Computational design

HARMONY HOMES

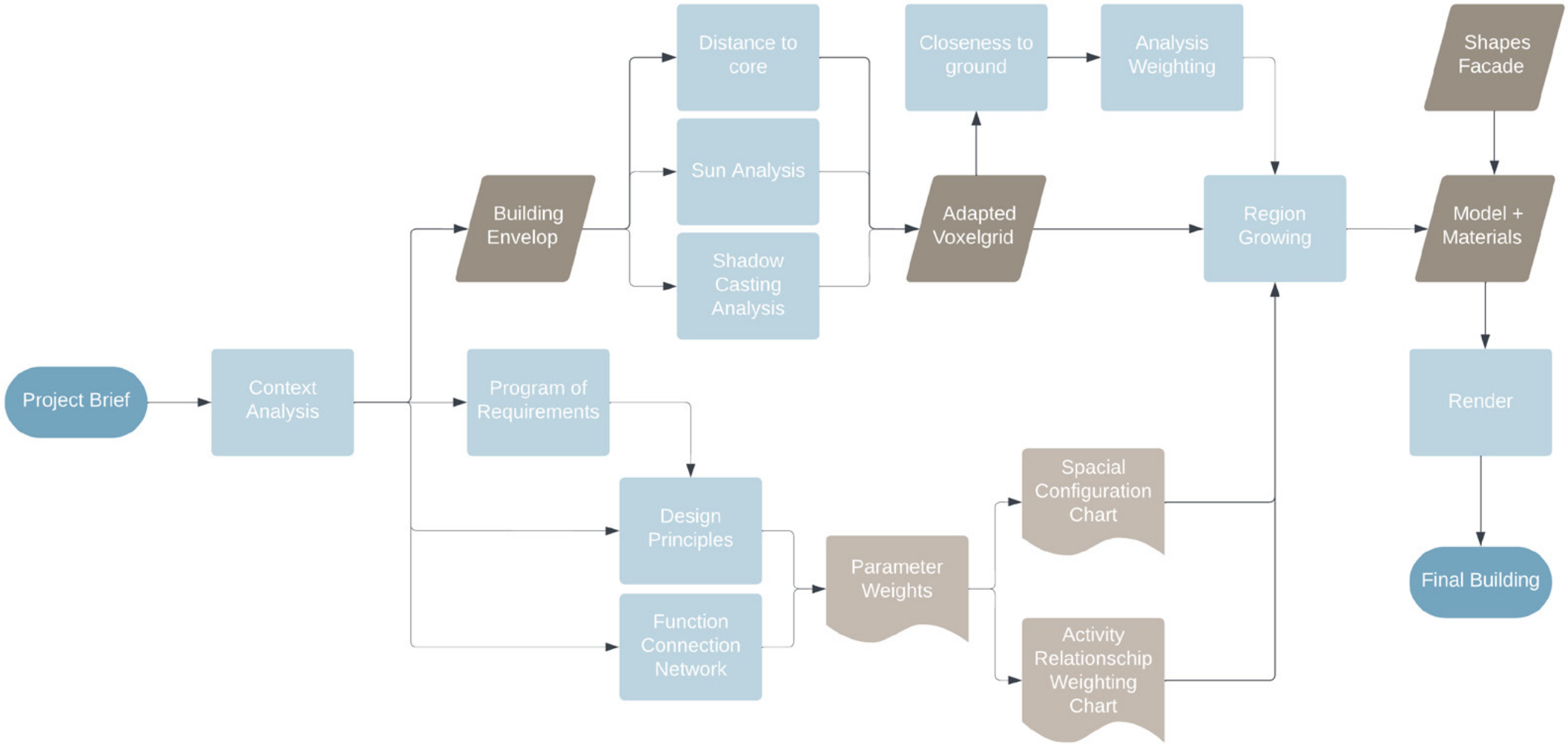
A PROJECT BY SHERLOCK HOMES

Suzanne Erenst, Lapo den Hollander, Bart Koppejan // BK7083 // 2024 // TUDelft

Contents

Design Process

Design Process



Assignment

Multifunctional building

Computationally designed

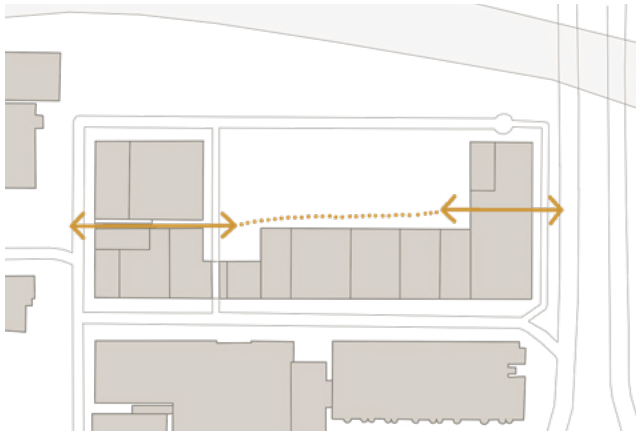
Well-connected to surroundings

Students

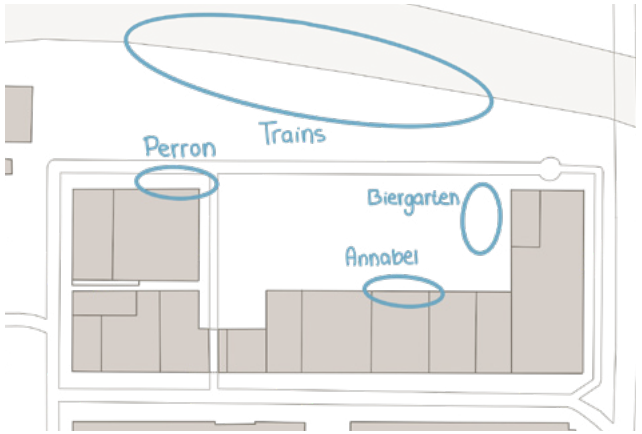
Starters

Elderly

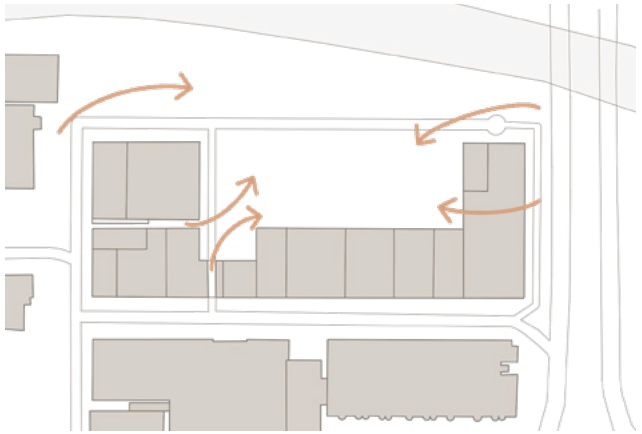
Content Analysis



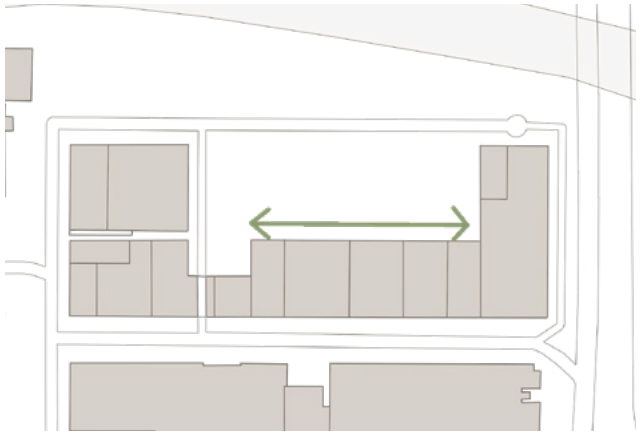
The “Yellow route”



Noise disturbance

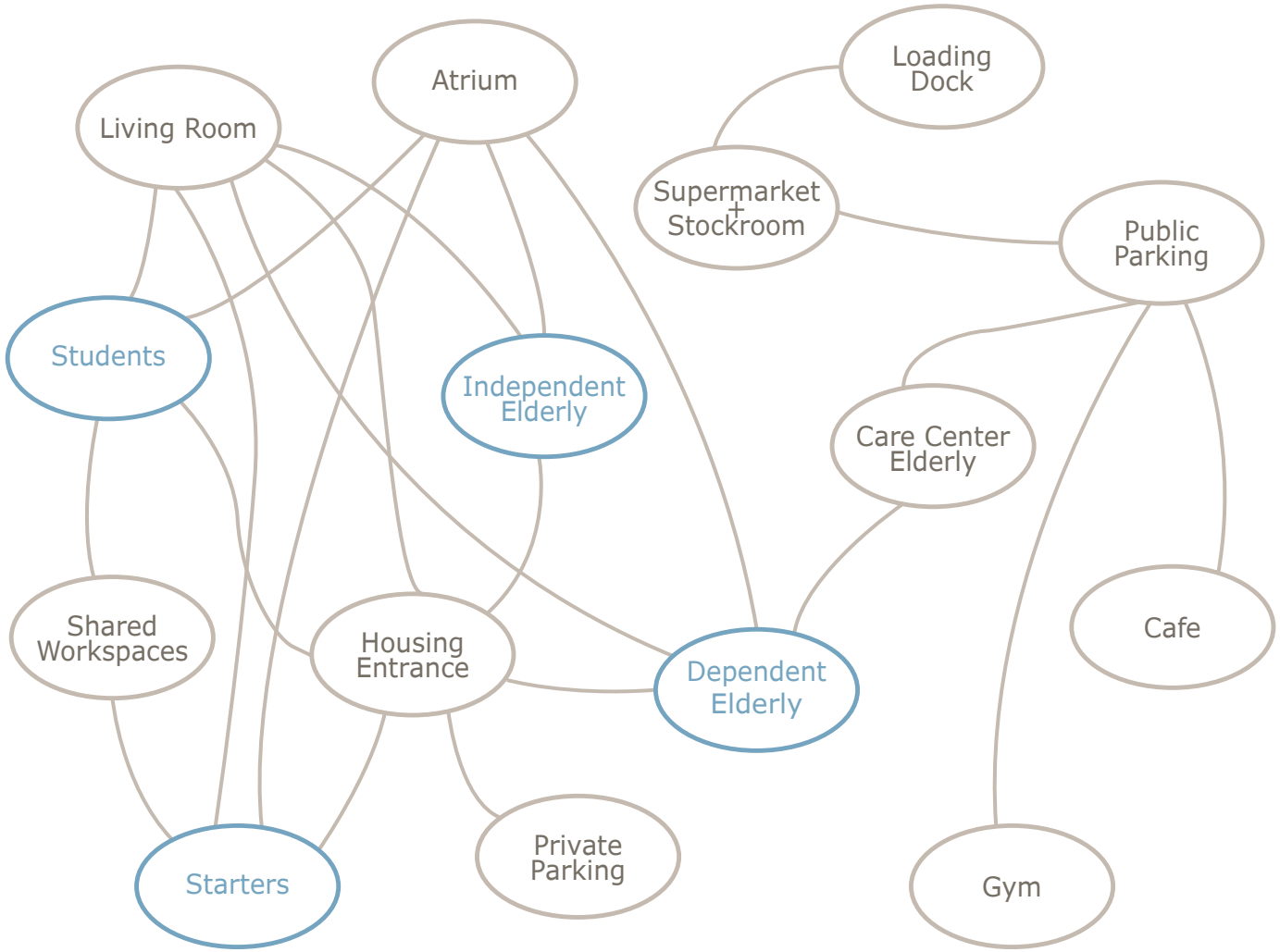


Area entries



Street with entrances

Design Principles



Connect Yellow route

Keep noise sensitive functions away from noisy areas

Open space in front of entrances from clubs

Shared semi-private areas

Atrium to get light into the building

two towers

Rainwater harvesting

Program Of Requirements

Housing Requirements

Student units: 400 units
- Studios:
  - With bathroom and kitchen
  - >25 m^2
- Multi-units:
  - Shared bathroom, kitchen and living room
  - 8 students per unit
  - >25 m^2

Elderly units: 200 units
- 50-80 m^2
- Dependent eldery connected to care center
- Accessible by elevator
- Exterior space with sunlight

Starter units: 200 units
- Accessible by stroller
- 25 m^2 per person
- up to two adults and 2 children
- With a bathroom, kitchen, 3 room apartment

Additional Requirements

Mobility
- 1.33 private bike parking spot per resident
- 0.8 private car parking spot per dwelling
- 10 publicly accesible bike parking spots per 200 m^2 of other programs
- 4 publicly accessible car parking spots per 200 m^2 of other programs

Environment
- Area for greenery equal to total plot surface area
- System for rain water harvesting in the building

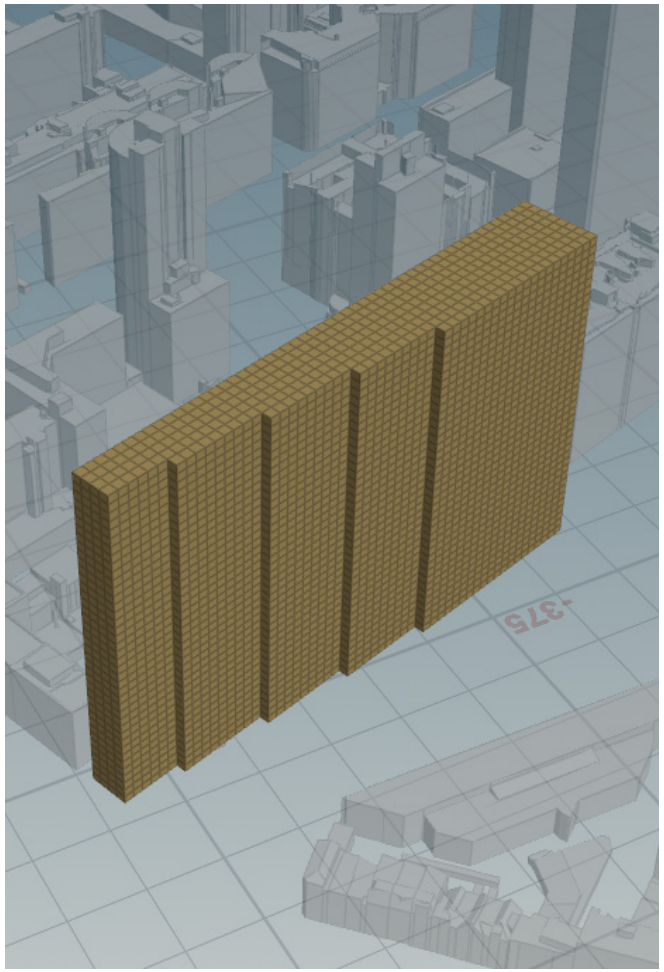
Surroundings
- No blind facades on ground level
- Preserve existing functions of surroundings
- Keep biergarten intact
- Distance of 10 meter to other facades

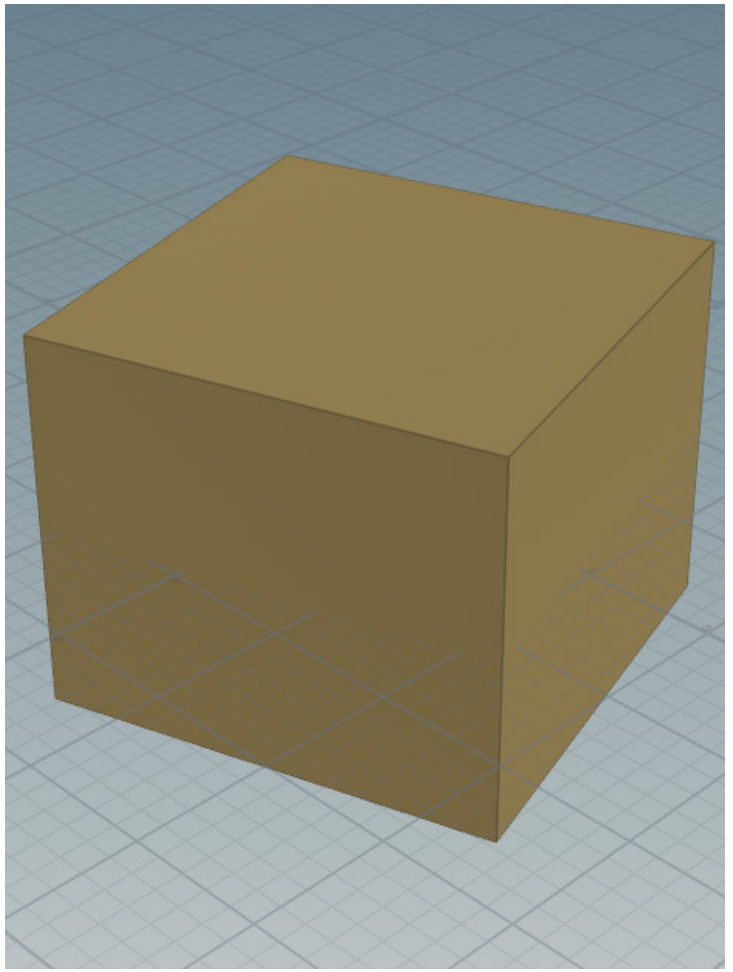
Computational goals
- Minimise blocking of sunlight on other buildings
- Maximise social integration
- Balance between privacy and community

REL Chart

	Housing entree	Starter houses	Student houses	Private parking	Independent elderly houses	dependent elderly houses	shared workspaces	Supermarket and stockroom	Care center elderly	Cafe	Loading dock- supermarket	gym	Outside	Public parking
Housing entree	self	5	5	4	5	5	3	2	3	1	1	2	2	1
Starter houses	5	Self	3	4	3	3	3	2	1	2	1	2	3	2
Student houses	5	3	Self	3	2	2	4	2	1	3	1	2	2	1
Private parking	4	4	3	Self	4	3	1	1	1	1	1	1	2	1
Independent elderly houses	5	3	2	4	Self	4	2	4	3	2	1	2	3	2
dependent elderly houses	5	3	2	3	4	Self	2	4	4	2	1	2	3	2
Shared workspaces	3	3	4	1	2	2	Self	2	2	2	1	2	2	3
Supermarket and stockroom	2	2	2	1	4	4	2	Self	4	3	5	3	2	4
Care center elderly	3	1	1	1	3	4	2	4	Self	2	1	3	4	3
Cafe	1	2	3	1	2	2	2	3	2	Self	3	3	3	3
Loading dock- supermarket	1	1	1	1	1	1	1	5	1	3	Self	2	3	2
Gym	2	2	2	1	2	2	2	3	3	3	2	Self	3	4
Outside	2	3	2	2	3	3	2	2	4	3	3	3	Self	2
Public parking	1	2	1	1	2	2	3	4	3	3	2	4	2	Self

Building enveloppe

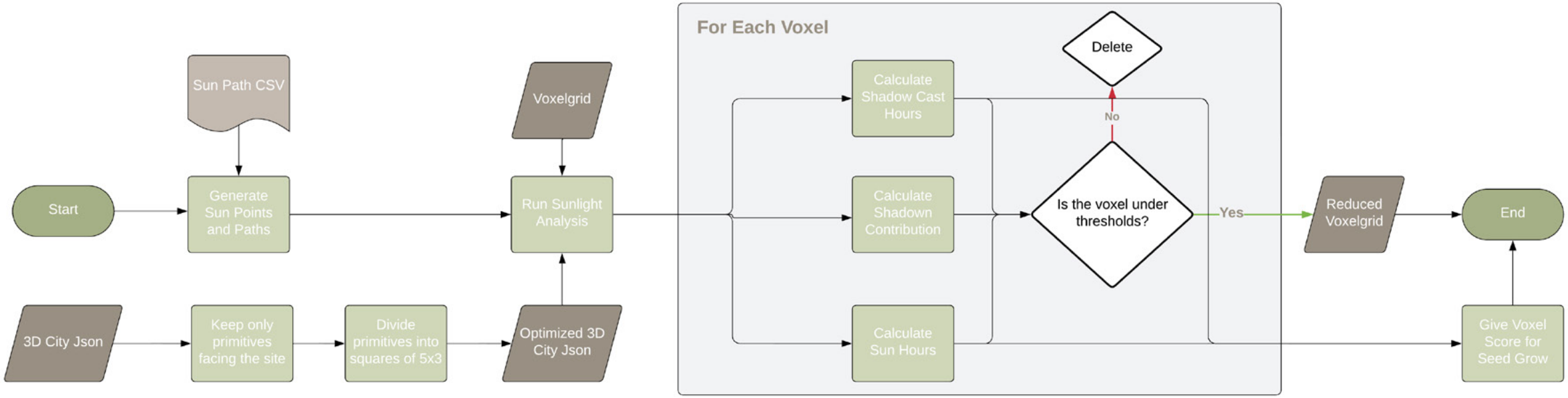




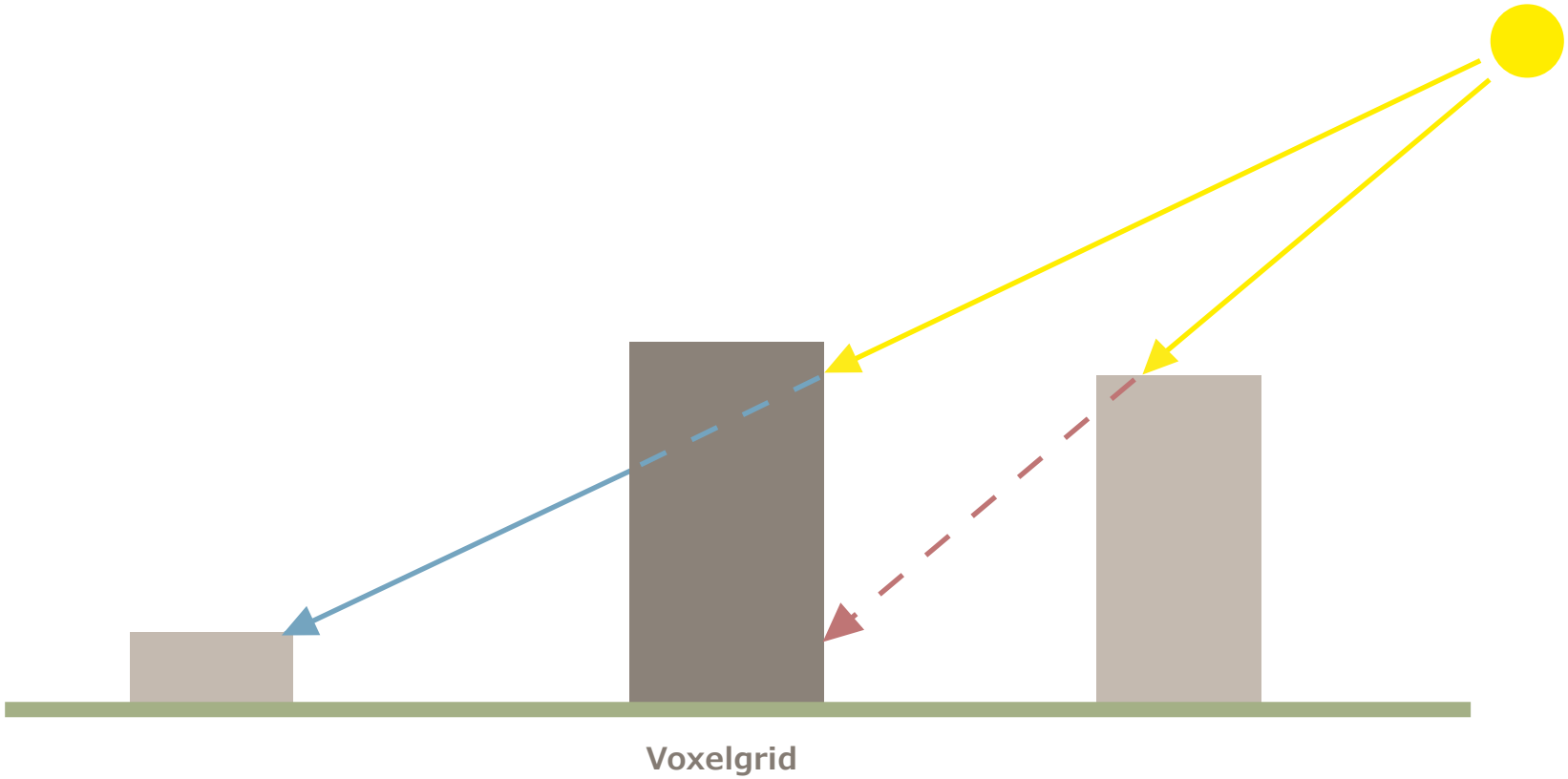
Shaped by:

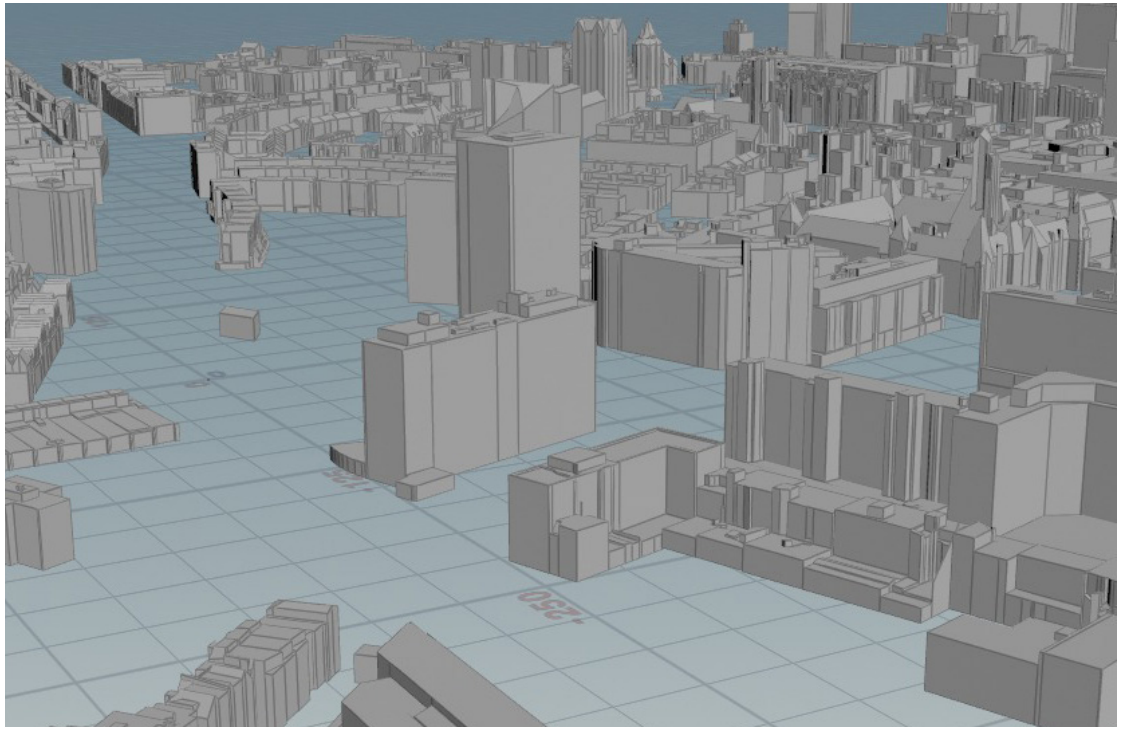
- Yellow route from biergarten
- Amount of floors
- Voxelsize
- Threshold values for sun and shadow analysis

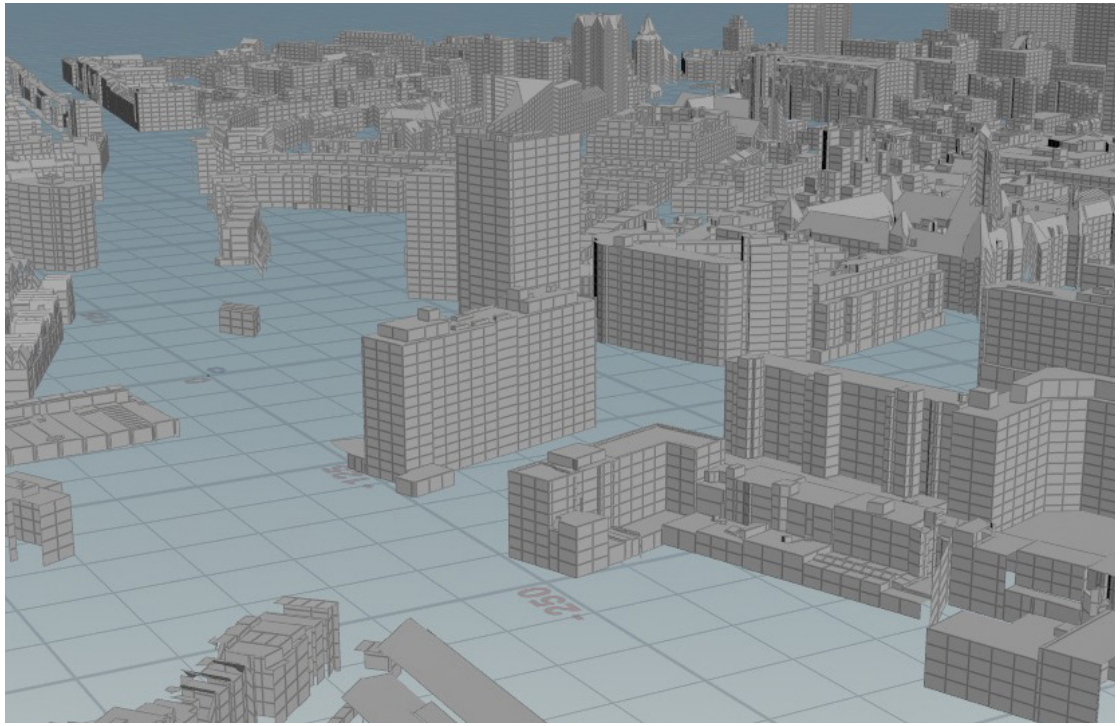
Sunlight analysis

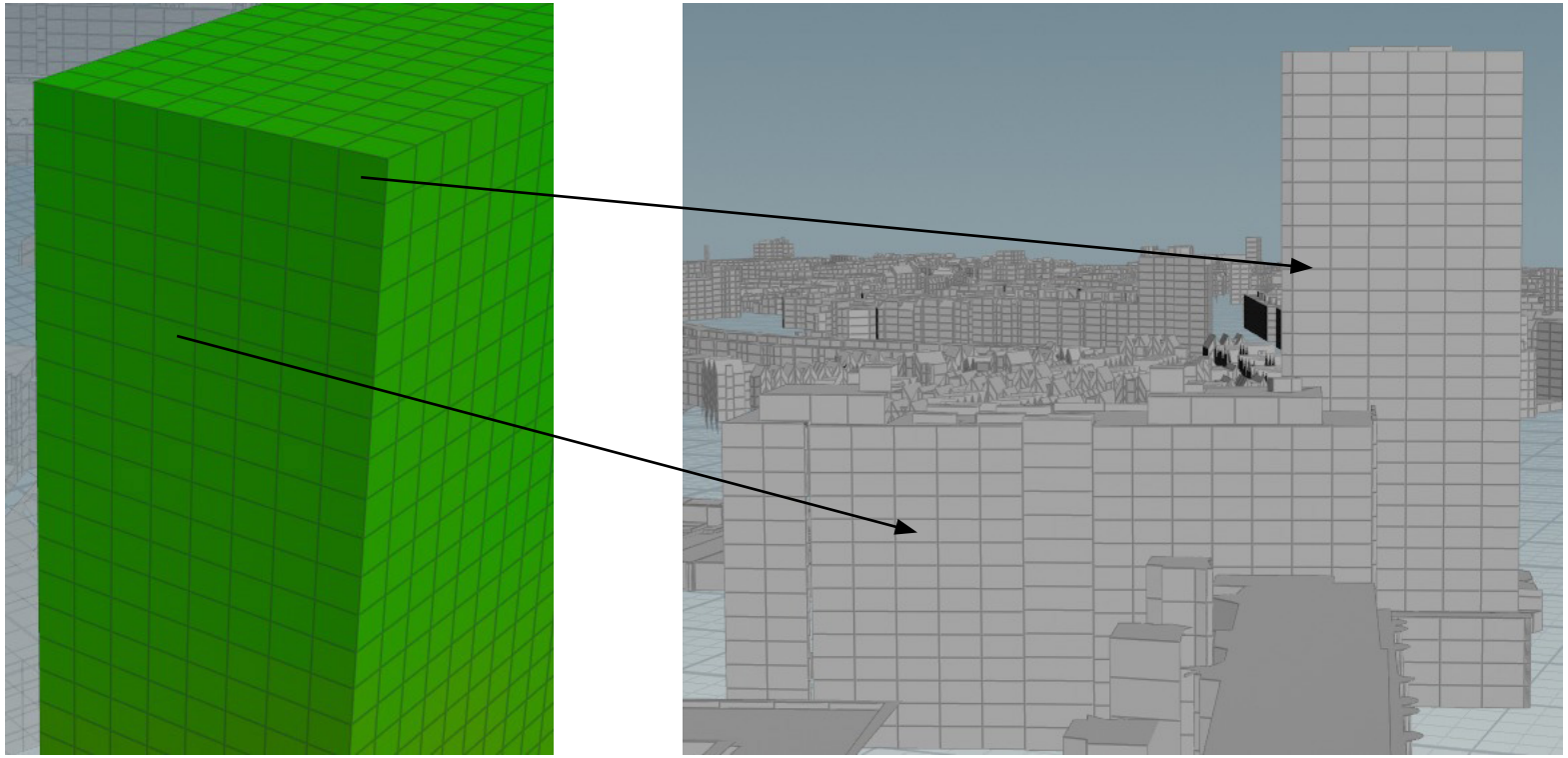


Shadow Casting Analysis



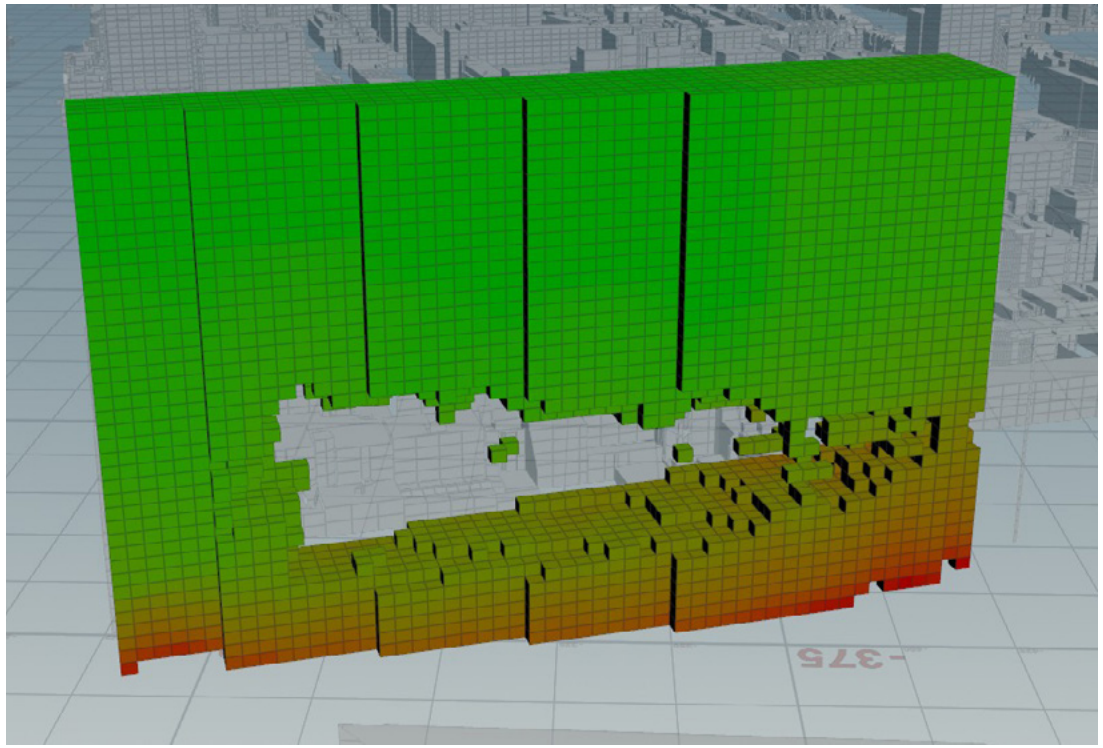




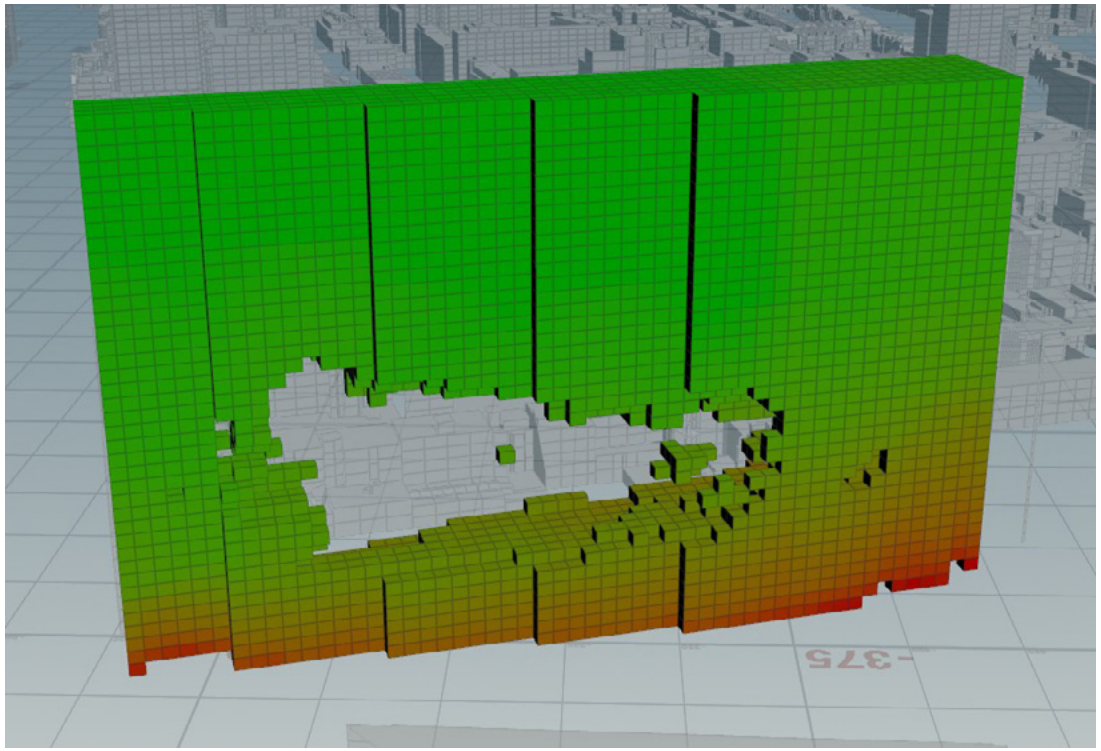


nd = amount of sunhours for day d

1/nd = shadow contribution score for voxel

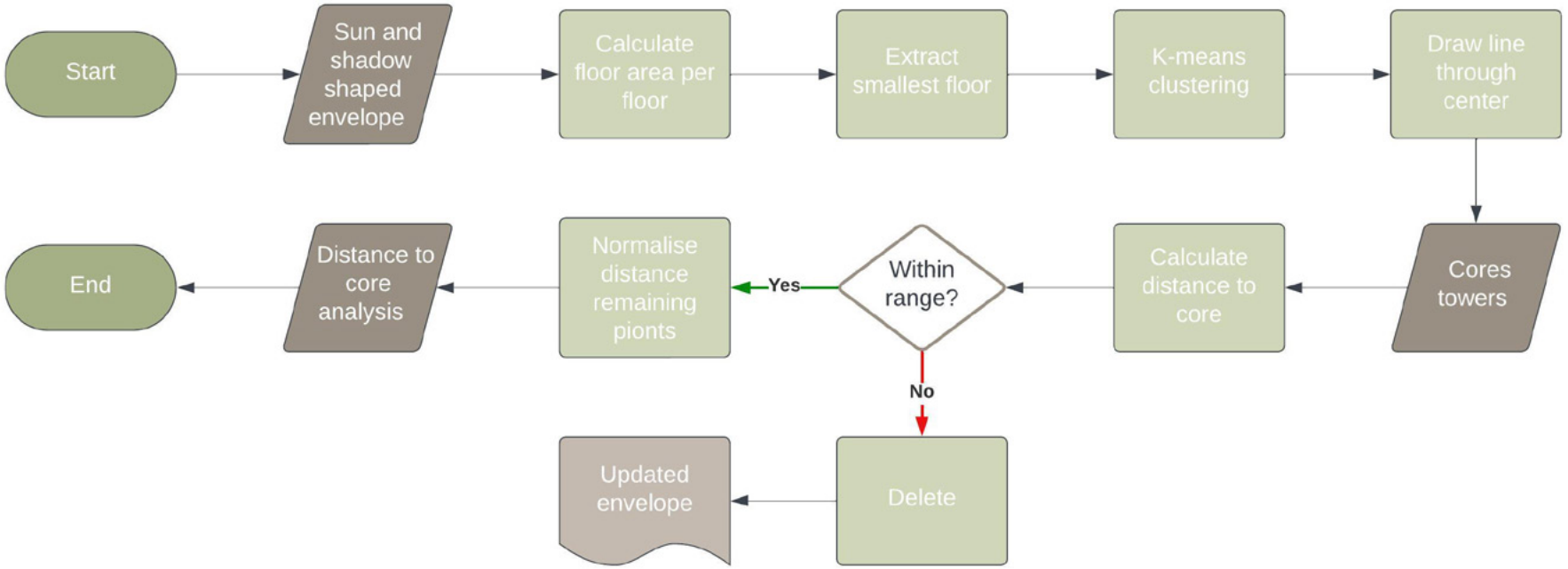


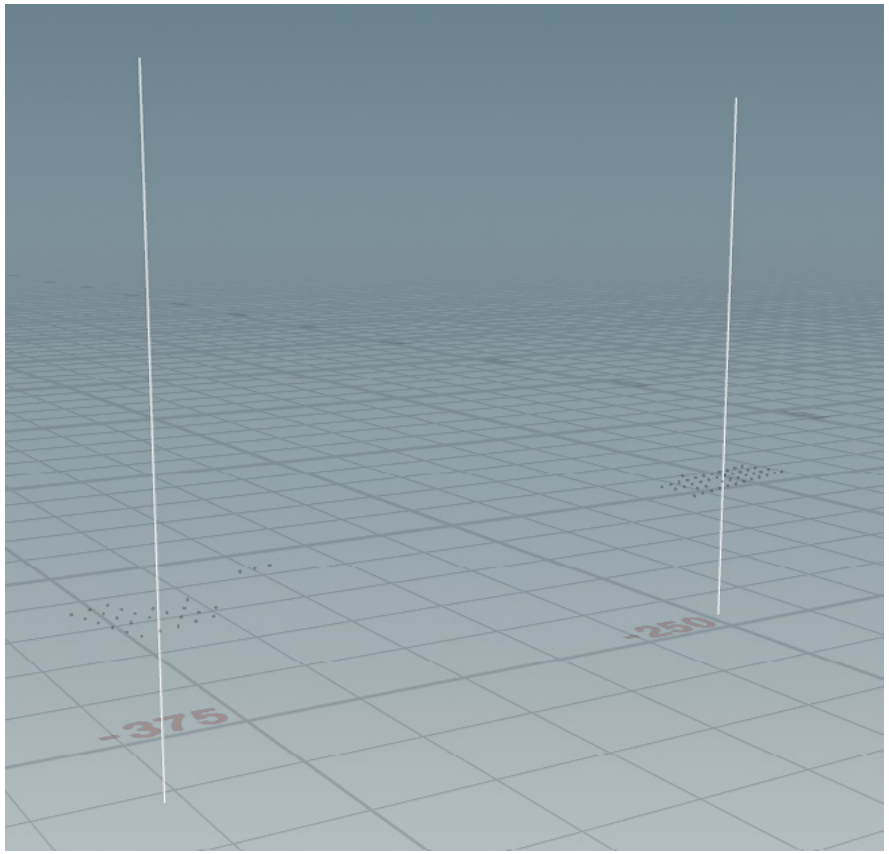
Blocking score

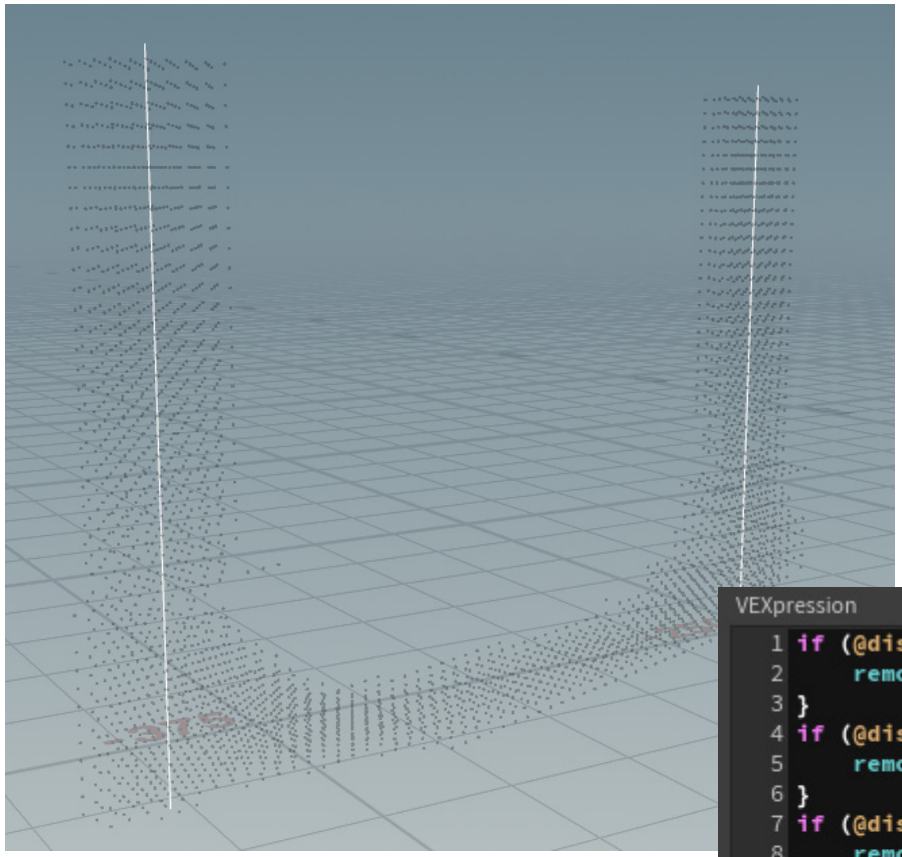


Shadow contribution score

Distance to core analysis





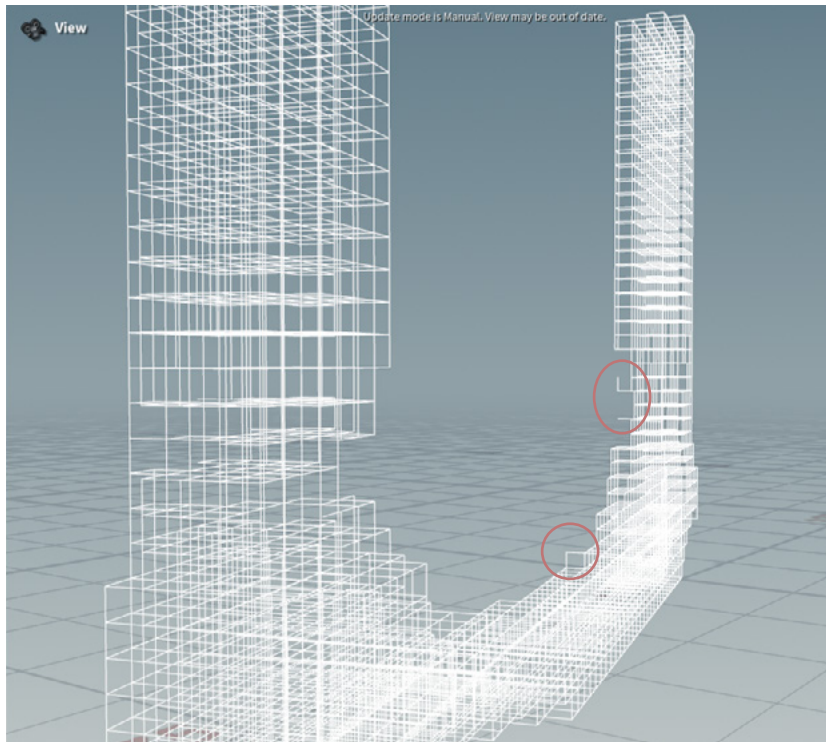


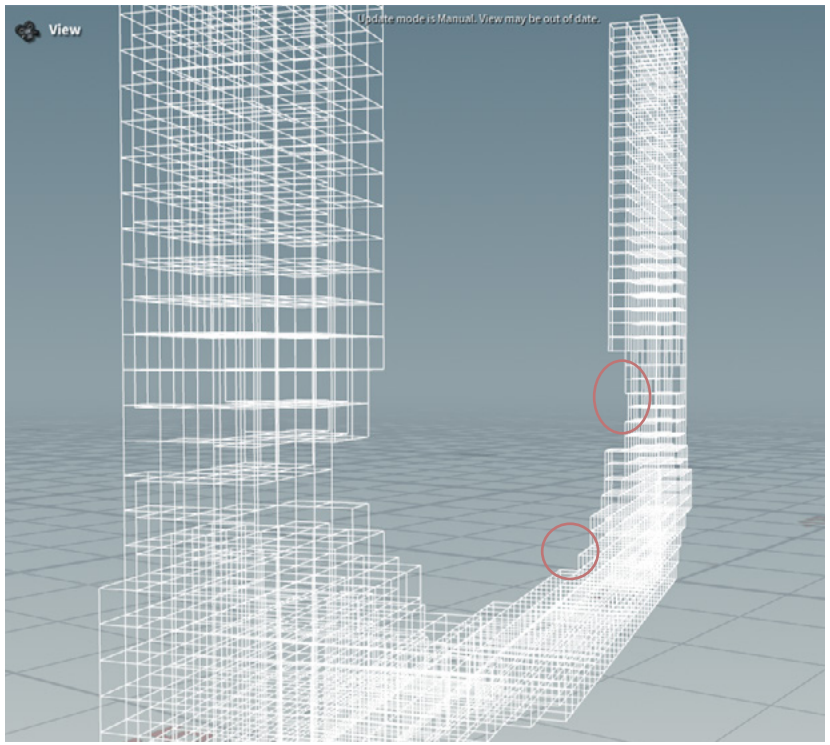
VEXpression

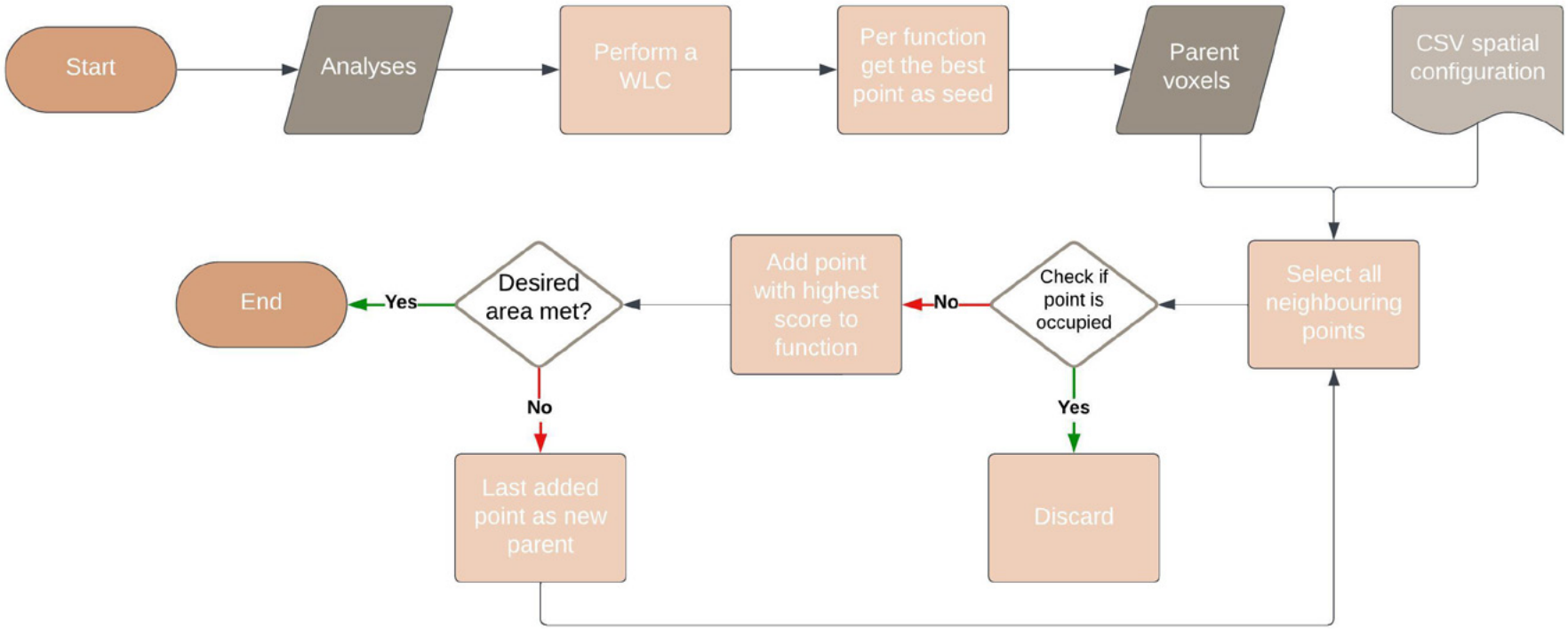
```
if (@dist > 12 && @P.y > 30.6) {
    removepoint(geoself(), @ptnum);
}
if (@dist > 15 && @P.y > 24.6) {
    removepoint(geoself(), @ptnum);
}
if (@dist > 22 && @P.y > 16.6) {
    removepoint(geoself(), @ptnum);
}
if (@dist > 30 && @P.y > 10.6) {
    removepoint(geoself(), @ptnum);
}
if (@dist > 40 && @P.y > 7.6) {
    removepoint(geoself(), @ptnum);
}
```

Closeness to ground analysis

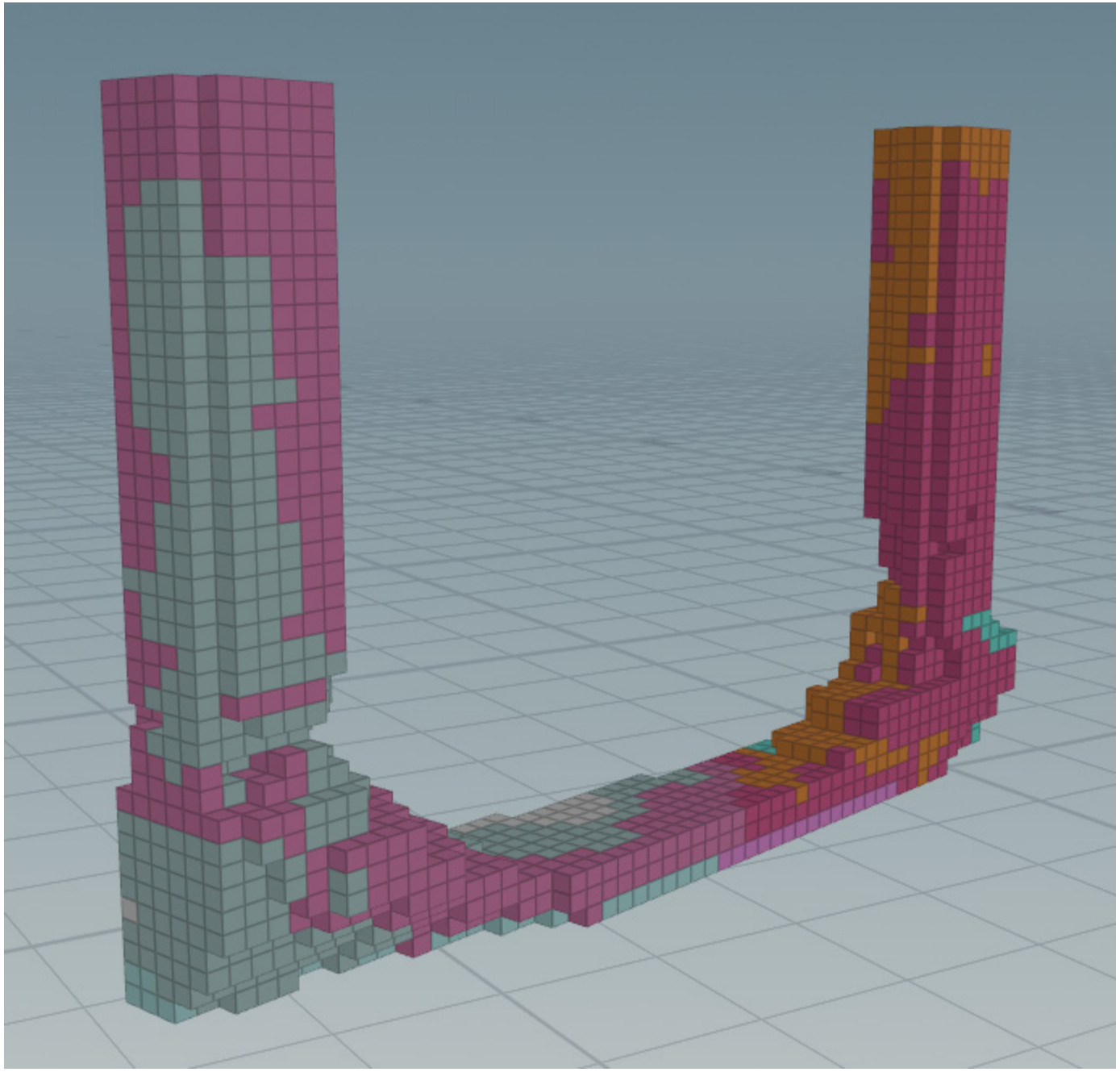
Function Growing






Start
Analyses
Perform a WLC
Per function get the best point as seed
Parent voxels
CSV spatial configuration
Select all neighbouring points
Check if point is occupied
No
Add point with highest score to function
Desired area met?
Yes
End
Yes
Discard
No
Last added point as new parent

Find Shape

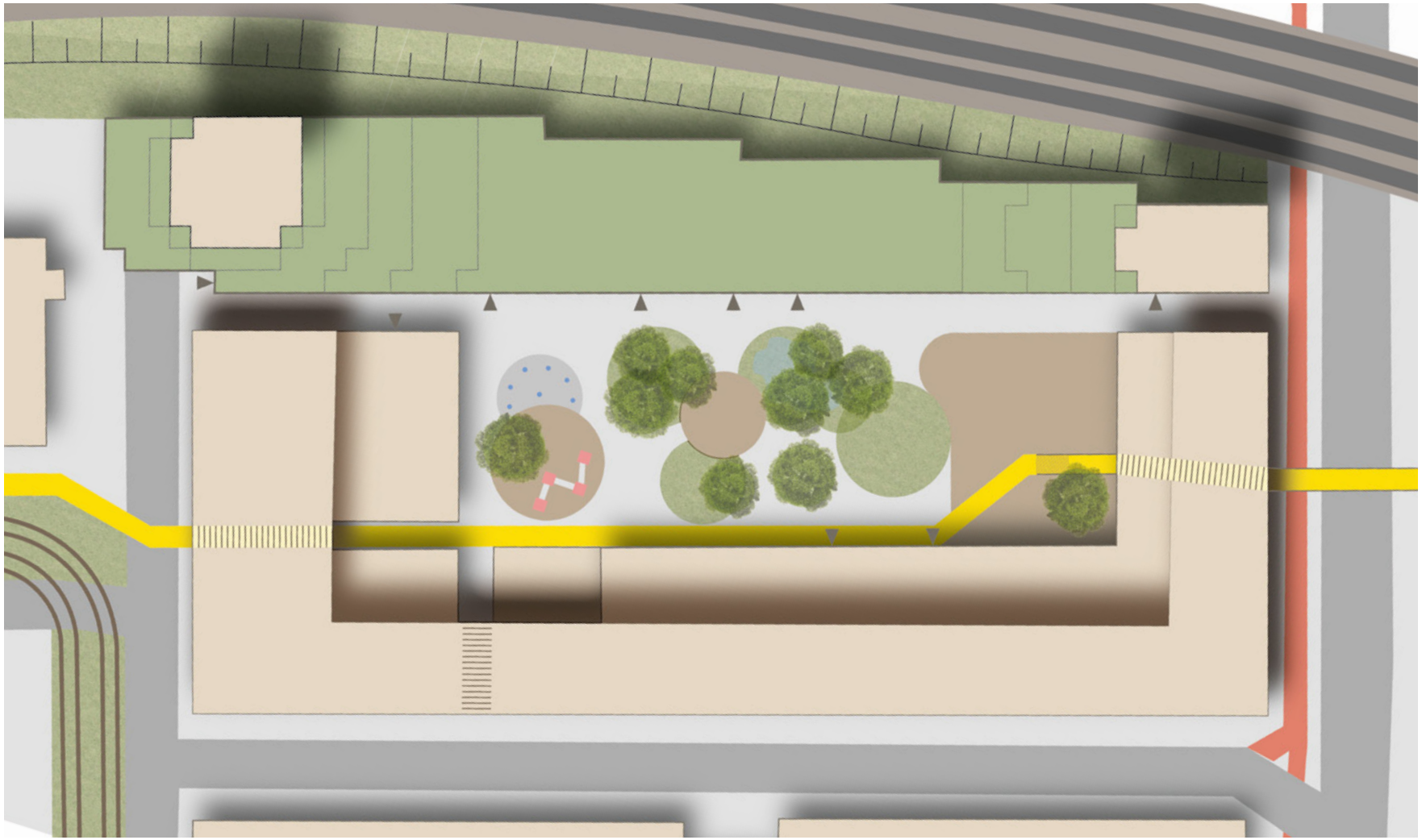


Procedurality

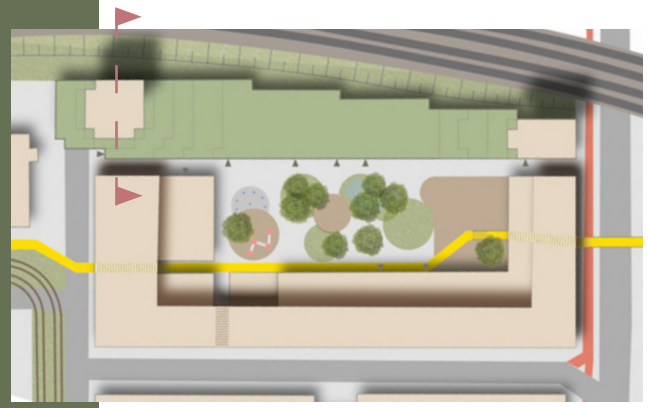
Variables:
Voxel width
Voxel height
Amount of floors
Sun hits minimum
Shadow contribution maximum
Floor area per function
Weights
“flatness”

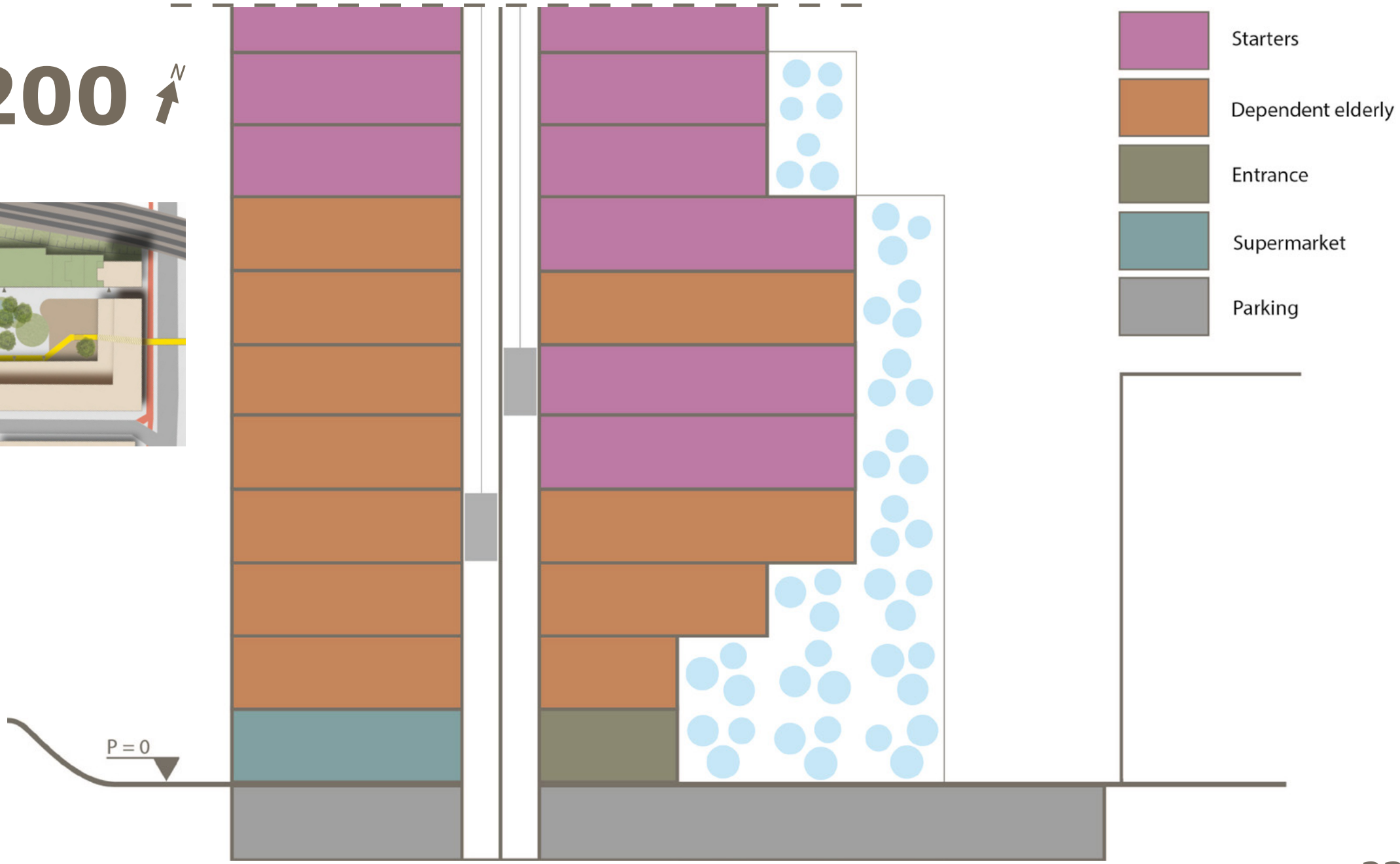
Minimum and maximum radius windows topfloor and first floor

1:500

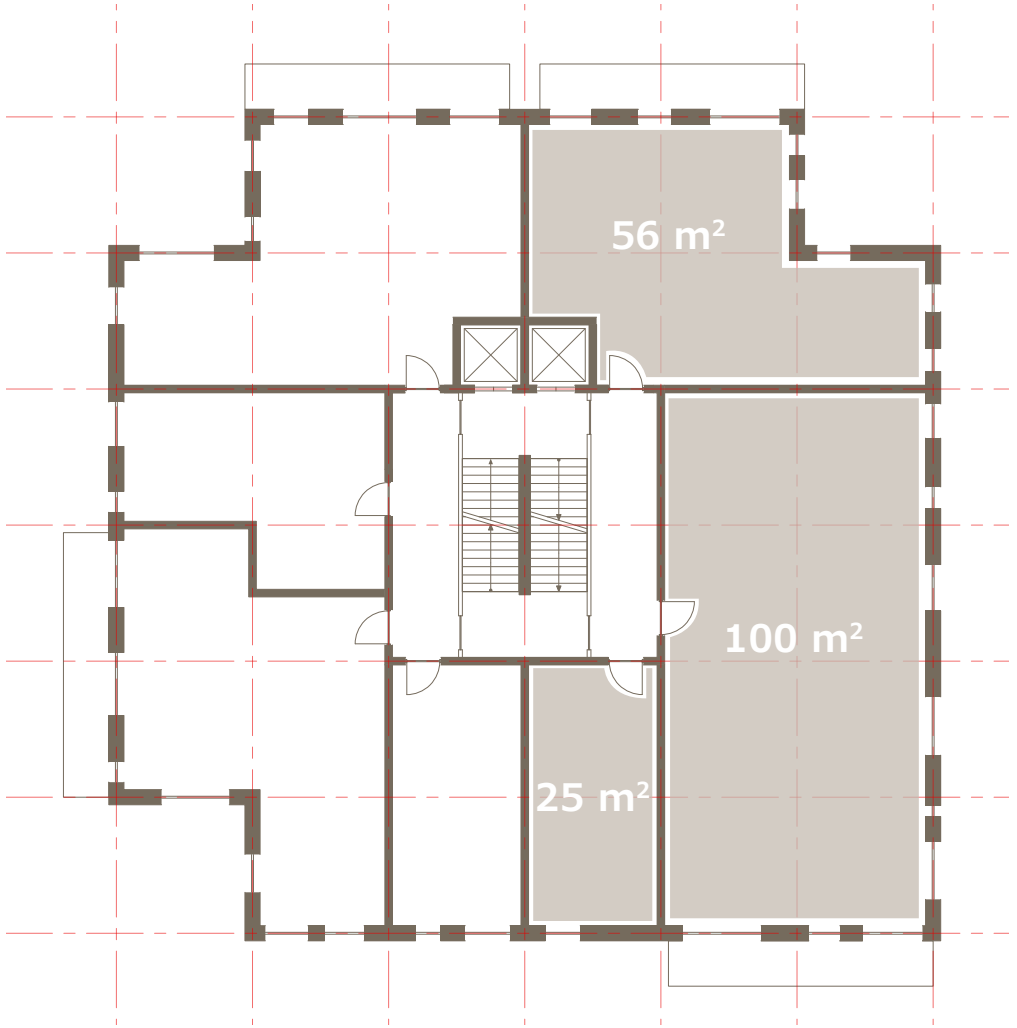


1:200

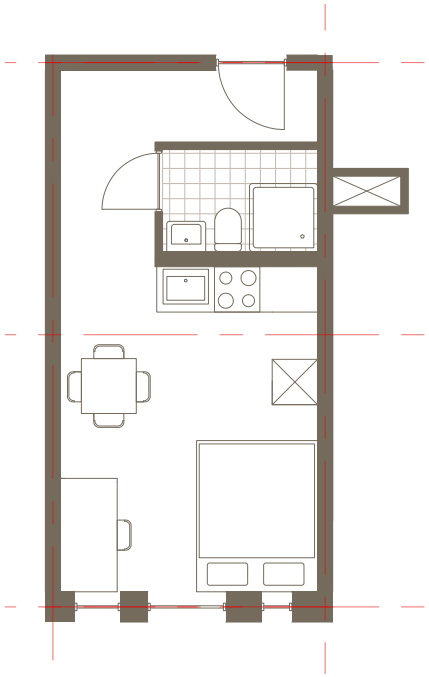



N
Starters
Dependent elderly
Entrance
Supermarket
Parking
P = 0

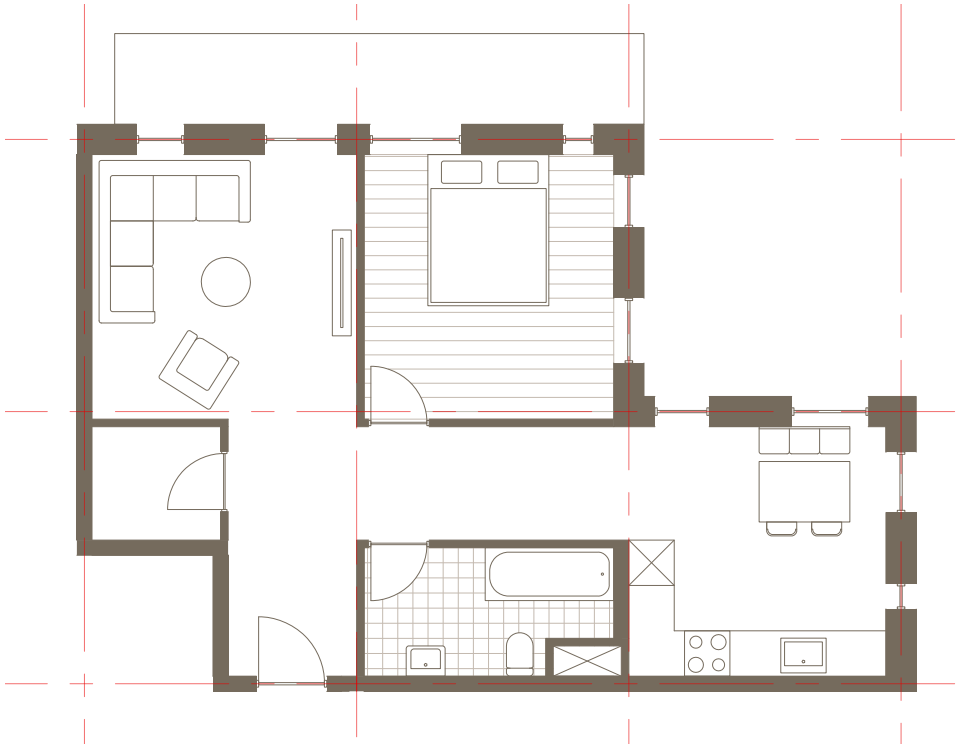
Tower 1:200



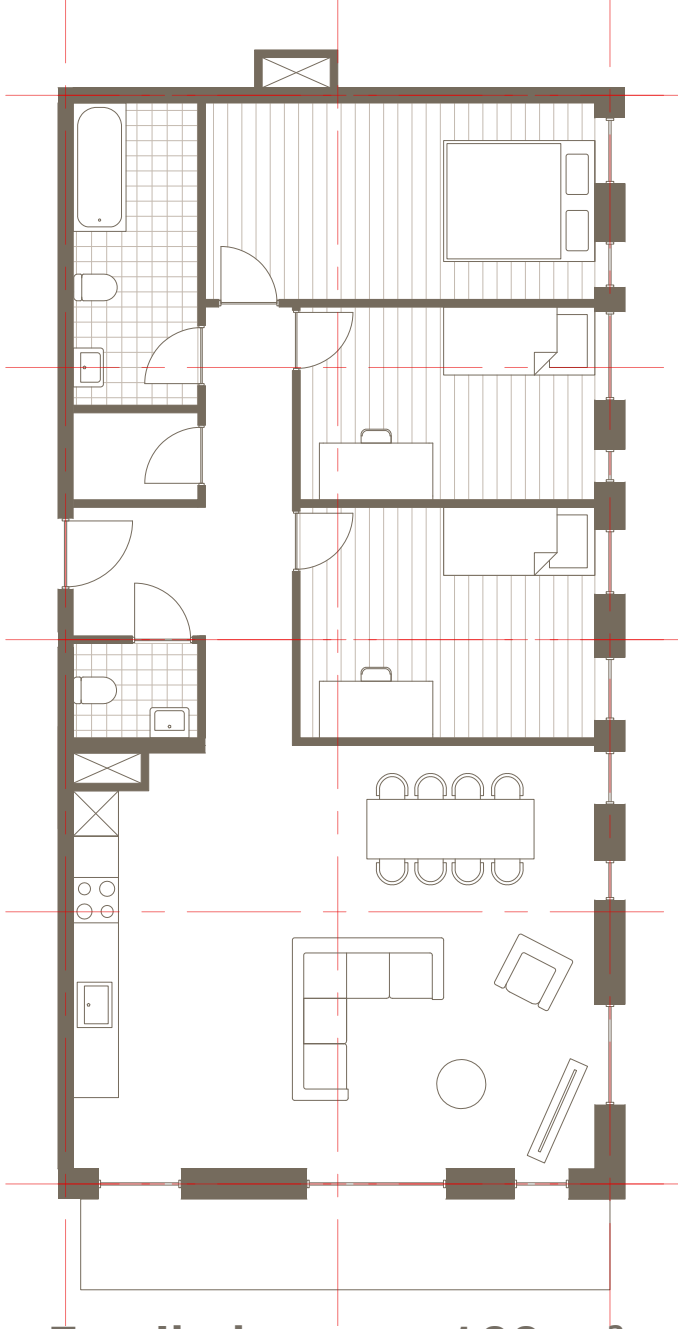
Module plans 1:100



Student home ~ 25 m²

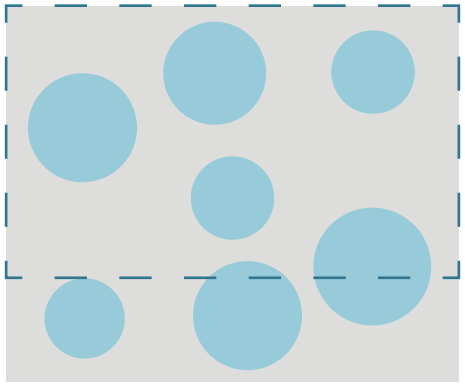


Elderly home ~ 56 m²



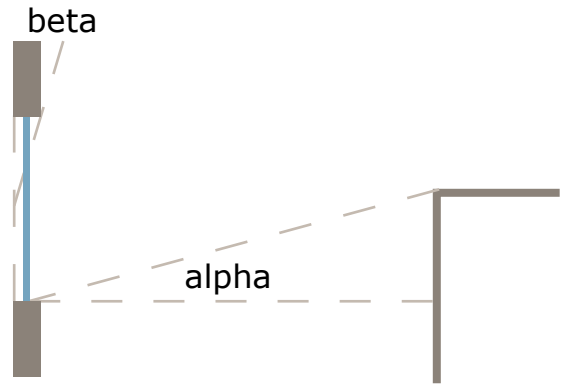
Family home ~ 100 m²

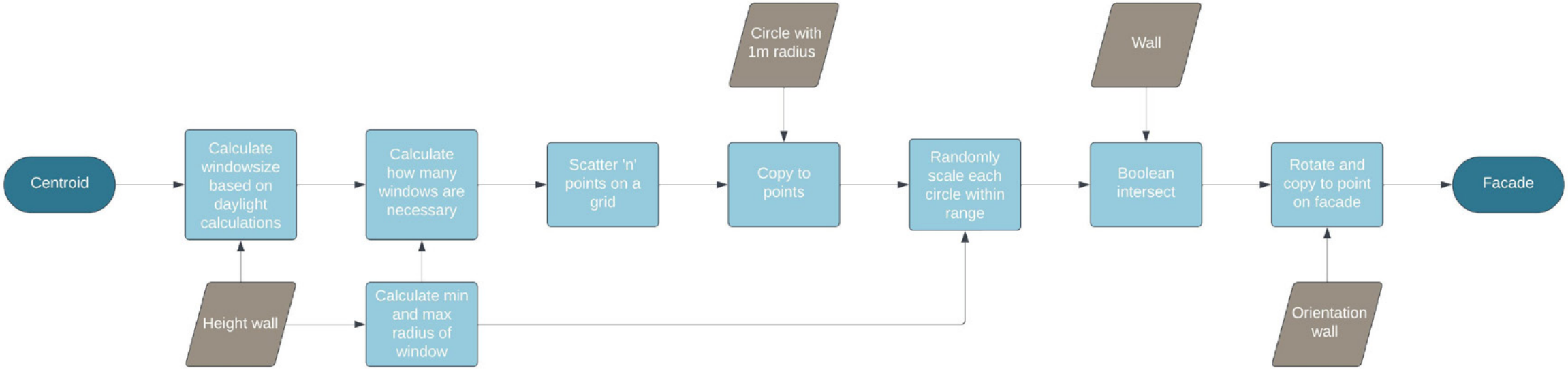
Concept



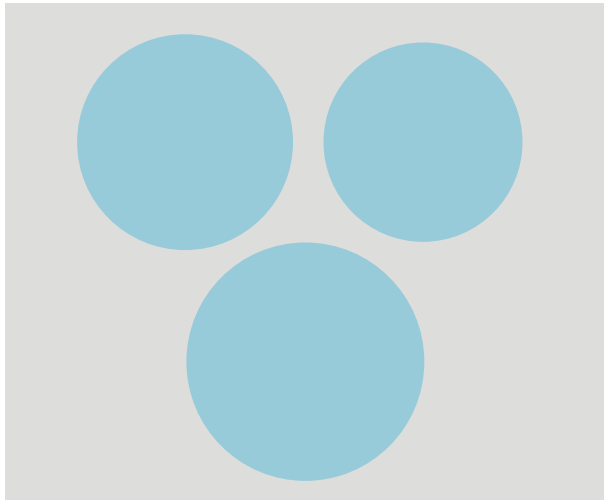
r min,top
r max,top
r min,firstfloor
r max,firstfloor
area = pi * r2
n_windows = ceil(Ae / area)

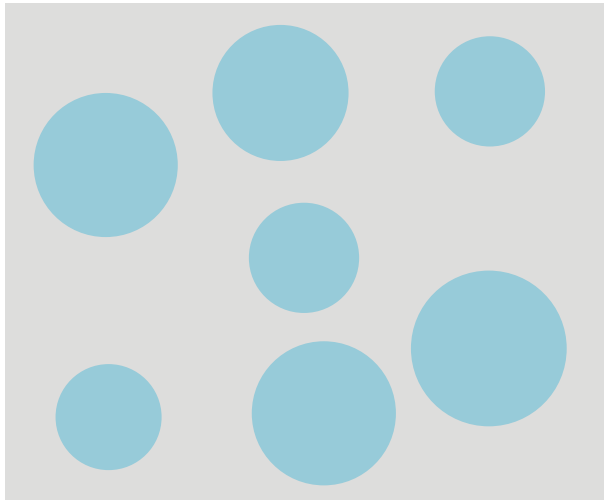
Ae = Ad * Cb
Ad = total_area * (1 - 0.6 / floorheight)
Cb = 0.7

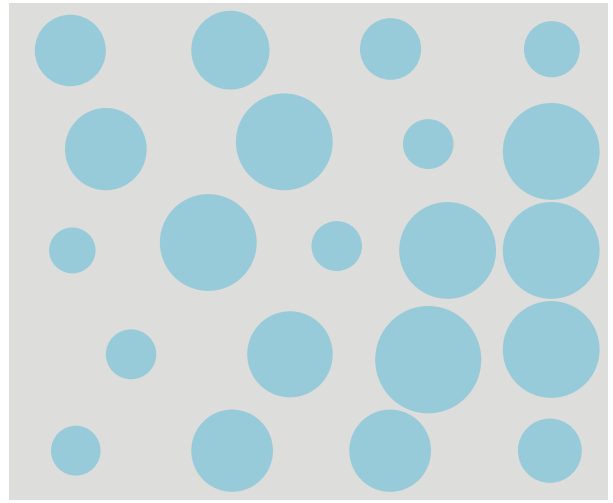


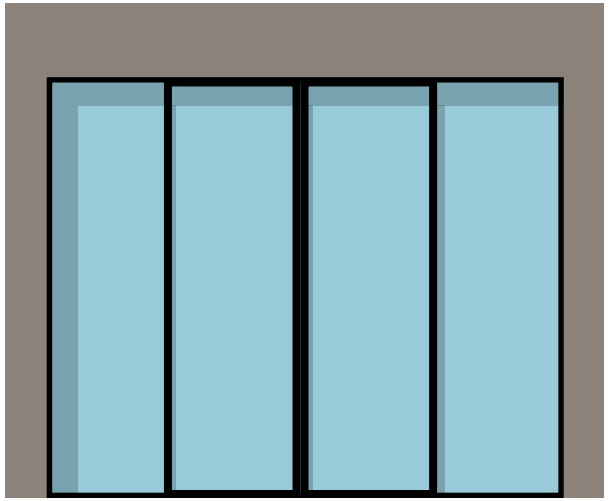


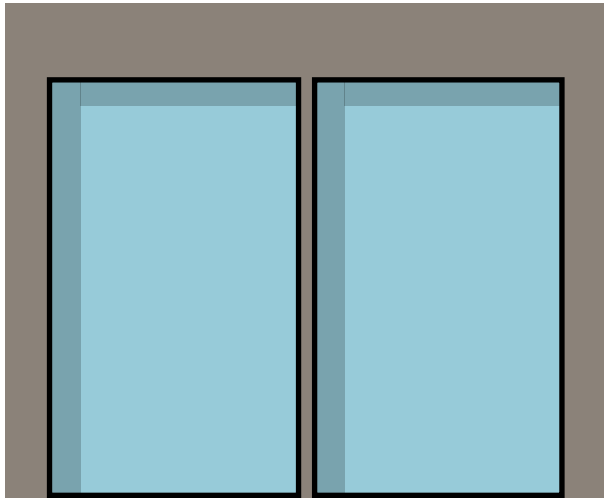
Tiles

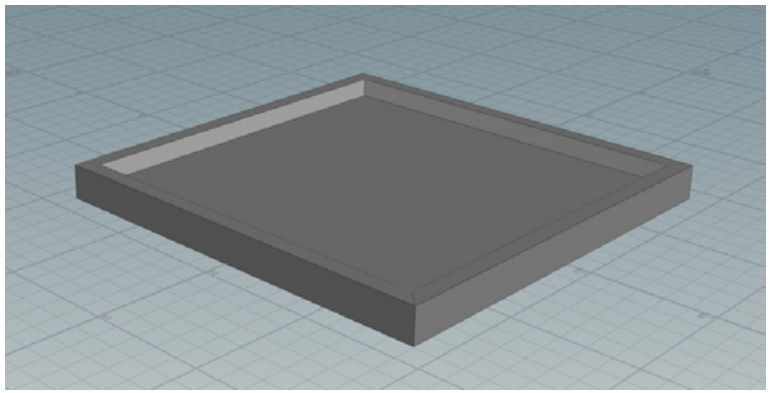












Final project

Thank You