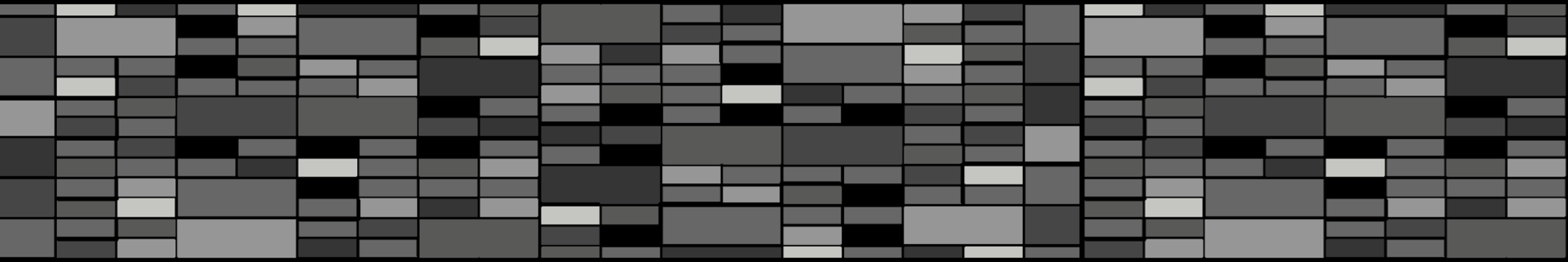
Integration of an Algorithmic BIM Approach in a Traditional Architecture Studio



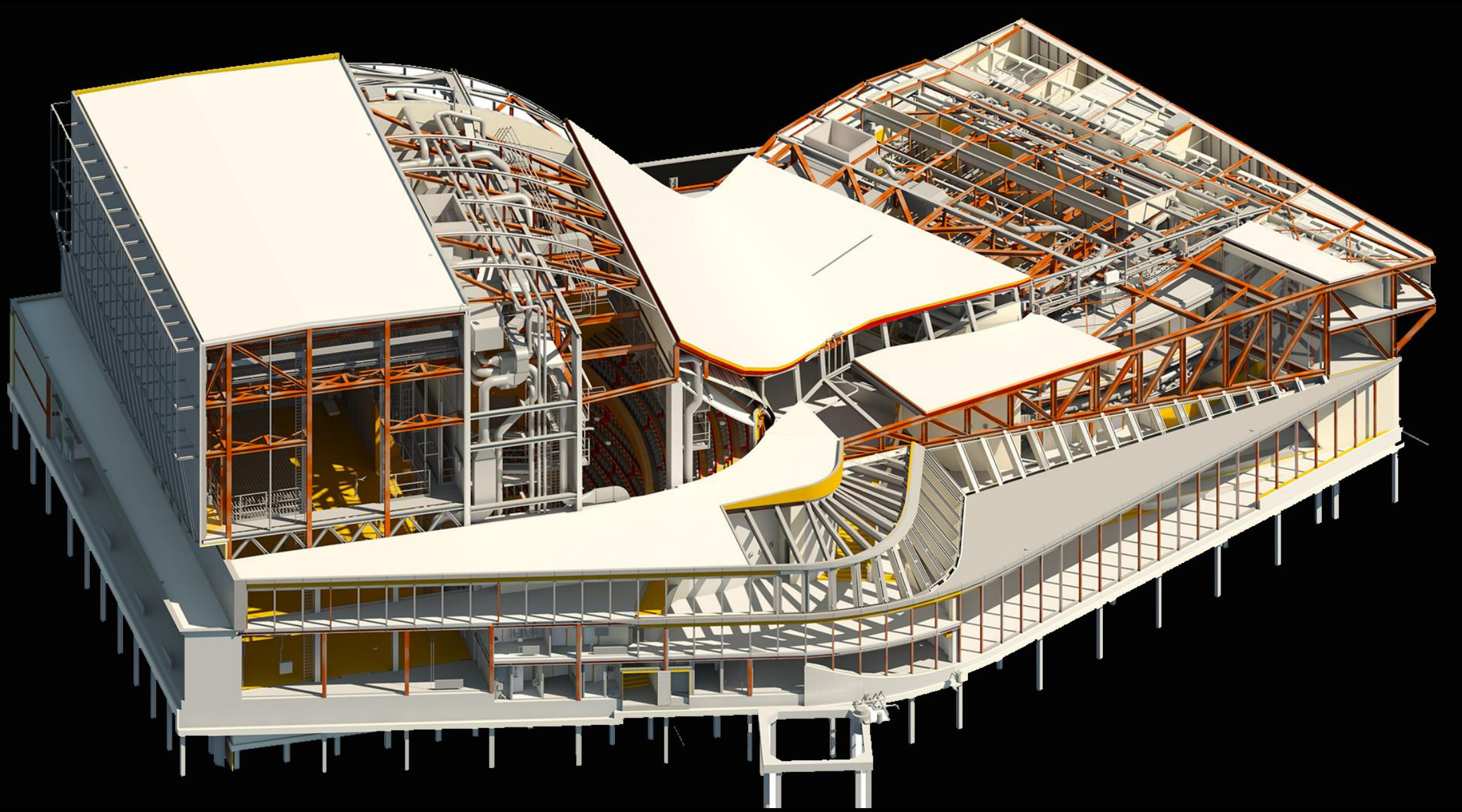
Inês Caetano
António Leitão

ines.caetano@técnico.ulisboa.pt
antonio.menezes.leitão@técnico.ulisboa.pt

CAADRIA 2017: PROTOCOLS, FLOWS AND GLITCHES

AD: Algorithmic Design

BIM: Building Information Modeling



http://www.archdaily.com/555446/theatre-de-stoep-unstudio

ABB: Algorithmic-Based BIM

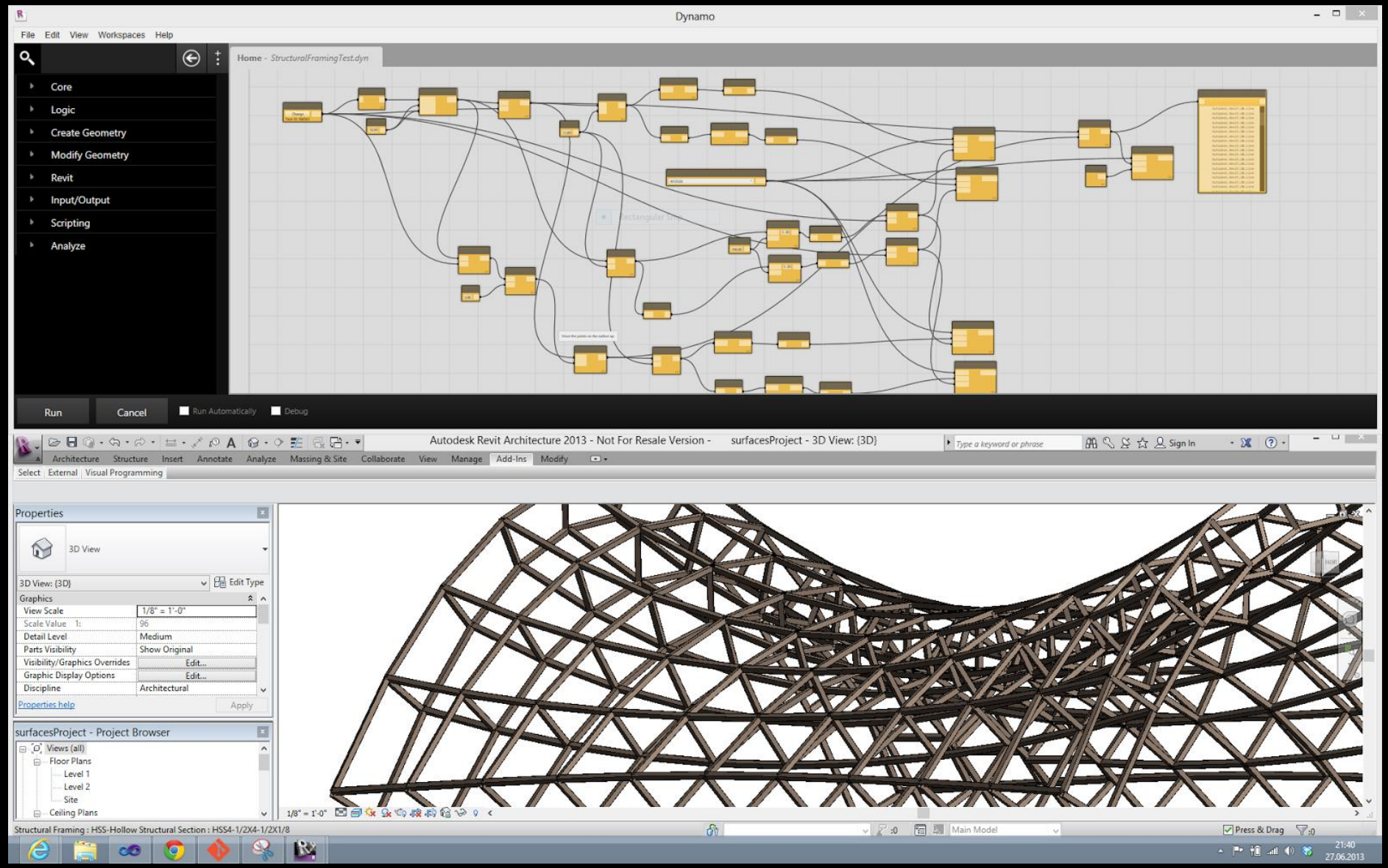


ABB: Algorithmic-Based BIM

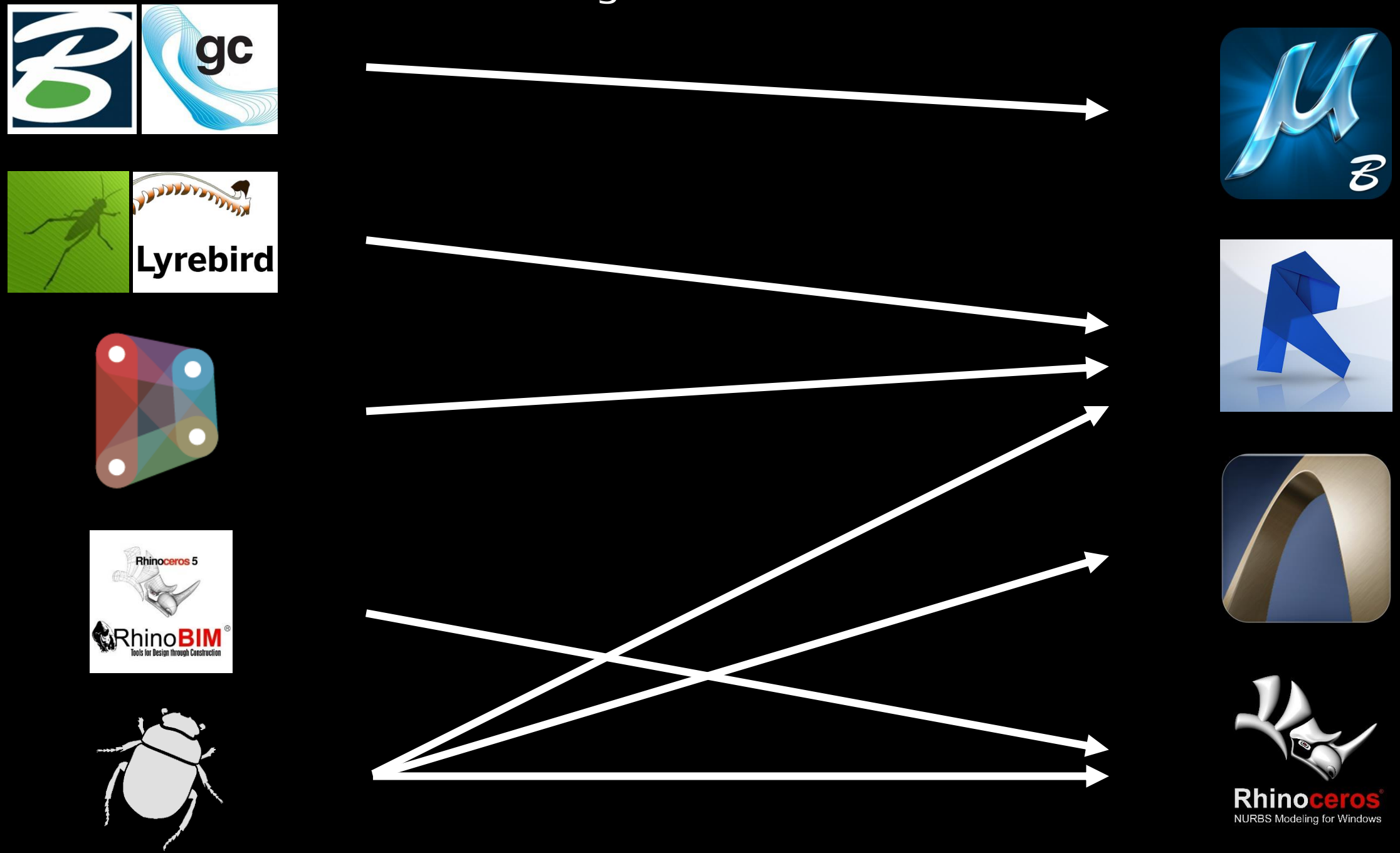
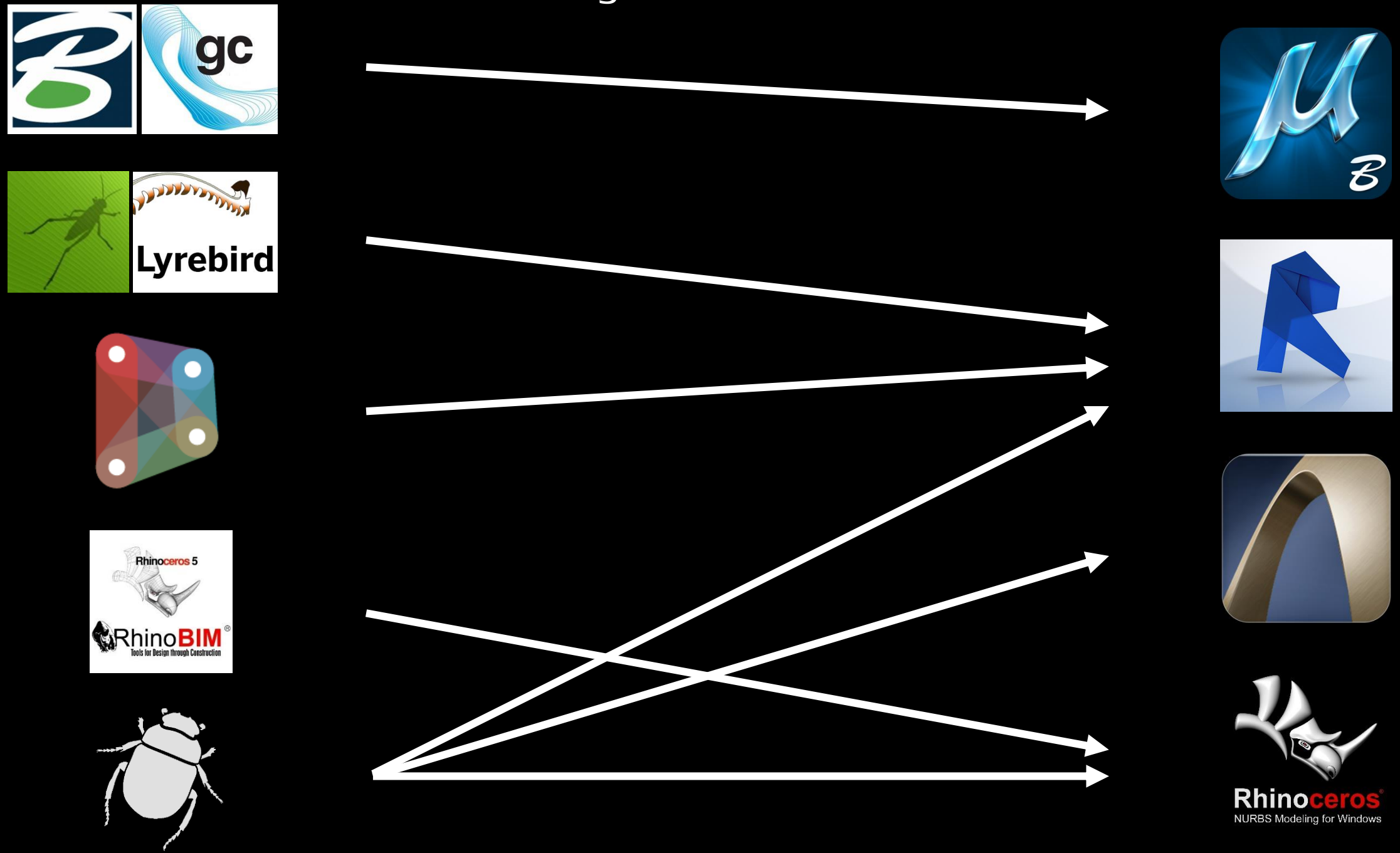


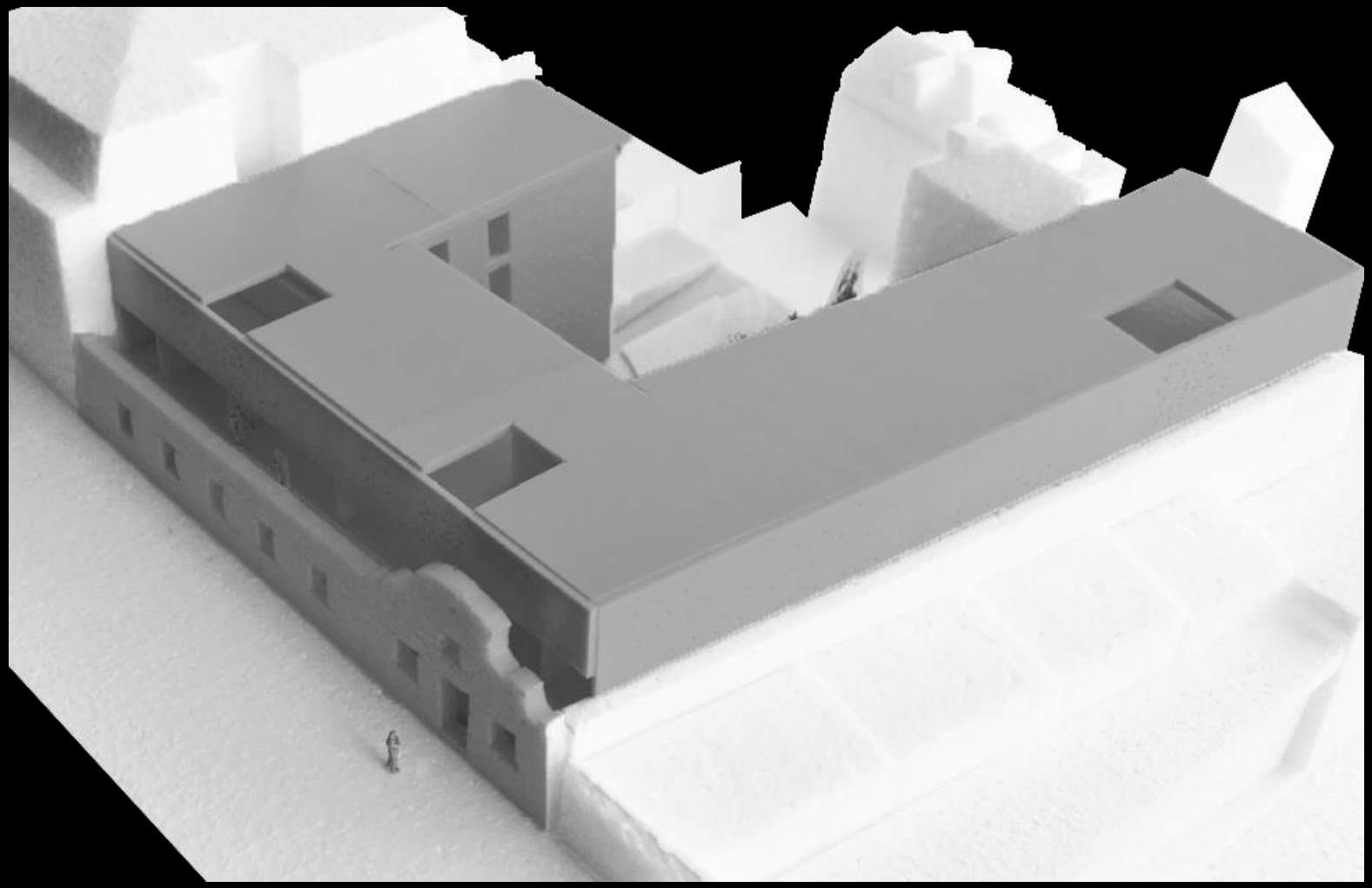
ABB: Algorithmic-Based BIM



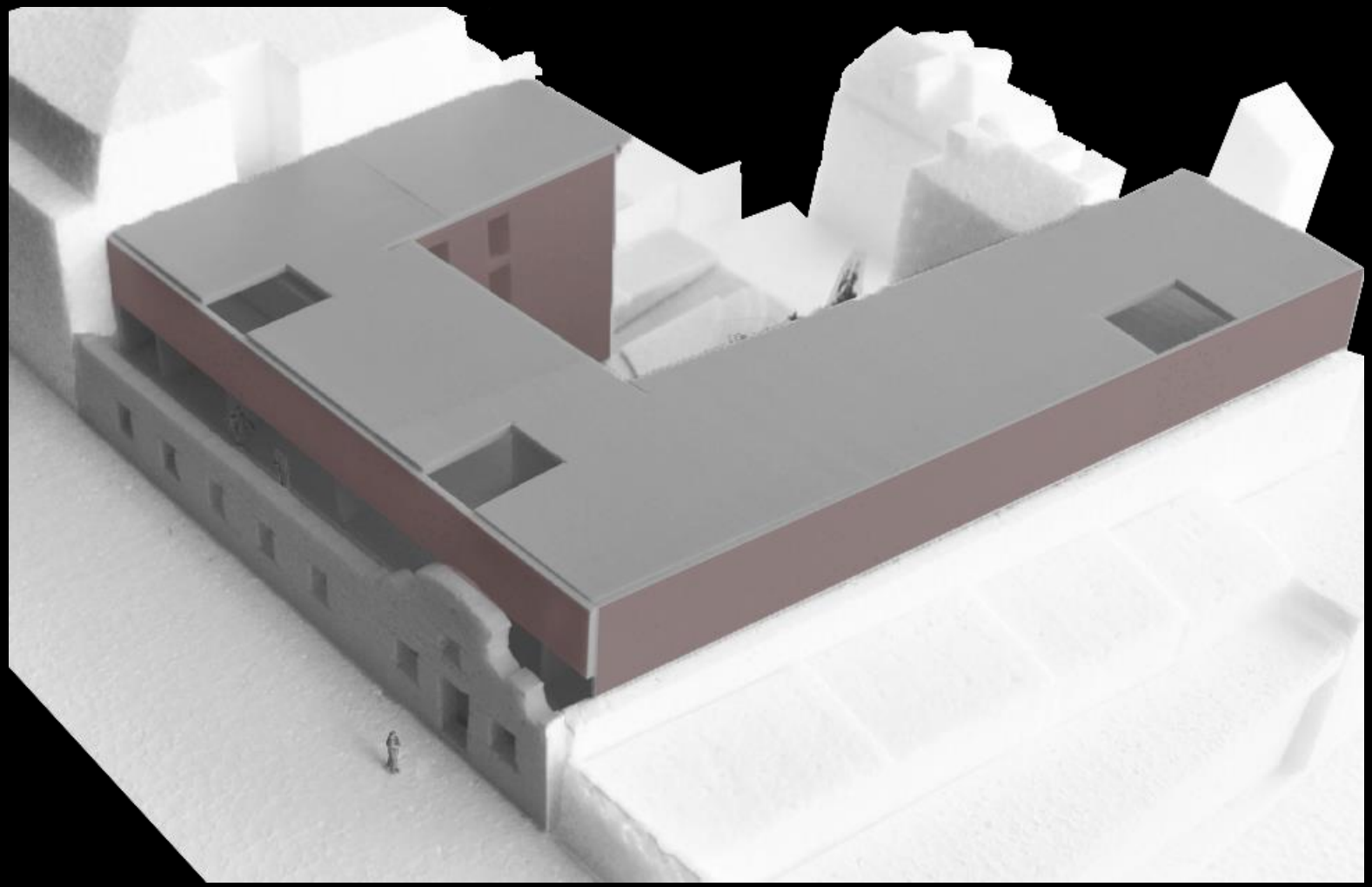
atelier dos remédios

+ FORA

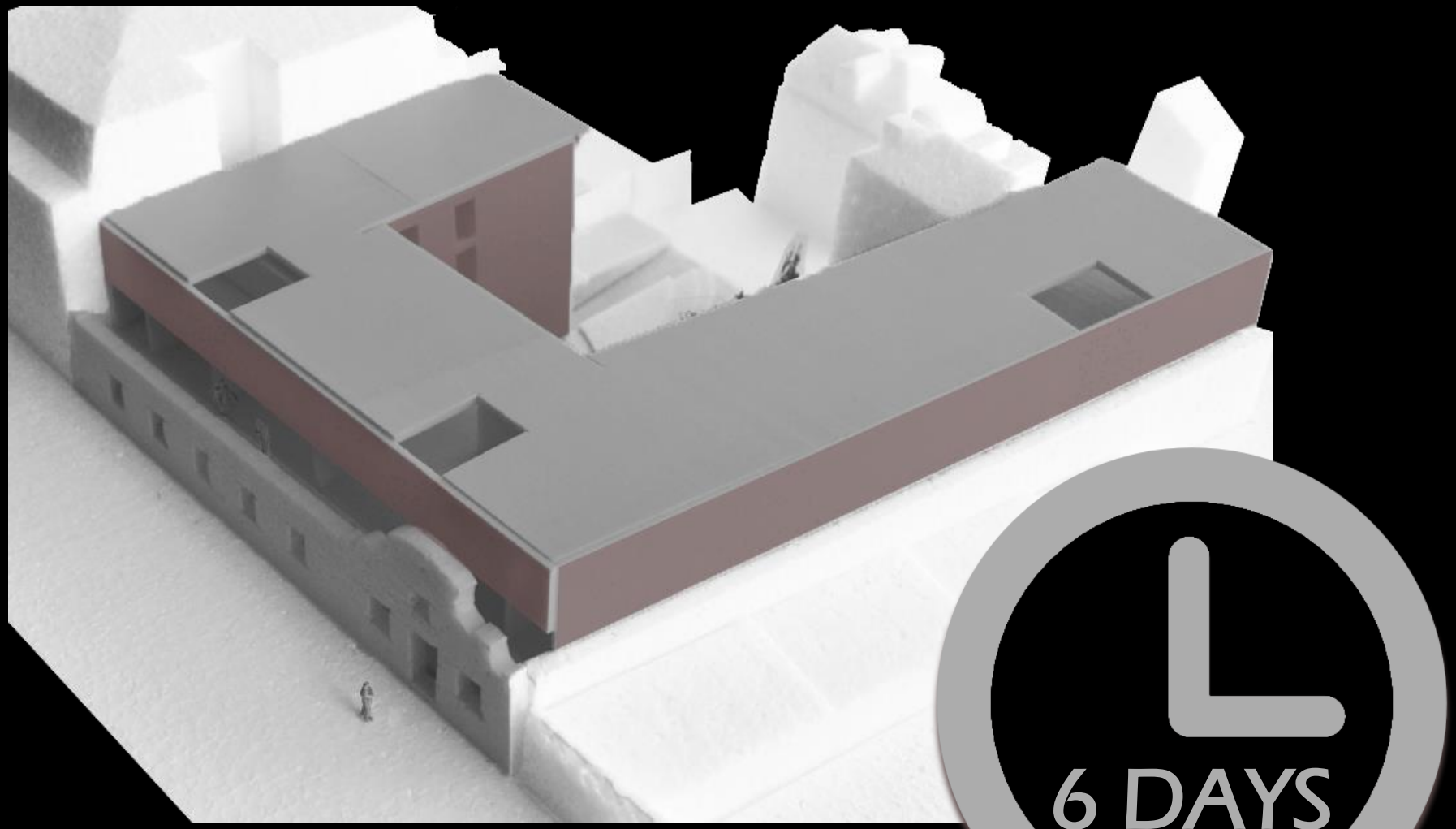
Parametric Facades



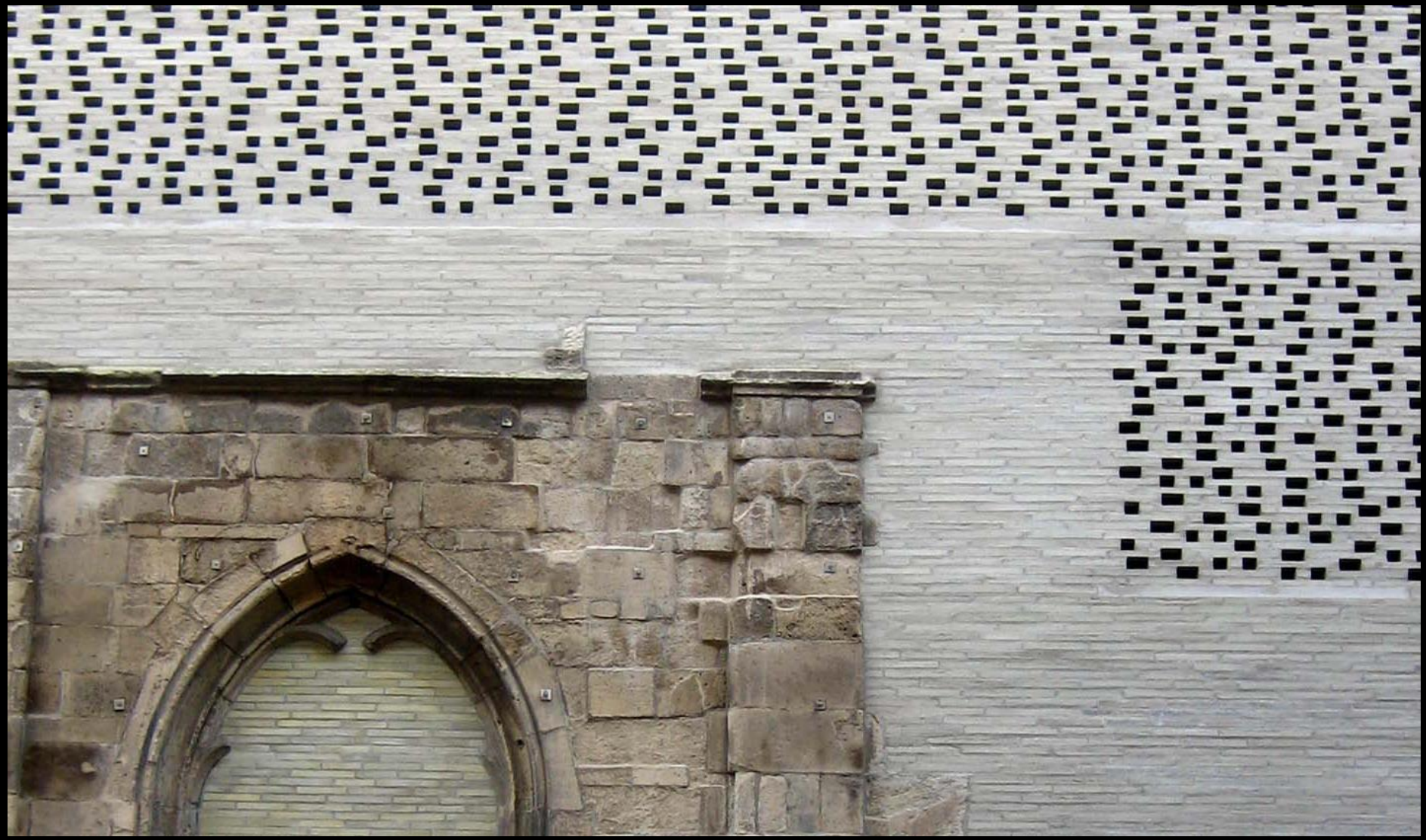
Parametric Facades



Parametric Facades

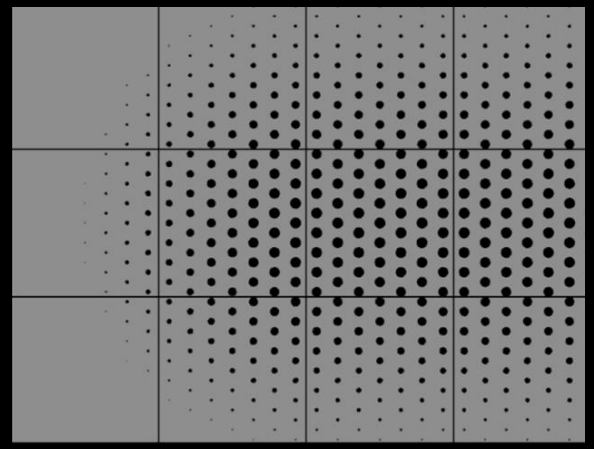


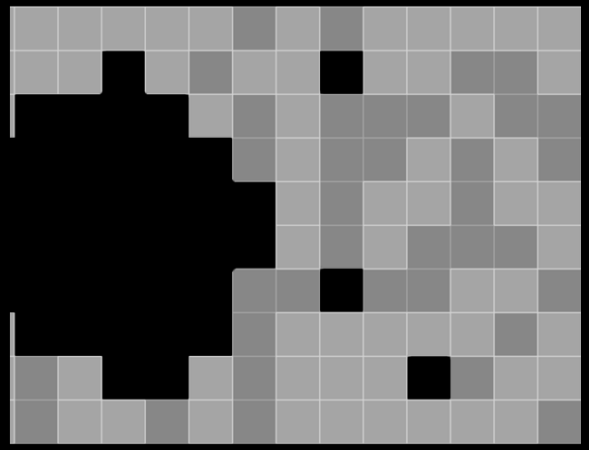
Concept *permeability-opacity*

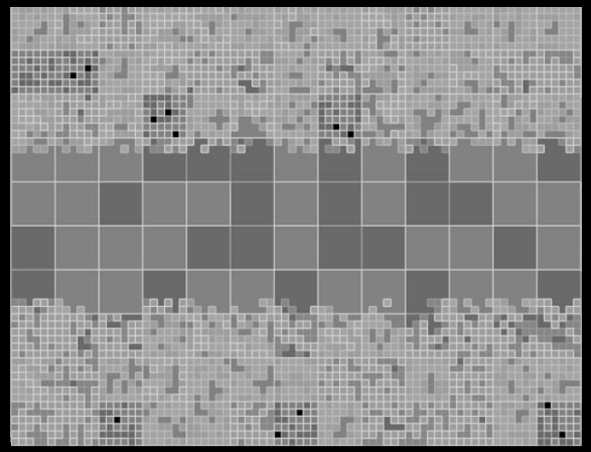


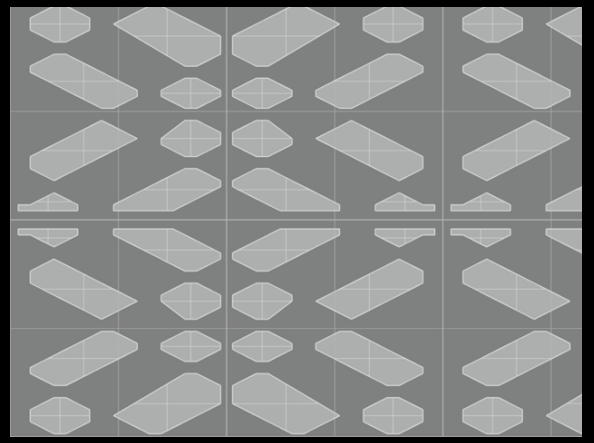
Kolumba Museum| Peter Zumthor

Concept *permeability-opacity*

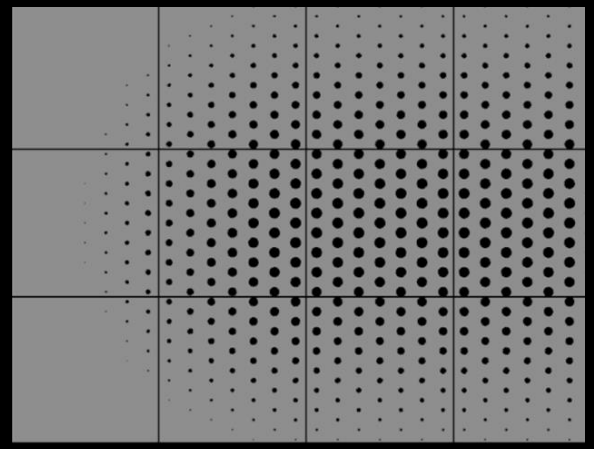


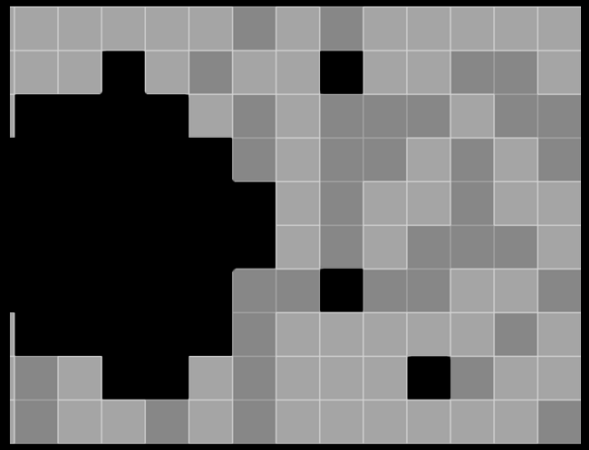


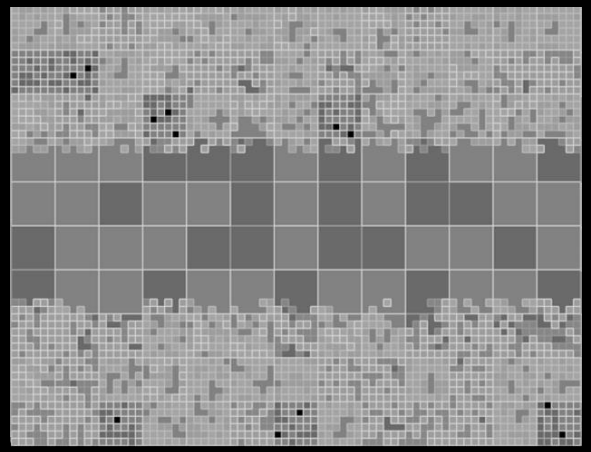


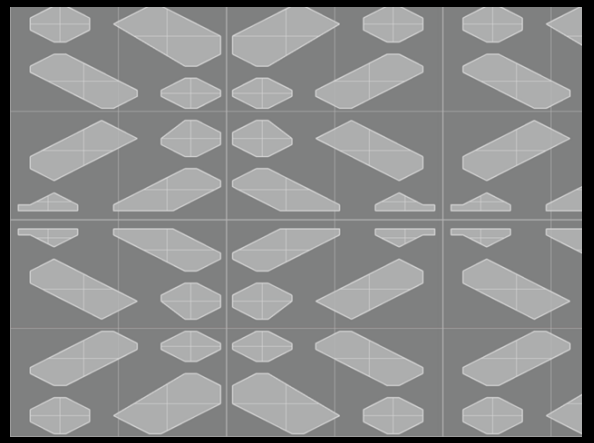


Concept *permeability-opacity*

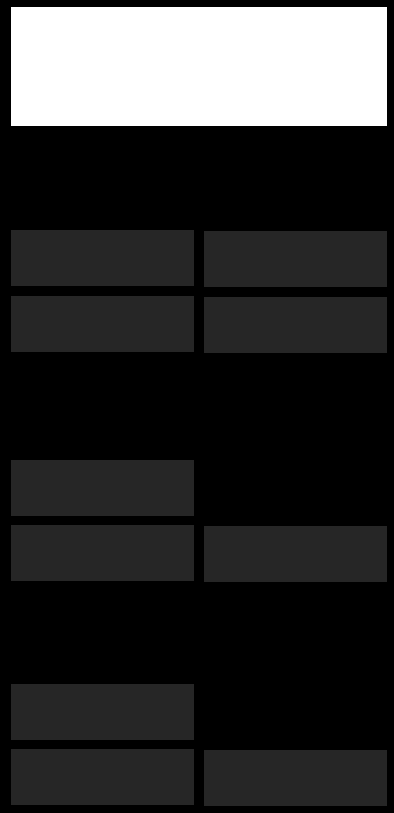


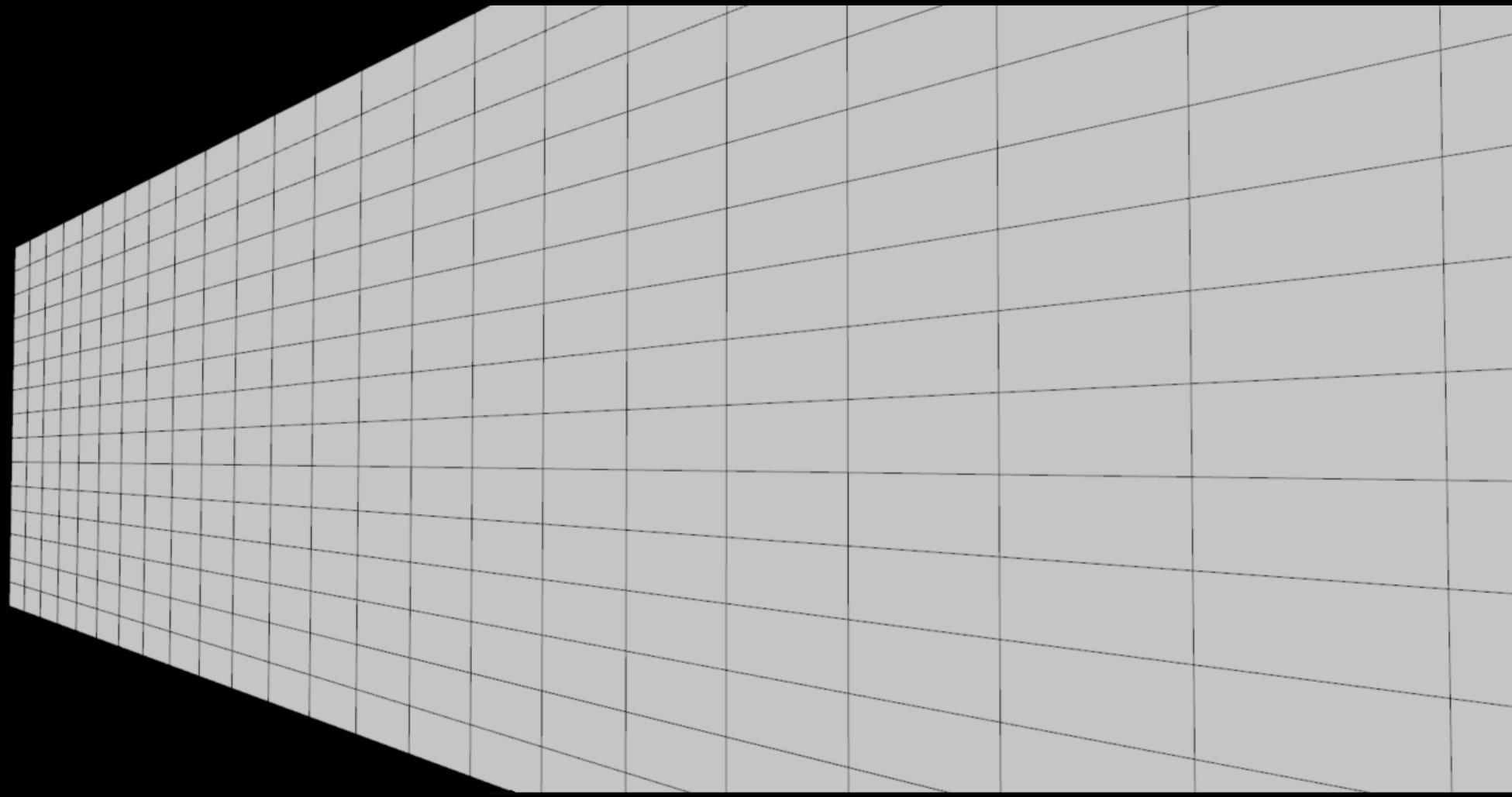




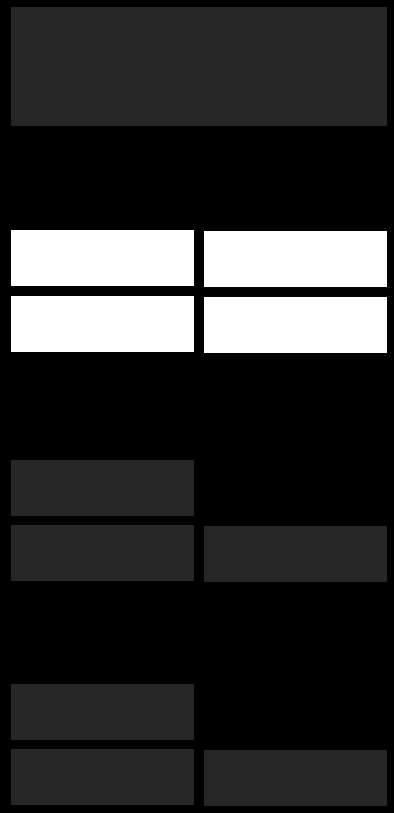


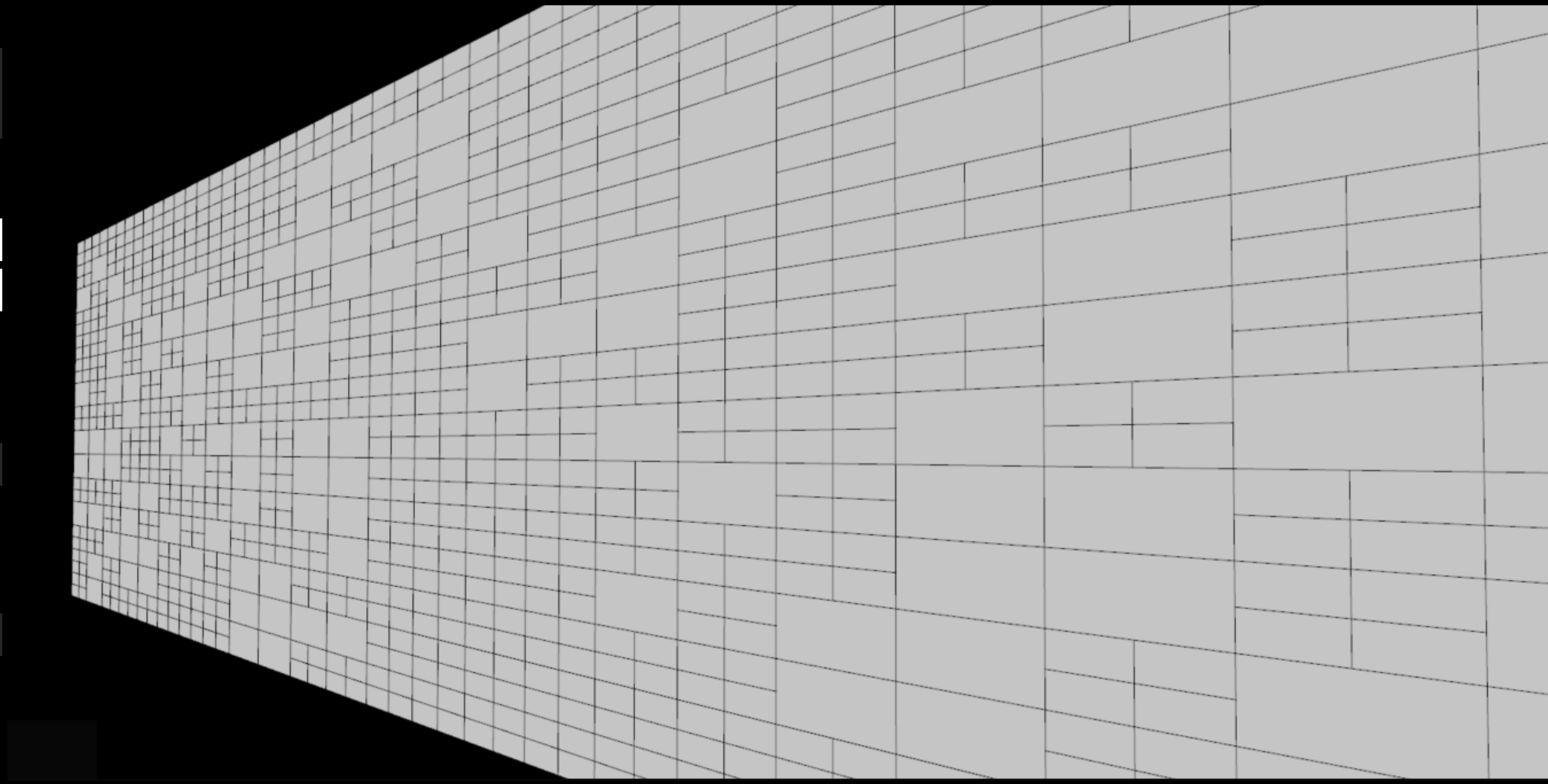
Pattern Development



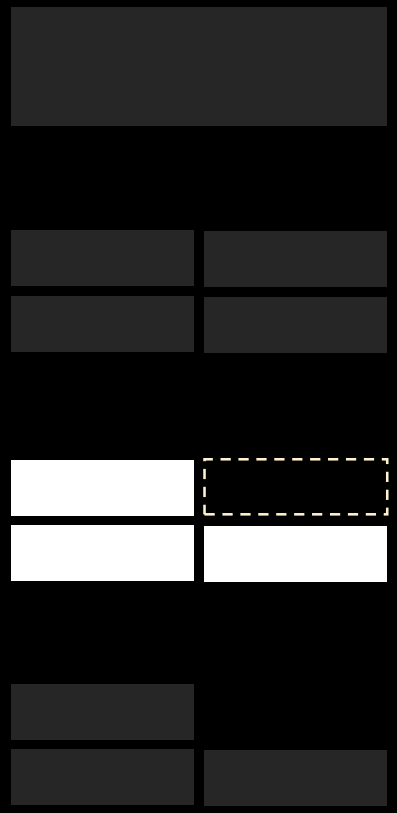


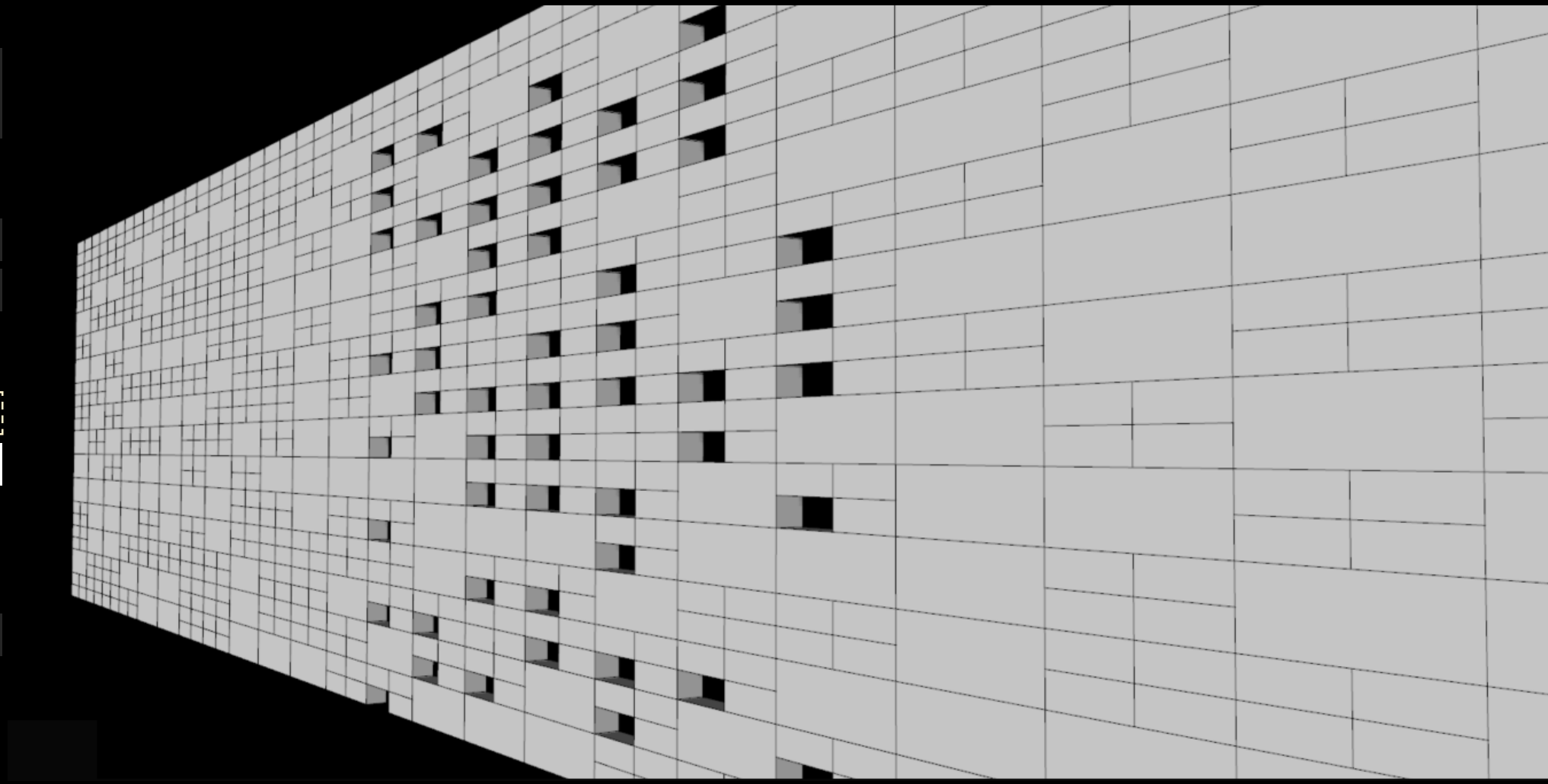
Pattern Development



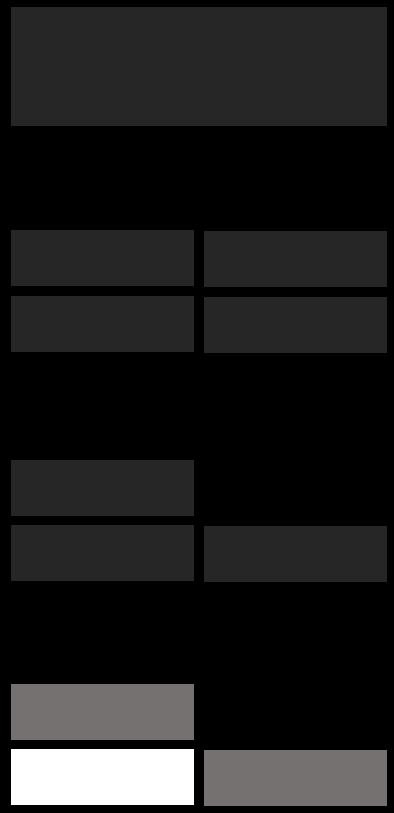


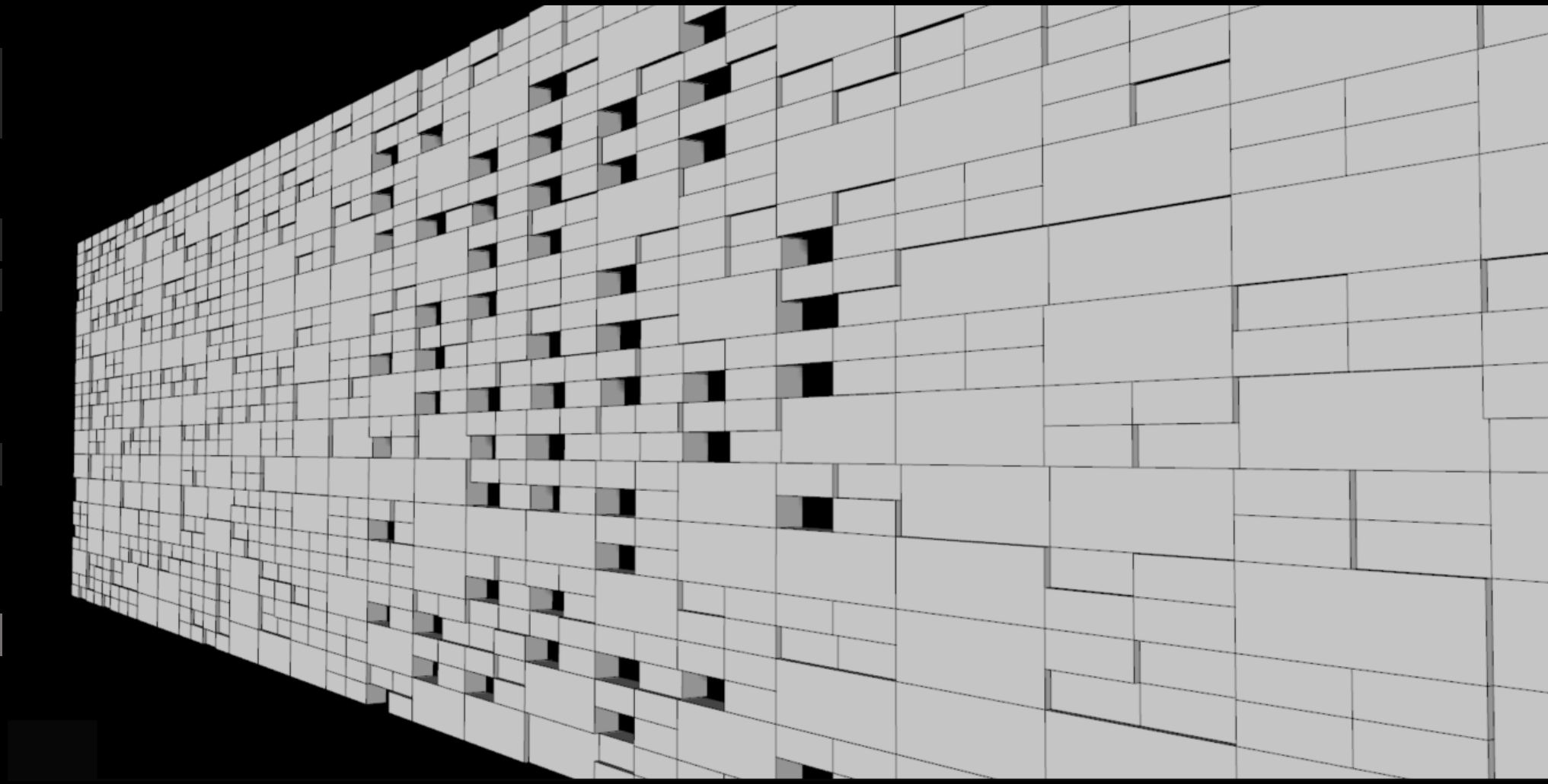
Pattern Development





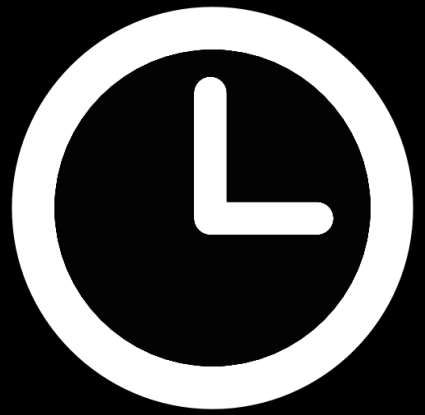
Pattern Development





CAD / BIM portability

CAD tool



QUICKLY GENERATE
SEVERAL SOLUTIONS

BIM tool

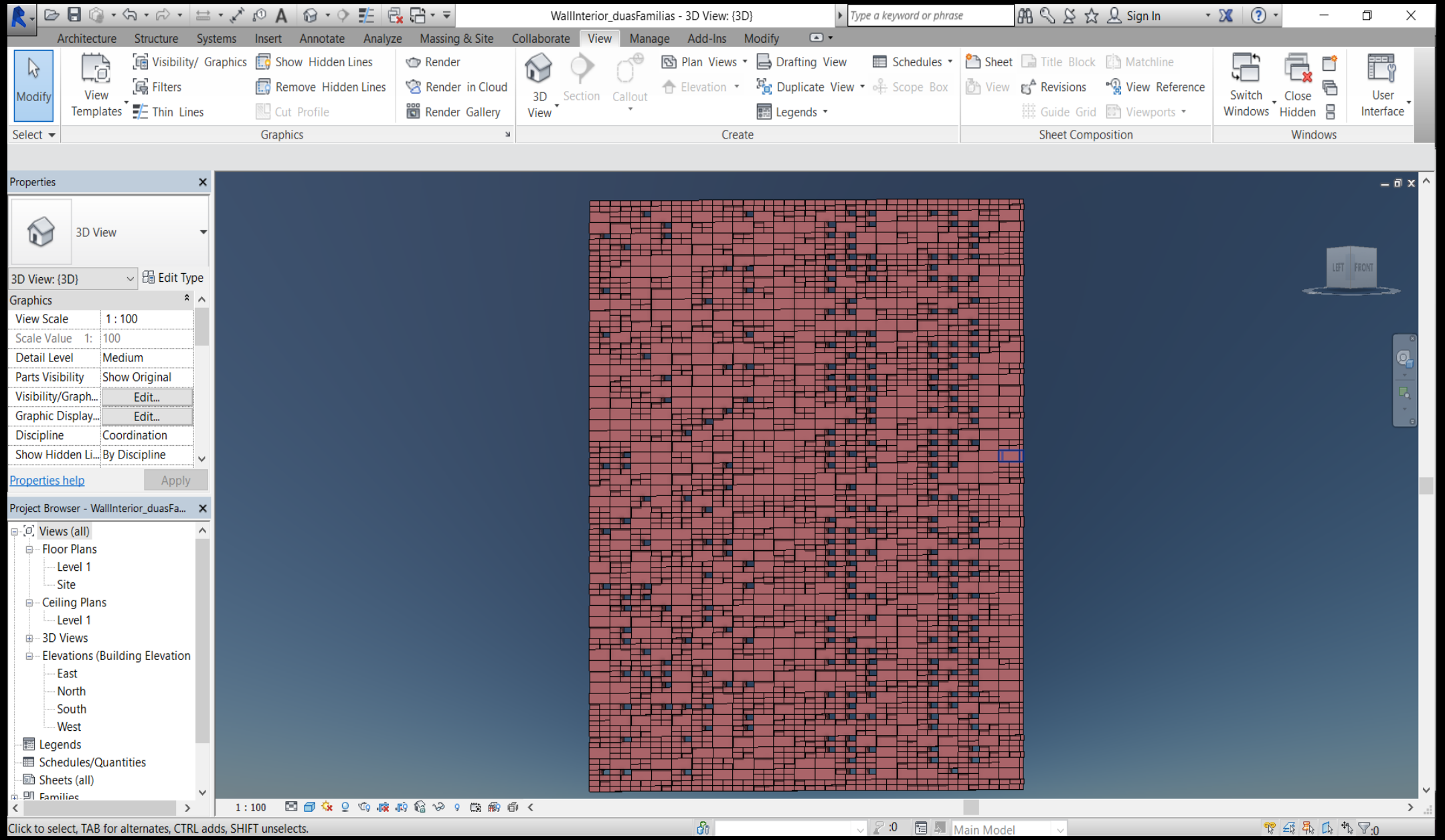
CAD / BIM portability

CAD tool

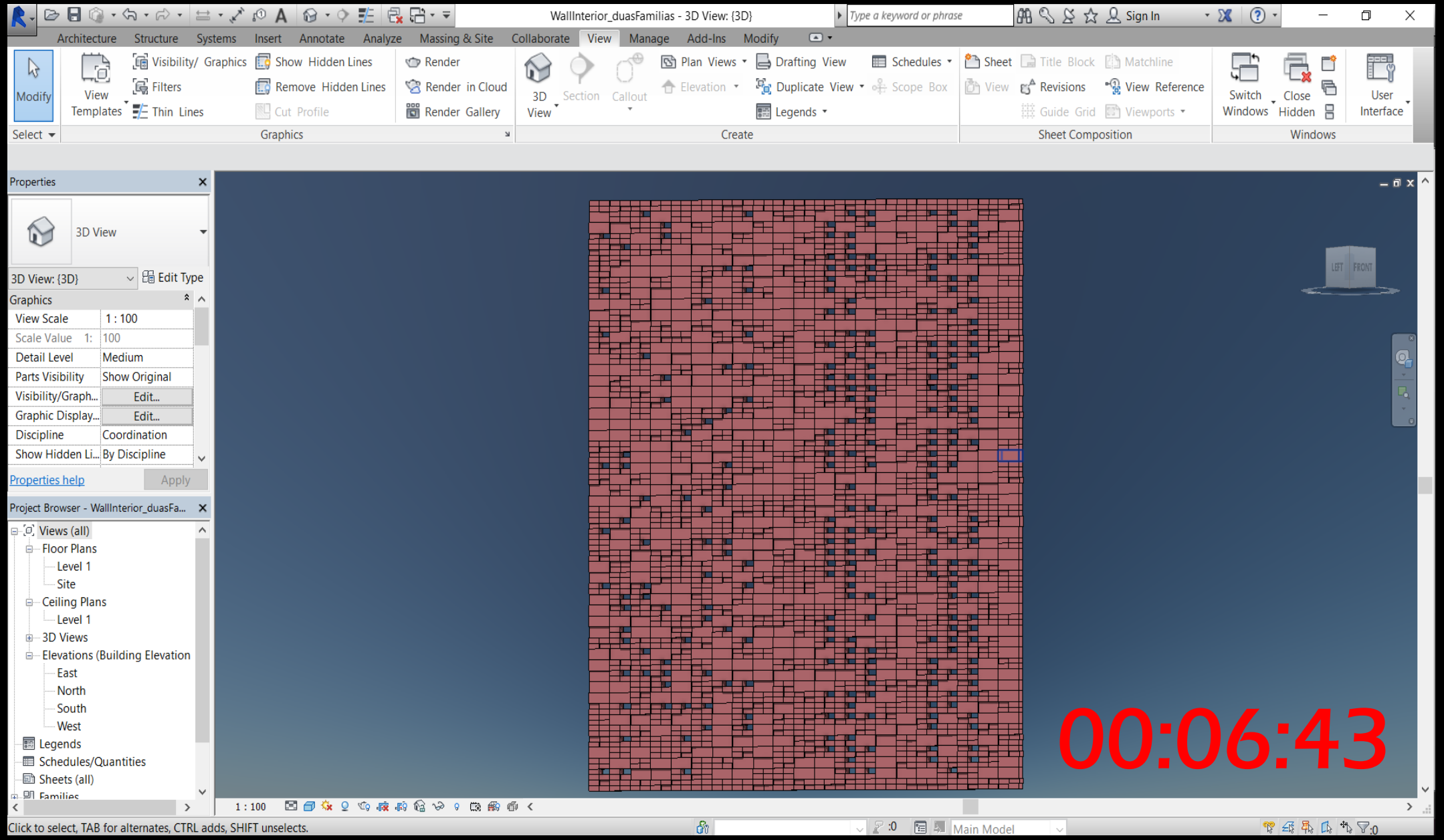
QUICKLY GENERATE
SEVERAL SOLUTIONS

BIM tool

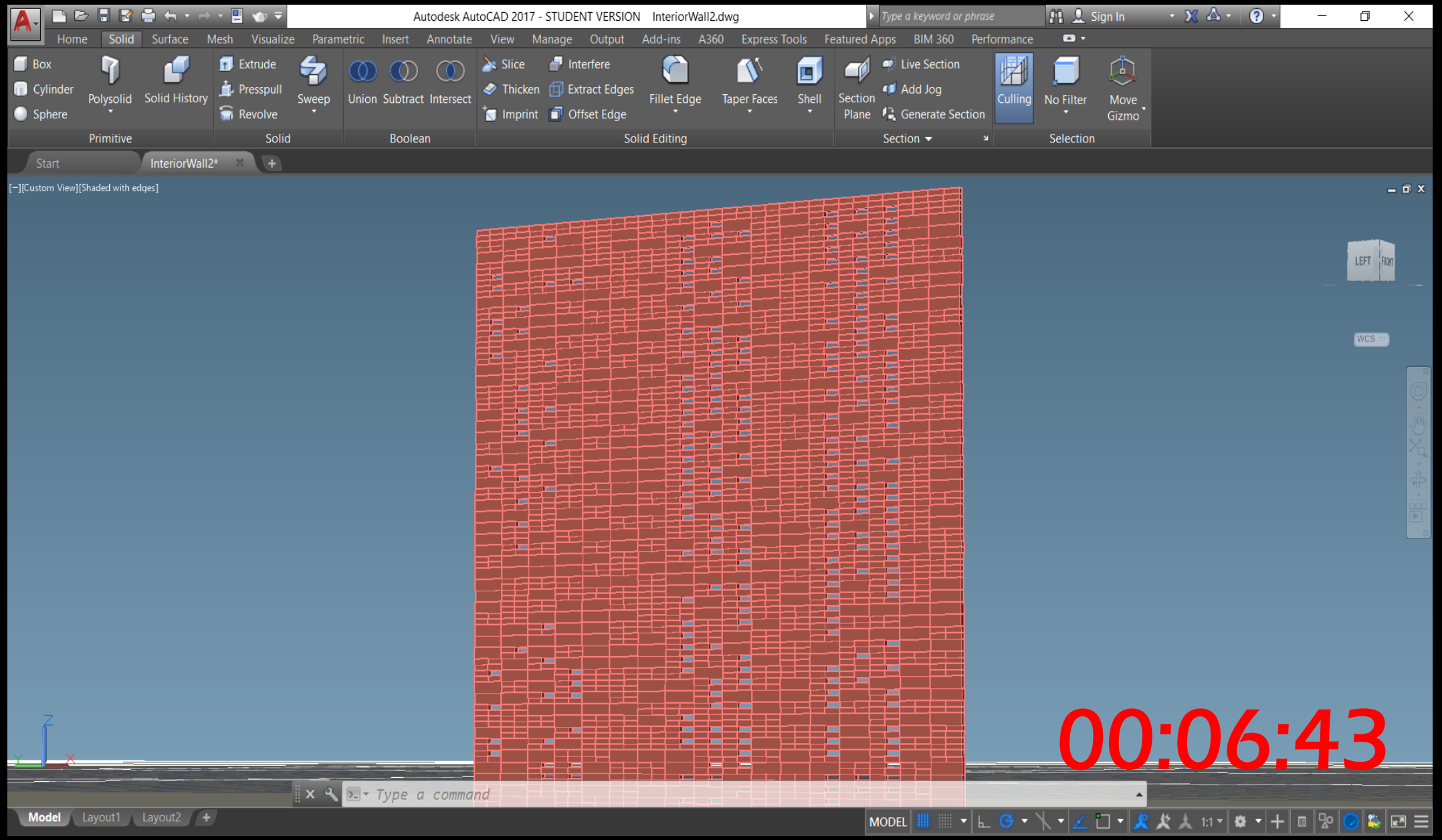
Pattern Development



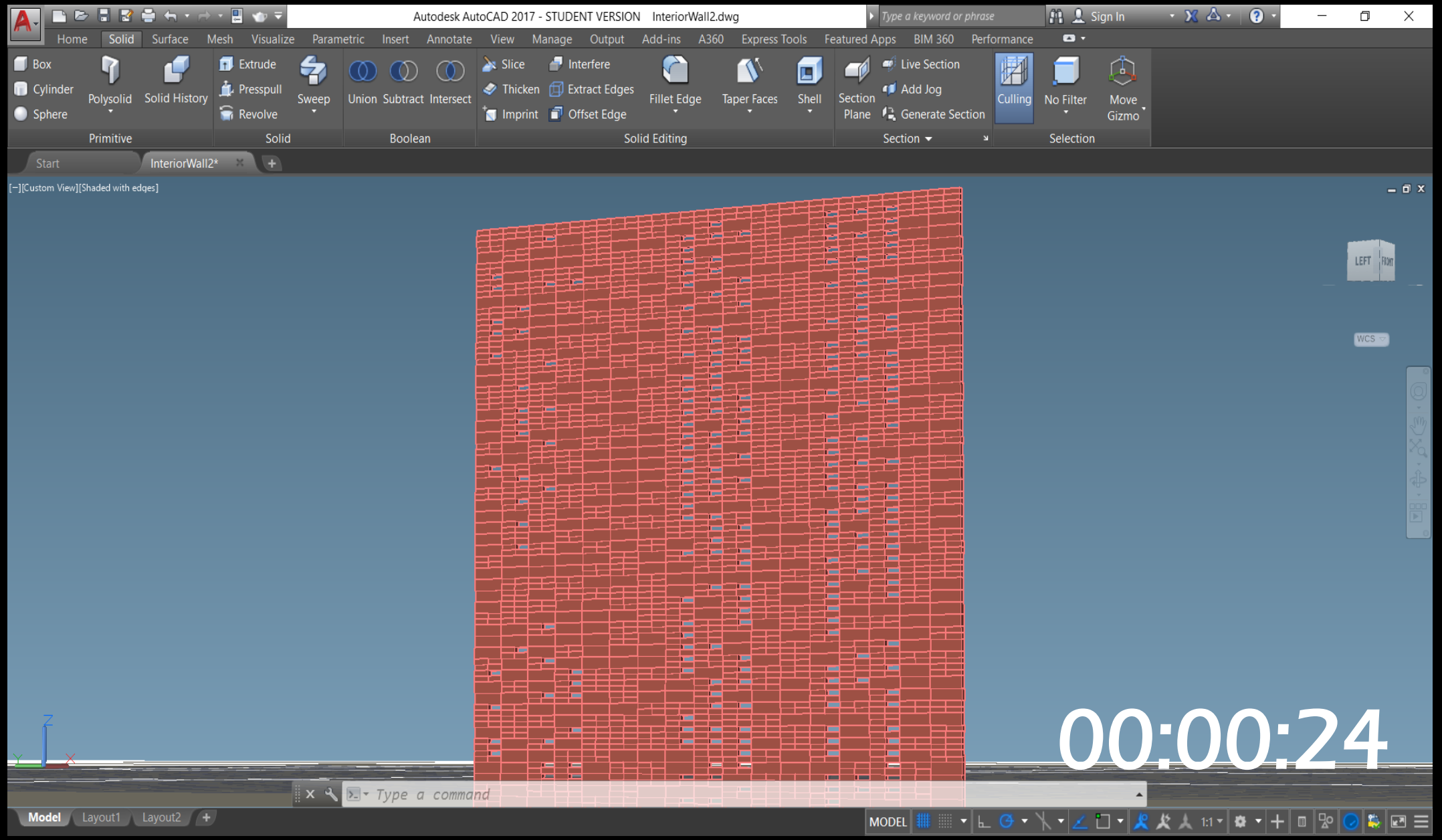
Pattern Development



Pattern Development



Pattern Development



CAD / BIM portability

QUICKLY GENERATE
SEVERAL SOLUTIONS

CAD tool

BIM tool

CAD / BIM portability

CAD tool

QUICKLY GENERATE
SEVERAL SOLUTIONS

BIM tool

CAD / BIM portability

CAD tool

QUICKLY GENERATE
SEVERAL SOLUTIONS

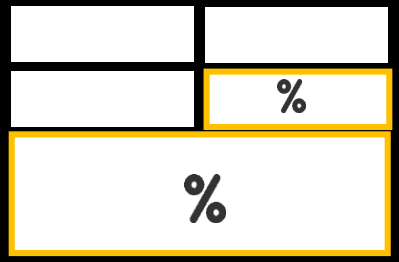
BIM tool

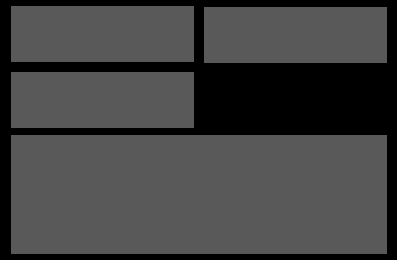
BIM families ✓

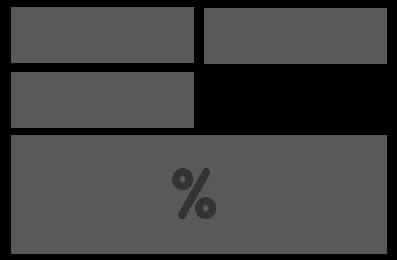
Information ✓

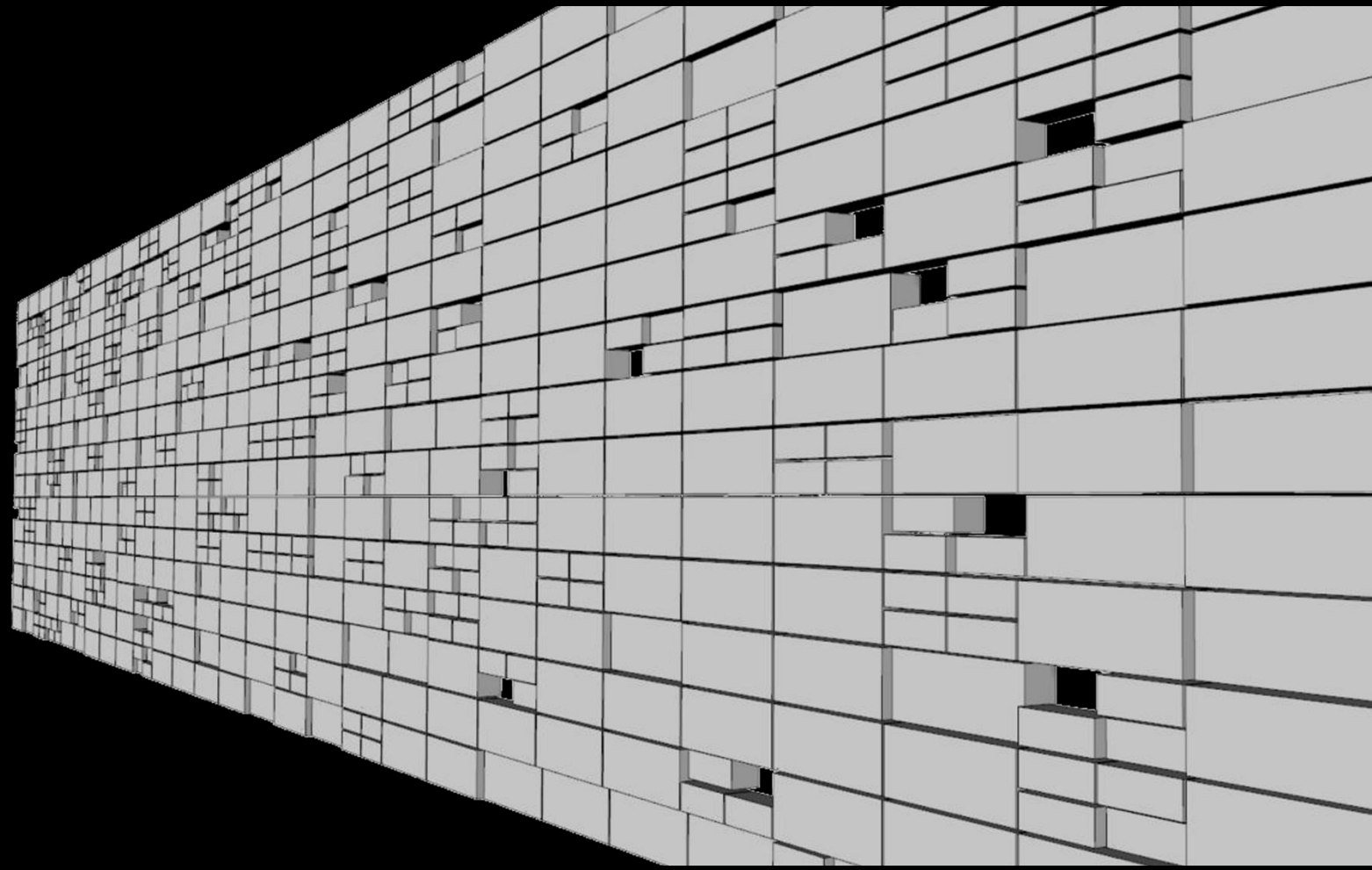
Dimensions ✓

Advantages of an algorithmic approach

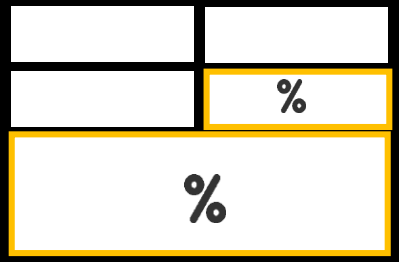


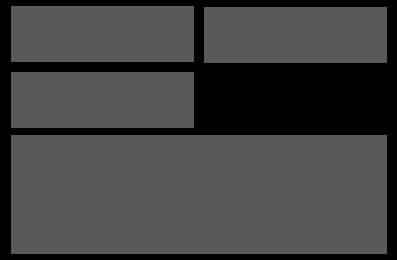


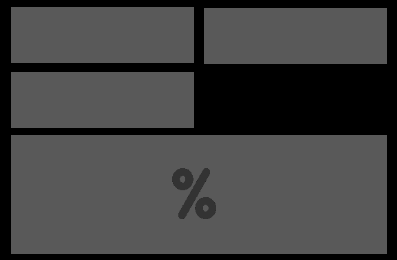


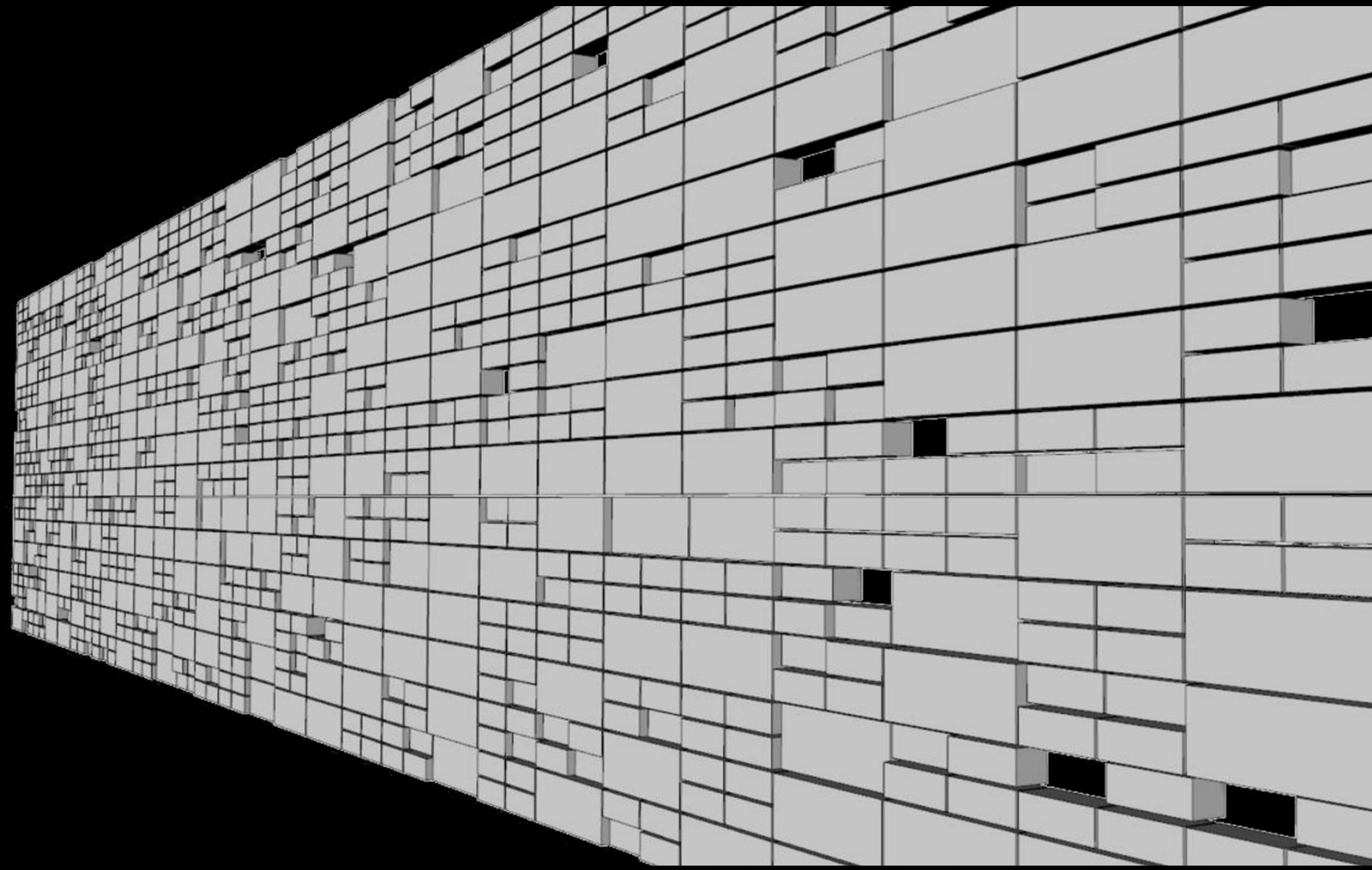


Advantages of an algorithmic approach

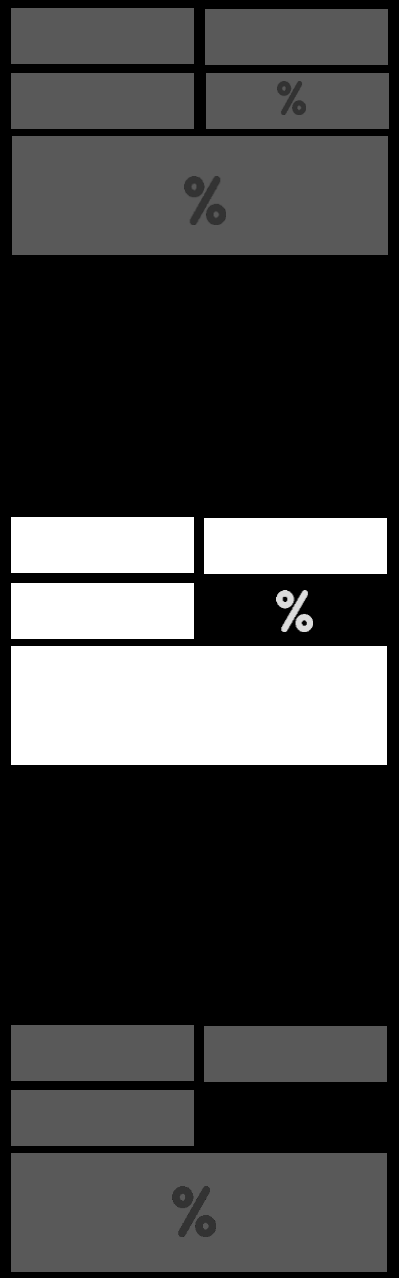


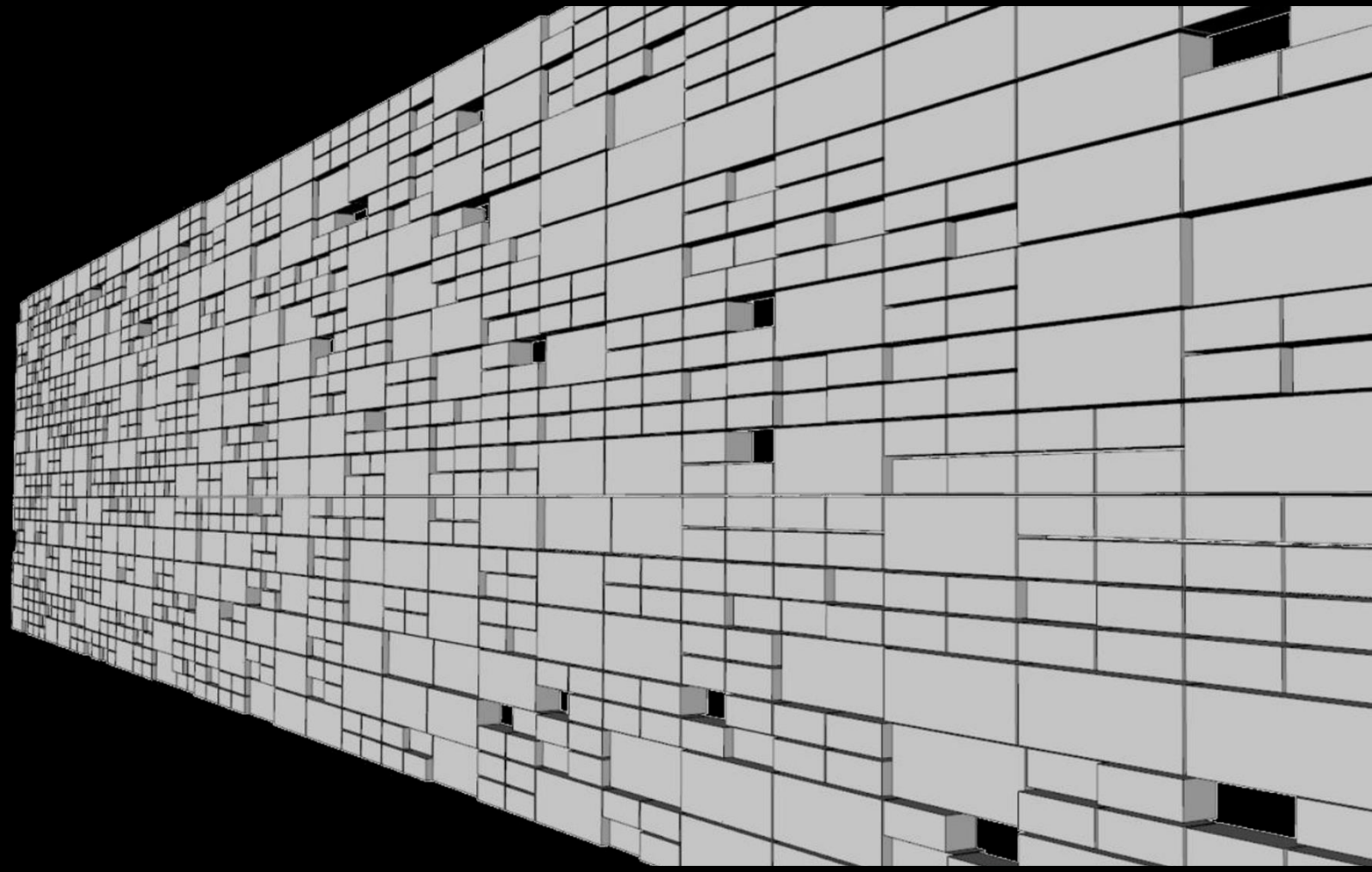




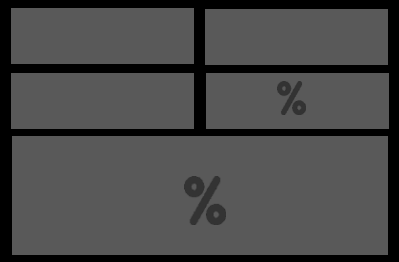


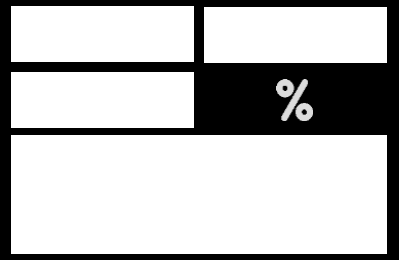
Advantages of an algorithmic approach

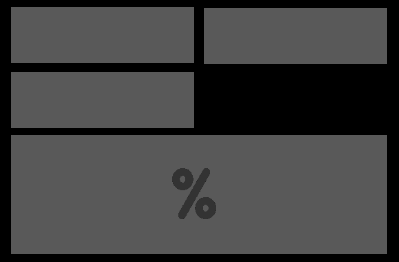


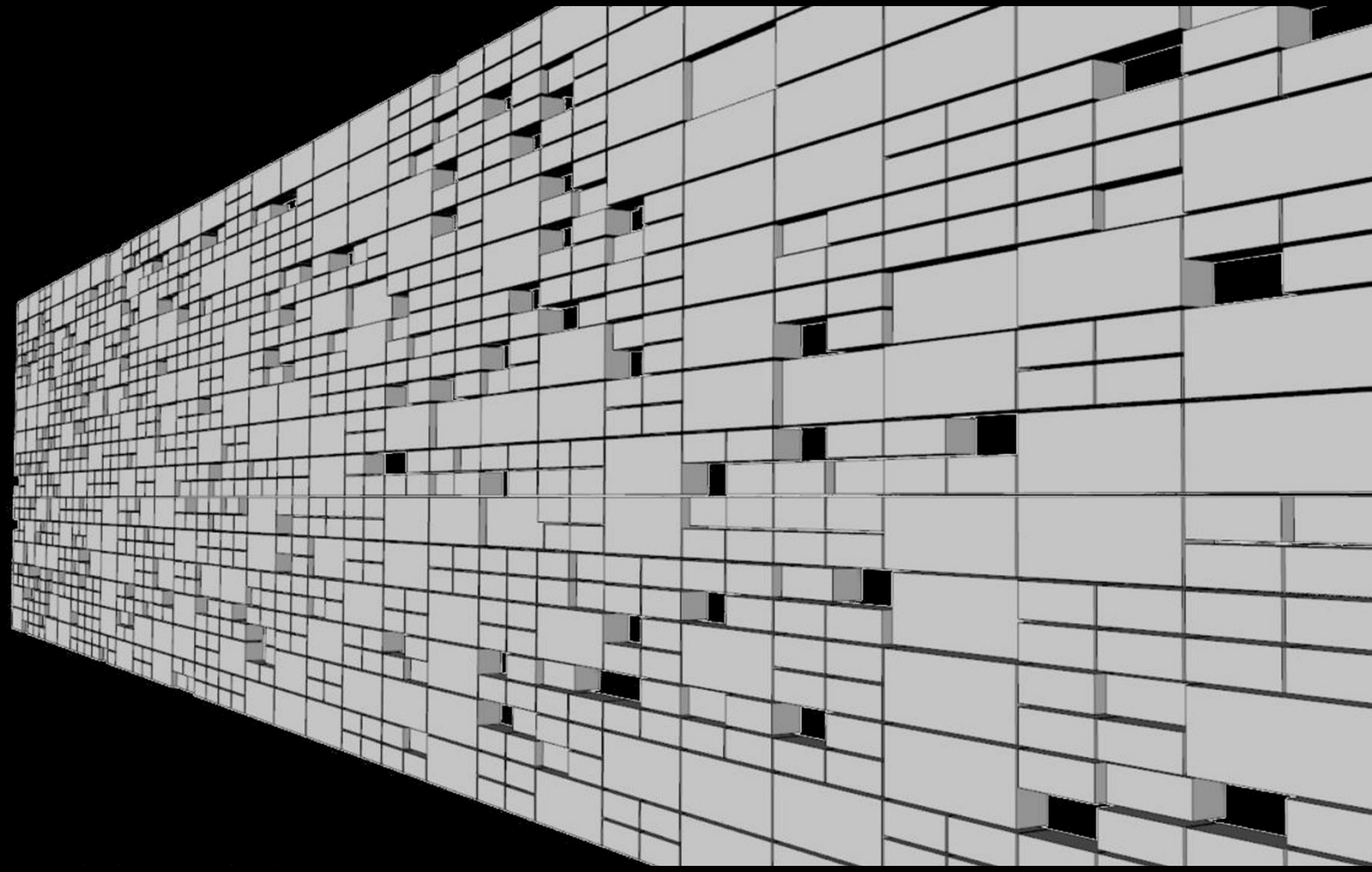


Advantages of an algorithmic approach

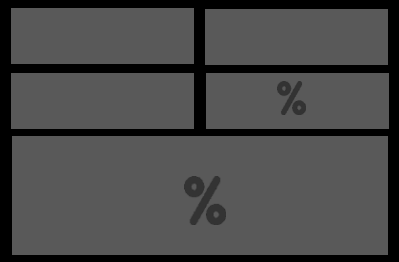


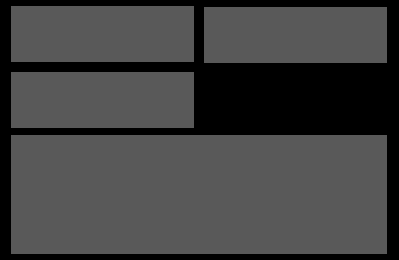


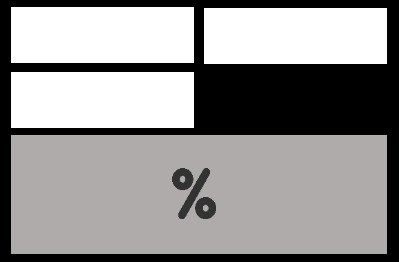


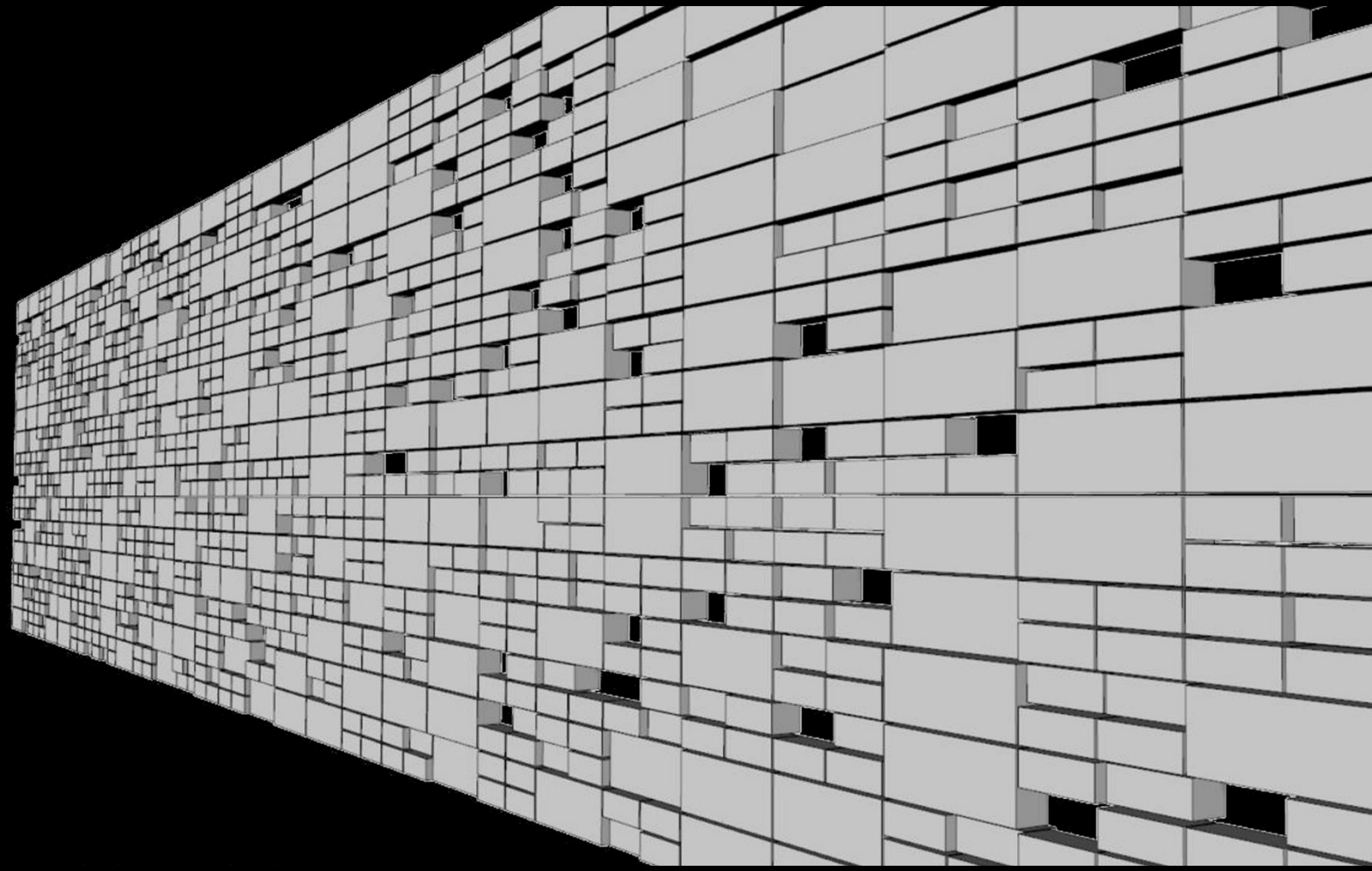


Advantages of an algorithmic approach

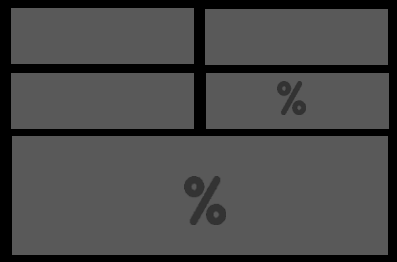


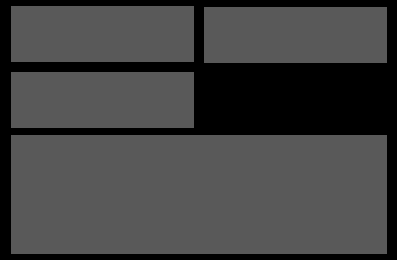


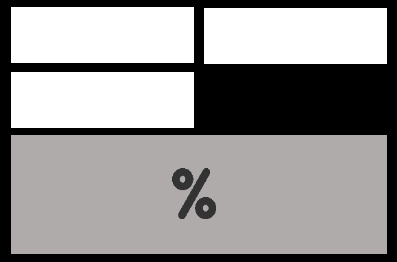


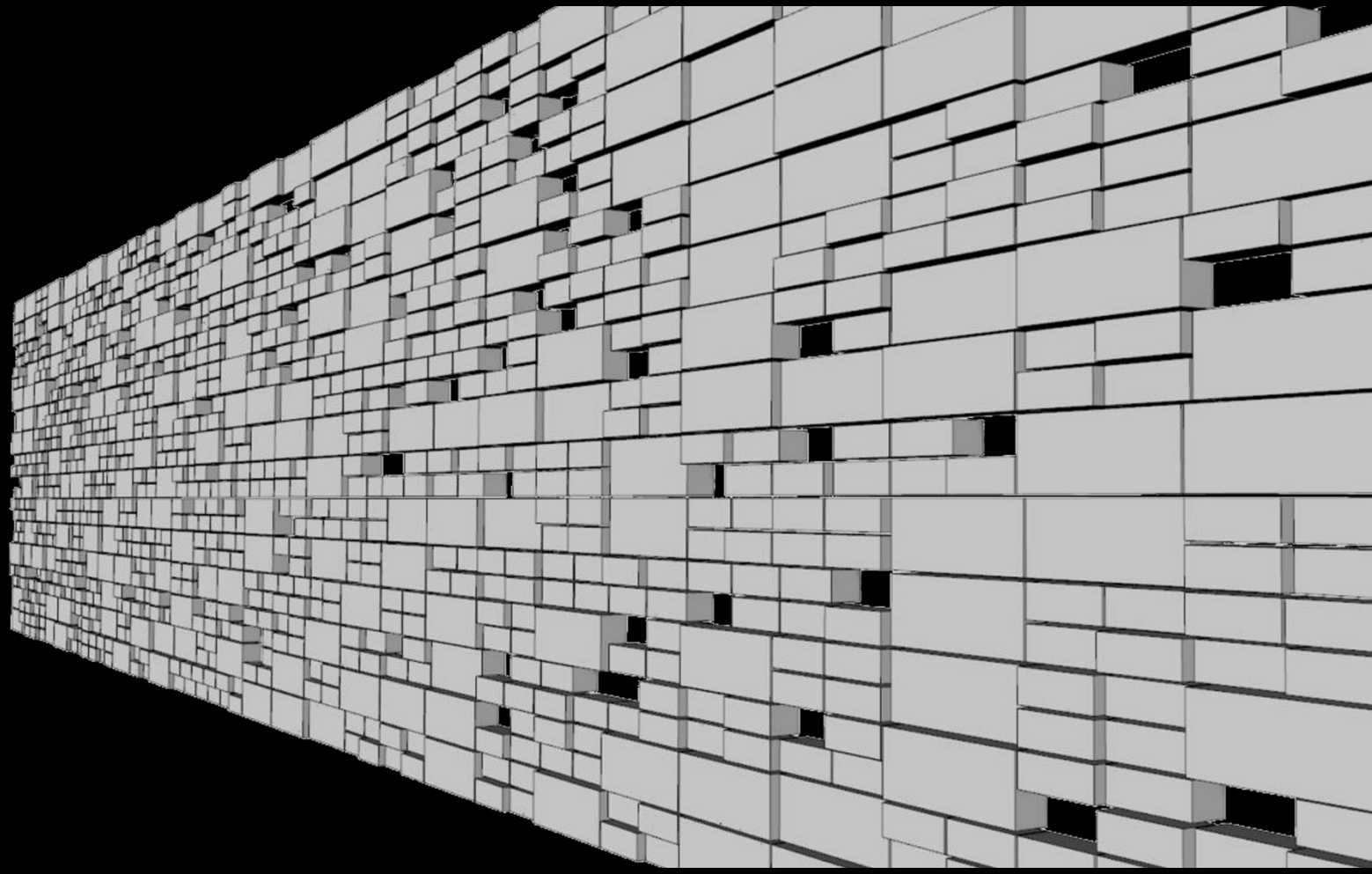


Advantages of an algorithmic approach

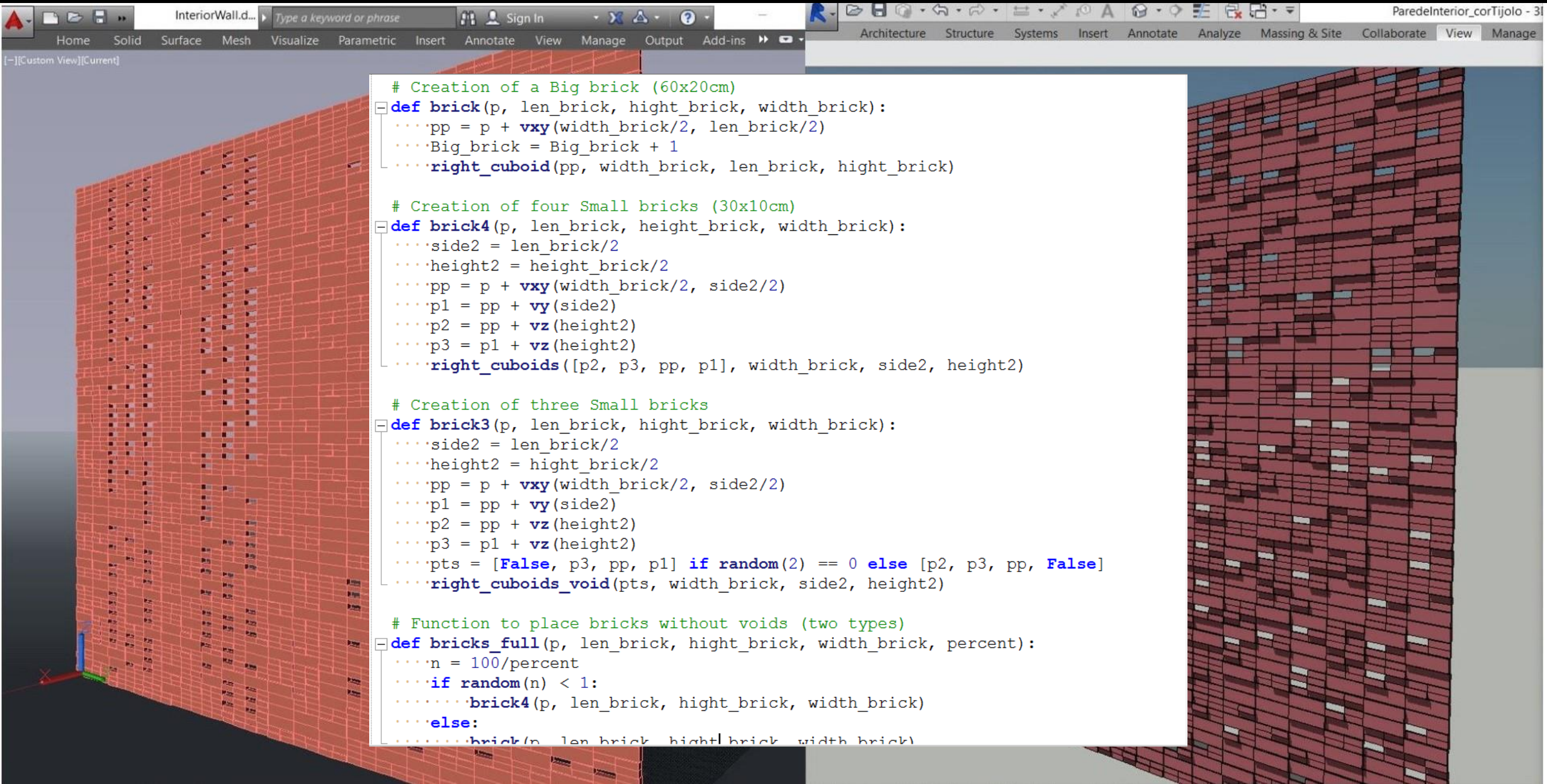








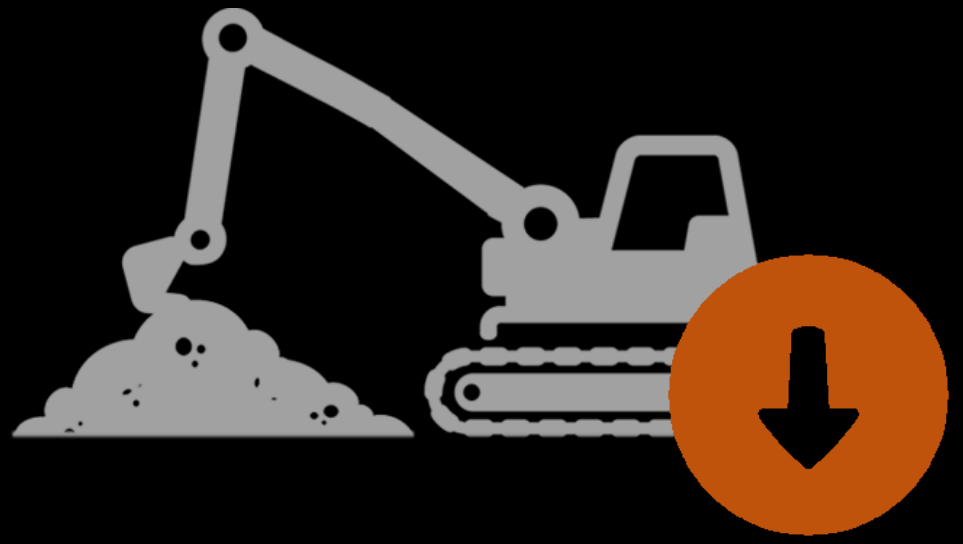
CAD / BIM portability



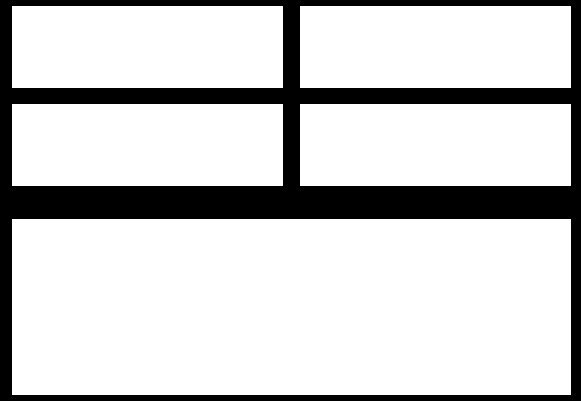

```
# Creation of a Big brick (60x20cm)
def brick(p, len_brick, hight_brick, width_brick):
    pp = p + vxy(width_brick/2, len_brick/2)
    Big_brick = Big_brick + 1
    right_cuboid(pp, width_brick, len_brick, hight_brick)

# Creation of four Small bricks (30x10cm)
def brick4(p, len_brick, height_brick, width_brick):
    side2 = len_brick/2
    height2 = height_brick/2
    pp = p + vxy(width_brick/2, side2/2)
    p1 = pp + vy(side2)
    p2 = pp + vz(height2)
    p3 = p1 + vz(height2)
    right_cuboids([p2, p3, pp, p1], width_brick, side2, height2)

# Creation of three Small bricks
def brick3(p, len_brick, hight_brick, width_brick):
    side2 = len_brick/2
    height2 = hight_brick/2
    pp = p + vxy(width_brick/2, side2/2)
    p1 = pp + vy(side2)
    p2 = pp + vz(height2)
    p3 = p1 + vz(height2)
    pts = [False, p3, pp, p1] if random(2) == 0 else [p2, p3, pp, False]
    right_cuboids_void(pts, width_brick, side2, height2)

# Function to place bricks without voids (two types)
def bricks_full(p, len_brick, hight_brick, width_brick, percent):
    n = 100/percent
    if random(n) < 1:
        brick4(p, len_brick, hight_brick, width_brick)
    else:
        brick(p, len_brick, hight_brick, width_brick)
```

Manufacturing Stage

Manufacturing Stage

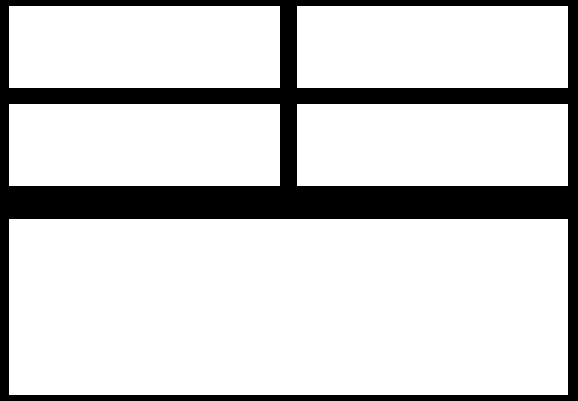


Manufacturing Stage



2 Sizes of Bricks

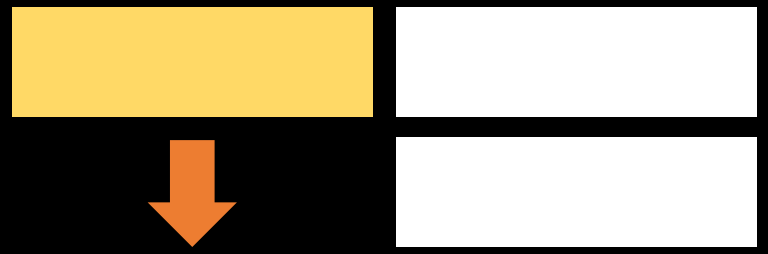
Manufacturing Stage



2 Sizes of Bricks

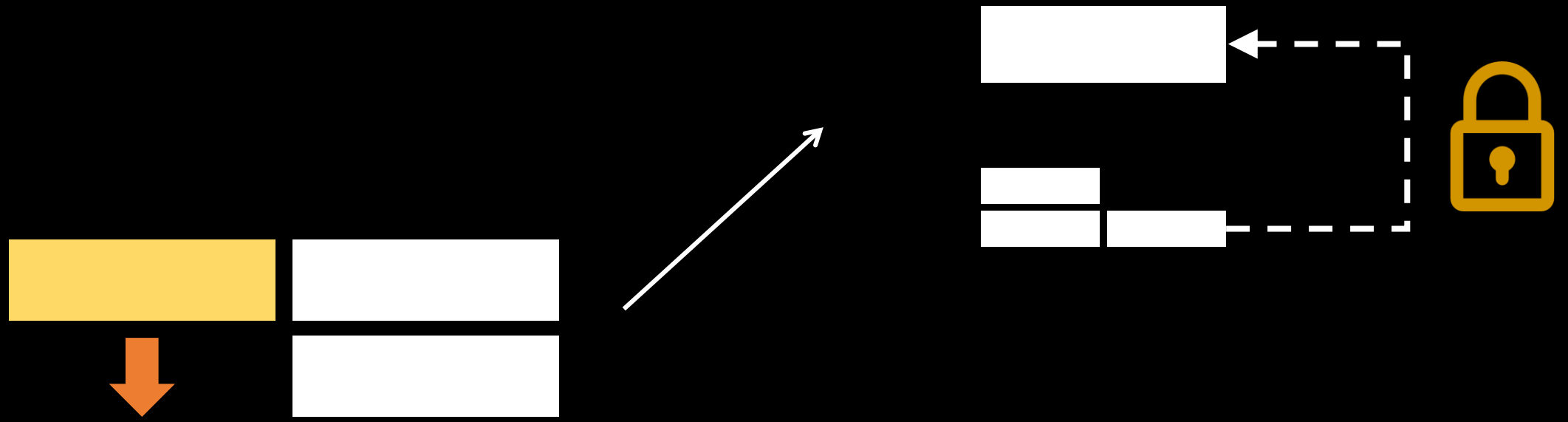
CHALLENGES

Façade voids



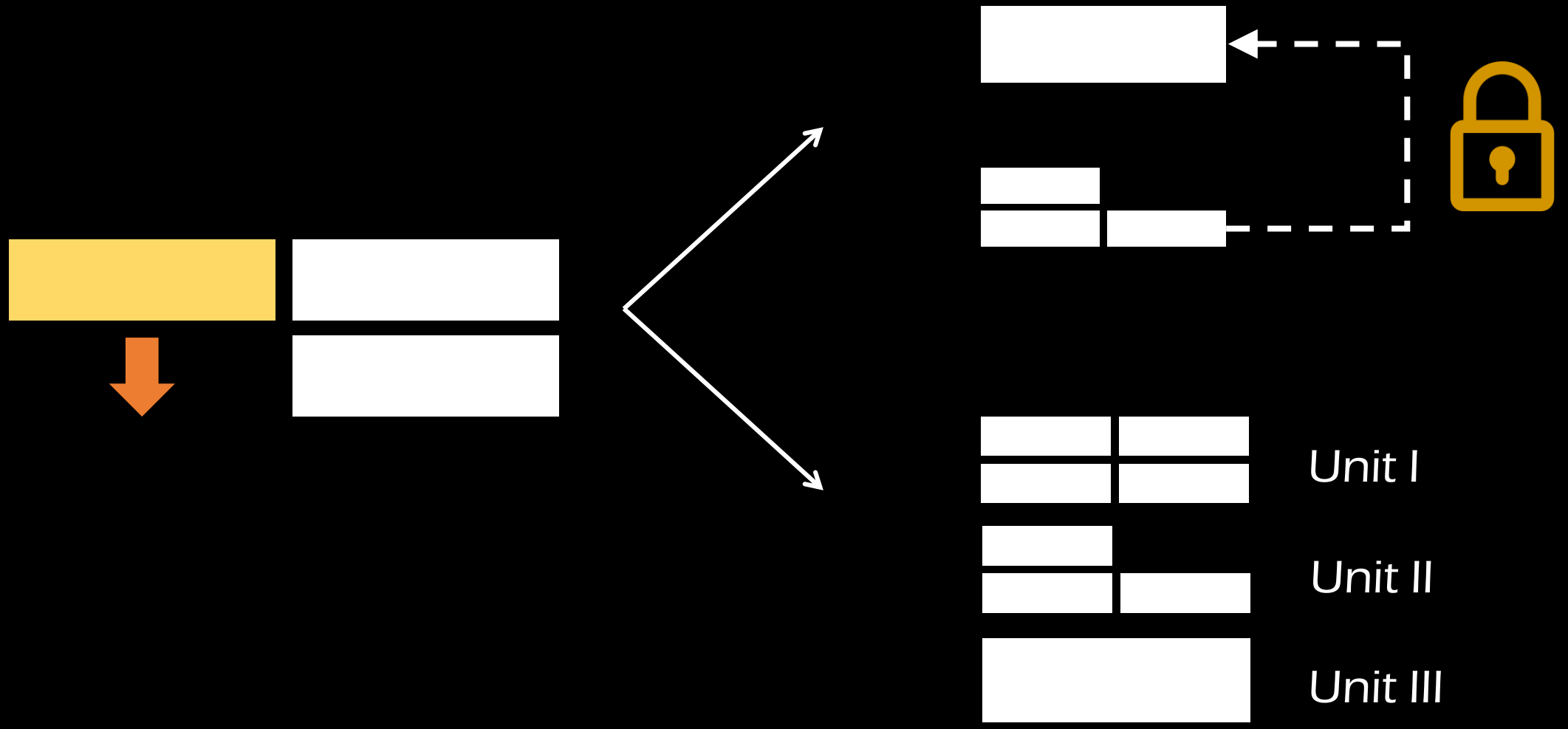
CHALLENGES

Façade voids



CHALLENGES

Minimize number of moulds



CHALLENGES

Minimize number of moulds

POSSIBLE VARIATIONS

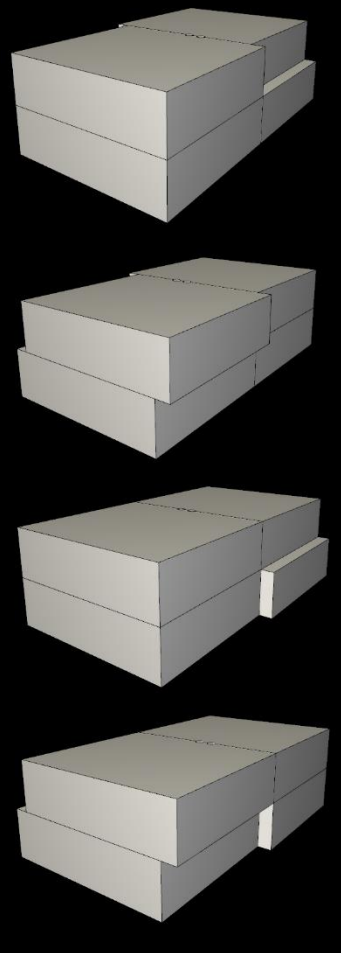
TYPOLOGY

A

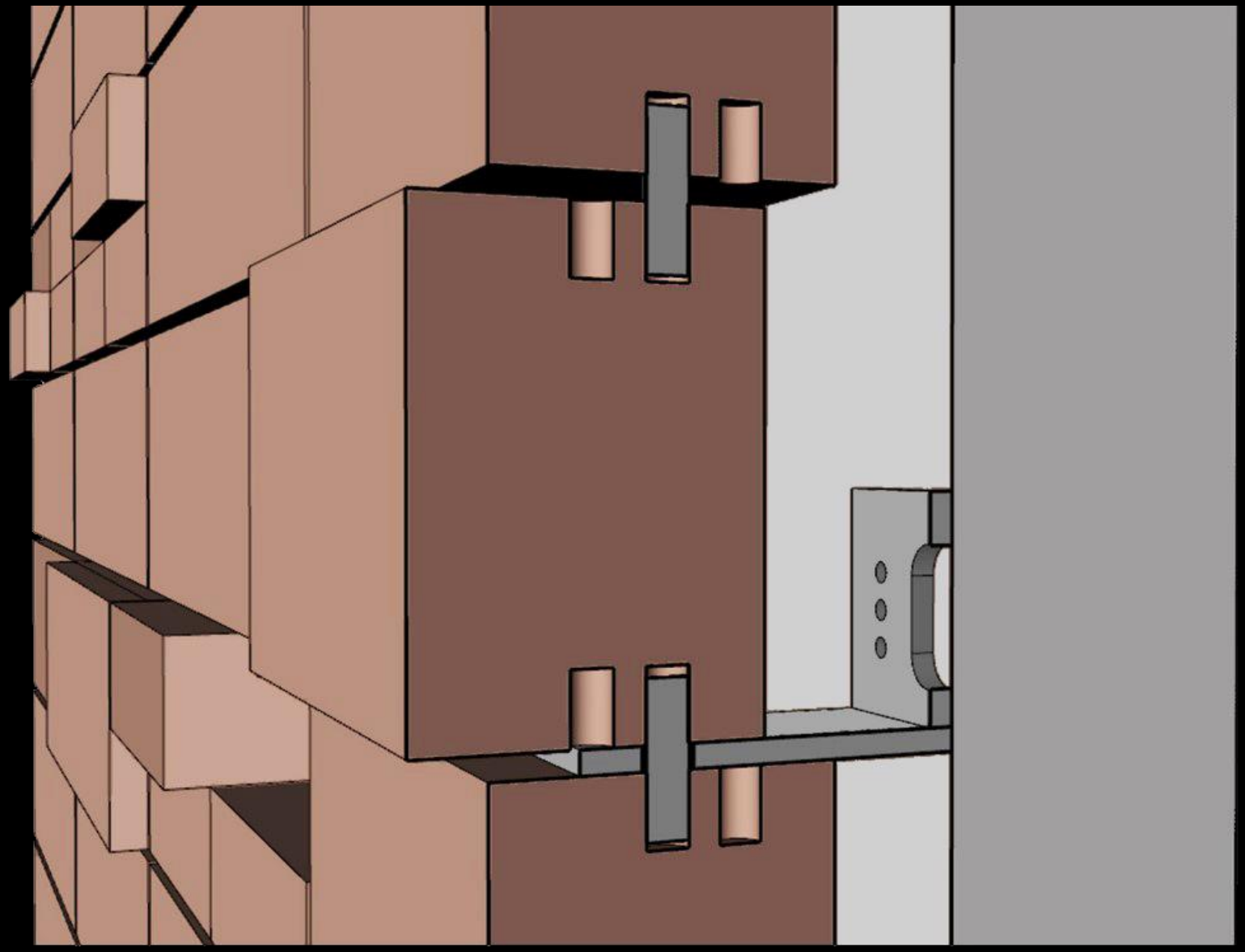
B

C

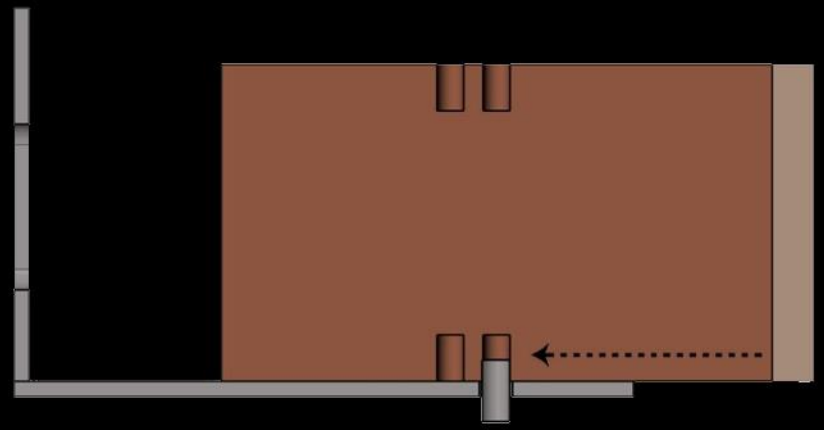
D



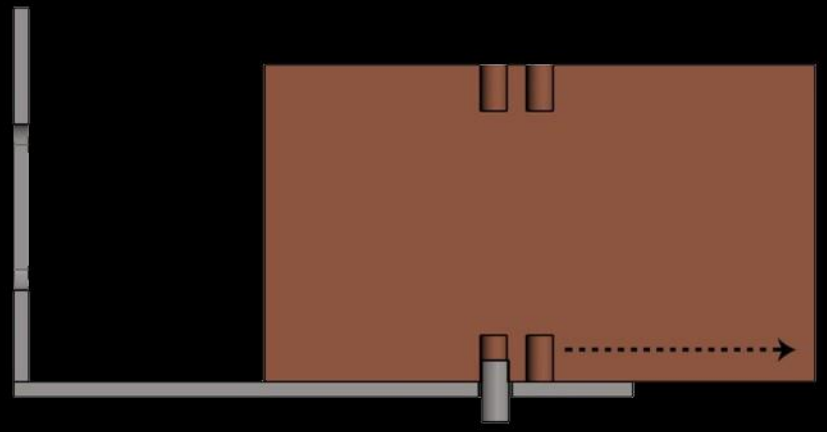
PLACEMENT OF THE BRICKS



PLACEMENT OF THE BRICKS



PLACEMENT OF THE BRICKS



TALLYING FOR FABRICATION

TYPOLOGY	QUANTITY
	1619
	387
	224
	224
	45
	51
	61

TYPOLOGY	QUANTITY
	385
	99
	20
	28
	95
	28
	100

REGULAR BRICK

PROTRUDED BRICK

CONCLUSIONS

Advantages of an algorithmic approach

QUICKER CHANGES

MORE DESIGN OPTIONS

IMPROVED RESULTS

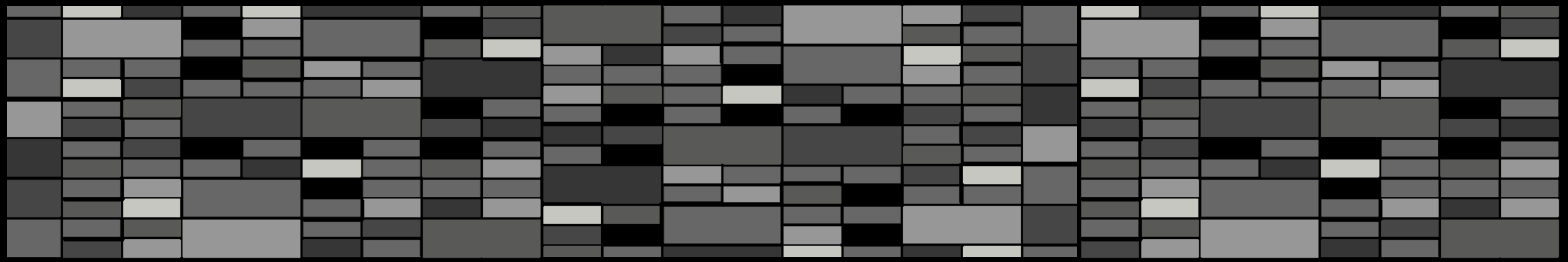
BETTER DESIGN PRECISION

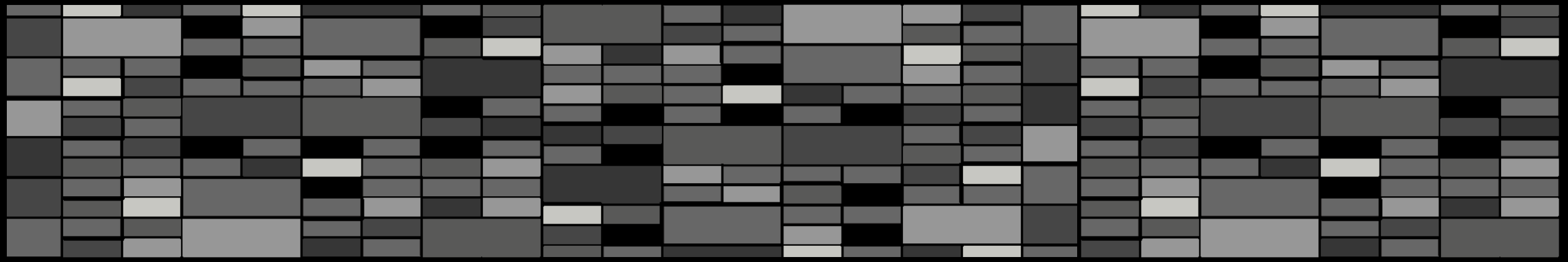
LESS EFFORT

SHORTER TIME

FEWER MODELING LIMITATIONS

FEWER REPETITIVE TASKS



THANK YOU

Inês Caetano ines.caetano@técnico.ulisboa.pt
António Leitão antonio.menezes.leitão@técnico.ulisboa.pt