Dynamically Scoped Layer Activation
- > Layered Generic Functions
- > Layer-in-class and class-in-layer

ContextL

```
(deflayer employment-layer)

(with-active-layers (employment-layer)
  ... contained code ...)

(define-layered-class employer
  :in-layer employment-layer ()
  ((name :initarg :name
         :layered-accessor employer-name)))
```

ContextPy

- > Python
- > Dynamically Scoped Layer Activation
- > Decorators
- > Layer-in-class

ContextPy

```
class Slab:

    @around(a3DLayer)
    def generate(self):
        return extrusion(surface_from(self.path),
                          self.thickness)

    @around(a2DLayer)
    def generate(self):
        return self.path
```

ContextJ

- > Java (source-to-source compiler)
- > Dynamically Scoped Layer Activation
- > Reflection API
- > Layer-in-class

ContextJ

```
class Employer{  
    String toString() {  
        return "Name: " + name;  
    }  
  
layer Address {  
    String toString() {  
        return proceed() + "; Address: " + ...  
    }  
}
```

Lambic

- > Common Lisp
- > Predicate Dispatch
- > Activation with Predicates
- > Different modularization

Lambic

```
(defgeneric factorial (n)
  (:predicates < = >))
```

```
(defmethod factorial (n)
  (:when (> n 0))
  (* n (factorial (- n 1)))))
```

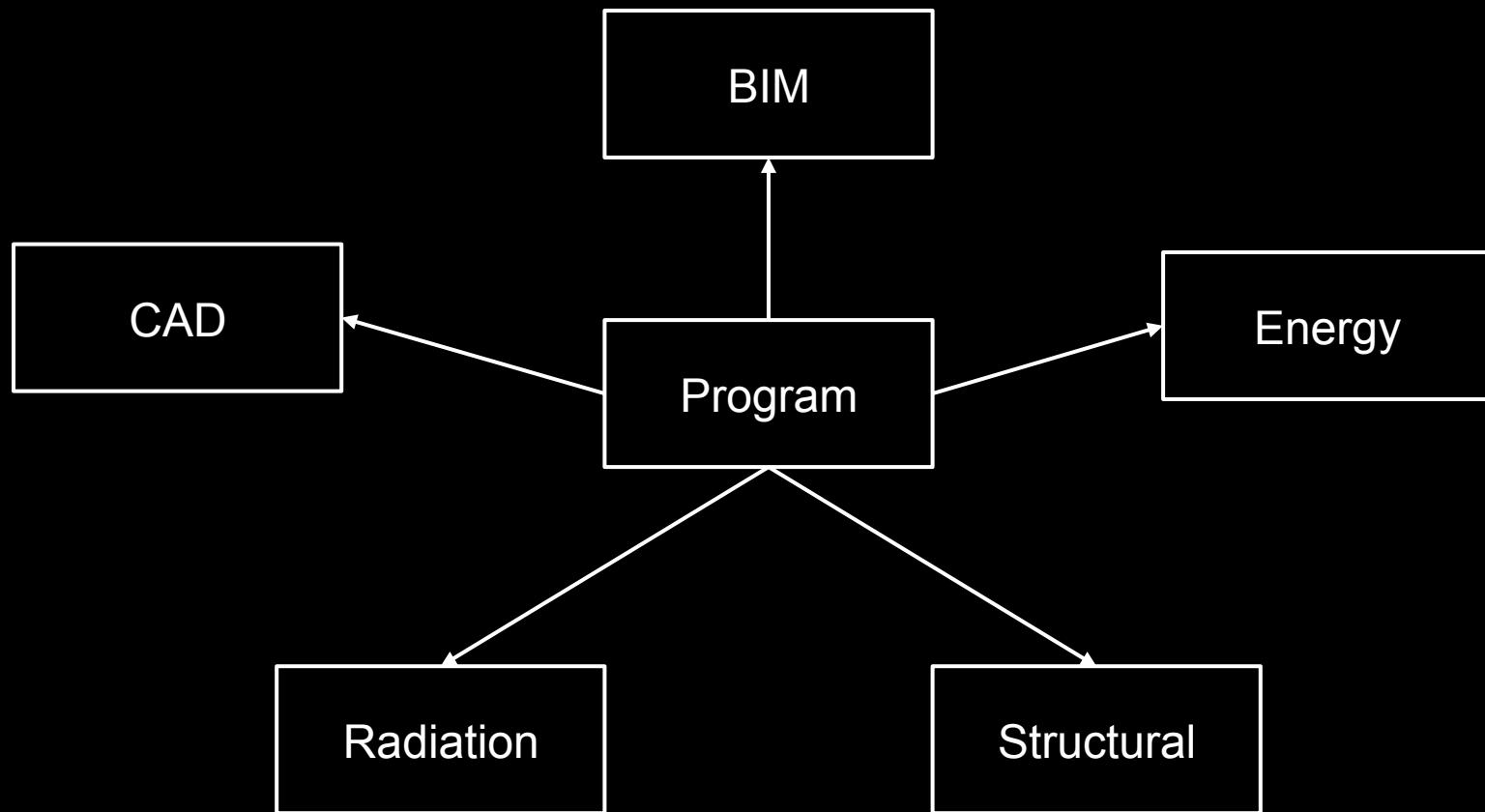
```
(defmethod factorial (n)
  (:when (= n 0))
  1)
```

Comparison

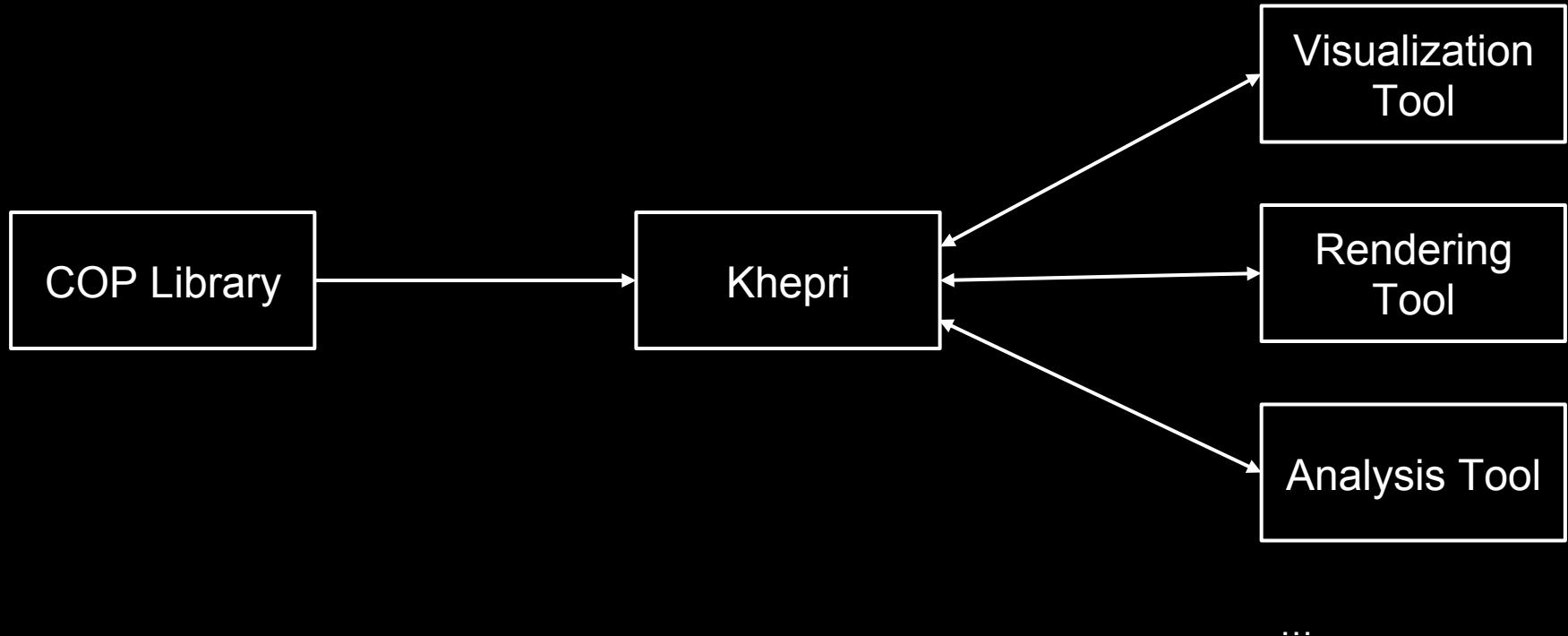
	Base Language	Implementation	Layer Activation	Modularization
ContextL	Common Lisp	Library	DSA	LIC, CIL
ContextScheme	Scheme	Library	DSA	CIL
ContextErlang	Erlang	Library	Per-Agent	Erlang Modules
ContextJS	JavaScript	Library	Open Implementation	LIC, CIL
PyContext	Python	Library	DSA, Implicit	CIL
ContextPy	Python	Library	DSA	LIC
ContextJ	Java	S2S	DSA	LIC
JCop	Java	S2S and Aspect	DSA, declarative, conditional	LIC
EventCJ	Java	S2S and Aspect	DSA	LIC
JavaCtx	Java	Library and Aspect	DSA	LIC
ContextR	Ruby	Library	DSA	LIC
ContextLua	Lua	Library	DSA	CIL
ContextS	Smalltalk	Library	DSA, indefinite	CIL
Ambience	AmOS	Library	DSA, global	CIL
Lambic	Common Lisp	Library	-	-
Subjective-C	Objective-C	Preprocessor	Global	LIC

Context-Oriented Algorithmic Design

COAD



Implementation

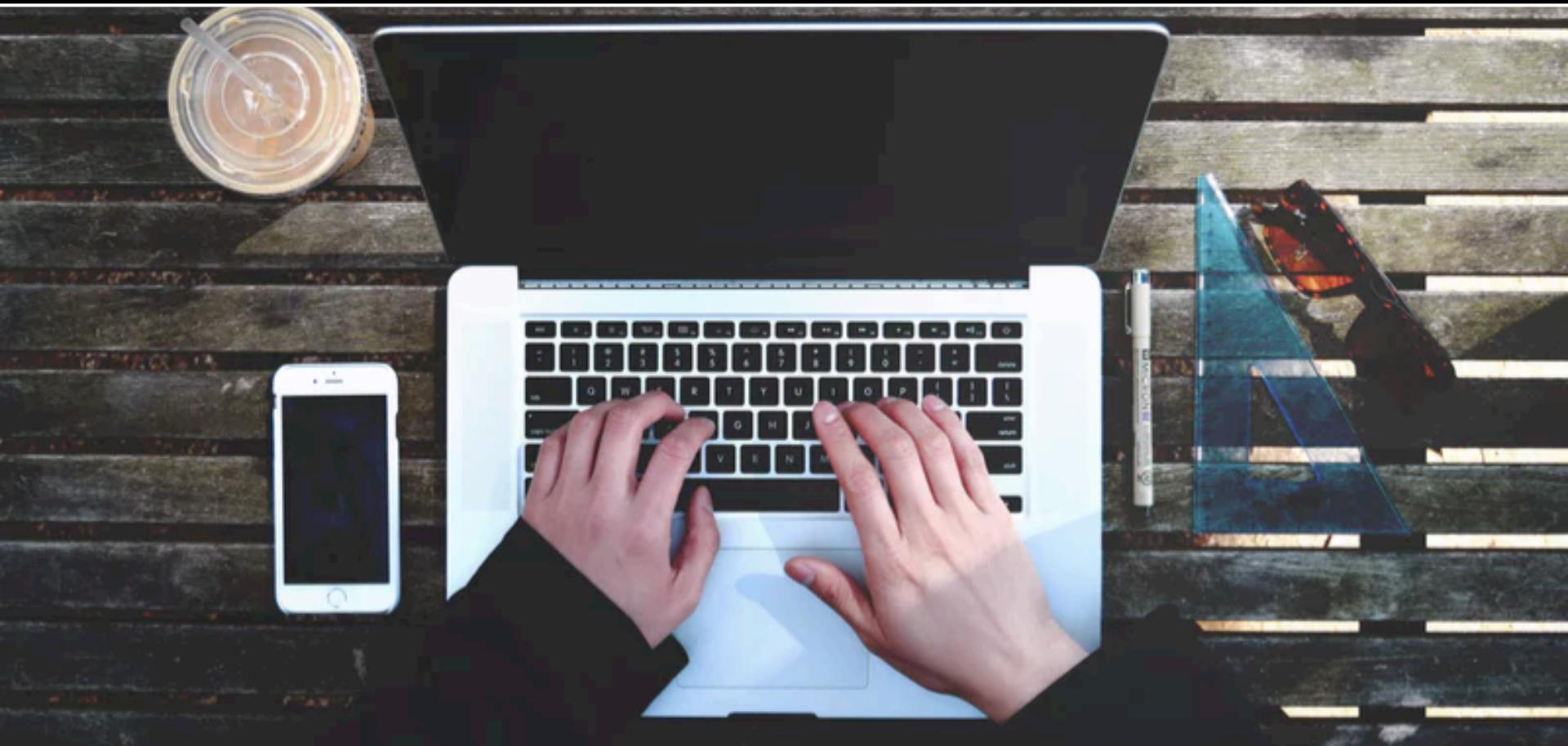


Implementation

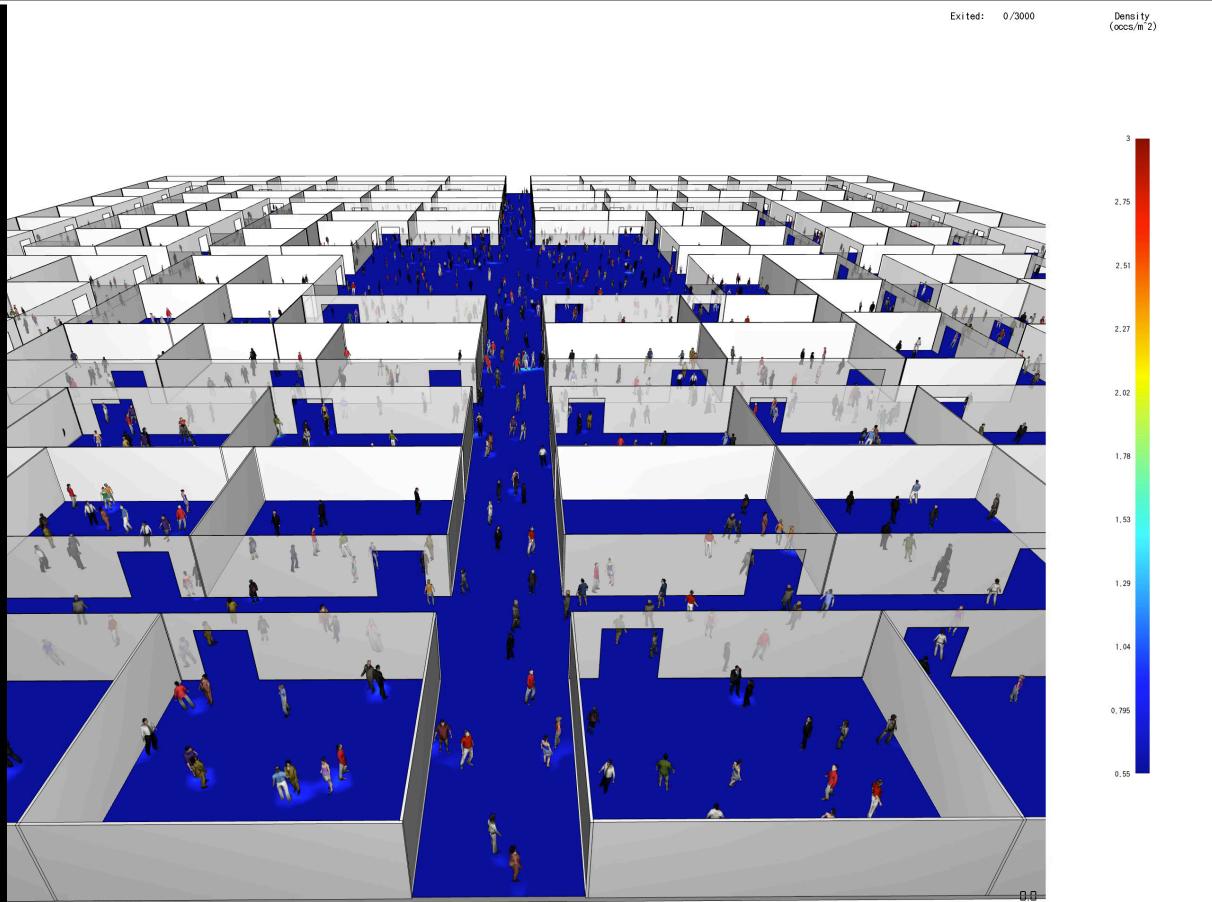
- > ContextScheme (adapted to Racket)

- > Khepri

Case Study



Case Study



Original Version

```
(define (shop-2d ...)  
  (...  
   (line ...)  
   (rectangle ...)  
   ...) )
```

```
(define (shop-3d ...)  
  (...  
   (right-cuboid ...)  
   ...) )
```

Original Version

```
(define (mall . . . shop . . .)
  (...  
    (shop . . . ))  
  . . . ) )
```

COP Version

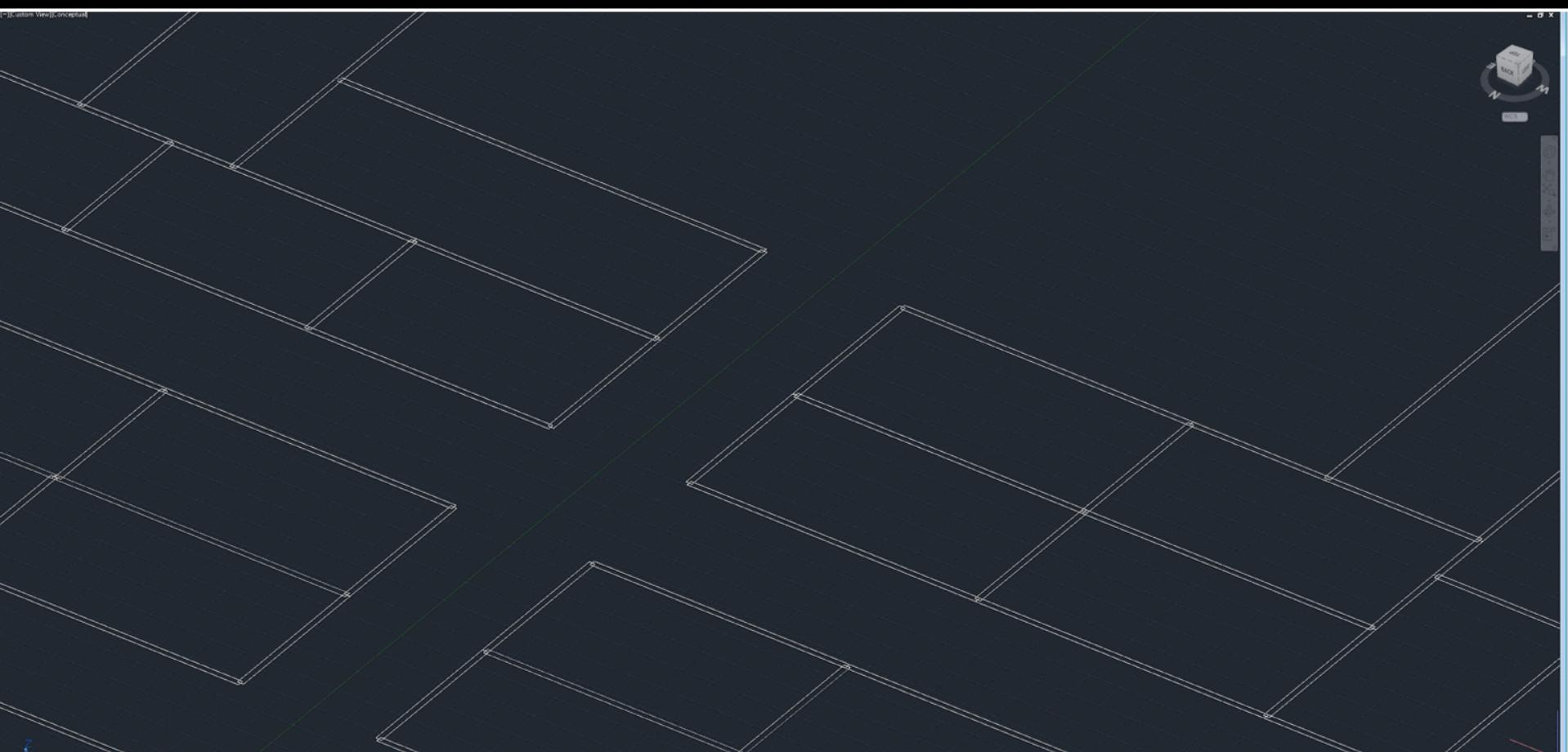
```
(define (shop p v l w)
  (...
   ((wall) p0 p1 wall-thickness wall-height)
   ...
   ((door) ((wall) p2 p3 wall-thickness wall-height)
           p4 p5 wall-thickness door-height))))
```

COP Version

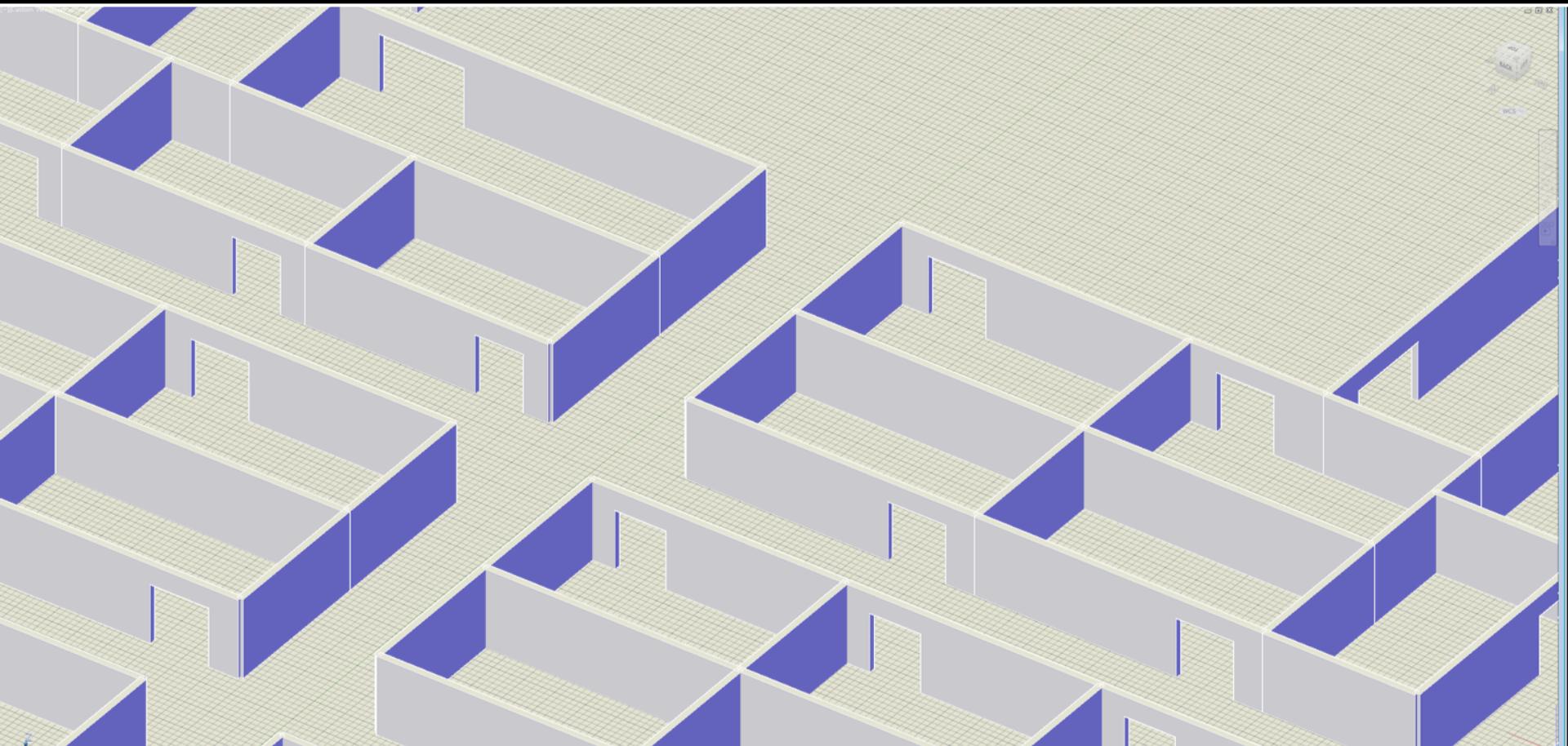
(with-layers (2D)

(mall (xy 0 0) 100000 12000 25000 7000 7000 4))

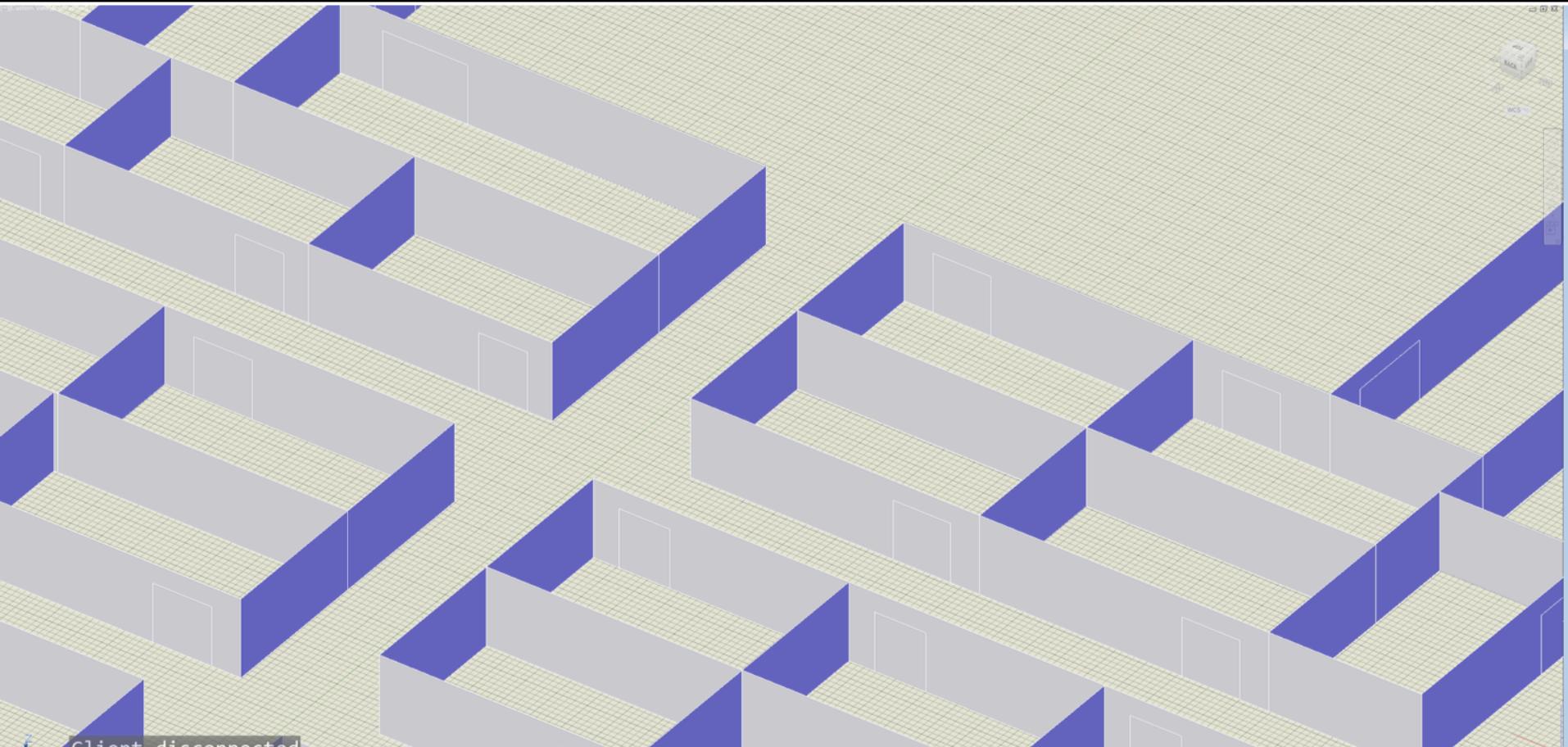
COP 2D



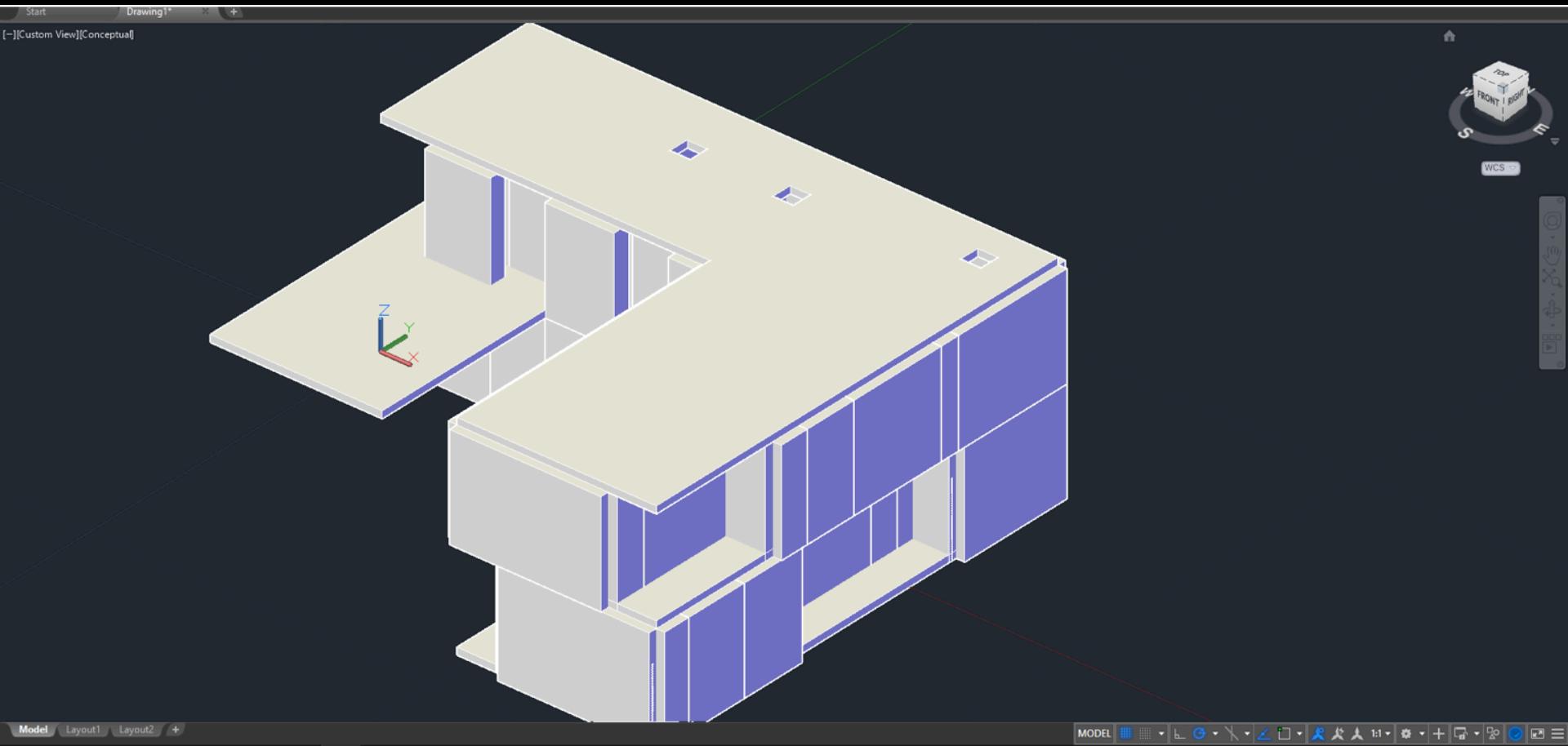
COP 3D



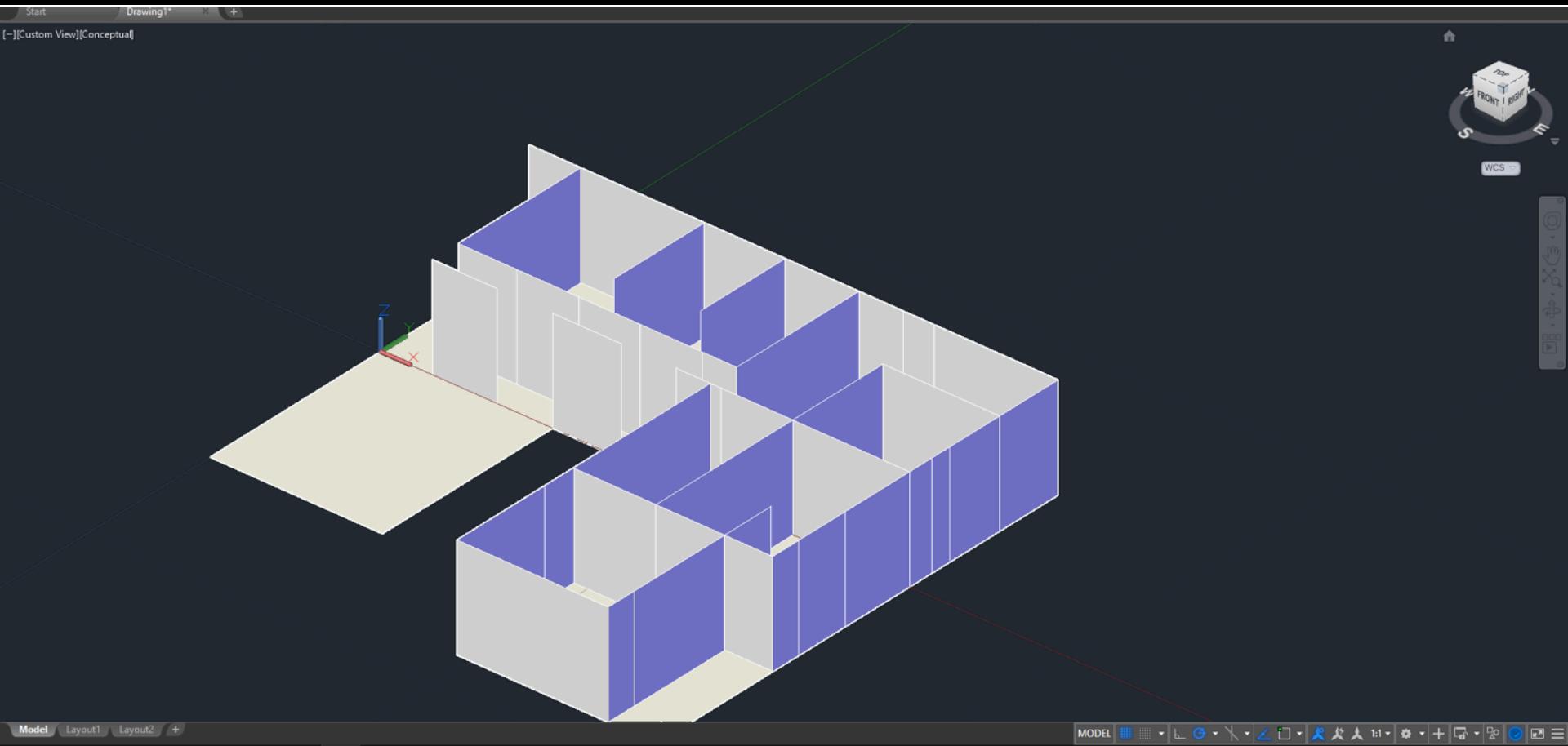
COP Analysis



Partial Modelling



Partial Modelling



Advantages

- > Reduces the code
- > Dynamic context change

Evaluation



Evaluation

- > Higher-order or not?
- > Implicit or Explicit Activation?
- > Performance?

Future Work





Thank you!

Questions?

bruno.b.ferreira@tecnico.ulisboa.pt
antonio.menezes.leitao@tecnico.ulisboa.pt

<https://algorithmicdesign.github.io/>