

JOSHUA PARKER

SELECTED WORK

2007 -

JOSHUA PARKER, M. ARCH., SYRACUSE  
UNIVERSITY SCHOOL OF ARCHITECTURE (SUSOA),  
B.S. ELECTRICAL ENGINEERING, UNIVERSITY OF  
WASHINGTON (UW), CERTIFICATION, INSTITUTE OF  
ADVANCED ARCHITECTURE OF CATALONIA (IAAC)

## CONTENTS

### ZHUHAI PORT PAVILION

5

Project Scope: Architectural Design of City planning museum for Zhuhau Port Authority | Collabotors: Ingame Architecture, Yang Zhang (principal in charge) | Role: Consultant for Ingame Architecture, Lead designer | Contribution: Concept/schematic design, computational design, design development

### UPLIFT AIRSHIP HANGER

9

Project Scope: Design Competition, Airship Hanger, Anhui, China | Collabotors: OPEN Architecture, Chinese Academy of Building Research (CABR), LI HU (principal in charge) | Role: Consultant for OPEN, Lead designer | Contribution: Concept/schematic design, form-finding, computational design

### PROTOCELL

15

Project Scope: metadesign, collective housing system | Collabotors: OPEN Architecture, Li Hu | Role: Lead designer, researcher

### CHENJIACI CULTURAL VILLAGE AND PARK REDEVELOPMENT

21

Project Scope: Redevelopment of Cultural park & Square of Chan's Ancestral Temple, Guangzhou, China | Collabotors: Node Architecture and Urbanism, Doreen Liu (principal in charge), Jiebin, Huang, Shaoli Lin | Contribution: Concept/schematic design, masterplan for landscape/park nodes | Role: consulting architect for Node Architecture and Urbanism, in charge of landscape nodes

### GUANGXI MUSEUM OF FINE ARTS

25

Project Scope: Design / Analysis of Building envelope | Collabotors: NODE Architecture and Urbanism | Roles: Design Consultant for NODE | Contributions: Concept/schematic design, massing, incharge of landscape and building envelope.

### SPIRALLING: PARKING GARAGE

27

Scope: Custom parametric Tool for parking garage study | Collabotors: OPEN Archtiecture, LI HU | Contribution: Consultant for OPEN Architecture, Computational designer, form-finding

### WEAVING: URBAN FURNITURE / SHADING

29

Scope: Custom parametric Tool | Collabotors: Kristiana Leniart | Contribution: Computational design

### SAMPLING: JAZZ TOWERS, BAOTOU

31

Scope: residential towers | Collabotors: OPEN Archtiecture, LI HU (Principal in charge) | Role: Consultant for OPEN | Contribution: Custom design tool

### SOURCE CODE, UPLIFT HANGER

33

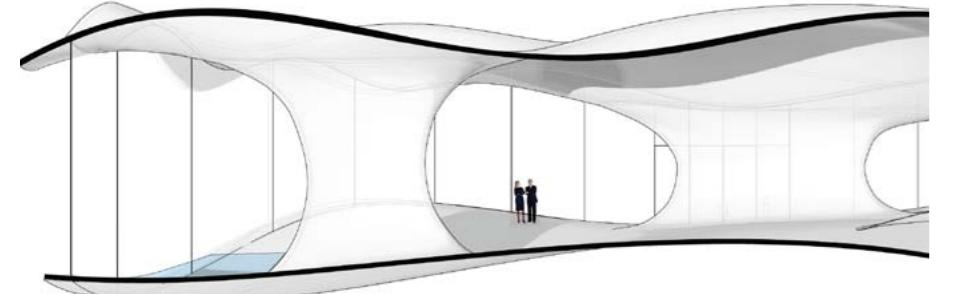
## ZHUHAI PORT PAVILION

Pavilion to Showcase the 5-10 year development plan for the 7 port areas in Zhuhai. The plan of Gaolan Port is at the center of the display with the Wanshan Port being a supplementary. Requirements include dark rooms with circular model spaces 15m and 4.5m in diameter, Large display wall, cinema, lounge and general public space. Design to invoke the ocean and port and Zhuhai's position as a international port.

Project asked for a formal response to "sea and port". Continuous free-form surface provide language and medium to produce spaces evocative of flowing sea water while guiding people through continuous sequences of soft expansions and contractions that serve as exhibition space. the visitor experience is of being underwater. the language and meaning of port is also incorporated to describe moments where the undulating ground plane connects with ceiling to enclose ovoidal spaces that become opportunities for office space, bathrooms, cinema, etc.

---

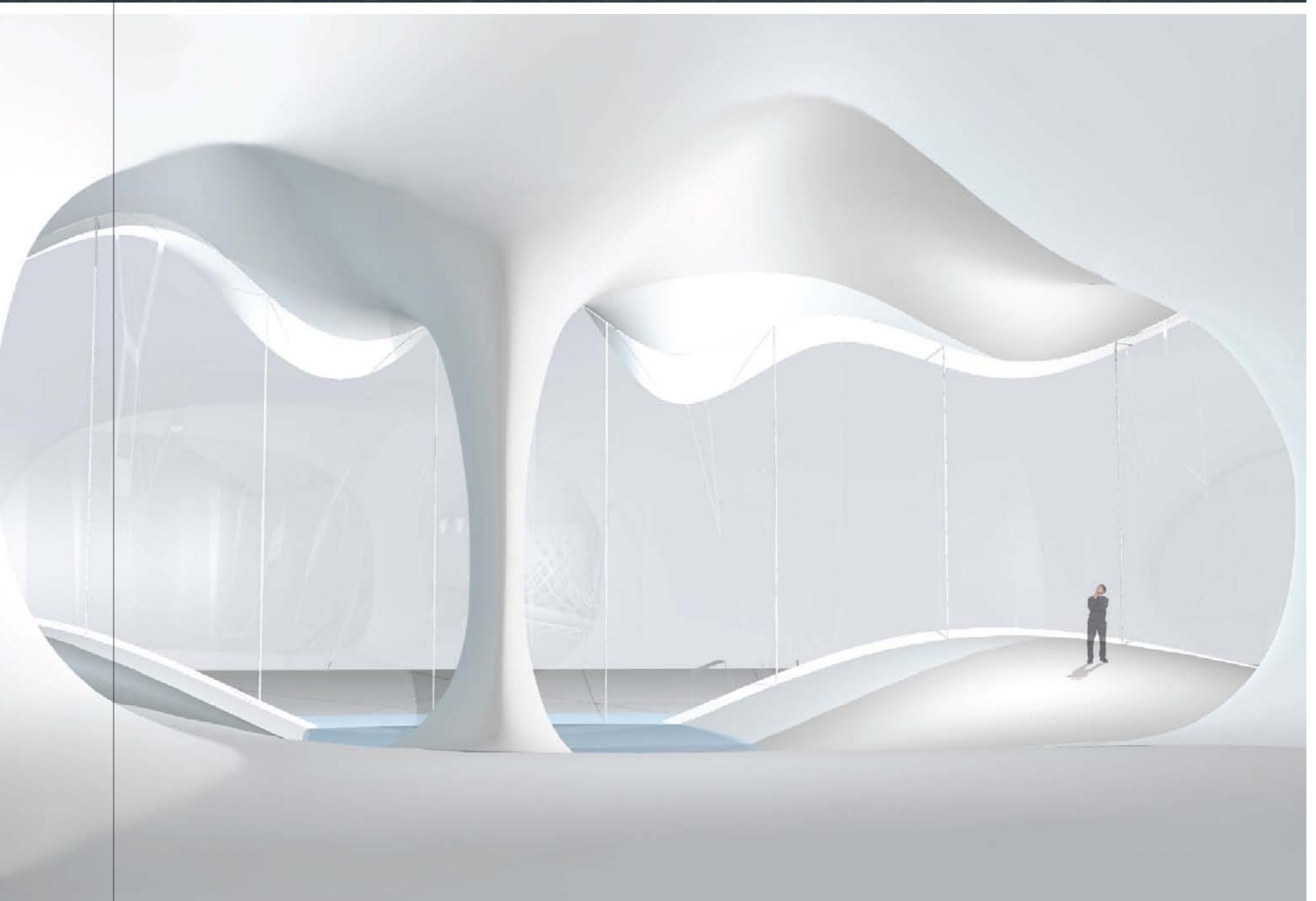
Project Scope: Architectural Design of City planning museum for Zhuhai Port Authority |  
Collaborators: Ingome Architecture, Yang Zhang (principal in charge) | Role: Consultant for Ingome  
Architecture, Lead designer | Contribution: Concept/schematic design, computational design, design  
development



1.

1. diagrammatic building section  
2. contour in plan, site area is approx. 4,000 square meters.  
opposite: Renderings from street and interior view.

images and text courtesy of Ingome office

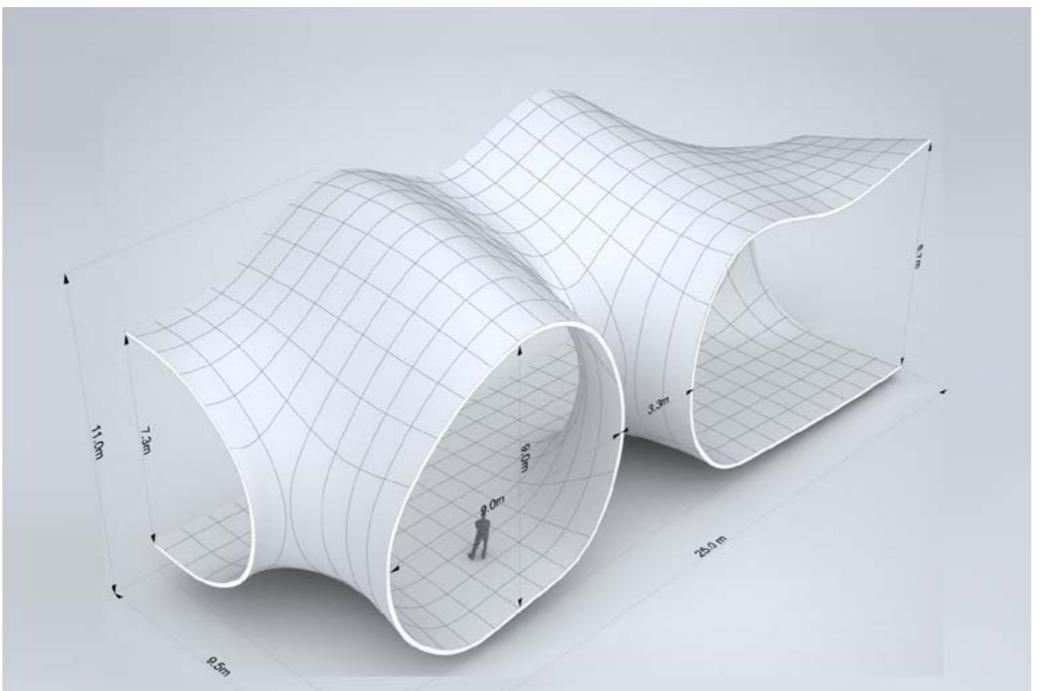


Program was divided into two groups. For the most part public activity space, lounge and exhibition occupies the main interstitial space. Remaining program: offices, bathroom, cinema, dark rooms, etc. occupies ovoidal spaces created by surface deformations. Design was a process of placing circular program in plan, taking the centers, and radius, as location and size of surface ports. A port pulls ground and roof planes together, effectively disturbing the surface as it self adjusts according to structural criteria. The surface, in turn, pulls back stretching and elongating the circles, creating ovoidal spaces. The cumulative effect of all ports is a heavily undulated surface pair. The frequency and amplitude of irregular waveforms depends on incremental spacing of ports and ground ceiling height.

The thin-shell structure is realizable with reinforced concrete shells and plywood formwork. This has proved to be an economical and efficient solution used by free-form structures in recent years, most notably by Mutsuro Sasaki working in collaboration with Toyo Ito. Of particular relevance as design and structural precedent was the Kakamigahara Crematorium. Based on design precedents and structural heristics, we can anticipate surface thickness of 200 mm. This creates an exceptionally thin and light structure (see img below). The large overhang is additionally supported by steel columns that interrupt otherwise continuous glass infill that wraps the structure at the edge minus an offset that also varies according to structural requirements.

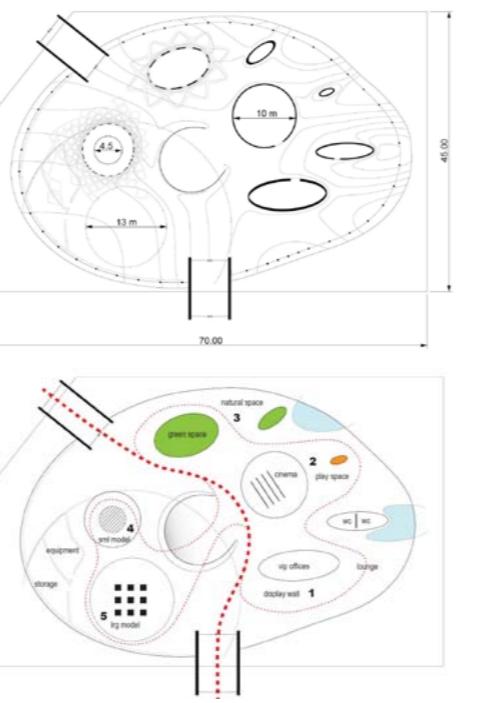
The structure would require variable reinforcement to support the distributed vertical load of approximately 1 KN/m<sup>2</sup>, as well as the horizontal wind and seismic loads. The arrangement of reinforcement can be determined with calculus-based optimization software, which calculates incremental compression and bending forces along surface, then produces a reinforcement map. According to Sasaki, the ideal construction solution also includes steel plates to better hold the form and counter the loss of rigidity due to cracking. He describes a "hybrid construction of steel plates and concrete". Plates would act both as formwork for concrete and then fixed in place for a hybrid cross section.

A custom paneling tool was built with java/processing using open source geometry library (igeo, by Saturo Sugihara) to interrogate form and explore the possibility of perforating the shell in places to create visual and physical access through vertical surfaces. Tool defines a uv diagram and maps performance patterns onto surface using simple algorithms and exposing design parameters and enable real-time collaborative use by architects and engineers.

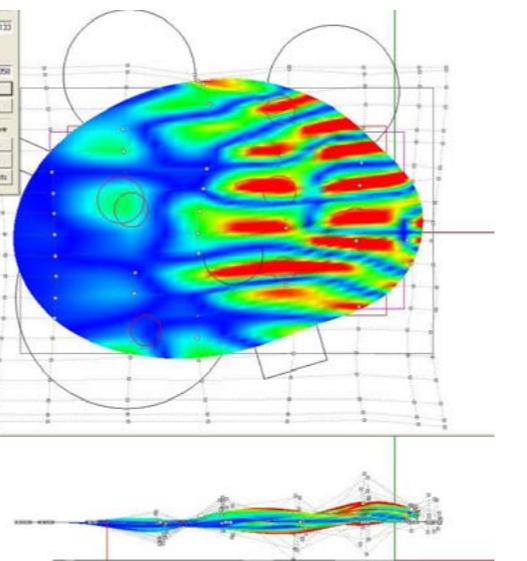


7

Below: floor plan contours, and programmatic relation to primary circulation paths.

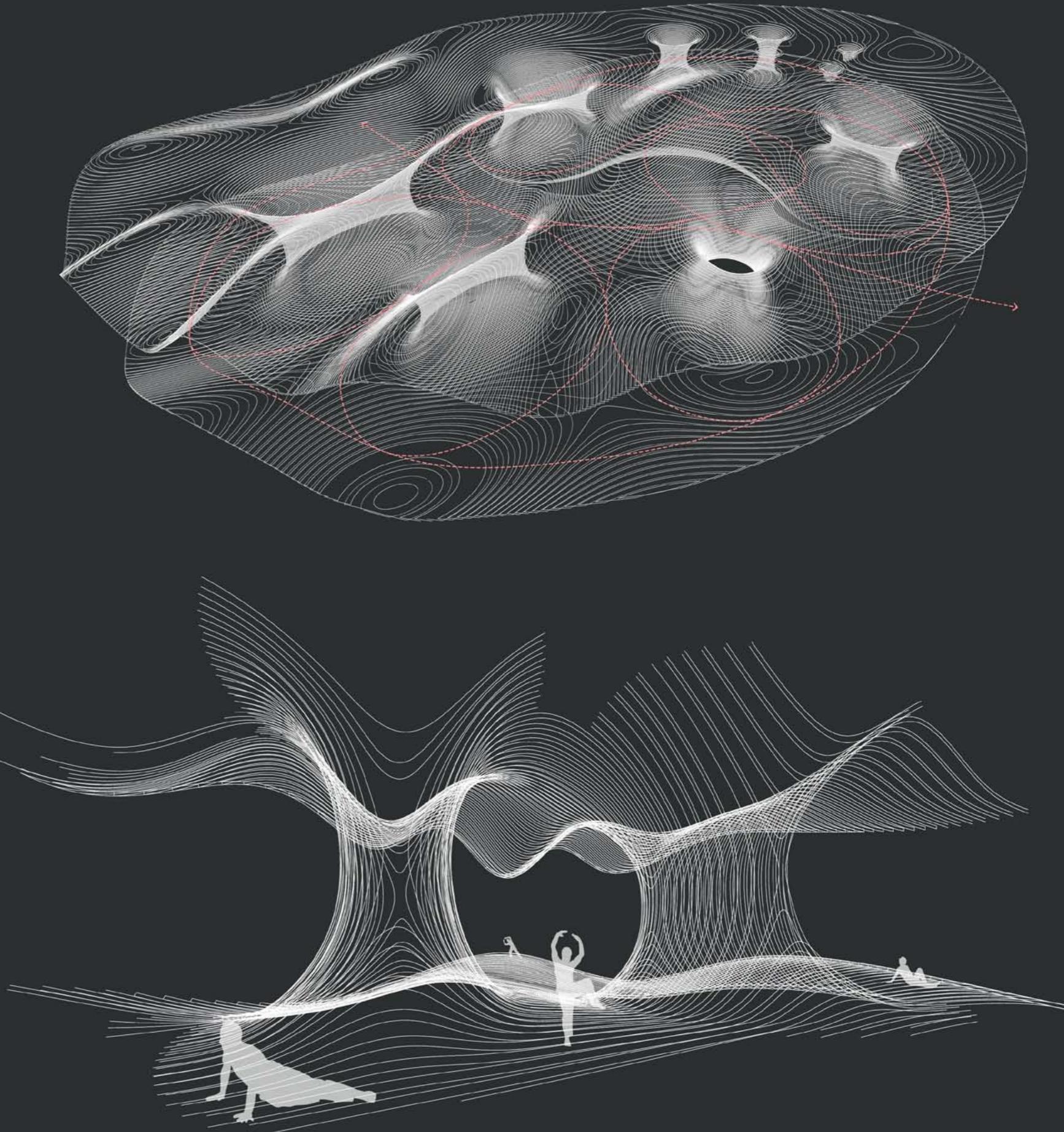


Left: Sectioned axon showing scale and formwork.  
Below: Curvature analysis, blue indicating areas of minimal curvature, and red, areas of maximum.



Top: Contours and interior circulation.

Below: lateral contours express form and indicate possible structural approach.



8

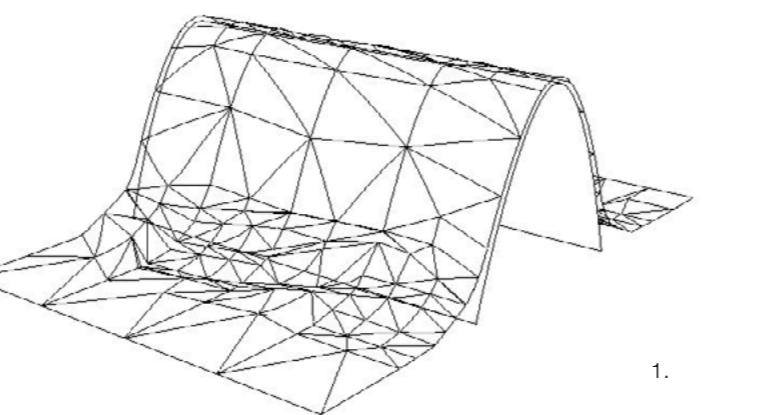
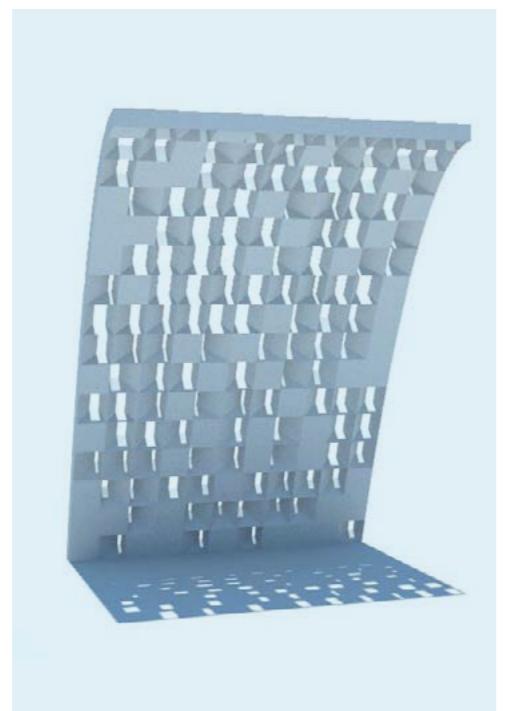
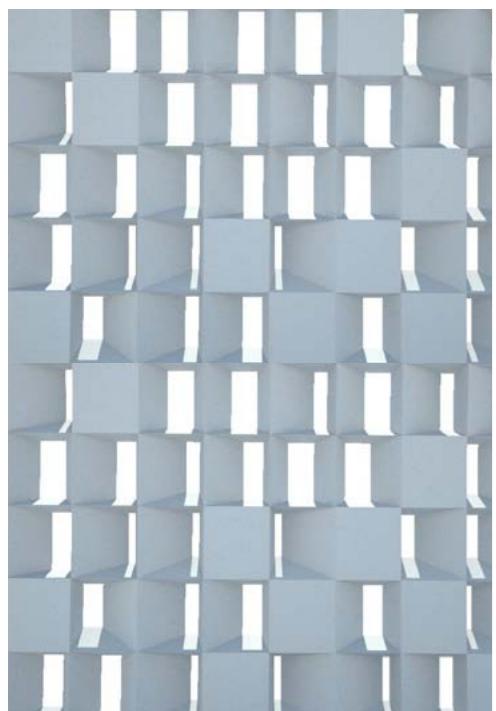
## UPLIFT AIRSHIP HANGER

Layers of earth are thrust upward by the geological process of uplift exposing a spatial pocket in the earth for airship operations while generating unexpected opportunities for secondary function as form becomes manifest as a “natural” landform that seemlessly integrates with site both visually and systemically as a member of a self-sustaining networked ecology of tree-covered mountains and landscapes.

Prefabricated concrete frame and panel system realizes an innovative and fully integrated structural skin that carries vertical loads of structure, roof installations and ceiling crane while oposing lateral forces with highly optimized shape that minimizes material while expressing the combined efficiency of structure, economy and form.

Flexible and highly efficient structural framework minimizes maintenance and construction costs while permitting a highly differentiated and modular system of cuts and openings that modulate natural light, facilitate ventilation and rain water management, as well as generate sufficient energy to realize a zero mainanence green roof and fully integrated building system that produces all of the energy it consumes.

Project Scope: Design Competition, Airship Hanger, Anhui, China | Collabortors: OPEN Architecture, Chinese Academy of Building Research (CABR), LI HU (principal in charge) | Role: Consultant for OPEN, Lead designer | Contribution: Concept/schematic design, form-finding, computational design



1. uplift formation concept  
opposite: aerial view of landform

2-3. possible lighting variation achieved with integrated paneling system.

text and images courtesy of OPEN Architecture  
more at <http://openarch.com/task/1454>



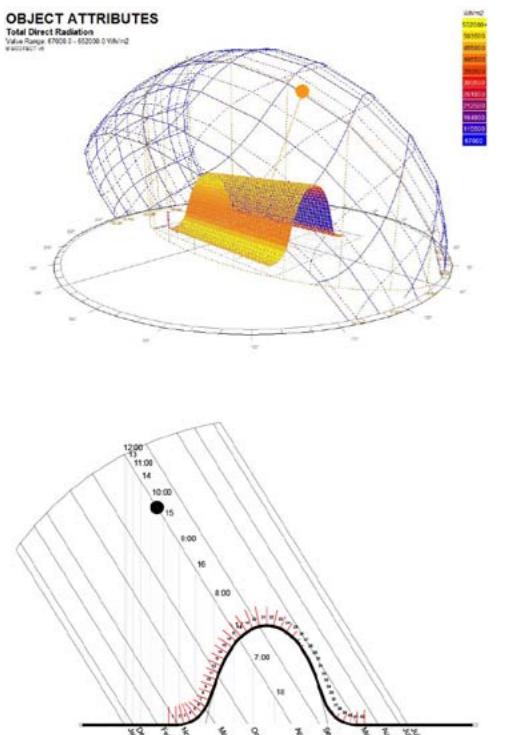
above: site photos of tree covered landscape



Prefabricated concrete frame and panel system realizes an innovative and fully integrated structural skin that carries vertical loads of structure, roof installations and ceiling crane while opposing lateral forces with highly optimized shape that minimizes material while expressing the combined efficiency of structure, economy and form.

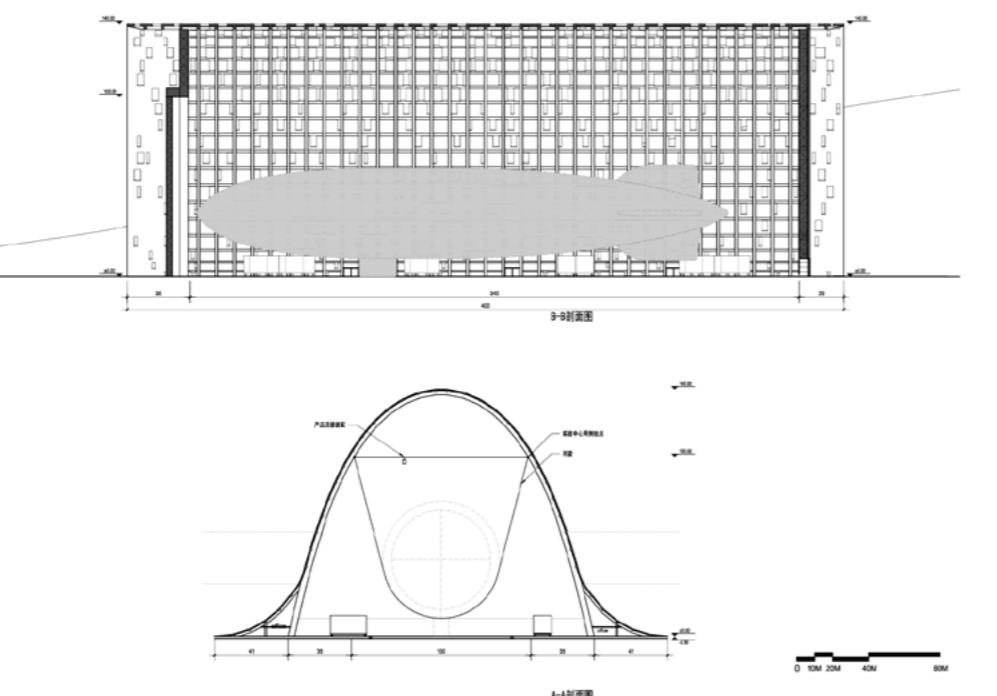
Primary structure consists of 64 concrete ribs that are arrayed along the length, and which follows catenary curve. This highly efficient form allowed us to minimize material and cost, while supporting an exceptionally thin and lightweight skin. It was important that the skin be expressed as lifting up and not visually oppress the space below so this apparent lightness was critically important architectural criteria met with structural intelligence.

Lateral support beams run the length of the structure, stabilize it against lateral forces and establish gridded framework for prefabricated panels and integrated energy systems. Framework and panel system comprise the structural skin, which follows catenary form of load-bearing ribs and peals away at base to express itself as a continuation of the ground plan, i.e. layer of earth lifted up from it. This separation of skin from primary structure also creates an interstitial space that runs the length of the hanger to be occupied by offices, storage and secondary program.

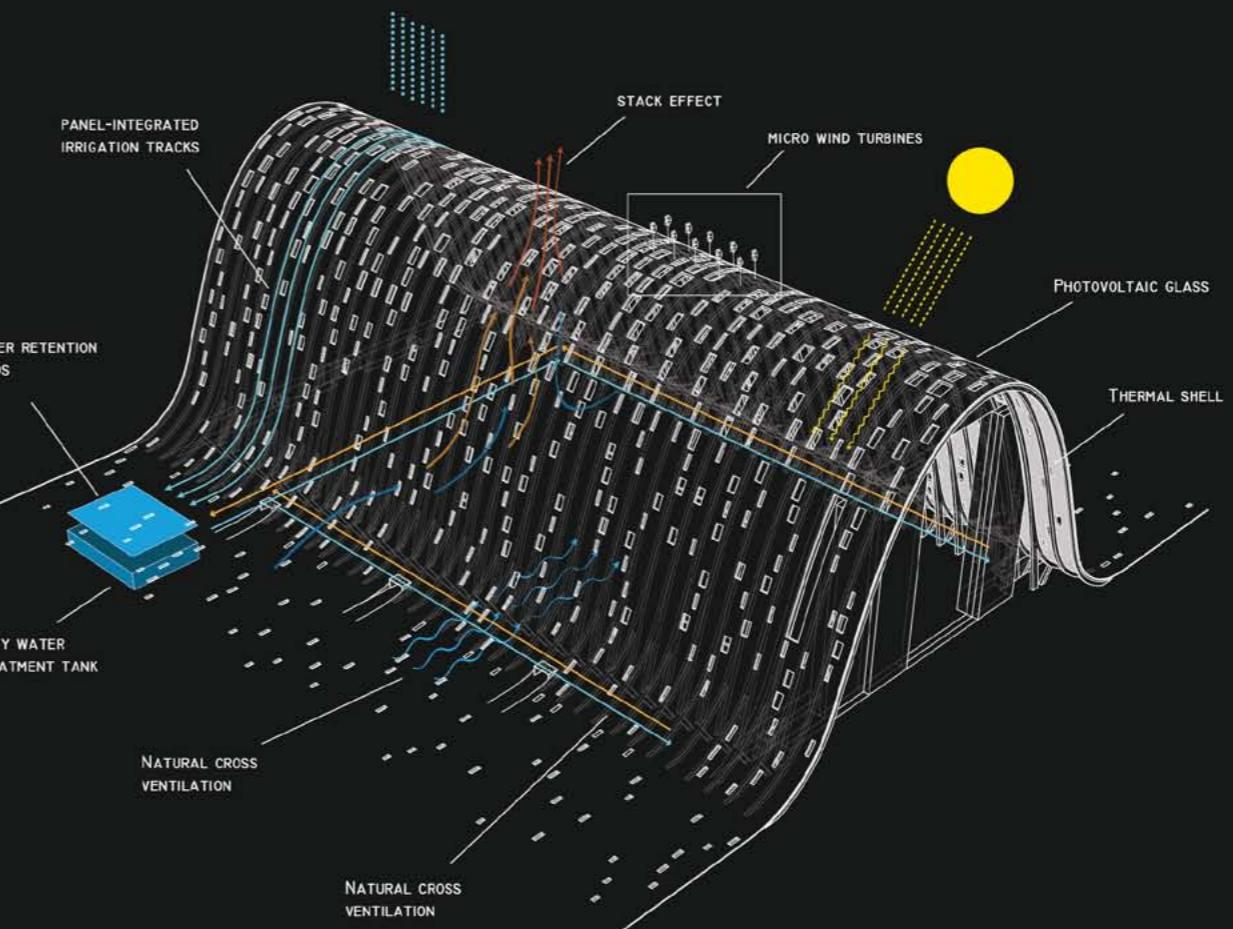


Above: seasonal solar exposure of build skin.

Right: Sections. Primary structure can be seen to follow a catenary form. Skin can be seen to separate at base creating interstitial spaces.

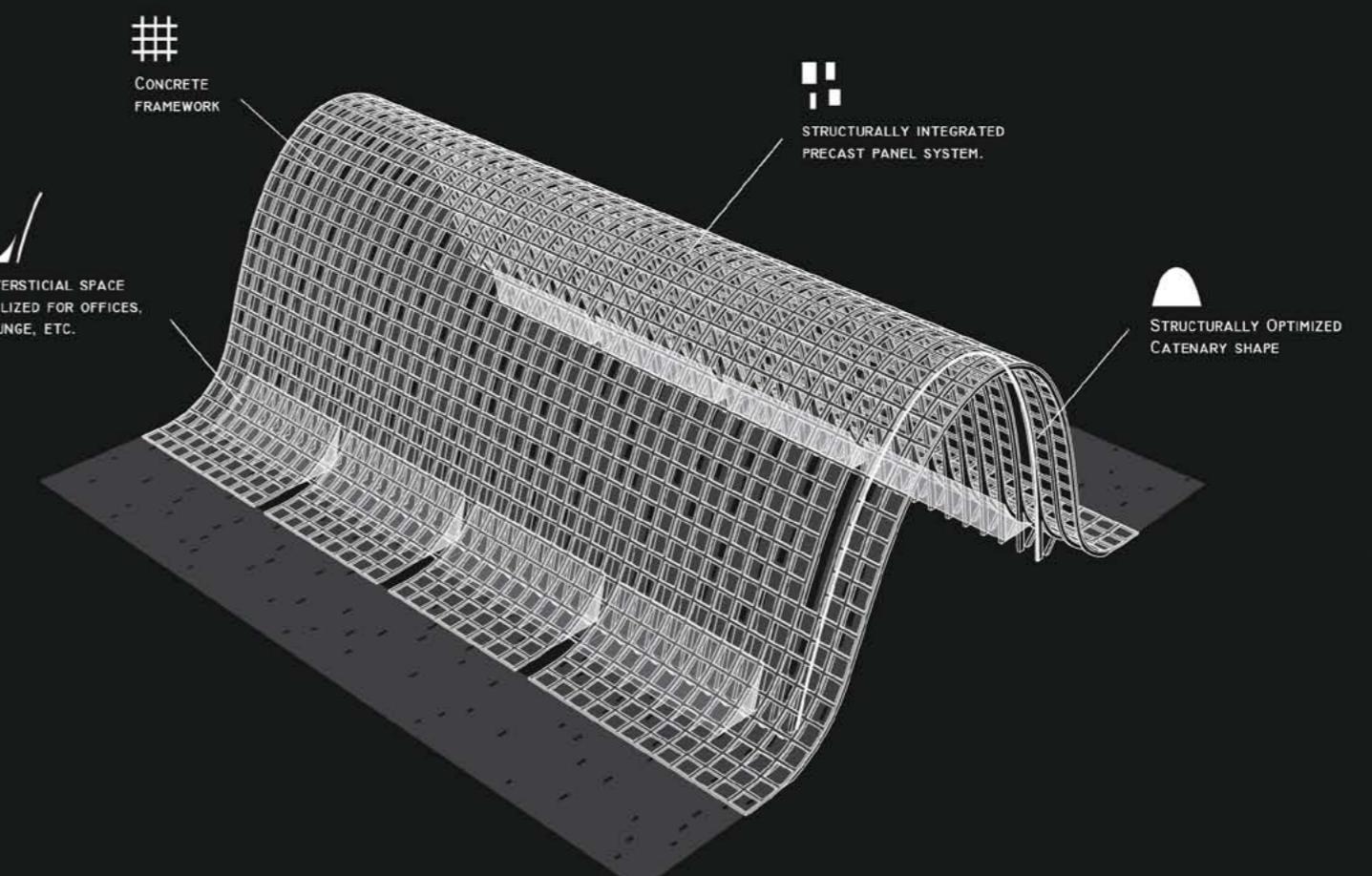


Structure at ends also frame the hanger doors, consisting of four segments, which run along tracks, supported by truss at the top and in ground at the bottom. As doors open they penetrate the skin, where omissions of lateral supports in the framework can be seen. This did not compromise structure as framework was sufficiently redundant to allow for the omission of beams in places where door penetrates skin.



passive energy systems

modular system of cuts and openings that modulate natural light, facilitate ventilation and rain water management, as well as generate sufficient energy to realize a zero maintenance green roof



structural framework

minimizes costs while permitting a highly differentiated and modular system of cuts and openings.

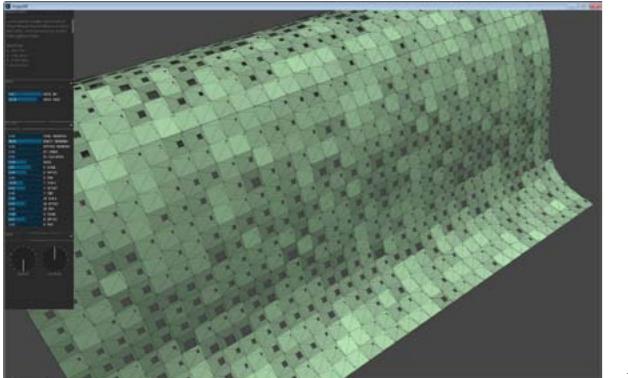
Flexible and highly efficient structural framework minimizes maintenance and construction costs while permitting a highly differentiated and modular system of cuts and openings that modulate natural light, facilitate ventilation and rain water management, as well as generate sufficient energy to realize a zero mainanence green roof and fully integrated building system that produces all of the energy it consumes.

Using solar analysis software we determined the annual direct and indirect exposure of each panel. This data informed the arrangment of solar collection, diffuse light control and green roof programing. Modulation of light was acheived with 13 discrete panels types, which permit direct and diffuse light to penetrate skin. Penetration pattern responds most directly to solar profile and grade, and is then adjusted to satisfy programmatic mapping of recreational trails, activity patches, and maintenance tracks.

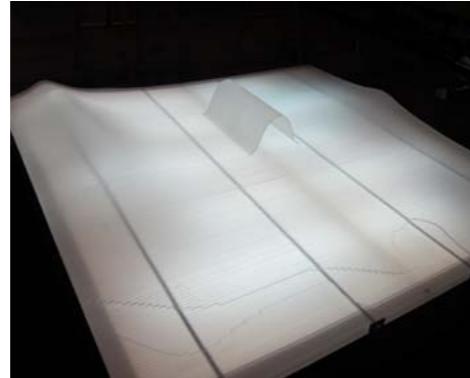
Indigenous trees and ground cover are specified for landscape at the base and provide shading for trail network that extends outward and up the structure from the base, connecting greenroof ecology and public park services with surrounding forest and hills. Trees extend partway up structure, and ground covers the entire surface. Top soil and grass is supported by ecoweb mesh installed on top of structural skin.

Interior lighting requirements are met during day with system of cuts and opening in the skin. This also produces a controlled and highly dramatic lighting effect that registers time of day and quality of natural light. During night artificial lights produce a light pattern on the exterior skin.

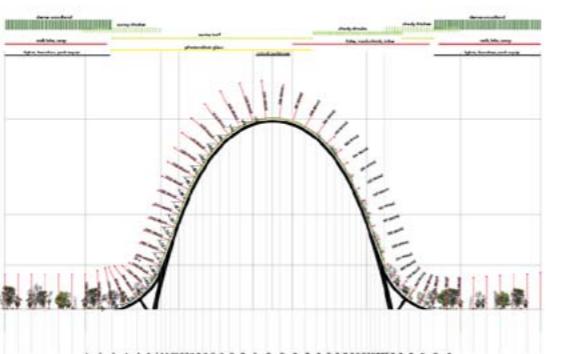
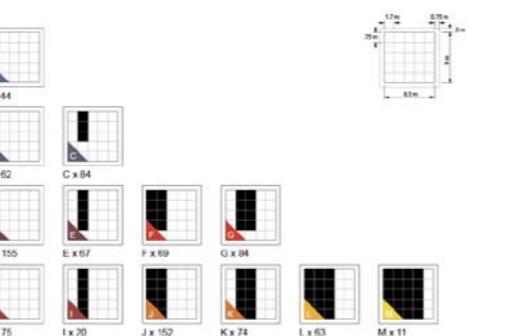
Custom software tool was developed to search design space specified by parameters describing the panelized system of cuts and opening to meet lighting requitements and exposre lighting effects in real-time. Alternate system of apperatures, shown above, produce slight more dynamic expression of light control that better registered the relationship between skin and environment, however effect came at a cost: added complexity to the prefabricated paneling system.



1.



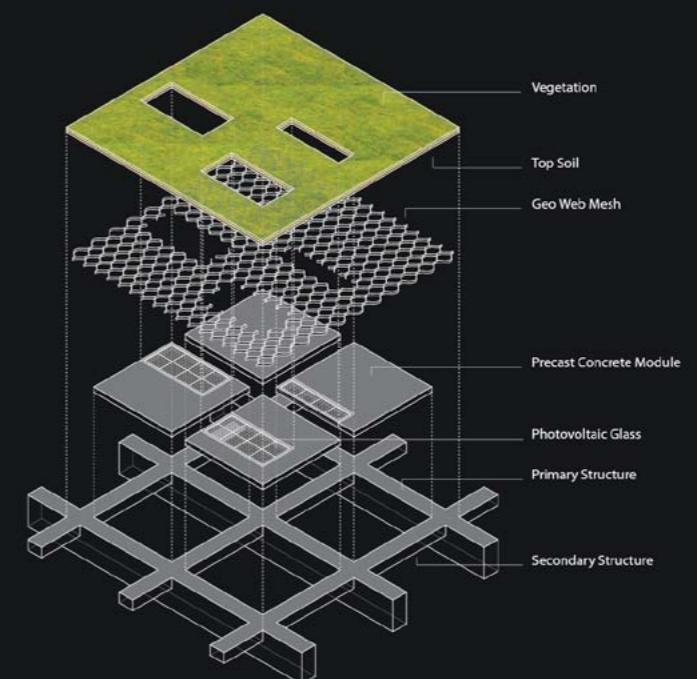
2.



Above: Modules and encoding scheme based on solar exposure and program mappings.

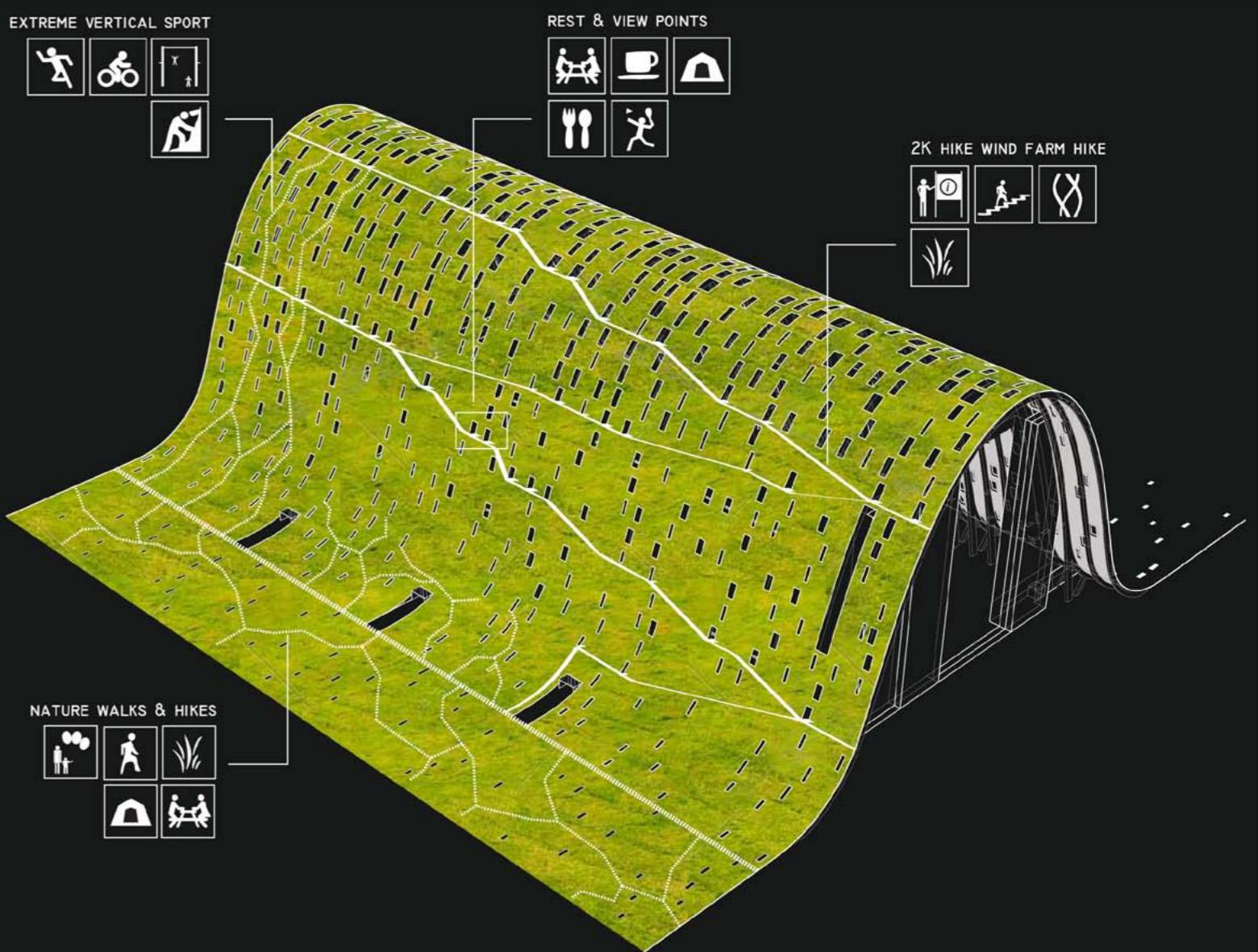
1. Bottom Left: screenshot of custom paneling tool generates perforation pattern based on solar data, as well as light and program constraints.

2. Bottom Right: images of landform model



#### SELF-SUFFICIENCY, PROGRAMMED GREEN ROOF

left: diagramatic detail of integrated construction layers. prefabricated panels with pre-installed solar glass fit onto frame. eco-web mesh is installed ontop of frame/panel system. hardware not shown.



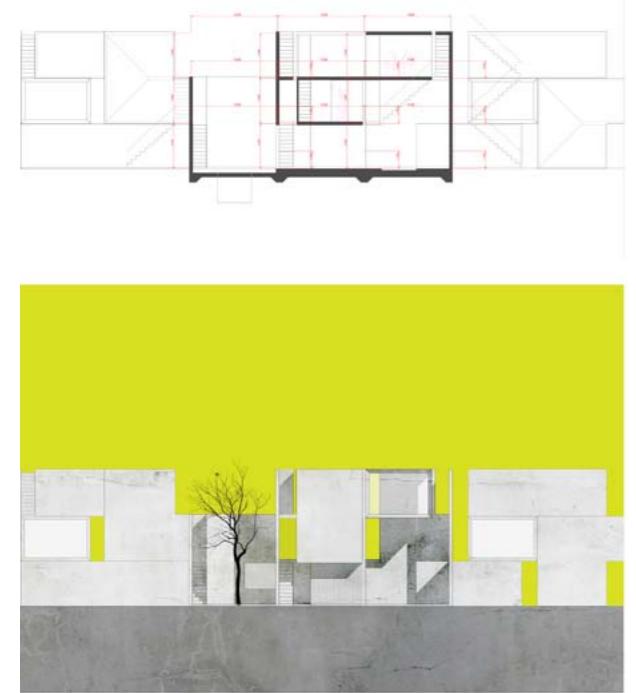
above: programmatic diagram of green-roof. program is mapped on to surface according to grade, plantings, and sun exposure.

## PROTOCELL

Research project initiated by OPEN Architecture proposal for Netdragon commune At the virgin beach front area where Min River meets the Pacific Ocean, and adjacent to Fuzhou Changle Airport, Netdragon Websoft Company's new headquarters building is currently under construction, embedded in it, the grand visions for the future of the company. Nearly 700 employees and their families will move from the center of Fuzhou city to this new campus. A 50,000m<sup>2</sup> land which used to be an eel farm will become their new living quarters. This is a very interesting and young community of people. Their work is to create wonderful virtual worlds that entertain millions of people. In real life and on this land of their new dreams, they need a unique living area and a brand new collective life style.

The Netdragon Village project is an architectural experiment investigating how to restructure relationships between people and nature, and also relationships amongst people themselves. The project seeks to create a ecological habitat for the future at a comparatively low cost.

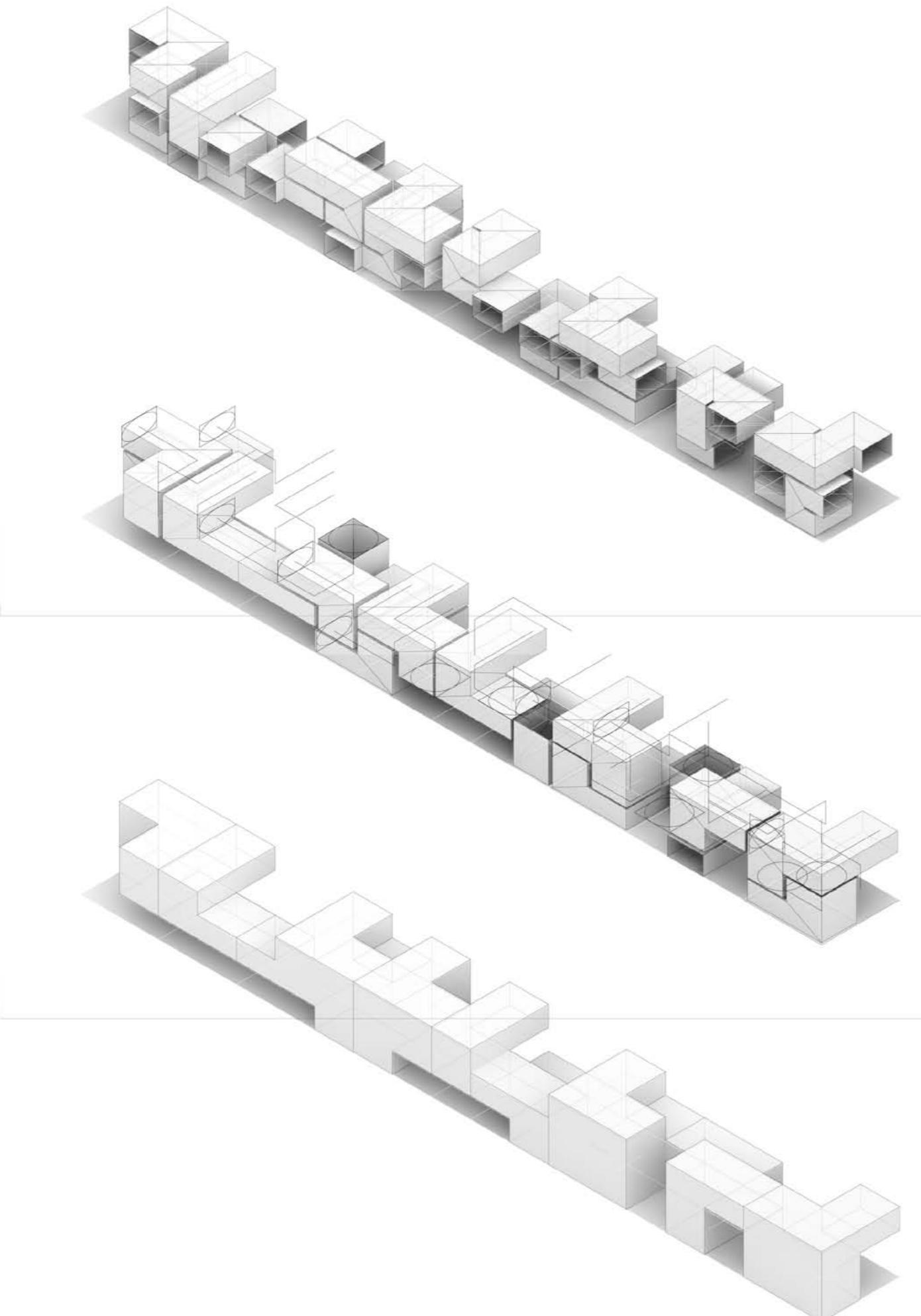
Project Scope: metadesign, collective housing system | Collaborors: OPEN Architecture, Li Hu | Role: Lead designer, researcher



1. self-organized arrangement  
of approximately 100 four-cell  
housing units.



x100 habitat

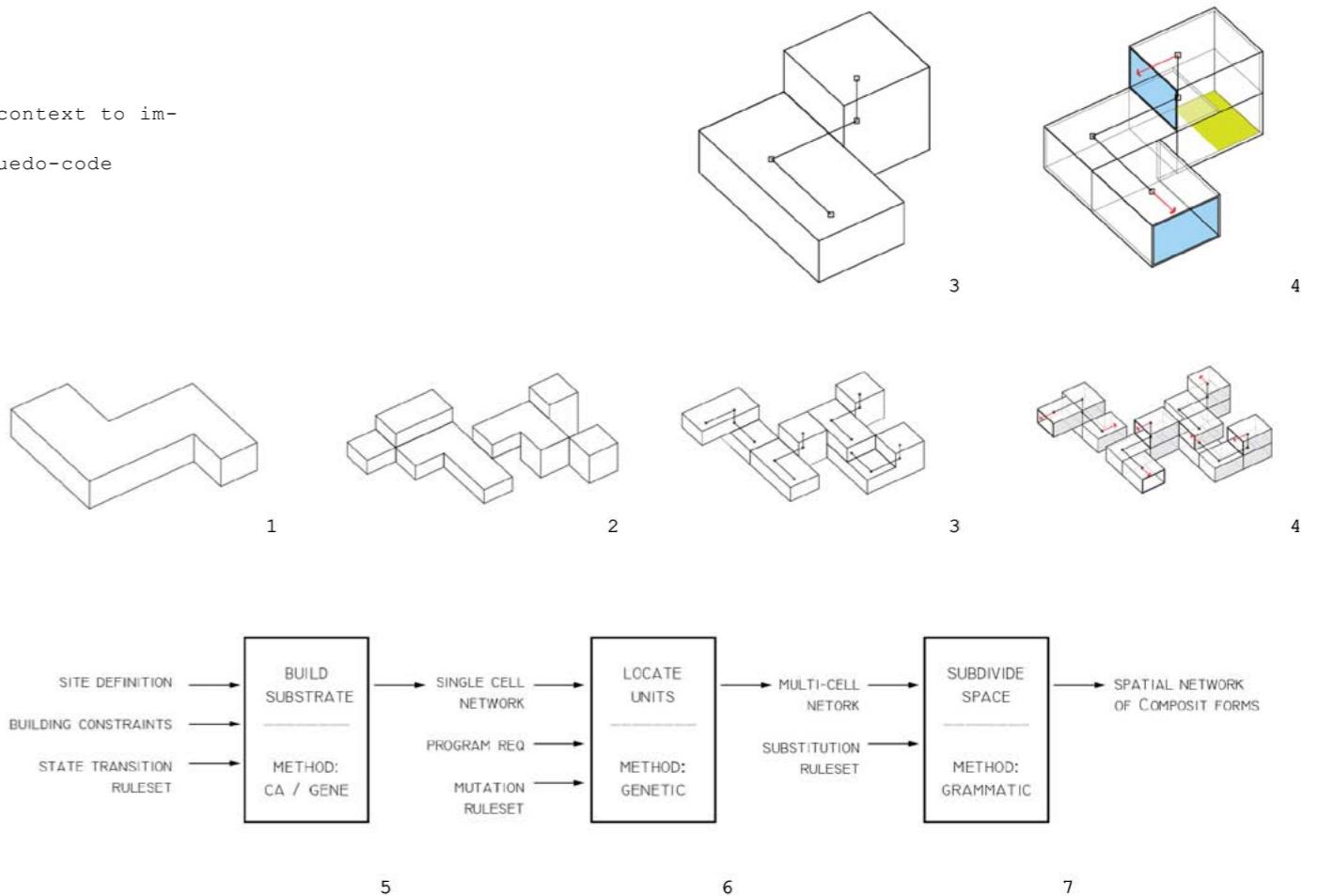


above: exploded axon of x1 unit  
opposite: floorplans, grnd, 2, 3

The algorithm, initially implemented in rhino-script, can be divided into three isolated stages. The first distributes cellular mass according a CA logic and optimizer that adjusts massing for circulation paths and other encoded constraints like site, topographic features, etc. The first stage requires as input an initial massing condition as cellular automata proceeds as incremental refinement.[1] First stage is also responsible for constructing a cell network, ie. each cell maintains state and keeps track of neighboring cells. This network is passed to stage two along with further building constraints like program, floorspace allocation, etc. The second stage subdivides single-cell mass into multi-cell clusters or linked cell chains. This is accomplished by establishing an initial condition of randomly placed conditional units and defining a fitness function that eval-

uates the placement and situation of the unit and simply discards losers and ignores winners. This is a weak genetic optimization algorithm that is more brute force than anything else, but it does find solutions in reasonable amount of time. The final stage embeds building intelligence into cell clusters, places utility core, and adapts unit to basic building constraints... All stages are intended to be a guided meta-design processes, in which designers participate in real time by making subjective decisions and feeding them back into the system.

1. boundary condition
2. massing after 1st pass
3. cellular chains form
4. cells self adapt to local context to improve access and views.
- 5-7. schematic diagram and pseudo-code
7. resultant housing unit



```

//input
site boundary
module size
far
porosity
avg height
ruleset obj

//calc 2nd order params
site area from bounds
floor area as site_area*far
mod as fir_area/mod_size
seeds as mods/porosity
levels as avg height
spl as seeds/levels
cvg as spl*mod_size/site_area

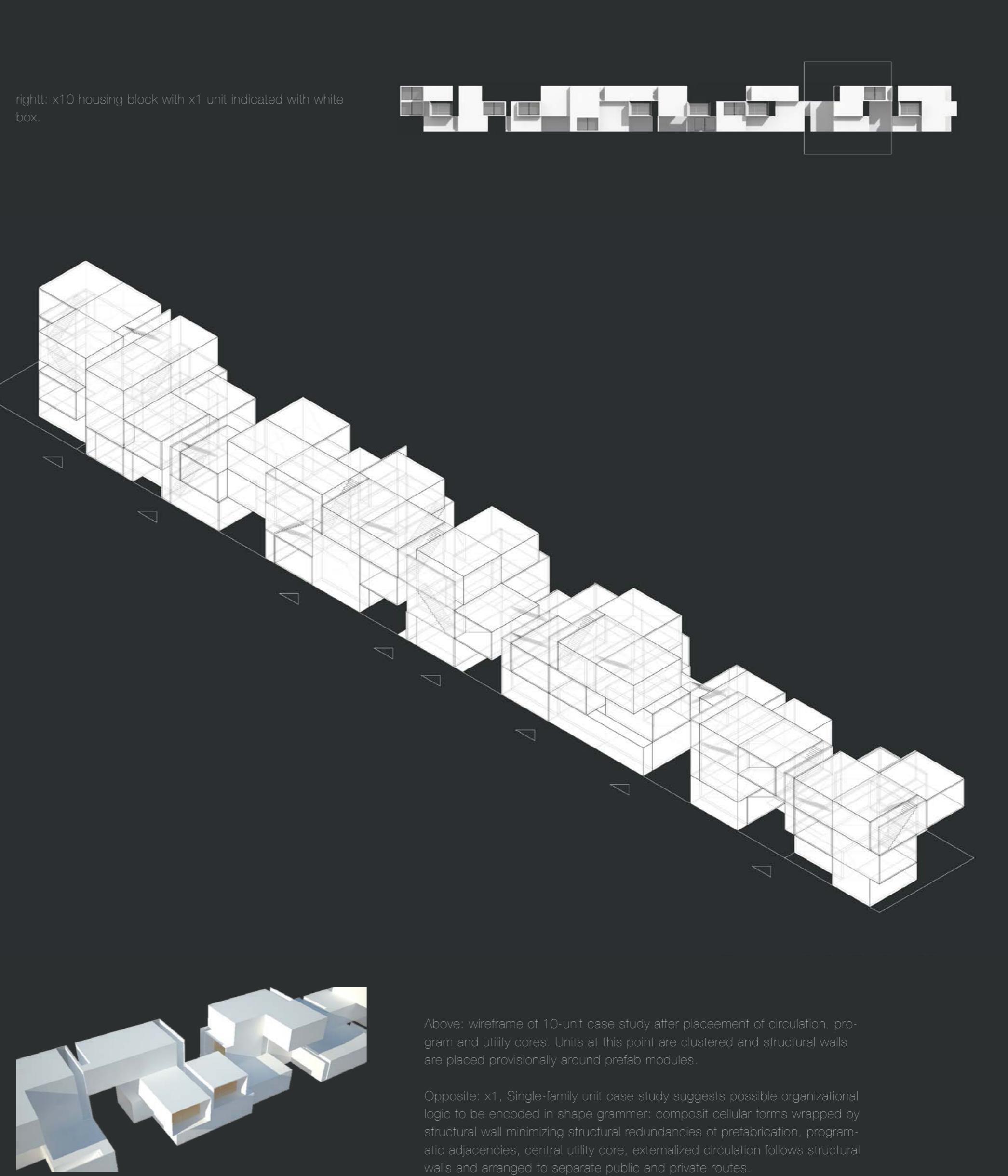
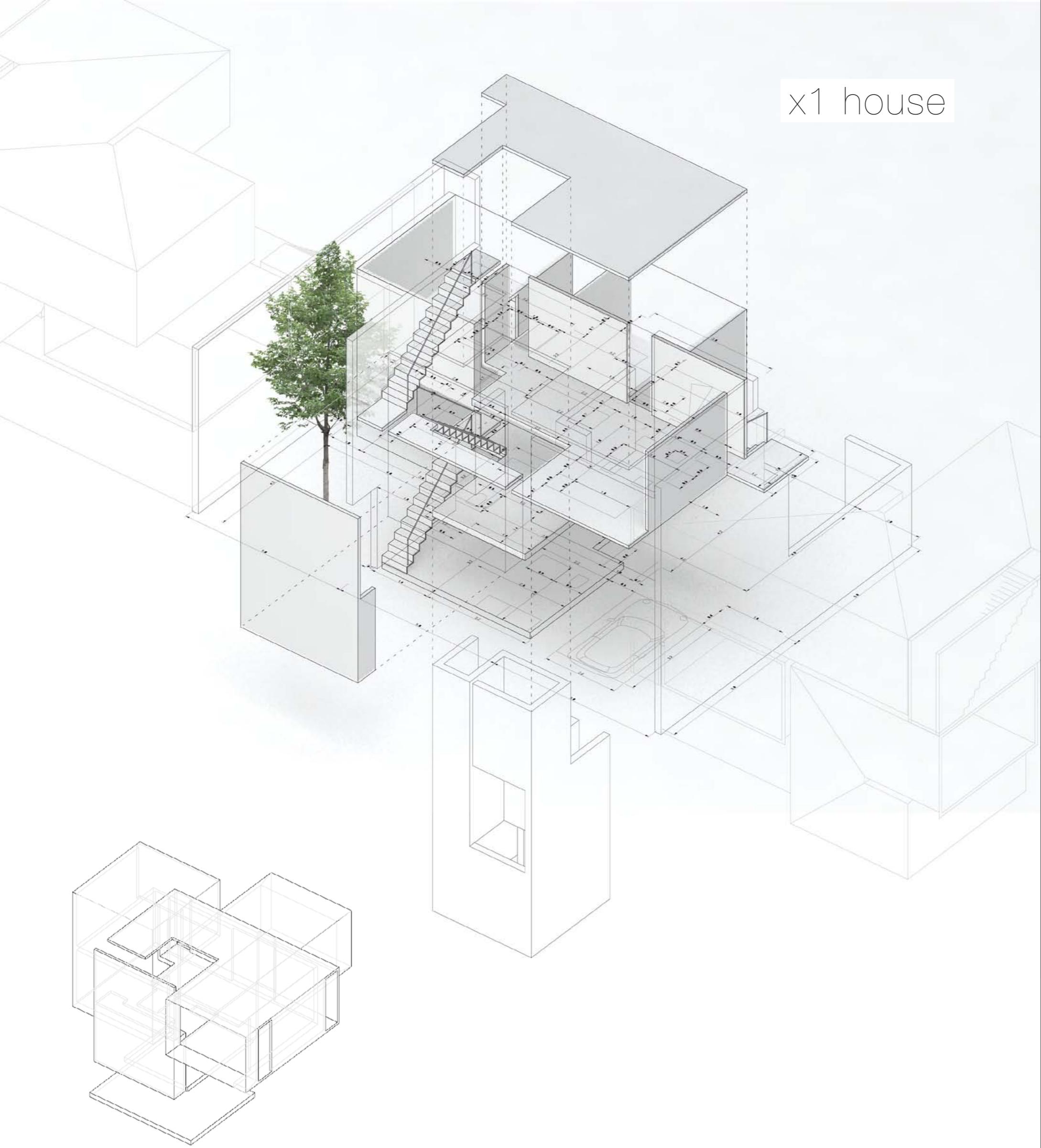
//build
create 2d pt grid
create cells from pts
assign random type
link cells to neighbors
define neighborhoods
//refine

//input
array of linked cells
num of 6-cell units
num of 12-cell units
max height
ruleset obj

//solve
find candidate snake
Apply ruleset
(necessary conditions)
repeat until all found

//optimize
pick snake at random
calculate fitness
apply ruleset
repeat

//output
for each ordered cell set
connect pts with polyline
sweep polyline
  
```



## CHENJIACI CULTURAL VILLAGE AND PARK REDEVELOPMENT

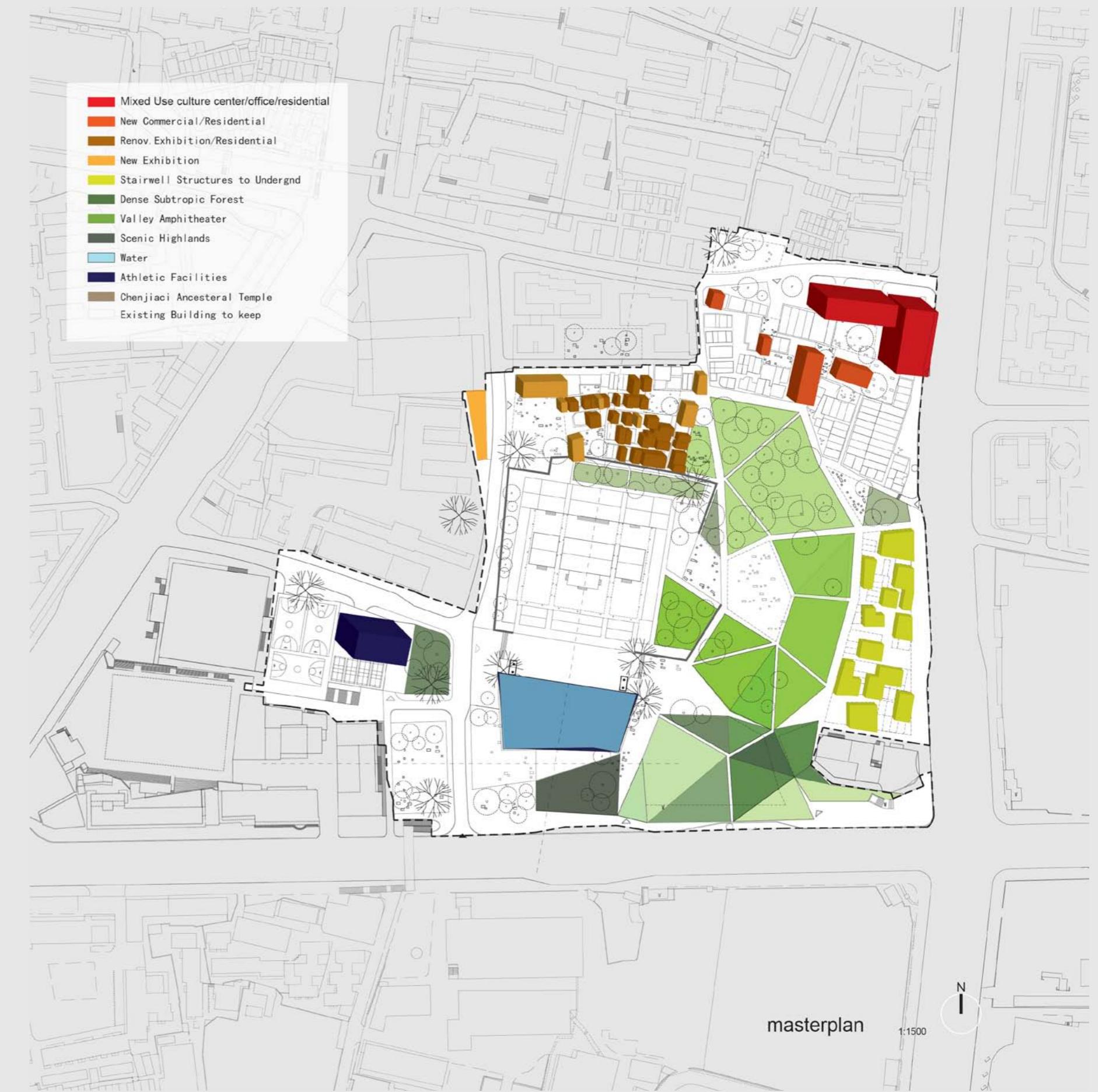
The cultural square of Chan's Ancestral Temple (CAT) is intended to be a multi-programmed tourist site that is rich both in historical architecture and civic culture. Our design proposes to divide the land around the CAT, the core of the region, into five spaces with different programs: history space, exposition space, culture space, sports space and civic activities space. Through rebuilding, revitalizing and transforming the five spaces, we aim at creating a new spatial organization that consists of one core, one museum, one stripe, one street and multiple points, restoring the historical features of the CAT to the largest extent and revitalizing this area through tourist and commercial propositions. [text by node]

Project Scope: Redevelopment of Cultural park & Square of Chan's Ancestral Temple, Guangzhou, China | Collaborators: Node Architecture and Urbanism, Doreen Liu (principal in charge), Jiebin, Huang, Shaoli Lin | Contribution: Concept/schematic design, masterplan for landscape/park nodes | Role: consulting architect for Node Architecture and Urbanism, in charge of landscape nodes



top: diagram of park zones  
right: views from park  
opposite: masterplan

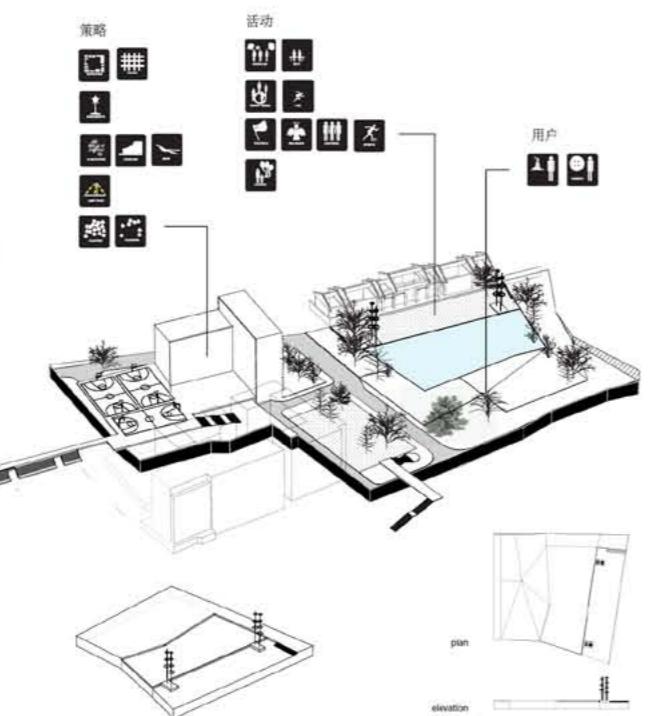
images courtesy of NODE Architecture & Urbanism





高地

升起的石块地下形成了以陈家祠为背景的框景，并引人入内。  
在地面上游人可以享受着无遮挡的景色，更可一览陈家祠的滋味。  
在的地方，游人可放风筝、打羽毛球等活动。小路引着人到野餐和休息的理想地点。

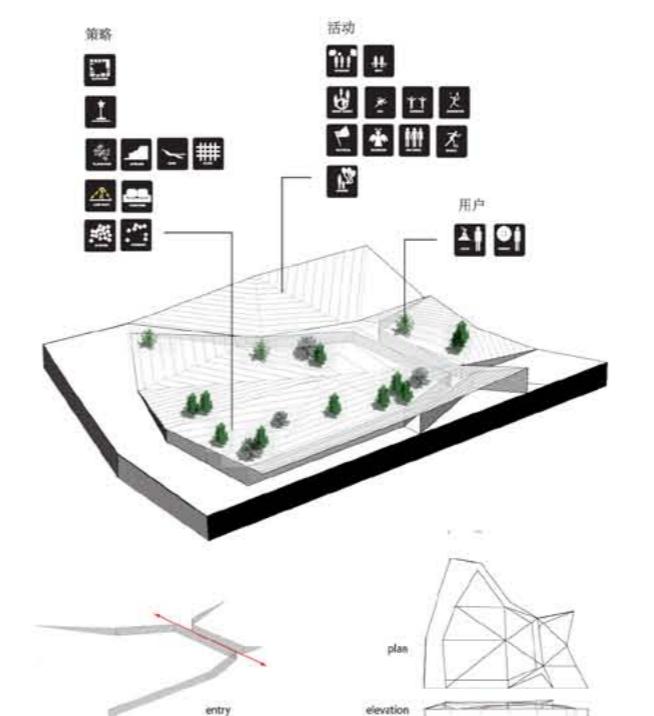


no.2

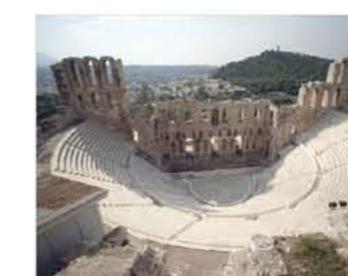


高地

升起的石块地下形成了以陈家祠为背景的框景，并引人入内。  
在地面上游人可以享受着无遮挡的景色，更可一览陈家祠的滋味。  
在的地方，游人可放风筝、打羽毛球等活动。小路引着人到野餐和休息的理想地点。

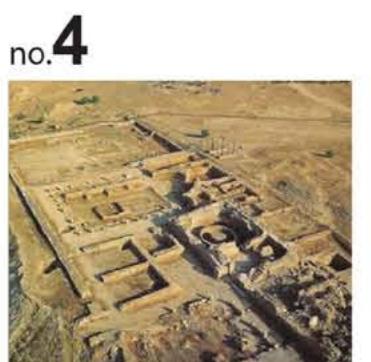
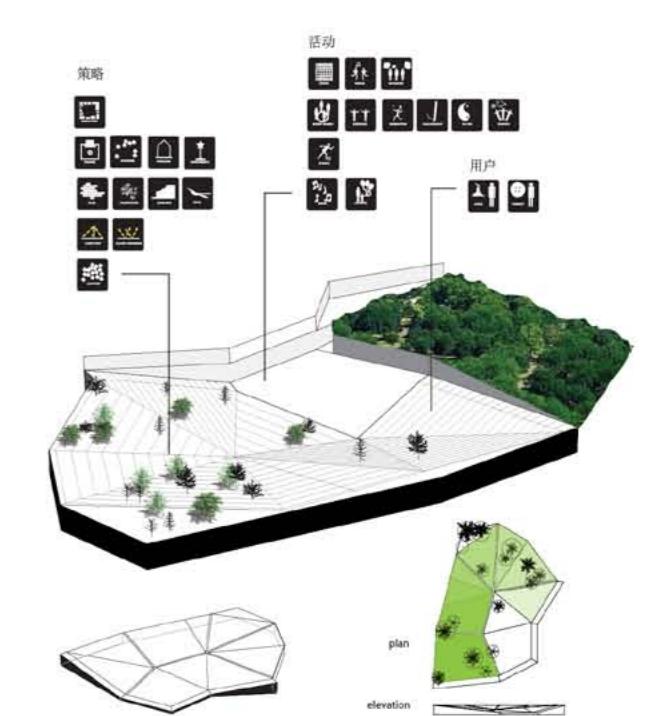


no.3



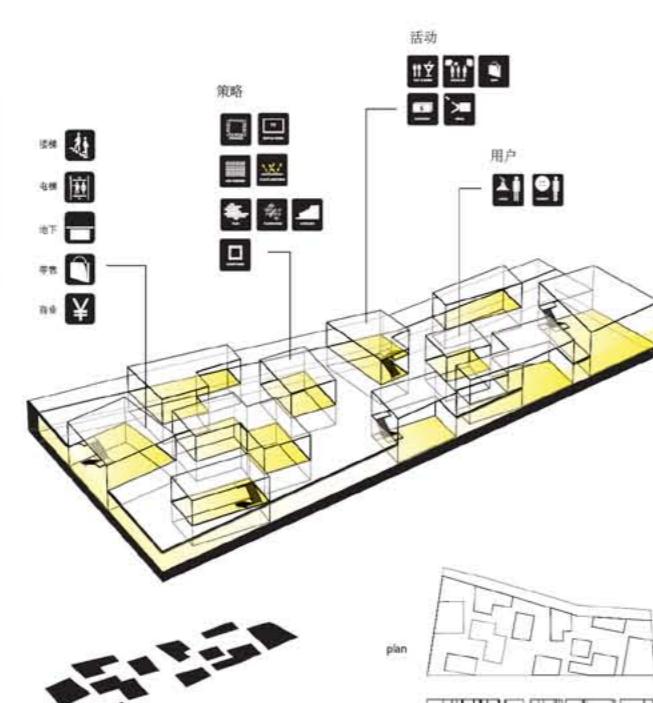
自然剧场

坡地自然景观形成了一个以陈家祠为中心的大型广场。广场可以举办大型活动、集体运动和游戏，是一个理想的社交场所。  
陈家祠被变成一个美丽的背景衬托着这个社交广场。围墙借用园林中的框景手法，不光可以让外行人窥视陈家祠的，也可让祠中游人观看广场的情景。



广场

通透的结构形成地面与地下的连接。  
晚上，地底下的照明使这结构变成一个景观中的发光体，在日间自然光投射临时的地下购物中心。在街道上结构阻挡行人的视线，惹起行人的好奇心。  
绿化屋顶把结构融入景观中。历史的痕迹使这些结构与临近的文化区内的建筑物形成关系。



no.5



现有森林

树冠提供最佳的遮阴处。  
这里是退休人员聚集、交际、运动和下棋的好地方。  
弯曲的小路穿过大森林形成自然步行径并穿插于各个休闲空间中。

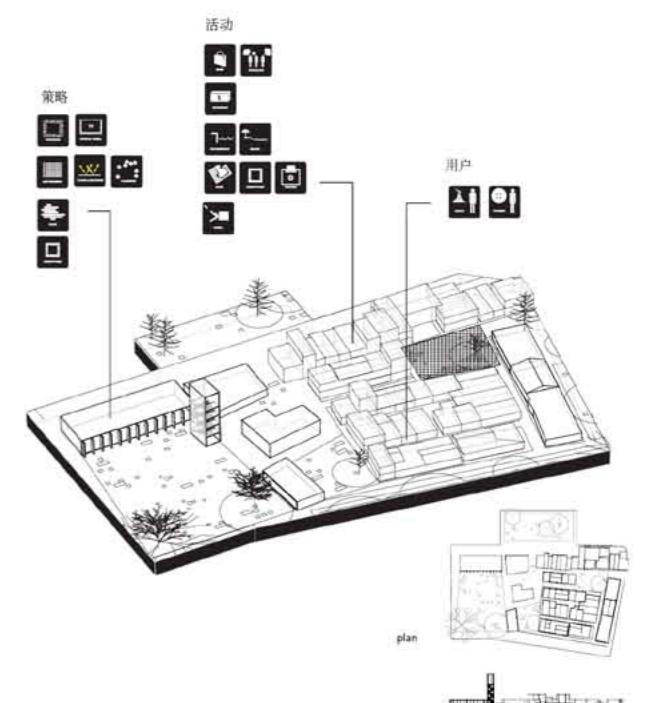


no.6



自然剧场

坡地自然景观形成了一个以陈家祠为中心的大型广场。广场可以举办大型活动、集体运动和游戏，是一个理想的社交场所。  
陈家祠被变成一个美丽的背景衬托着这个社交广场。围墙借用园林中的框景手法，不光可以让外行人窥视陈家祠的，也可让祠中游人观看广场的情景。



## GUANGXI MUSEUM OF FINE ARTS

36,000 m<sup>2</sup> museum of cultural art comprising 18,000 m<sup>2</sup> of Exhibition Space and just under 3000 m<sup>2</sup> of interior public space. Remaining area split between public facilities, office space and storage. 35% of total volume was burring and 65% above ground. The Museum is designed to integrate with the terrain, i.e. to embed part of the museum functions into the mountain, then follow the moutain slope to organize the circulation from the waterfront to the mountain top, thus create the impressive cantilevered building feature and form the corresponding axis at the head of the "dragon vein".

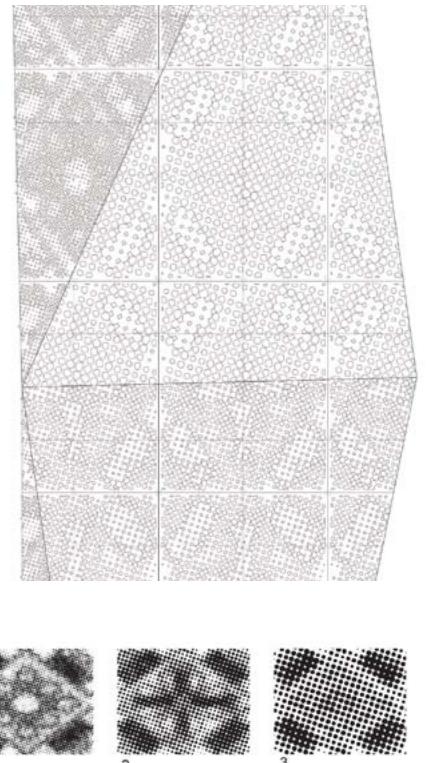
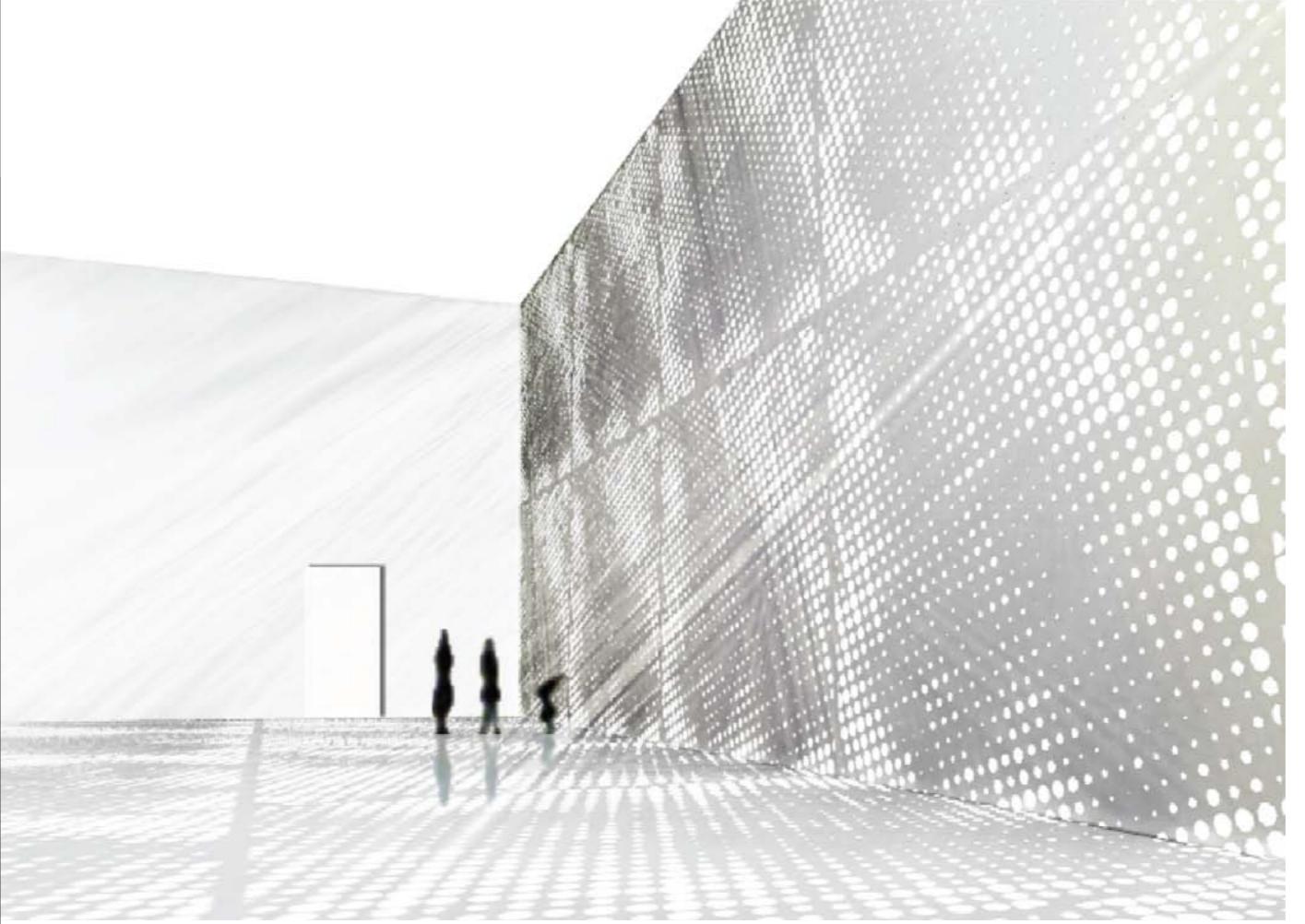
I participated in development of concept, schematic design and initial massing, In charge of design and development of building Envelope and landscape as single continuous triangulated skin. Perforation pattern derived from traditional rug patterns and constructed with 2.5 x 4m perforated aluminum panels. Quilted effect from triangulation also recalls traditional rug pattern. pattern tiles are approx. 8x8m and consist of 8 perforated aluminum panels, 2 of which are unique. 3 unique pattern tiles of varying transparencies then yield a total of 6 unique aluminum panels.

Project Scope: Design / Analysis of Building envelope | Collaborators: NODE Architecture and Urbanism | Roles: Design Consultant for NODE | Contributions: Concept/schematic design, massing, incharge of landscape and envelop design

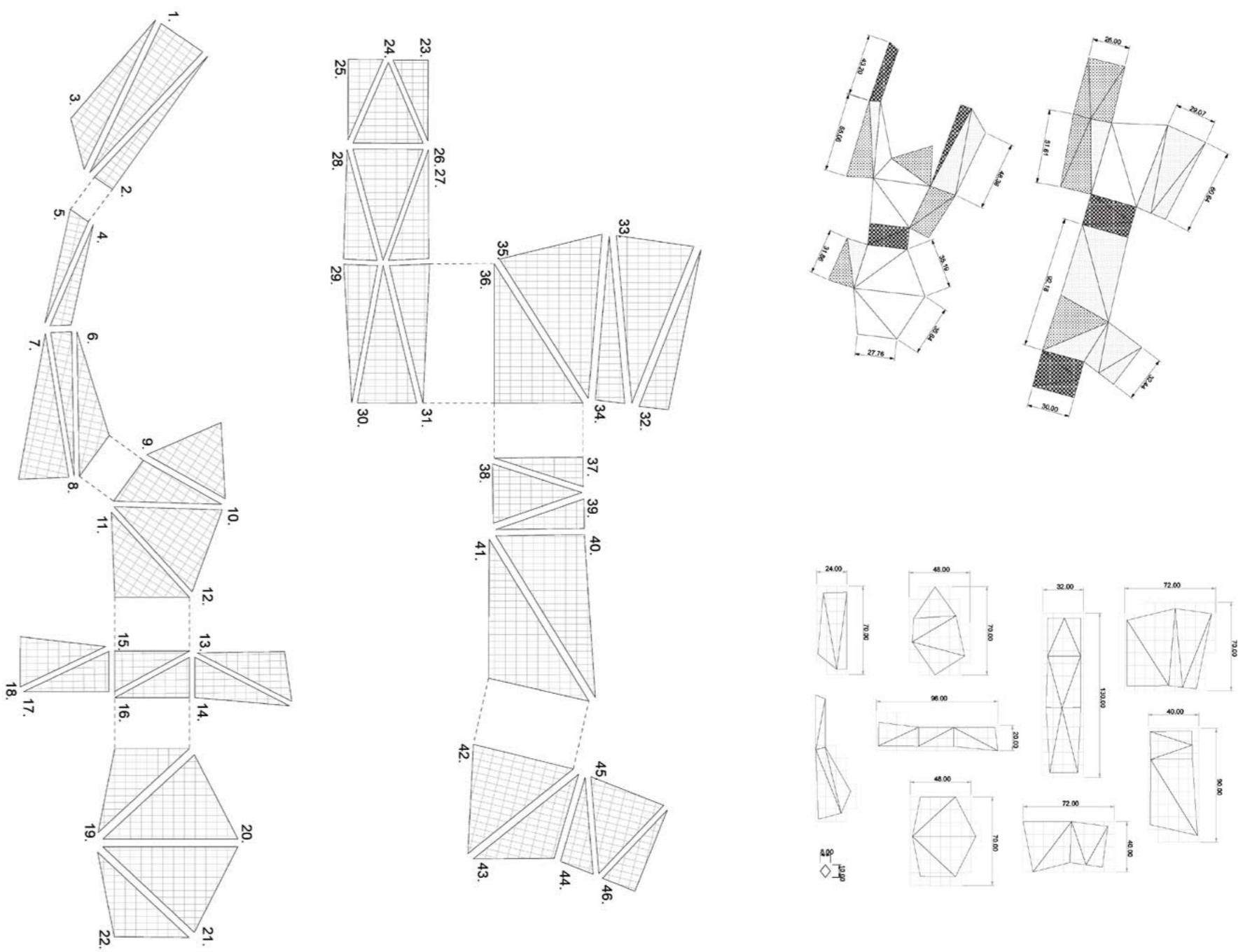
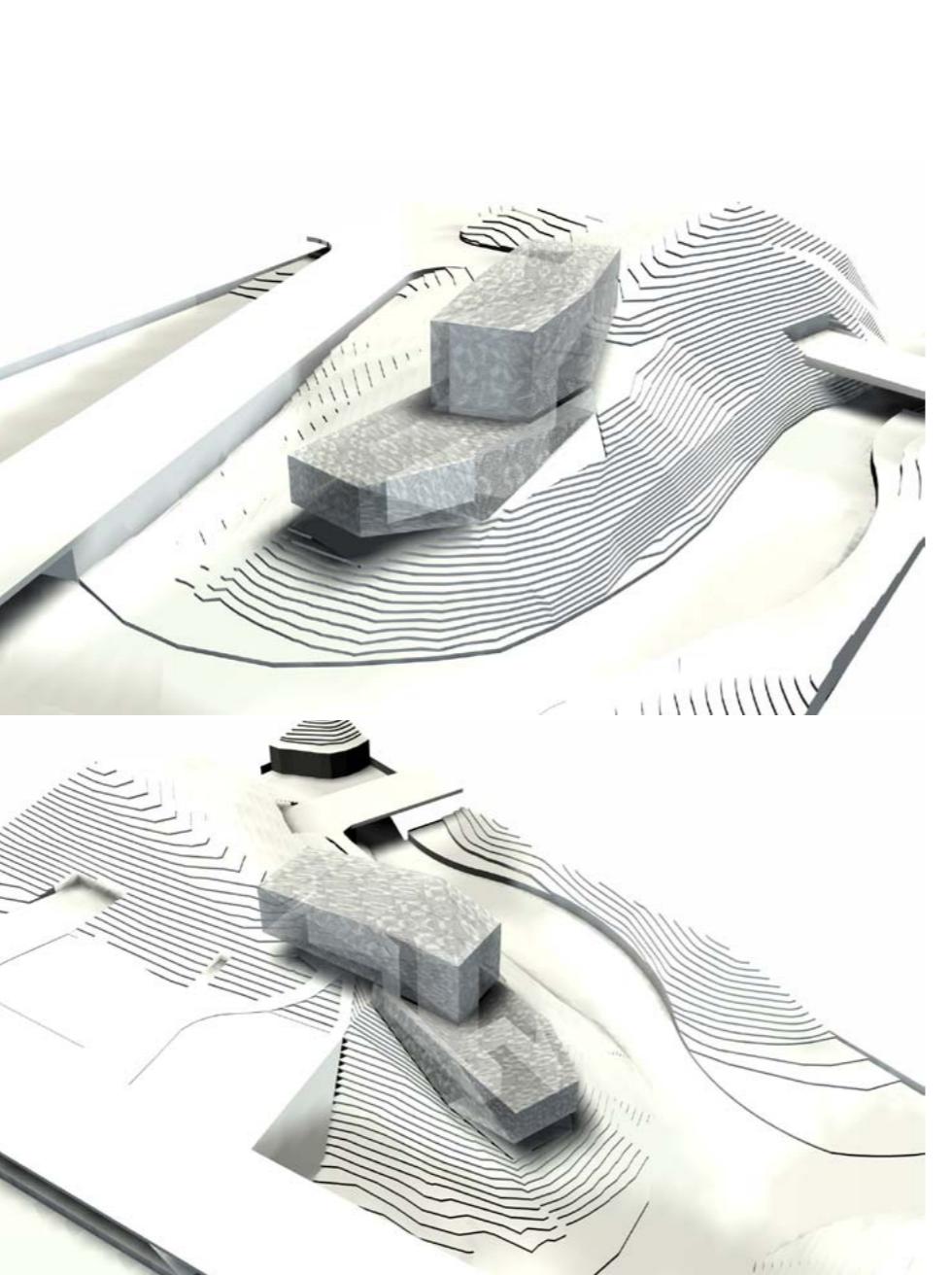


right: bird's eye views of massing and skin  
opposite: skin unfold with panel and pattern mappings.

images courtesy of NODE Architecture & Urbanism



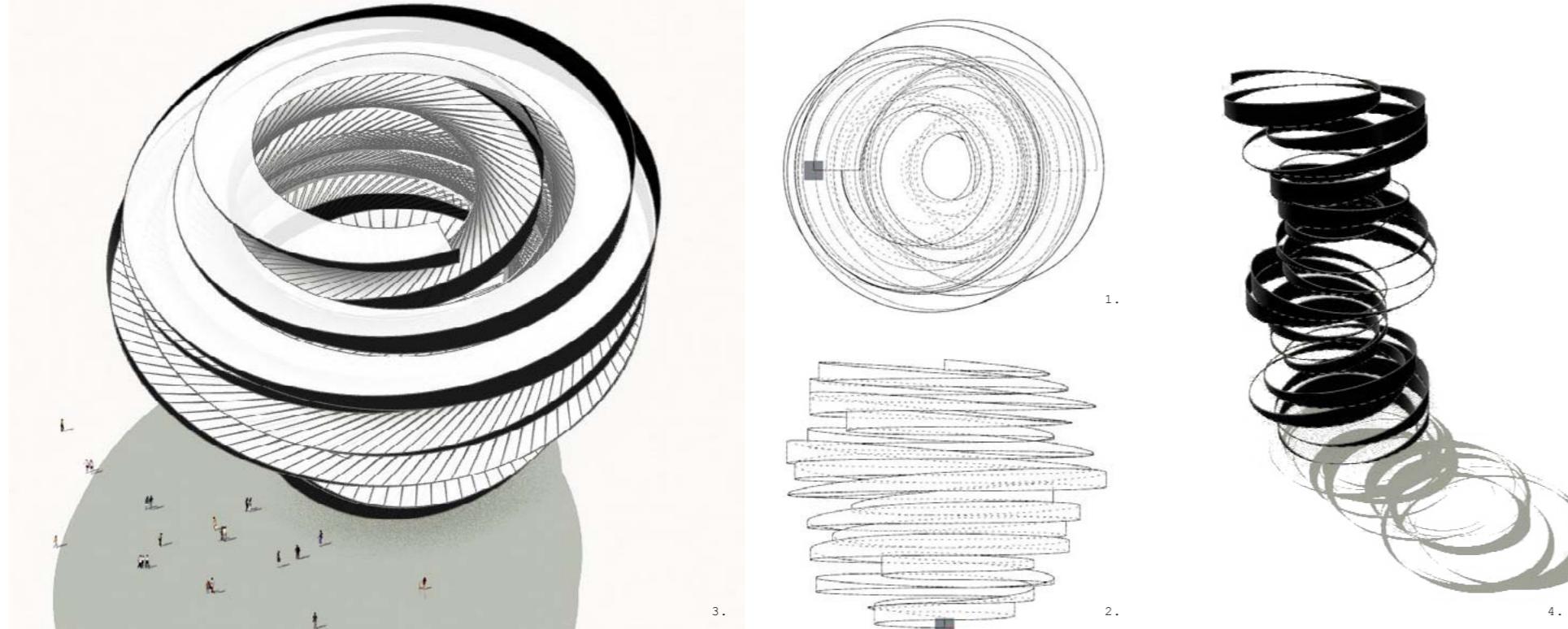
三个独特的穿孔模块来自于纺织品上的图案。这些铝模块从表达性、文化和标志性平铺有序地附在建筑表面上。这三个模块，每块需要由8块2x4米的面板组成。每个图案主要由两种独一无二的图案组成，三个穿孔的模板就共有六种不同的图案组成。



## SPIRALLING: PARKING GARAGE

Custom spiralling tool developed for parking garage Form-finding excercise with grasshopper visual programming rhino plugin. Algorithm based on simple Heuristic spiraling algorithm by Aranda Lasch: Specify radius. Then, At each increment, Rotate point around center by x Degrees and step in z direction by distance Y. Connect Ordered set of points with arc segments or spline. Road-like surface is Defined by offseting each point radially producing a second set of points corresponding to inner edge of the spiralling surface. Agerage radius is constrained as symbolic function of height. Randomness is introduced by offseting points along radius at each increment. Vertical wall surfaces also follow the leading edge, the height of which varies as a function of sine. Other exposed parameters include initial radius, vertical step, rotation increment, and randomness, as well as material thickness. Structural intellegence was not requested. model was used primarily to explore design space and communicate to client possibilities of spralling form.

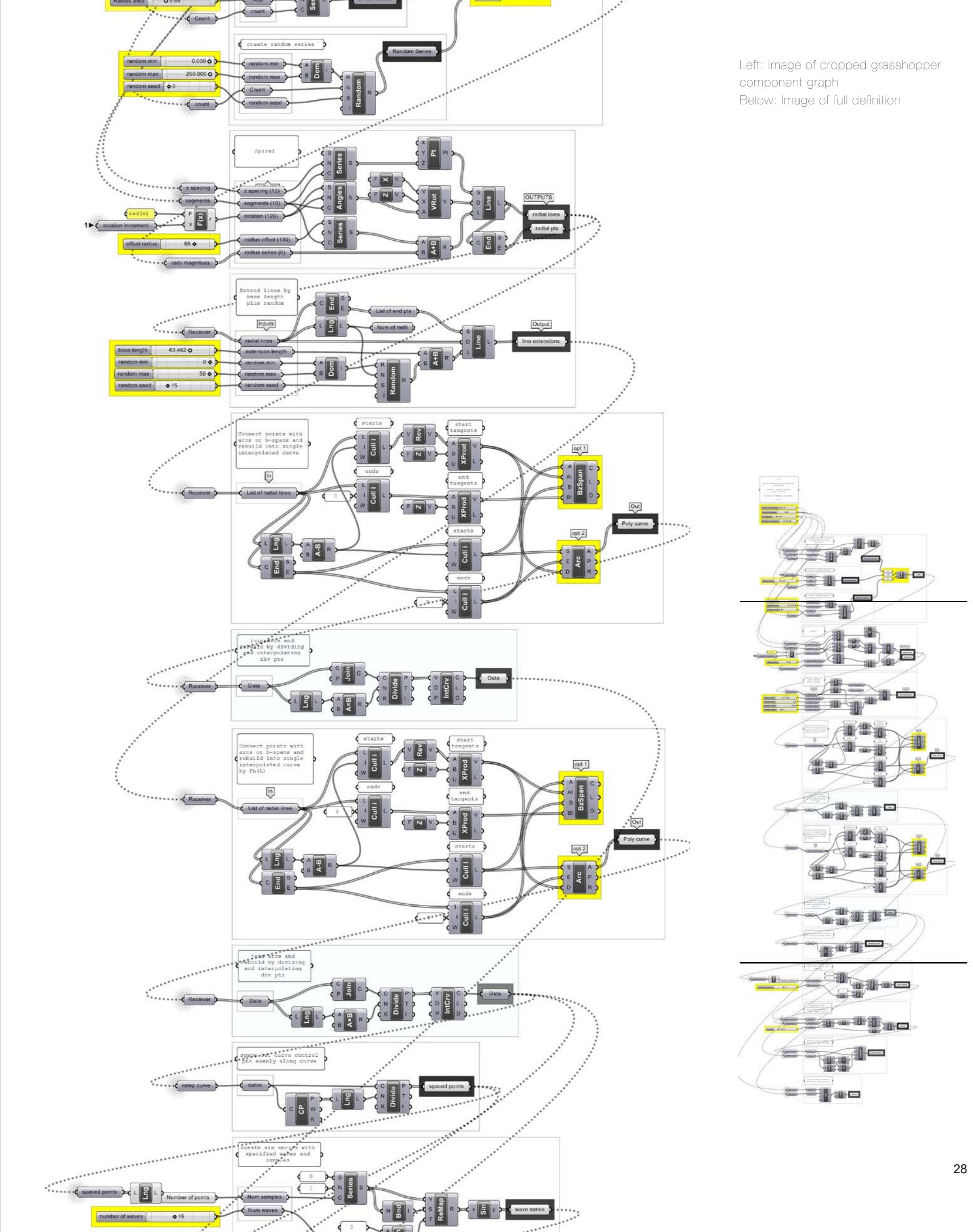
Scope: Custom parametric Tool for parking garage study | Collabortors: OPEN Archtecture, LI HU | Contribution: Consultant for OPEN Architecture, Computational designer, form-finding



1-2. Above: plan and elevation of test spiral

3. Concept rendering of test form  
4. Compression and wave frequency of edge surface ensures high degree of structural overlapping.

opposite: grasshopper screenshot illustrating inner and outer asynchronous spiralling.

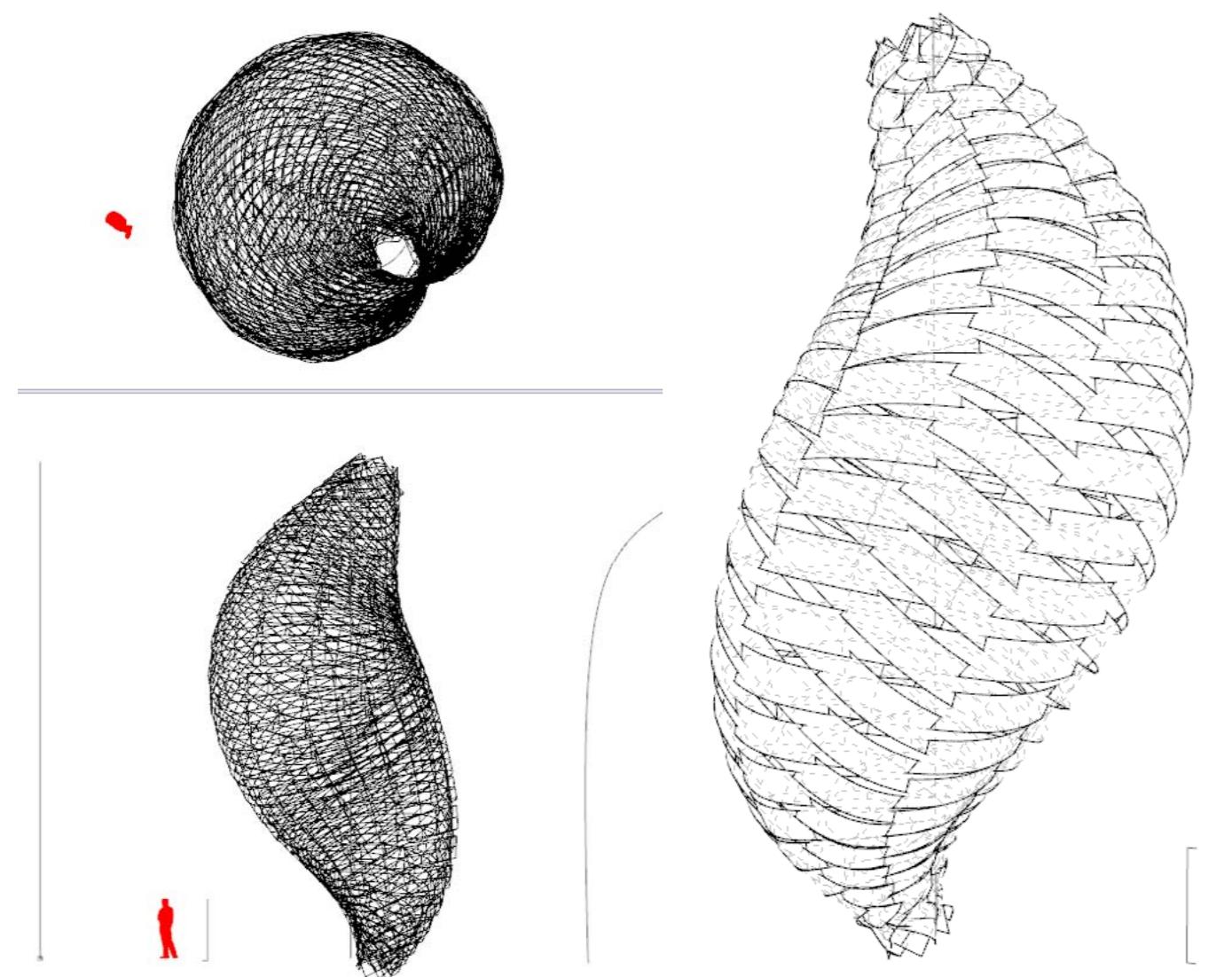


Left: Image of cropped grasshopper component graph  
Below: Image of full definition

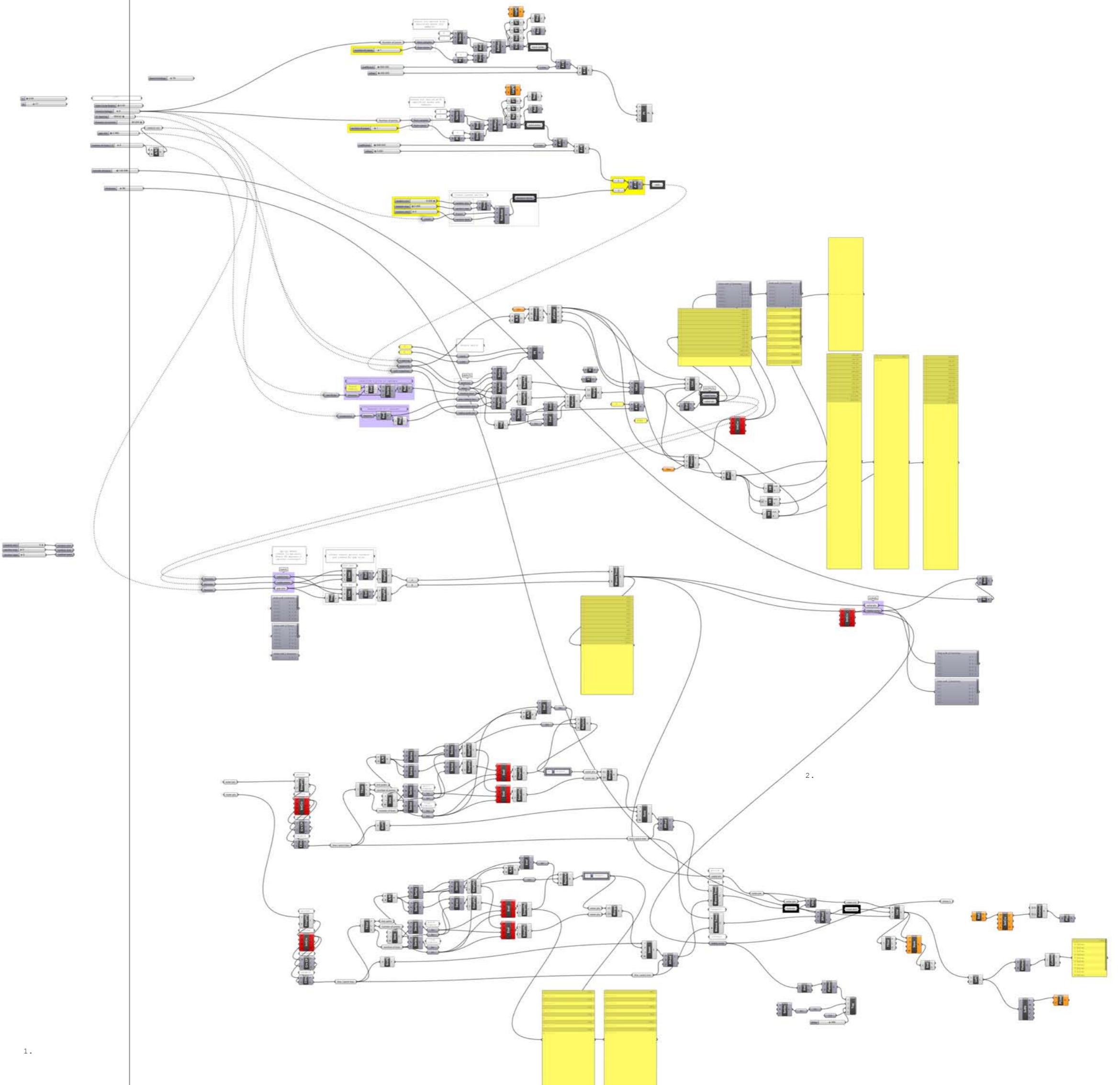
## WEAVING: URBAN FURNITURE / SHADING

Weaving component Designed/implemented with grasshopper, visual programming rhino plugin for urban design furniture charrette in collaboration with Kristiana Leniart (AECOM), and later generalized as multi-scalar generic weaving tool. Custom Algorithm builds on spiralling algorithm used in early study. Defines x number of spirals starting every  $360/x$  degrees in both directions. spirals offset radially at points of overlap to avoid intersection. This approximates a basic basket weave pretty well, however fails to orient strip surface normals perpendicular to base form. This functionality was added and Algorithm was further developed to allow for symbolic and geometric input to define base form.

Scope: Custom parametric Tool | Collaborators: Kristiana Leniart | Contribution: Computational design



1. Test form following s-curve with sinusoidal expansion and interlocking weave
2. Test form for shading device with non-overlapping strips



## SAMPLING: JAZZ TOWERS, BAOTOU

Jazz Towers is one of the many urban typologies within the BTCBD City of Plazas. The project responds to issues of gated development by placing a horizontal commercial perimeter block to surround the ten vertical residential towers, which allows both protection and accessibility. An open plaza harnessed by the horizontal band and on the roof of ground floor parking structure is inserted in the center. It provides public space for the residents on a higher ‘ground’, as well as linking to the social network of open spaces of the master plan.

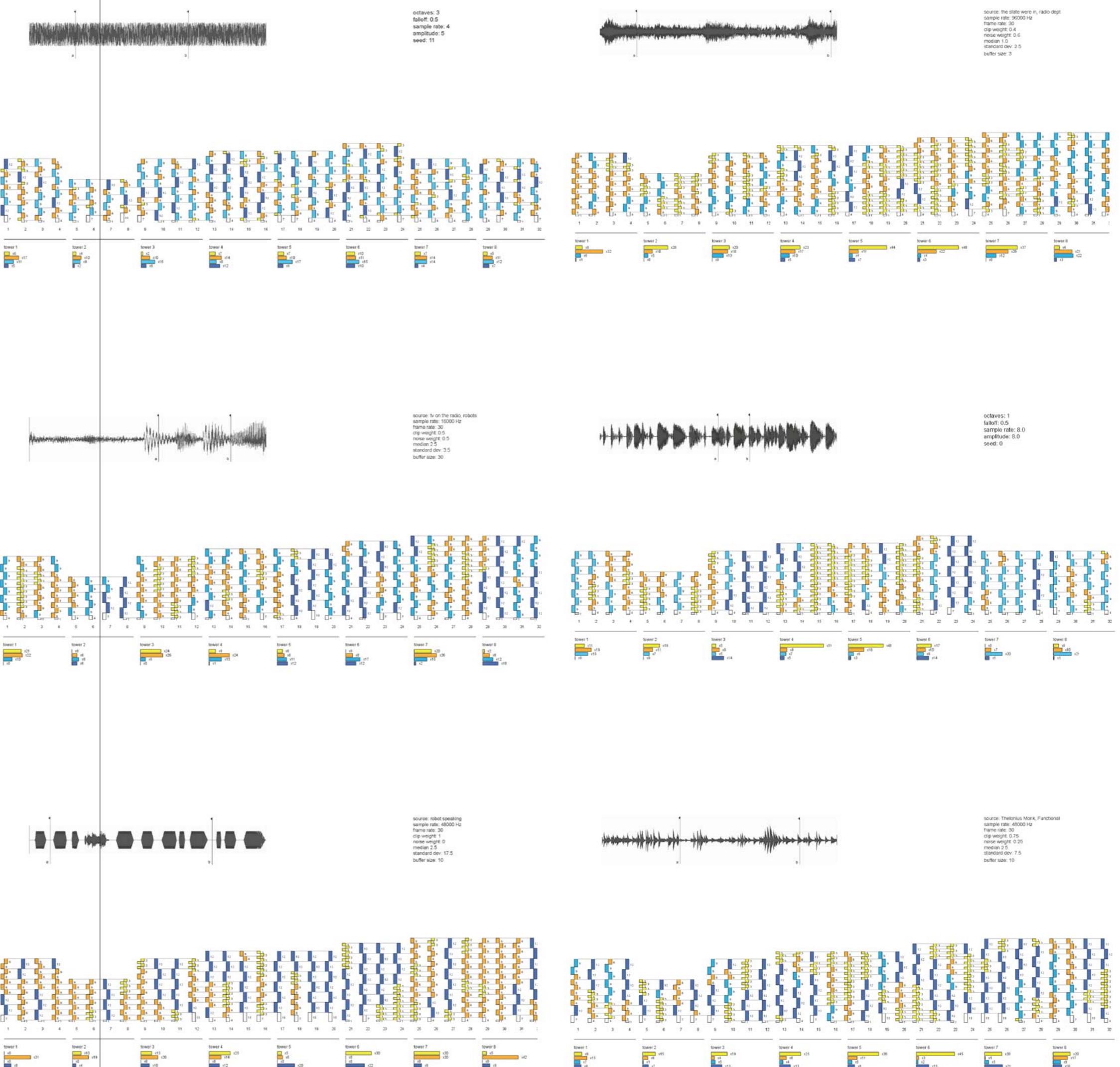
The towers, measured 18m x 18m with central cores, are proposed with open plan for flexible unit layouts. The corners of the towers cantilever out 1.5m to provide each space with its own unique perspective increasing connection from the interior space to the city. Inspired by the spontaneous yet systematic structure of Jazz music, the cantilever view-boxes are composed on the elevations of the towers through parametric calculation of Jazz musical sound wave. The musical rhythmic façade allows spacio-temporal movement by placing strategic architectonic elements in space to influence human moving and visual patterns. When one meanders around the “Towers of Interlocking Views”, it evokes a sequence of harmonic spaces as well as the dynamic and free energy of urban life.

Scope: residential towers | Collaborators: OPEN Architecture, LI HU (Principal in charge) | Role: Consultant for OPEN | Contribution: Custom design tool



1. building elevations  
2. architectural model

text and images courtesy of OPEN Architecture  
more at <http://openarch.com/task/1456>



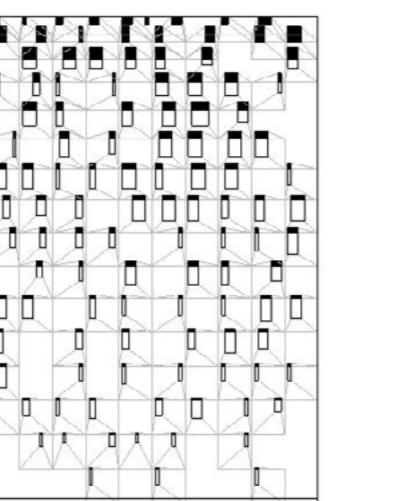
## **SOURCE CODE SAMPLE, UPLIFT HANGER**

Code written for Interactive paneling tool developed for use in design of structurally integrated prefabricated concrete panel system for design competition of airship hanger in anhui, china for competition submission completed in collaboration with OPEN Architecture and the Chinese Academy of Building Research (CABR)

Uses the following processing libraries:

controlP5 - gui toolkit

- <http://www.sojamo.de/libraries/controlP5/>  
Proscene - 3d scene library
  - <http://code.google.com/p/proscene/>  
Objloader - 3d object loader
  - <http://code.google.com/p/saitoobjloader/>



right: perforation pattern on skin segment.

33

```

*****tutes
Joshua Parker
011

tive paneling tool developed for use in
of structurally integrated prefabricated
e panel system for design competition
hip hanger in anhui, china for competition
ion completed in collaboration with OPEN
ture and the Chinese Academy of Building
n (CABR)

ntated with Simple Model-View-Controller
attern:

s a wrapper for controlP5, which supplies
er interface toolkit
olListener - recieves ui events, ids and
s them to controller
ller - contains methods for updating data
& controlling view(s)*
contains the scene(empty space and camera),
ted 3d base model, and data model defining
system. View calls draws to screen via
ate interface, though only one viewType
plemented: View3d, a basic 3d view.
- data model defining panel system, composed
oclass skin, which is composed of panels.
s are basically two sets of parameters: one
defines the panel's size and position in
, and another that defines the size and
ive position of a single rectangular void
e panel. Panel also contains data from ecotect
the amount of direct and diffuse radiation
panel recieves.

e following processing libraries:
P5 - gui toolkit
//www.sojamo.de/libraries/controlP5/
e - 3d scene library
//code.google.com/p/proscene/
er - 3d object loader
//code.google.com/p/saitoobjloader/
ted License: GPL, V3
Natures Lab | www.fnl.com

*****processing.core.*;
controlP5.*;
processing.opengl.*;
controller.Listener;
controller.Controller;
model.Hanger;
view.*;
gui.*;

class Apernatures extends PApplet{

private static final serialVersionUID = 1L;

//CONSTANTS
int U = 37; //number of modules in U direction
int V = 42; //number of modules in V direction
float D = 10f; //panel length/width dimension, D

float T = .5f; //thickness, T of frame (meters)
float SUBDIVS = 20f; //subdivide panel for snapping

//path to tabular data from ecotect analyses
String PATH = "data/ecotectData.txt";

//objects
Listener myListener;
Controller myController;
Hanger myhanger;
View myview;
Gui mygui;

public void setup(){
size(1280, 1020, PGraphicsOpenGL.OPENGL);
background(0);
noStroke();
}

//smooth();
//MODEL: Parametric Data for panel system
float snap = (float) ((D-(T*2))/SUBDIVS);
myhanger = new Hanger(this, U, V, D, T, snap,
PATH);

//VIEW: draws a representation of model to
screen
myview = new View(this, myhanger);

//GUI: needs to access view scene for drawing 2d
gui on top of 3d scene
mygui = new Gui(this, myview);

//CONTROLLER: updates model & views in response
to ui events routed to it
myController = new Controller(this, myhanger,
mygui, myview);

//LISTENER/ROUTER: receives ui events, ids and
routes them to controller
myListener = new Listener(this, myController);

}

public void draw(){
background(50);

//Continually redraw view and gui via controller
myController.update();
}

//required to run as application
public static void main(String args[]) {
PApplet.main(new String[] { "--present", "Apern-
tures" });
}

//send ui events to listener, so it routes them to
controller
public void controlEvent(ControlEvent theEvent){
(myListener.controlEvent(theEvent));
public void mousePressed(){myListener.
mousePressed(mouseX, mouseY);}
public void keyPressed(){myListener.
keyPressed(key);}
}

*****Controller Class
MVC Controller contains methods for updating data
model & controlling view(s)*

Part of:
Aper[na]tures
Author: Joshua Parker
Date: 2011

Inherited License: GPL, V3

*****package controller;

import processing.core.PApplet;
import processing.core.PVector;
import model.Hanger;
import view.View;
import gui.Gui;

public class Controller{
//The parent PApplet
PApplet p;

//PROPS
private Hanger myhanger;
private View myview;
private Gui mygui;

//CONSTRUCTOR
public Controller(PApplet parent, Hanger Myhanger,
Gui Mygui, View Myview){
p = parent;
myhanger = Myhanger;
mygui = Mygui;
myview = Myview;
}

// PUBLIC METHODS /////////////////////////////////
////////////////////////////

//gui noise increment
public void setNoiseInc(float v){
myhanger.setInc(v);
}

//set noise seed
public void setNoiseSeed(int v){
myhanger.setSeed(v);
}

//set coefficient with index i
public void setPanelXcf(int i, float C){
myhanger.setXcf(i, C);
}

//public void moveCameraByTheta(float v){
PVector v0;
float r = 500;
float phi = 0;
float theta = 0;
//r = myview.scene.camera().sceneRadius();
//r = 500;
v0 = myview.scene.camera().position();
phi = (float) Math.atan(v0.y/v0.x);
theta = PApplet.radians(v); //azimuth 0-2PI
setCameraPosition(phi, theta, r);
}

//public void moveCameraByPhi(float v){
PVector v0;
float r = 500;
float phi = 0;
float theta = 0;
//r = myview.scene.camera().sceneRadius();
//r = 500;
v0 = myview.scene.camera().position();
theta = (float) Math.acos(v0.z/r);
phi = PApplet.radians(v); //azimuth 0-2PI
setCameraPosition(phi, theta, r);
}

//public void setCameraPosition(float phi, float
theta, float r){
float x = r*(PApplet.sin(theta)*PApplet.
cos(phi));
float y = r*(PApplet.sin(theta)*PApplet.
sin(phi));
float z = r*PApplet.cos(theta);

PVector v = new PVector(x,y,z);
PVector v2 = new PVector(0,0,0);
PVector v3 = new PVector(0,0,-1);

myview.scene.camera().setPosition(v);
myview.scene.camera().setOrientation(theta,
phi);
myview.scene.camera().setUpVector(v3,true);
myview.scene.camera().lookAt(v2);
}

//public void focusGui(){
myview.scene.disableMouseHandling();
myview.scene.disableKeyboardHandling();
}

//public void focusScene(){
myview.scene.enableMouseHandling();
myview.scene.enableKeyboardHandling();
}
}

```

34



```

//add group 2 controllers
cp5.addSlider("noise inc")
.setId(1)
.moveTo(g2)
.setPosition(slideX,slideDY*1)
.setRange(0.0f,1.0f)
.setValue(.4f);

cp5.addSlider("noise seed")
.setId(2)
.moveTo(g2)
.setPosition(slideX,slideDY*2)
.setSize(slideW,slideH)
.setRange(0,50)
.setValue(8);

//add group 3 controllers
cp5.addSlider("total radiation")
.setId(5)
.moveTo(g3)
.setPosition(slideX,slideDY*1)
.setSize(slideW,slideH)
.setRange(0,10)
.setValue(0);

cp5.addSlider("direct radiation")
.setId(6)
.moveTo(g3)
.setPosition(slideX,slideDY*2)
.setSize(slideW,slideH)
.setRange(0,10)
.setValue(0);

cp5.addSlider("diffuse radiation")
.setId(7)
.moveTo(g3)
.setPosition(slideX,slideDY*3)
.setSize(slideW,slideH)
.setRange(0,10)
.setValue(0);

cp5.addSlider("by grade")
.setId(8)
.moveTo(g3)
.setPosition(slideX,slideDY*4)
.setSize(slideW,slideH)
.setRange(0,10)
.setValue(0);

cp5.addSlider("by elevation")
.setId(9)
.moveTo(g3)
.setPosition(slideX,slideDY*5)
.setSize(slideW,slideH)
.setRange(0,10)
.setValue(0);

cp5.addSlider("noise")
.setId(10)
.moveTo(g3)
.setPosition(slideX,slideDY*6)
.setSize(slideW,slideH)
.setRange(0,20)
.setValue(0);

cp5.addSlider("x scale")
.setId(11)
.moveTo(g3)
.setPosition(slideX,slideDY*7)
.setSize(slideW,slideH)
.setRange(-1.0f,1.0f)
.setValue(1.0f)
.setNumberOfTickMarks(21)
.snapToTickMarks(true)
.showTickMarks(false);

cp5.addSlider("x offset")
.setId(12)
.moveTo(g3)
.setPosition(slideX,slideDY*8)
.setSize(slideW,slideH)
.setRange(-1,1)
.setValue(.1f)
.setNumberOfTickMarks(100)

cp5.addSlider("x trim")
.setId(13)
.moveTo(g3)
.setPosition(slideX,slideDY*9)
.setSize(slideW,slideH)
.setRange(0,3)
.setValue(0)
.setNumberOfTickMarks(11)
.snapToTickMarks(true)
.showTickMarks(false);

cp5.addSlider("y scale")
.setId(14)
.moveTo(g3)
.setPosition(slideX,slideDY*10)
.setSize(slideW,slideH)
.setRange(-3,3)
.setValue(1)
.setNumberOfTickMarks(100)
.snapToTickMarks(false)
.showTickMarks(false);

cp5.addSlider("y offset")
.setId(15)
.moveTo(g3)
.setPosition(slideX,slideDY*11)
.setSize(slideW,slideH)
.setRange(-1,1)
.setValue(.1f)
.setNumberOfTickMarks(100)
.snapToTickMarks(true)
.showTickMarks(false);

cp5.addSlider("y trim")
.setId(16)
.moveTo(g3)
.setPosition(slideX,slideDY*12)
.setSize(slideW,slideH)
.setRange(0,3)
.setValue(0)
.setNumberOfTickMarks(11)
.snapToTickMarks(true)
.showTickMarks(false);

cp5.addSlider("w scale")
.setId(17)
.moveTo(g3)
.setPosition(slideX,slideDY*13)
.setSize(slideW,slideH)
.setRange(-3,3)
.setValue(1)
.setNumberOfTickMarks(100)
.snapToTickMarks(true)
.showTickMarks(false);

cp5.addSlider("w offset")
.setId(18)
.moveTo(g3)
.setPosition(slideX,slideDY*14)
.setSize(slideW,slideH)
.setRange(-1,1)
.setValue(0)
.setNumberOfTickMarks(100)
.snapToTickMarks(false)
.showTickMarks(false);

cp5.addSlider("w trim")
.setId(19)
.moveTo(g3)
.setPosition(slideX,slideDY*15)
.setSize(slideW,slideH)
.setRange(0,3)
.setValue(0)
.setNumberOfTickMarks(11)
.snapToTickMarks(true)
.showTickMarks(false);

cp5.addSlider("h scale")
.setId(20)
.moveTo(g3)
.setPosition(slideX,slideDY*16)
.setSize(slideW,slideH)
.setRange(-3,3)
.setValue(1)
.setNumberOfTickMarks(100)

cp5.addSlider("h offset")
.setId(21)
.moveTo(g3)
.setPosition(slideX,slideDY*17)
.setSize(slideW,slideH)
.setRange(-1,1)
.setValue(0)
.setNumberOfTickMarks(100)
.snapToTickMarks(true)
.showTickMarks(false);

cp5.addSlider("h trim")
.setId(22)
.moveTo(g3)
.setPosition(slideX,slideDY*18)
.setSize(slideW,slideH)
.setRange(0,3)
.setValue(0)
.setNumberOfTickMarks(11)
.snapToTickMarks(true)
.showTickMarks(false);

//add group 4 controllers
cp5.addKnob("azimuth")
.setId(13)
.moveTo(g4)
.setPosition(20,20)
.setRadius(35)
.setAngleRange(2*PApplet.PI)
.setStartAngle((float)(.5*PApplet.PI))
.setRange(0,360)
.setValue(0)
.setNumberOfTickMarks(16)
.snapToTickMarks(true)
.setColorForeground(p.color(255))
.setColorBackground(p.color(40))
.setColorActive(p.color(255,255,0))
.setDragDirection(Knob.HORIZONTAL);

cp5.addKnob("elevation")
.setId(14)
.moveTo(g4)
.setPosition(110,20)
.setRadius(35)
.setAngleRange(PApplet.PI)
.setStartAngle((float)(1.5*PApplet.PI))
.setRange(1,179)
.setValue(0)
.setNumberOfTickMarks(8)
.snapToTickMarks(true)
.setColorForeground(p.color(255))
.setColorBackground(p.color(40))
.setColorActive(p.color(255,255,0))
.setDragDirection(Knob.HORIZONTAL);

cp5.addKnob("*****")
//Add Sidebar Accordion: *****
*****accordion = cp5.addAccordion("acc")
.setPosition(0,0)
.setWidth(200)
.addItem(g1)
.addItem(g2)
.addItem(g3)
.addItem(g4);
;

//initially open all groups
accordion.open(0,1,2,3);

//allow multiple groups open at a time.
accordion.setCollapseMode(Accordion.MULTI);

//Add keyboard shortcuts *****
*****cp5.mapKeyFor(new ControlKey() {public void keyEvent() {accordion.open(0,1,2,3);}}, 'o');
cp5.mapKeyFor(new ControlKey() {public void keyEvent() {accordion.close(0,1,2,3);}}, 'c');

//end init()

```

```

//function from https://forum.processing.org/
oscene-and-2d-drawing
private void saveState(Scene scene) {
    // Disable depth test to draw 2d on top
    p.hint(DISABLE_DEPTH_TEST);

    // Set processing projection and model-
rices to draw in 2D:
    // 1. projection matrix:
    float cameraZ = ((p.height/2.0f) / PApp-
(PApplet.PI*60.0f/360.0f));
    scene.pg3d.perspective(
        PApplet.PI/3.0f,
        scene.camera().as-
o(),
        cameraZ/10.0f,
        cameraZ*10.0f);

    // 2 model view matrix
    scene.pg3d.camera();
}

//function from https://forum.processing.org/
oscene-and-2d-drawing
private void restoreState(Scene scene) {
    // 1. Restore processing projection ma-
switch (scene.camera().type()) {
    case PERSPECTIVE:
        scene.pg3d.perspective(
            scene.cam-
ldOfView(),
            scene.cam-
pectRatio(),
            scene.cam-
ear(),
            scene.cam-
ar());
        break;
    case ORTHOGRAPHIC:
        float[] wh = scene.camera().w
WidthHeight();
        scene.pg3d.ortho(
            -wh[0],
            wh[1], wh[1],
            scene.cam-
ear(),
            scene.cam-
ar());
        break;
}
    // 2. Restore processing modelview ma-
scene.pg3d.camera(
    scene.camera().x,
    scene.camera().y,
    scene.camera().z,
    scene.camera().x,
    scene.camera().y,
    scene.camera().z,
    scene.camera().x,
    scene.camera().y,
    scene.camera().z);
    //Re-enable depth test
    p.hint(ENABLE_DEPTH_TEST);
}

***** data model defining panel system, composed
class

```

of subclass skin, which is composed of panels. Panels are basically two sets of parameters: one which defines the panel's size and position in space, and another that defines the size and relative position of a single rectangular void in the panel. Panel also contains data from ecotect about the amount of direct and diffuse radiation each panel receives.

Part of:

Aper[na]tures  
Author: Joshua Parker  
Date: 2011  
Inherited License: GPL, V3

---

\*\*\*\*\* package model;

```

import processing.core.PApplet;
import model.Skin;

public class Hanger{
    //The parent PApplet
    PApplet p;

    //composit skin
    public Skin myskin;

    //CONSTRUCTOR
    public Hanger(PApplet parent,
        int U,
        int V,
        float D,
        float T,
        float snap,
        String PATH){
        p = parent;
        //create skin
        myskin = new Skin(p, U, V, D, T, snap);
        //pass data path to skin so it can l
        data into panels
        myskin.loadData(PATH);
        //update skin pattern
        update();
    }

    //END CONSTRUCTOR
    /////////////////////////////////
    // PUBLIC METHODS //////////////////////

    /////////////////////////////////
    //set noise increment and update model
    public void setInc(float Inc){
        myskin.setInc(Inc);
    }

    //set noise seed and update model
    public void setSeed(int Seed){
        myskin.setSeed(Seed);
    }

    //set pattern coefficients and update model
    public void setXcf(int i, float C){
        myskin.setXcf(i, C);
    }

    //update Hanger Model
    public void update(){
        //update skin pattern
        myskin.update();
    }
}

***** Panel Class

```

Panel is subclass of Hanger data model defining panel system, composed of subclass skin, which is composed of panels.

Panels are basically two sets of parameters: one

which defines the panel's size and position in space, and another that defines the size and relative position of a single rectangular void in the panel. Panel also contains data from ecotect about the amount of direct and diffuse radiation each panel receives.

Part of:

Aper[na]tures  
Author: Joshua Parker  
Date: 2011

Inherited License: GPL, v3

```
*****  
package model;  
  
import processing.core.*;  
  
public class Panel  
{  
    //The parent PApplet  
    PApplet p;  
  
    //panel properties  
    public PVector[] coords;  
    public float x,y,w,h;  
//normalized  
    public float px, py, pd; //center-  
line of frame  
    public float t;  
//thickness of frame  
    public float ix, iy, id; //inside  
line, ie. panel  
    public float hx, hy, hw, hh; //perforations, ie.  
holes  
    public float snap;  
//panel subdivisions  
    public int pClr;  
//panel color  
  
    //imported data  
/*  
    private float avgDailyTotal;  
    private float avgDailyDirect;  
    private float avgDailyDiffuse;  
    private float grade;  
    private float elevation;*/  
  
    //normalized parameters  
    private float totalNorm;  
    private float directNorm;  
    private float diffuseNorm;  
    private float gradeNorm;  
    private float elevationNorm;  
    public float noiseVal;  
  
    //parameter weights  
    private float[] xcf;  
  
    //constructor  
    Panel(PApplet parent,  
          float px,  
          float py,  
          float pd,  
          float ix,  
          float iy,  
          float id,  
          float t,  
          float snap){  
  
        p = parent;  
  
        //println("new panel");  
  
        this.px = px;  
        this.py = py;  
        this.pd = pd;  
        this.ix = ix;  
        this.iy = iy;  
        this.id = id;  
        this.t = t;  
        //hx = this.hx;  
        //hy = this.hy;  
        //hw = this.hw;  
        //hh = this.hh;  
        this.snap = snap;  
    }  
}
```



```

///////////
    -1);
    p.directionalLight(126, 126, 126, 0, 0,
    //set view state
    public void setViewState(ViewState newState){
        this.viewState = newState;
    }
    //get current fill
    public boolean getFill(){
        return this.viewState.getFill();
    }
    setUpVector(v,true);
    myview.scene.camera().lookAt(v2);*/
    //turn off strokes, turn on surface fill
    public void enableFill(){
        //just want vertices so do not draw the
        3d model!!! this will slow sketch way down
        //myview.model.draw();
    }
    //turn off surface fill, turn on strokes
    public void disableFill(){
        this.viewState.disableFill();
    }
    //draw the view
    public void draw() {
        this.viewState.draw();
    }
}
*****  

View3d Class
View3d implements a basic 3d view
Part of:
Aper[n]tures
Author: Joshua Parker
Date: 2011
Inherited License: GPL, V3
*****  

package view;
import processing.core.*;
import saito.objloader.*;
import model.*;
public class View3d implements ViewState{
    //The parent PApplet
    PApplet p;
    //composite view
    public View myview;
    //flags
    boolean bFill = false;
    //CONSTRUCTOR
    View3d(PApplet parent, View Myview){
        p = parent;
        this.myview = Myview;
    }
    // PUBLIC METHODS ///////////////
    //turn off strokes, turn on surface fill
    public void enableFill(){
        bFill = true;
    }
    //turn off surface fill, turn on strokes
    public void disableFill(){
        bFill = false;
    }
    //get current fill
    public boolean getFill(){
        return bFill;
    }
    //draw the 3d view
    public void draw() {
        //setup lighting

```

```

    p.beginShape();
    p.vertex(vs[2].x, vs[2].y,
    Inherited License: GPL, V3
    ****
    package view;
    public interface ViewState{
        public void enableFill();
        public void disableFill();
        public boolean getFill();
        public void draw();
    }
    // PRIVATE METHODS /////////////
    //find hole vertices on face from a face
    vertices(vs),
    //face normals(ns), and face-space coordinates(cs)
    private PVector[] mapFace(PVector[] vs, PVector[]
    n, PVector[] cs){
        PVector[] pts = new PVector[0];
        for each corner, map 2d coord to 3d pt
        for (int i = vs.length-1; i >= 0; i--) {
            PVector pt = mapPt(vs[0],
            n[i], cs[i]);
            pts = (PVector[])PApplet.
            append(pts, pt);
        }
        return pts;
    }
    //find singe 3d pt from a vertice(v), normal(n),
    //and 2d relative coordinate(c)
    private PVector mapPt(PVector v, PVector n, PVec-
    tor c){
        float a = (float)Math.atan(n.z/n.y);
        a = PApplet.radians(90-PApplet.
        abs(PApplet.degrees(a)));
        float x,y,z;
        if(n.y < 0){
            x = v.x + c.x;
            y = v.y - c.y*PApplet.cos(a);
            z = v.z - c.y*PApplet.sin(a);
        }else{
            x = v.x + c.x;
            y = v.y - c.y*PApplet.cos(a);
            z = v.z + c.y*PApplet.sin(a);
        }
    }
    ****
    ViewState Class
    ViewState is an interface for various views
    Part of:
    Aper[n]tures
    Author: Joshua Parker
    Date: 2011

```

