Earthy 4.0

Anurag Sonar 5201756
Diederik Jilderda 4096649
Karin Backer 4552962
Pragya Chauhan 5319110

Mentors - Dr. Ir. Pirouz Nourian, Ir. Shervin Azadi, Ir. Hans Hoogenboom, Charalampos P. Andriotis, Ir. FR Schnater

Contents

1. Introduction

1.1. Design Problems and Potential

To get oriented on the challenges and possibilities in the design area, the demographics, infrastructure and urban conditions were analysed - initially in the entire class, and afterwards within each group. From these analyses (explained in detail in the next chapter) and the first impression of the situation, we identified community and healthcare as an important aspect that we would like to further study and develop. For the research, we used multiple sources as well as reports from the previous years' projects.

An analysis of the healthcare infrastructure in the refugee camp highlighted the following issues -

a. Disproportionate distribution of medical facilities in districts as per density of each district.
b. High pressure on major healthcare facilities as they are less in number
c. The walking distance to health care facilities is high in certain districts.

The first steps to form a design proposal stemmed from these points. They offered potential ideas as a starting point :

a. Offer a solution for both small and large scale of facilities.
b. Focus on a large scale, by designing a major healthcare facility.
c. Offer a solution that can be adapted to every locality

1.2. Design Vision

With these ideas, we formulated a design vision, which was to -

"Make primary healthcare more accessible to the people of Zaatari, by providing a flexible tool to design healthcare centers of different scales and functions based on specific local needs."

1.3. Design Proposal

The main concept of this project is to create an adaptable tool to locate and design a healthcare center, which will incorporate the scale, layout, functional and structural requirements along with constructability.

1.4. Urban Analysis

1.4.1. Site Anaysis

To analyse the urban context, we set up a number of queries regarding aspects we find important in selecting a location. Synthesis of these queries then results in a compounded map which puts all the information together. This provides the basis to select a location.

We analysed the urban context for population distribution, the build environment (in a Nolli map), the main infrastructure and access points to the refugee camp, and the current medical network. Synthesis of these maps indicates that suitable locations for new facilities are close to both the east and west access gates. This is logical since these areas are not covered by current hospitals yet, but are within the influence range of the access gates.

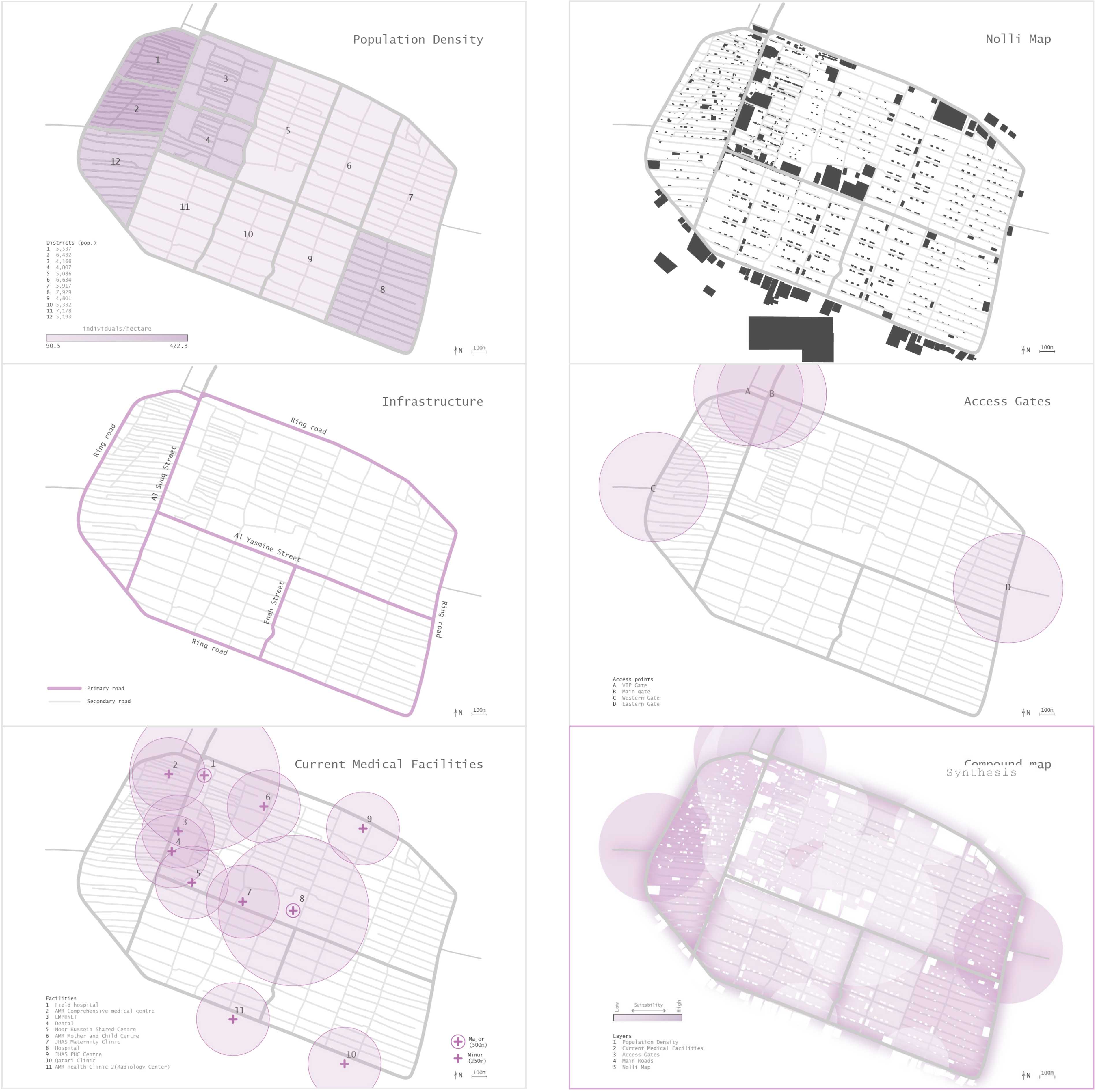


Figure 1. Urban analysis

1.4.2. Urban Medical Network

The larger vision of CARETHY+ is to provide a refugee camp the tools to improve the accessibility and quality of healthcare for refugees.

Healthcare facilities can be organised in several ways on an urban level. Centralized larger facilities can provide services in an efficient manner, including more specialized healthcare. Decentralized smaller facilities however improve the physical accessibility across an urban network by reducing travel distances in the camp.

We take advantage of these characteristics by proposing medical facilities on two levels.

Clinic:
Clinics are small facilities which are easy, fast and cheap to set up and provide basic healthcare.

Hospital:
Hospitals are larger facilities which provide both basic and specialized healthcare. Their size and centralized nature allows more efficient performance.

Figure 2 shows the improvement in medical accessibility of refugees when new facilities of both scales are added to the urban medical network.

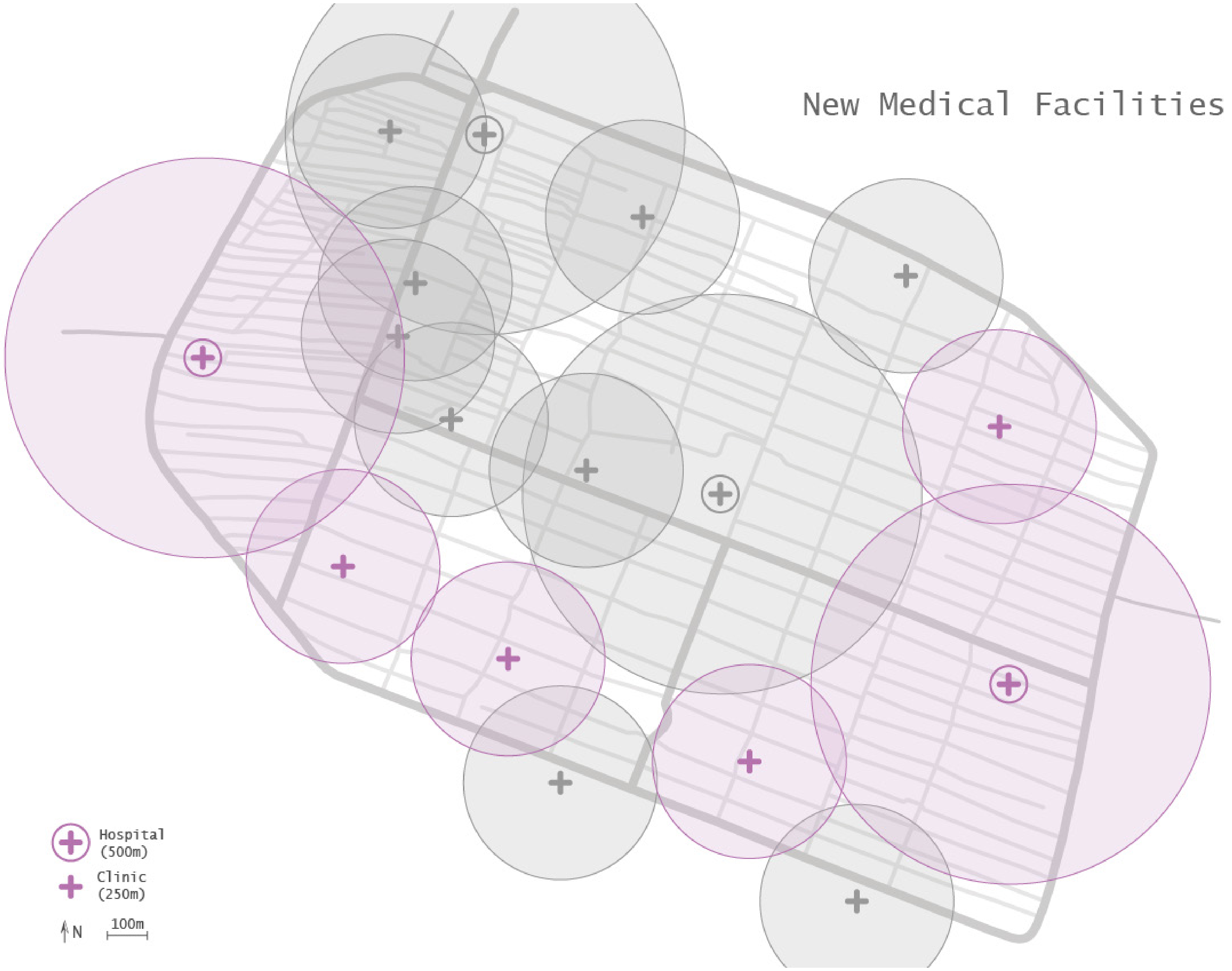


Figure 2. Map of Zataari

1.4.3. Clinics vs Hospitals

We just we defined medical facilities as two types: the clinic and the hospital. The next step is to define which functions they should provide and how the number of functions is to be determined.

Both clinics and hospitals provide basic care which comes in the form of Primary Healthcare Units. Additionally, hospitals may also provide specialized care: women's & children's care, oral healthcare, rehabilitation and emergency care. In case patients have to stay overnight, wards may also be necessary. This list is not exhaustive, and if other types of care are required (such as psychological care) then they can easily be adopted in the configuration process.

Since clinics are small-scale, a reception and a few auxiliary functions are all that are required to have a well-functioning facility. A full-fledged hospital however might also include additional functions such as a cafeteria and kitchen, a postal office, a pharmacy and even an educational center.

The size of the medical program depends on the number of people that a facility services (the population within its range of service). The rest of the program follows proportionally and can be adjusted to meet specific local needs.

We can derive the number of required healthcare units from international data. The Global Economy [3] has measurements of average number of specialists per 1000 population (which we will call 'level of service') by country, which we can use to both determine our level of service and compare this level to what can be considered the benchmark (western countries).

Of course it is unrealistic and unreasonable to demand a level of service comparable to western countries; resources such as money and staff are not infinite and not nearly as availabe in a refugee camp as in for example the Netherlands.

It is very difficult to pinpoint an appropriate level of service, so we set the value per 1000 population to a value we deemed reasonable. This value can of course be adjusted as desired by local decision-makers who have far more insight and experience in this regard than us.

An overview of functions, by urban scale level, and level of service of each respective function is provided in Figure 3. These describe the services a facility provides, however a more detailed program is required to make an actually functioning facility.

Functions	Clinic (small)	Hospital (large)	Service level: capacity per 1000 refugees
Reception	Yes	Yes	
Cafeteria		Yes	
Education		Yes	
Postal Office		Yes	
Pharmacy		Yes	
Auxiliary	Yes	Yes	
Primary Healthcare	Yes	Yes	0.3
Women's & Children's Healthcare		Yes	0.2
Oral Healthcare		Yes	0.1
Rehabilitation		Yes	0.2
Emergency Room		Yes	0.2
Ward		Yes	1.5

Figure 3. Functions by scale

1.5. Location Selection

Given our vision for an urban network of small and large facilities, together with the synthesis of spatial queries described earlier, we can determine a suitable location for the new facility. We choose a hospital scale facility to prove our configurational game (if it works on a large complex scale, it aso works on a smaller scale).

The darker the area in the map, the more overlap of criteria exists in this area (because we essentially perform a boolean operation). These locations are therefore the most desirable.

We selected the area near the eastern access gate as the location for our hospital. This area is well-accessible and has currently poor service (long distances to the nearest facility from this area of the camp).

Figure 4. Location Selection

1.5.1. Computational Tool

The first approach to urban analysis and location selection can be more refined. We aim for a more accurate approach using Grasshopper.

A criticism on the first approach is that there is little room for nuance, and that the importance of all aspects are weighed as equally important even though in reality certain aspects will be found more important than others. These weights are also not constant but depend on the specific situation.

The limitation regarding nuance can be illustrated with the example on the right. Proximity to the access gates is an aspect we deem important, however the returned boolean value of either 0 or 1 is too limiting. It makes sense to value closer proximity to the access gate more than low proximity, yet both return the same value. Similarly, stepping one foot outside the influence radius suddenly drops the returned value for the location to from a full one to zero.

We take a different approach to determine the desirability of a location. The concept is that the desirability of a location depends not only on what happens exactly at the location but also in its direct environment, in our case the range of the medical facility we want to place.

This means that in order to mathematically evaluate any location in the camp, integration of a density function f(x,y) over the service area A of a hospital has to be performed to arrive at a value that can be compared.

These density functions can be established for each criterion and do not have to be homogeneous in their distribution over the camp. A clear example is population density which is different for each district (and in reality even heterogeneous within each district, but we have no access to information of this level of detail). Evaluating a location in a high density area returns a higher value, which makes total sense.

To put this in a mathematical equation:

$$\sum_{i=1}^{n} \left(\int_A f_i(x,y)\, dA \right)$$

Essentially, for any location you integrate each density function over the area of service (the area covered by the hospital), and add them all up to put a value to the desirability of the location.

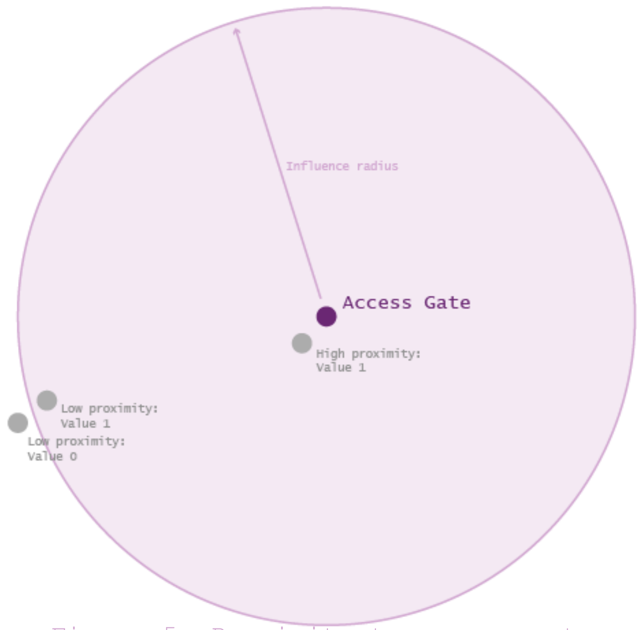


Figure 5. Proximity to access gate

1.5.2. Approximating density functions

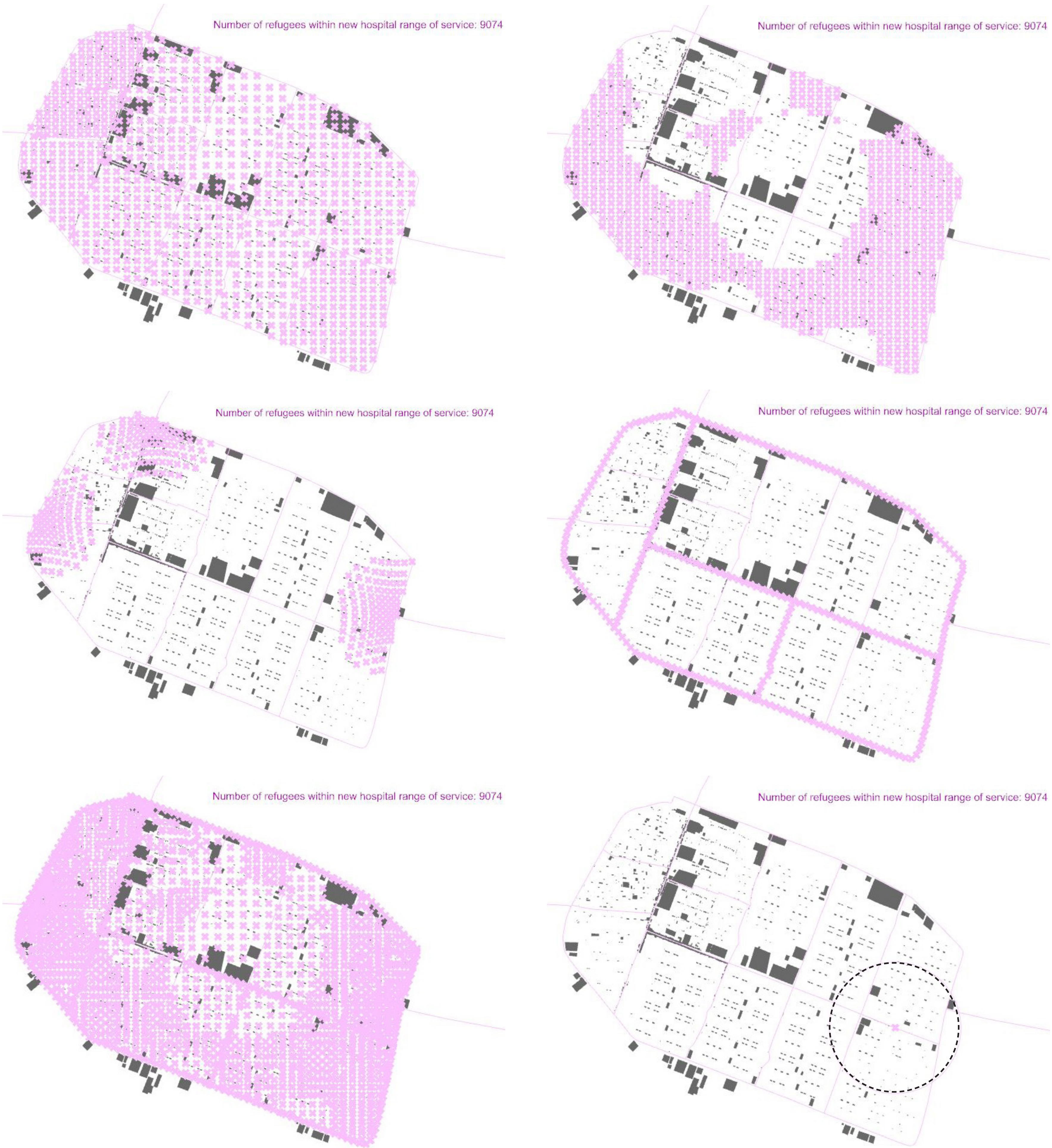
We approximate these density functions in Grasshopper by populating surfaces with points, the density and distribution of which can be independently tweaked. This allows both subtle transitions in influence zones and a control over relative important between the criteria.

The different density functions are shown here: population density and non-serviced area at the top, proximity to access gate and infrastructure in the middle and concluding with a synthesis of these layers (with the Nolli map subtracted from the point cloud) and location selection at the bottom.

The optimal location can be found with Galapagos, scanning the camp for the location where the most points fall within its range. Of course the point cloud is an approximation of a density function so the higher the resolution, the more accurate the result is (but also computationally heavy).

The script immediately calculates the number of refugees that live in the range of service of the hospital, allowing us to create a fluid workflow from urban analysis and location selection to setting up the program for the facility.

The conclusion is actually (but not completely unsurprisingly) the same as that of the manual approach. The difference is that decision-makers now have more control over the relative importance of criteria and that location selection is more scientific.

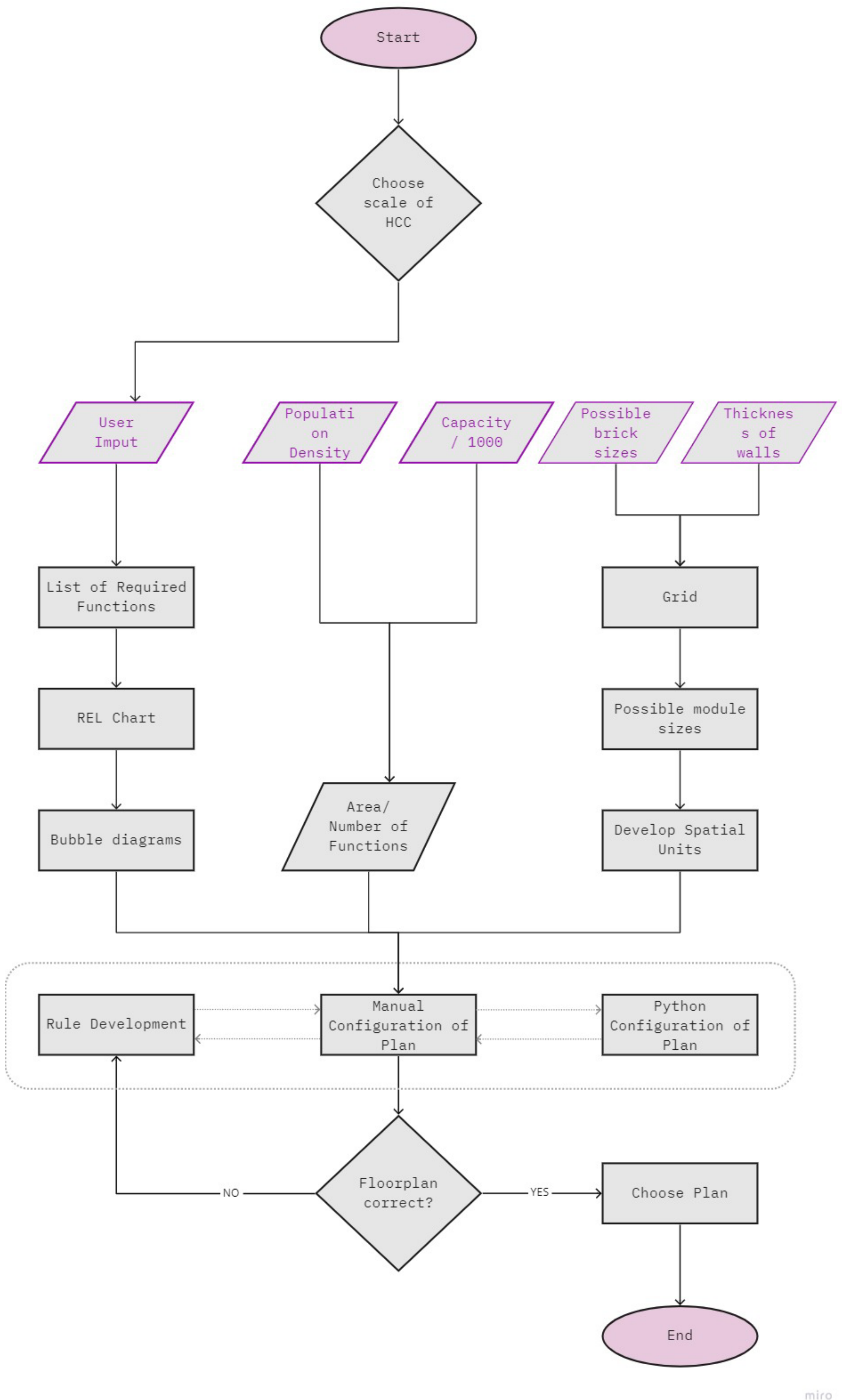


2. Configuration

2.1. Vision and Methodology

This sections covers the following aspects -

a. Developing the Program List
b. Deriving the areas and quantities of functions required.
c. Discussing the relationship between the functions.
d. Configuration of the game manually.
e. Configuration of the game computationally.



2.2. Process

2.2.1. Program Development

In Week 2, we listed functions that would be required in healthcare facilities. Since there were many functions which were difficult to segregate, we based our logic on three scales of facilities. The smallest would be the clinic, the medium size would be a community healthcare center and the largest facility would be a hospital. This is illustrated in the table below. However, in week 6, we fixed only two scales - the smallest and the largest scale to reduce the complexity of the project. After this conclusiom, we proceeded to the gamification.

	Functions	Clinic	Healthcare Center	Hospital
Basic Functions	Reception	Yes	Yes	Yes
	Cafeteria			Yes
	Education Hall		Yes	Yes
	Postal Office			Yes
	Pharmacy		Yes	Yes
	Auxilary	Yes	Yes	Yes
Specific Functions	Primary Healthcare Unit	Yes	Yes	Yes
	Women's and Children's Healthcare Unit		Yes	Yes
	Oral Healthcare Unit		Yes	Yes
	Rehabilitation		Yes	Yes
	Emergency Room		Yes	Yes
	Ward			Yes

Figure 6. Functions as per the different scales of facilities

2.2.2. Game Development

Why Gamification?

We imagine the project to allow a high degree of user participation in order to empower the displaced refugees and make the healthcare center highly customized to their needs. We do this by directly involving them in the design of their environment, based on the underlying premise that this could strongly improve the quality of the plan.

Figure 7. The team playing Carcassonne

Manual Development

The first inspiration for the gamification was 'Carcassonne'. After playing the actual game, we concluded that it was a good way to expand the program and could be played with a simple set of tiles and rules, as shown in Figure 8. However, while developing the functional program simultaneously, we realized that healthcare design has very strict requirements for distances and dimensions. To incorporate these within the game, the 'rooms' had to be rigid and programmed with fixed standard dimensions.

At first, the base dimension was 1.5m since it was the minimum required turning radius for the wheelchairs. When the project was modified to be a large-scale facility with a provision for wards, this dimension was changed to be 2.4m, which is the international standard for the corridor width in hospitals.

We then sketched all the possible modules with varying sizes, keeping the maximum number of modules in one direction as 3, which can be seen in Figure 9.

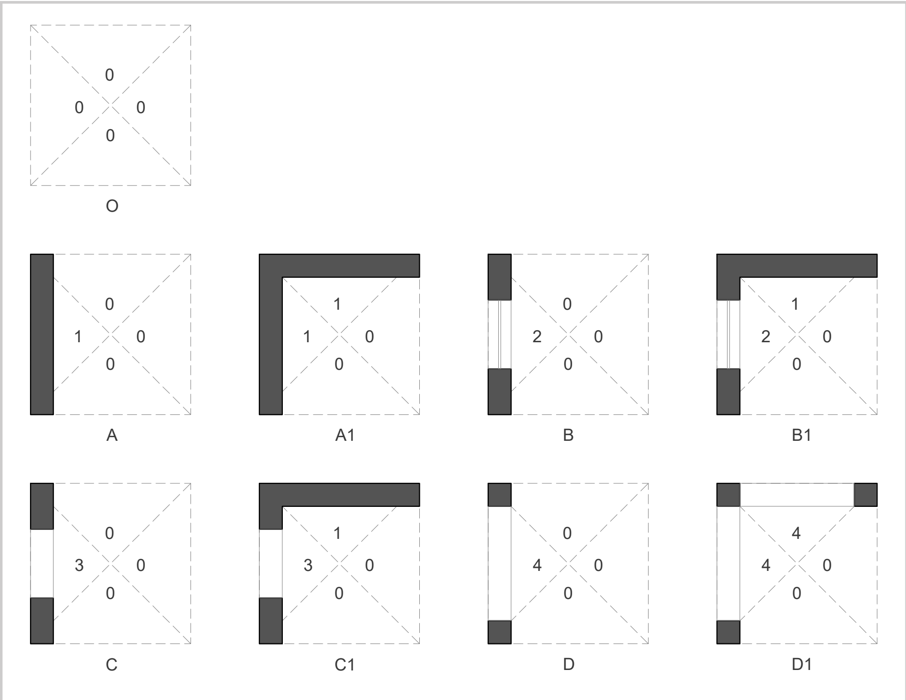


Figure 8. Intial Carcasonne Tiles

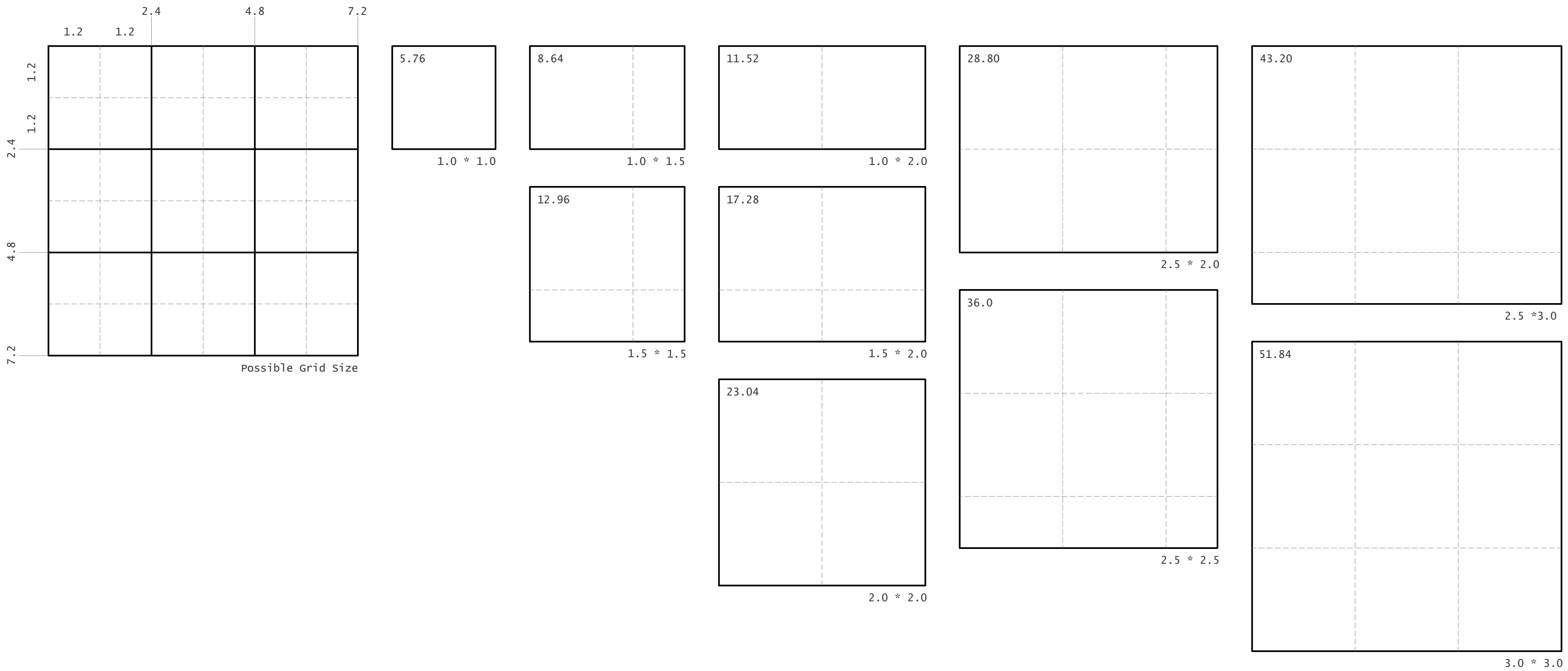


Figure 9. Module Sizes

By this time, we were using the Carcassonne logic only for assigning the walls, as shown in Figure 10. The numbers ‘0,1,2,3,4’ were assigned to the wall openings ‘fully open, no opening, window, door, open with columns only’. This idea was also rejected later since we discarded the concept of shared walls.

Moreover, this would be difficult to program with a second floor. In Week 7 we developed the staircase module and therefore also included the first floor in our configurations. Fucntionally, since we were not developing a ramp and accessibility was of great concern, only the ‘administrative zone’ was allowed to be placed on the first floor.

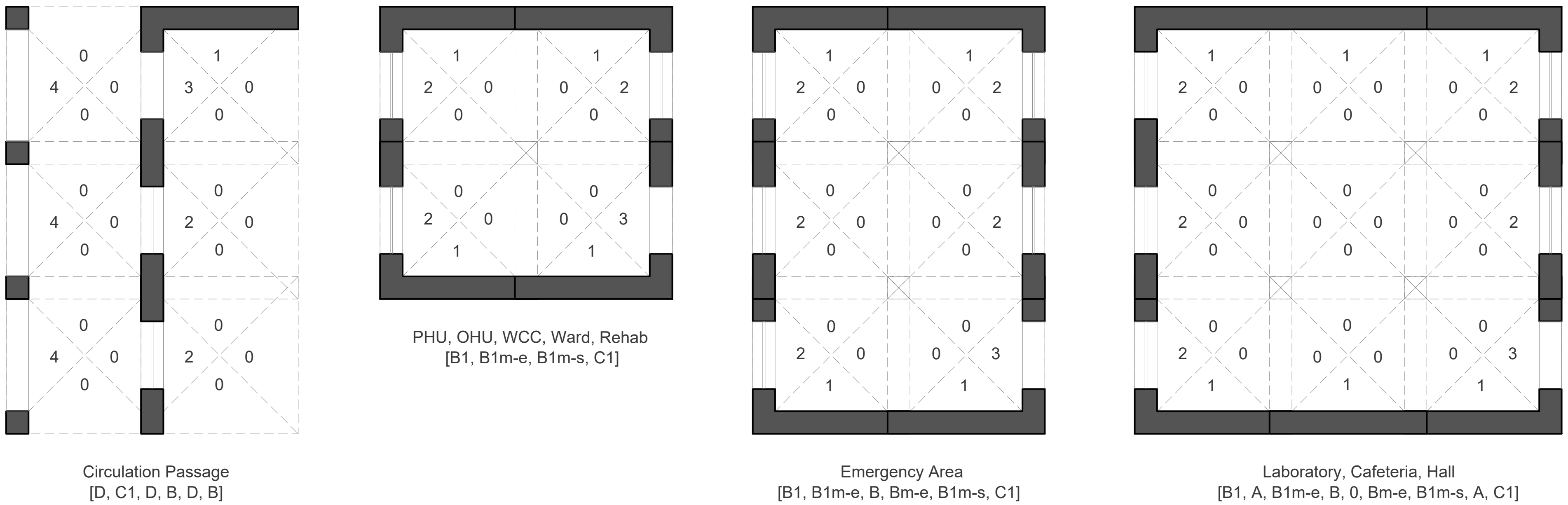


Figure 10. Spatial Units configured as Carsassonne

Once we had in mind some basic rules, we played the game several times manually to develop them more. Some key learnings from playing the game were the following -

Grouping of Functions

a. Promote staff efficiency by minimizing distance of necessary travel.
b. Group functions with similar requirements to have multi-purpose spaces.
c. The complexity of the functional program resulted in the segregation into three levels - Functions, Clusters and Zones.

Future Expansion

a. Plan should accommodate space for future expansion.
b. Leave open some grid blocks as a ‘pre-plan’ for future expansion.

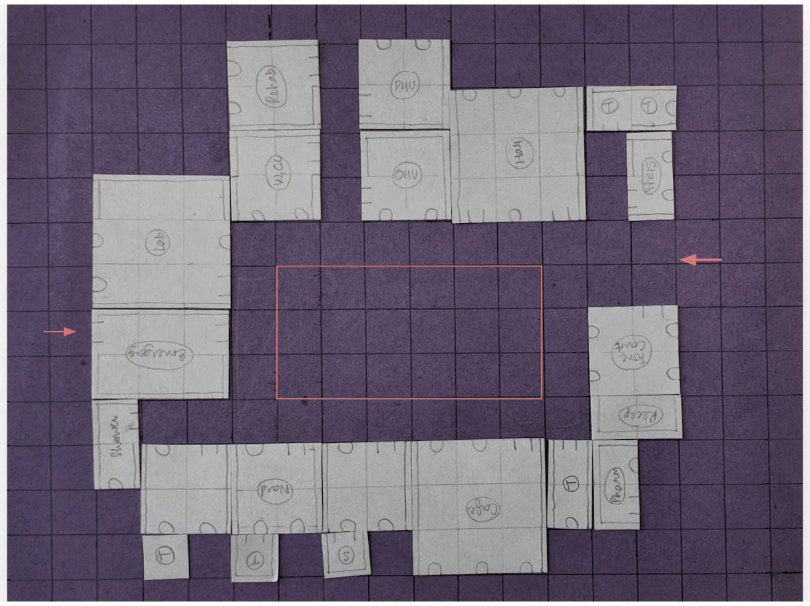
Accessibility

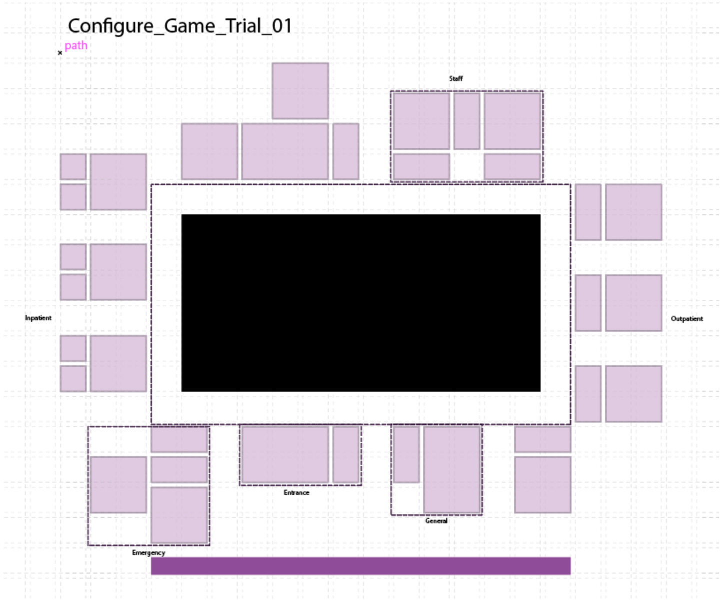
a. Since the building is usually closed at nights and on weekends, ensure that corridors are not left ‘open’ in the configuration.
b. The plan should not have too many level differences, for sake of accessibility on a wheelchair.
c. The plan should be designed for easy way-finding - the layout should not be confusing.

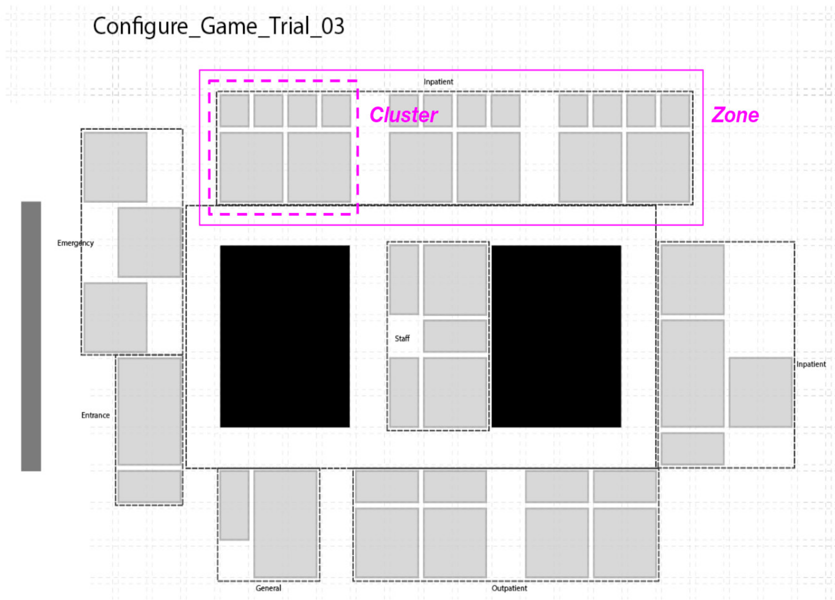
Environment

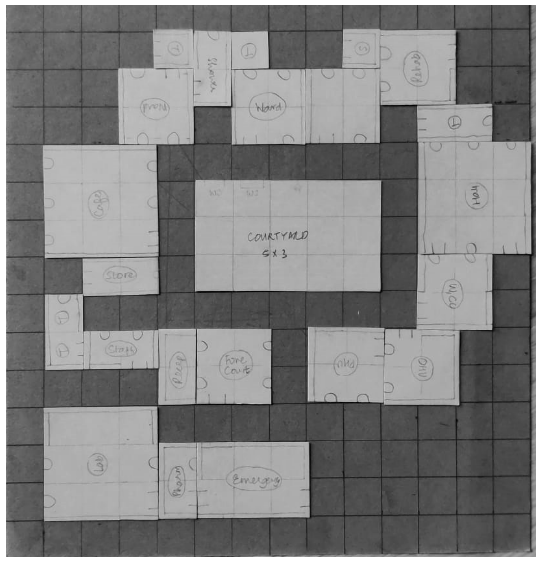
a. The building should promote a pleasant environment.
b. Spaces that allow natural light, ventilation and green elements are important for the configuration. This resulted in the ‘courtyard’ block.

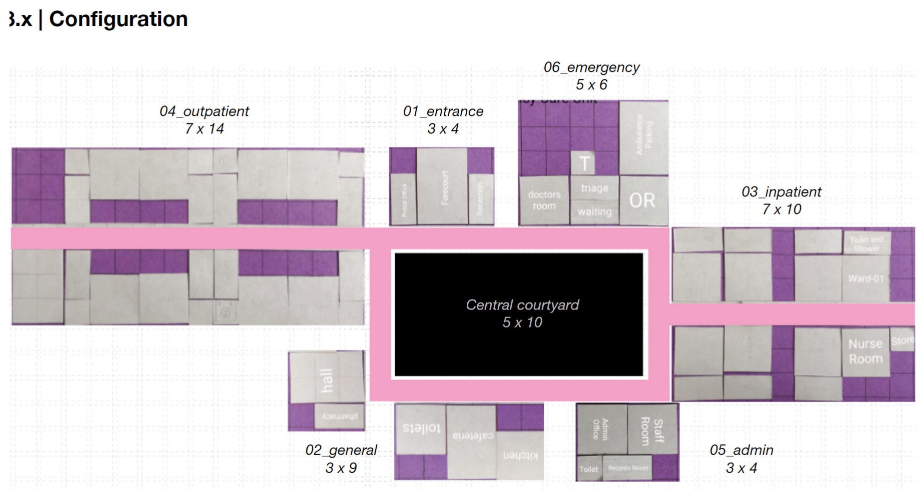
The various plans generated through the weeks are shown in thenext page.

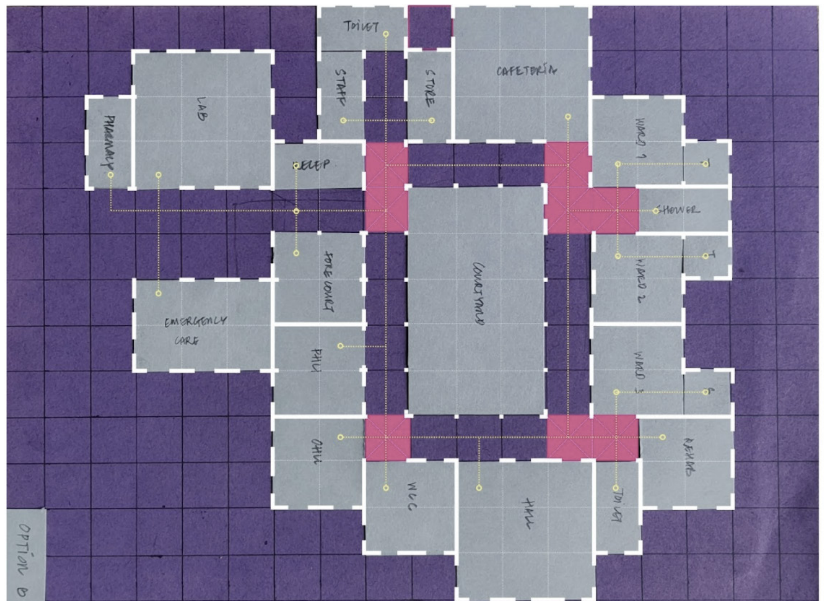


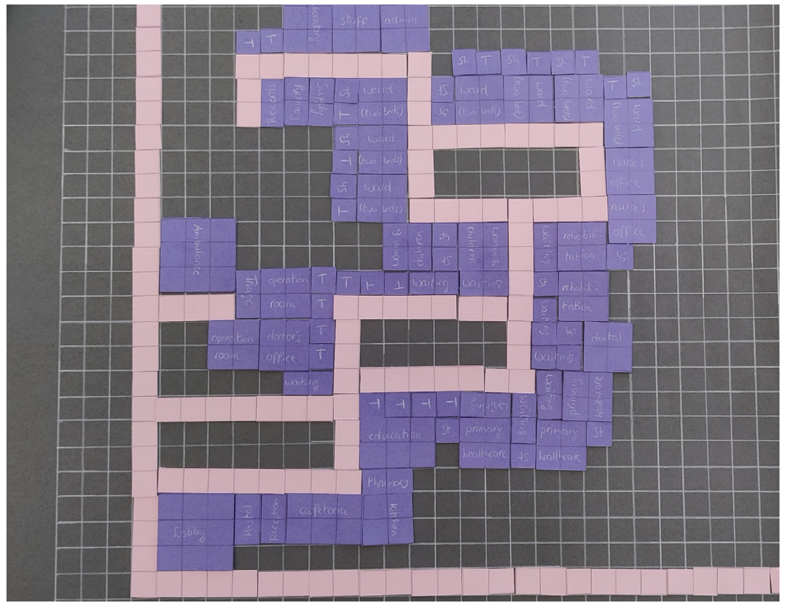












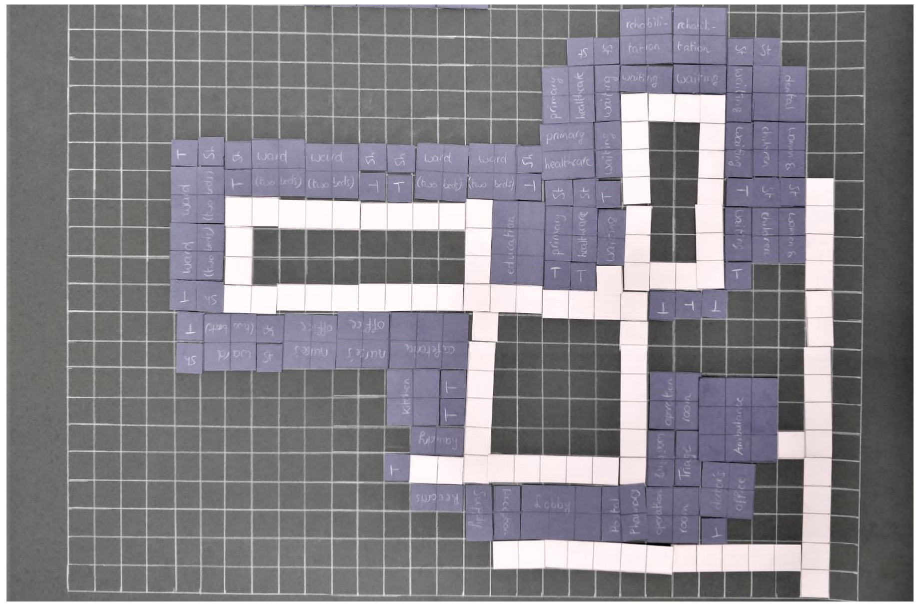


Figure 11. Developing Plan Configurations

Computational Development

We then began programming the game computationally. The game was initially configured based on the 'closeness of the zones' using the Kangaroo solver in Grasshopper. We realized that this did not provide much control over the placement of functions within the zones. Hence we discarded this method and instead worked on programming the game in python.

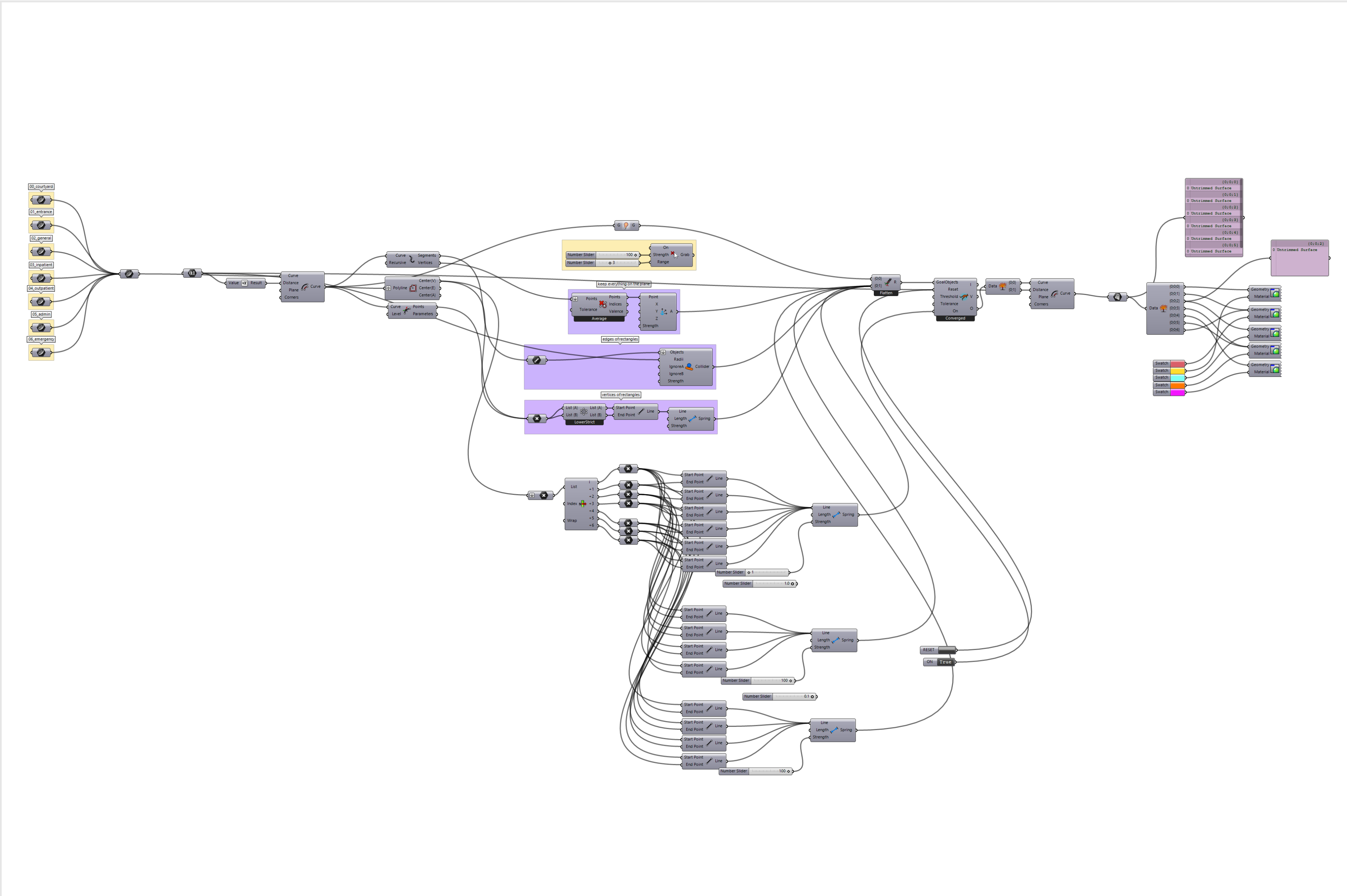


Figure 12. The Kangaroo solver script

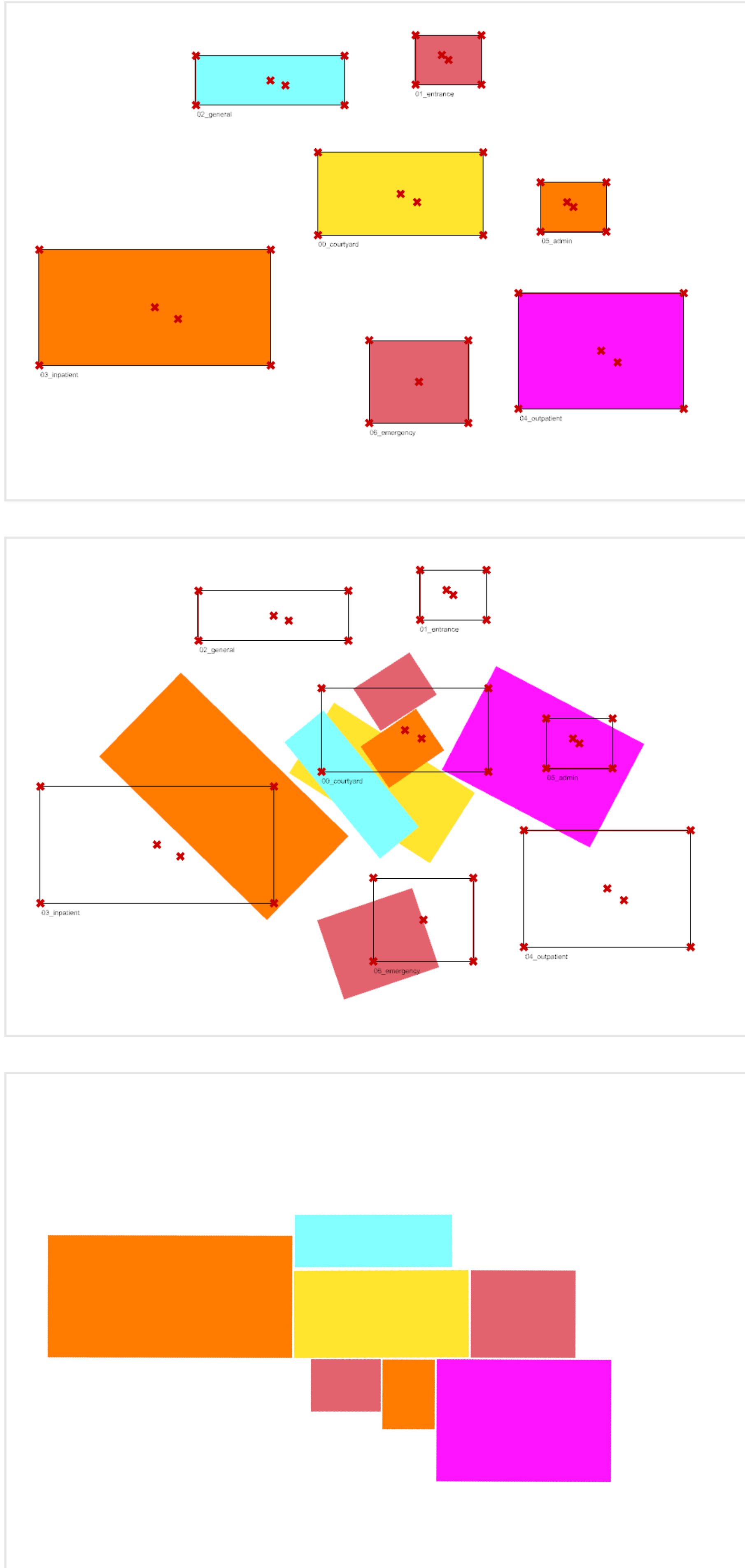


Figure 13. The results of the script

Game Rules

The game rules are informed by the python programming as well as the manual game. These rules are explained in detail in the next section. The following images show an important step in the process of gamification, which is defninng the 'move order'. It was important to prioritize the placement of functions while making a plan, in order to simplify the conplex plan.

MANUAL RULES OF THE GAME / PROGRAMMING OF THE COMPUTATIONAL GAME

Description of the Game

1. The player is given a combination of functions which are clubbed in their respective zones.
2. These functions are already laid out within tiles of 1x1, 1x2, 2x2, 3x2.
3. The aim is to achieve a functional health care center building of a required scale - which is achieved through configuring the functions by following all the rules of the game.
4. The configuration of functions is done by placing them zone-wise. This follows a priority list which is the following -
    a. *Context (predefined)*
    b. Entrance, Courtyard, General
    c. Emergency Care Unit
    d. Outpatient Department
    e. Inpatient Department
    f. Administrative Block
5. Within these zones, there is a specific 'move order' to place the functions in the order of their importance.
6. The game follows a system of learning, where the configuration can be evaluated after playing it once. This evaluation is based on the following aspects -
    a. Use less corridor space
    b. Improve lighting & ventilation
    c. Improve the hierarchy of sanitation areas
    d. Improve line of sight, smells

    Each time the player plays the game, the configuration can be improved with respect to these aspects.

Configuration of the Game

1. The game begins with the *'List of Functions'* which is developed as per requirements of the district. For this game, the list is predefined. To play for a different set of requirements, place the functions as per the *'Number required in our project case'* in the given list.

Move Order	**Rules**
0. Context	NA
1. Entrance, Courtyard, General A. Lobby Cluster a. Lobby b. Postal c. Reception B. Courtyard C. Cafeteria a. Kitchen D. Pharmacy, Education Hall E. Toilets	- Lobby is connected to the road and courtyard. - Toilets are connected to the courtyard. - Kitchen is connected to the cafeteria.
2. Emergency Care Unit A. Waiting Area B. Doctor's Office Cluster a. Doctor's Office b. Toilet C. Operation Room D. Triage E. Ambulance Parking F. Toilet G. Connect Ambulance Parking to the Main Road using corridor	- Waiting area is connected to the corridor. - Triage is connected to both the operating rooms and the ambulance parking. - The Doctor's office is connected to the waiting room. - The toilet is connected to the Doctor's office.
3. Outpatient Department A. PHU, OHU, Rehab, WCC Cluster a. Waiting Area b. Doctors Office c. Storage B. Toilets	- Place the functions attached to an extra courtyard. - Place PHU closer to the main corridor, place WCC further away. - Toilets are connected directly to the corridor space.
4. Inpatient Department A. Nurse's Office B. Ward Cluster a. Ward b. Shower, Toilet C. Storage	- Place the functions attached to an extra courtyard. - Storage spaces (common) are connected directly to the corridor space.

Figure 14. The Game Rules and Move Order

2.3. Final Product

2.3.1. Program

Grid Size

The grid size is the foundation from where space is allocated to the various program functions. If the base grid is too large, space will be wasted in rooms that are smaller than the grid, yet too small a grid is also undesirable.

In hospitals, having well-designed circulation is paramount. It is in fact so important that it forms the heart of floor plan design to guarantee good traffic flow and achieve a high degree of efficiency for both visitors (who are often not physically fit) and staff (nurses often have to walk several kilometers per day [1]).

Hospital circulation is used by various types of users. In order to guarantee enough space for the largest users (wheelchair users and wheeled stretchers), a minimum width of 2450mm is recommended [2] by health facility guidelines.

Based on this value, we take 2.4m as the free width of our grid. To include the width of the walls we consider a double tartan grid. This allows us to view every cell as a structurally independent unit, which gives a lot of freedom in the configuration process. Initially a wall thickness of 300mm was chosen, but after the addition of a second floor we had to conclude that a thickness of 450mm is structurally safer.

The resulting double tartan grid is shown in Figure 15. Figure 16 shows different aggregations of cells by which volume can be assigned to the various program functions.

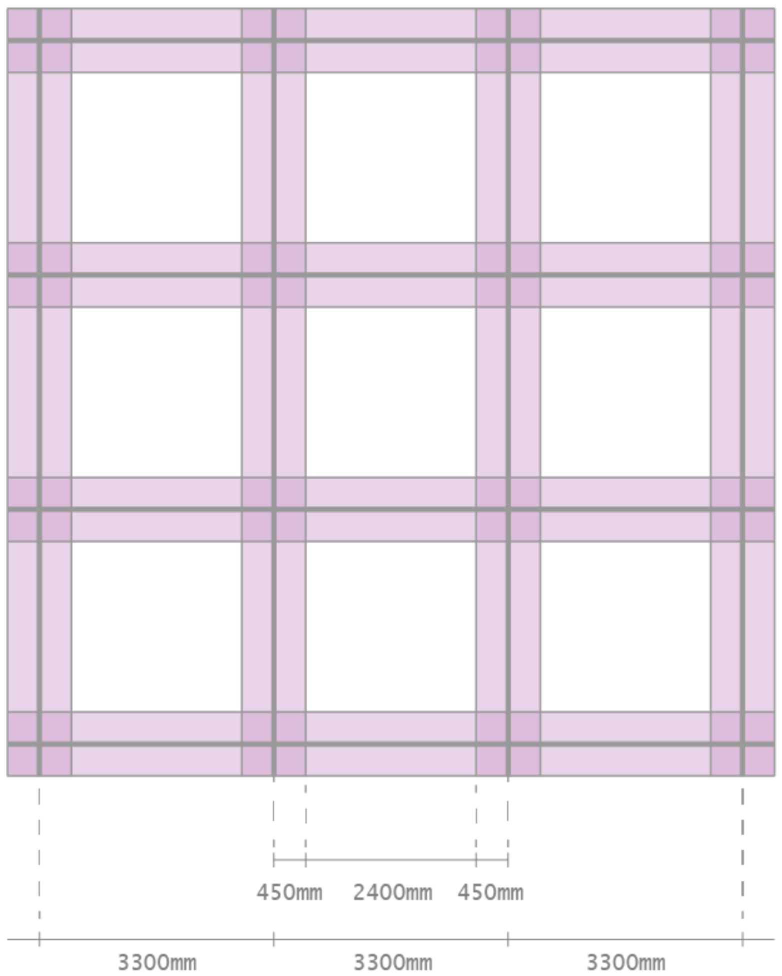


Figure 15. Double tartan grid

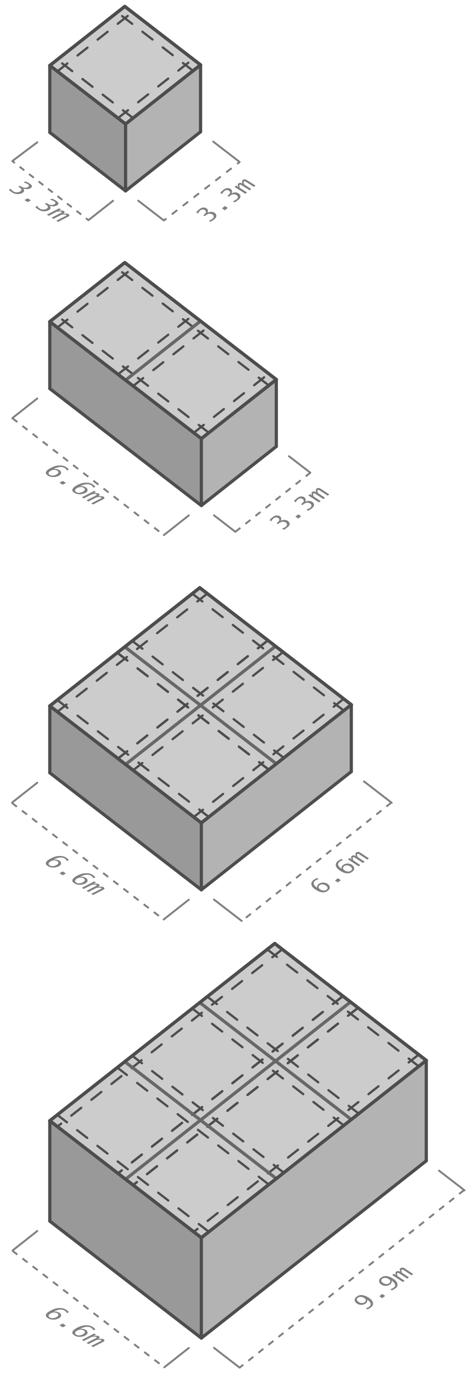


Figure 16. Volumes of aggregations

Program of Requirements

To provide structure to the overall configuration, the hospital program is divided into five distinct zones. Each zone clusters rooms together which perform a common function in the hospital and which have a similar profile regarding accessibility, privacy, etc.

General
The general zone forms the core of the hospital, providing an entrance lobby and a courtyard to which the other zones can be connected. It also contains general public functions such as a cafeteria, pharmacy, postal office and education room.

Emergency
The core of the emergency area is the triage. The triage area is where patients are assessed and sorted to determine the level of urgency and the type of care they require. Operation rooms are connected directly to the triage so that urgent cases can be treated near-immediately, while non-urgent cases and family members can make use of the waiting area.

Outpatient
All day-to-day care happens in the outpatient zone, where patients can visit their medical specialists. Every doctor's office has their own waiting area and storage unit, which together form a cluster.

Inpatient
Patients who have to stay overnight are taken in the inpatient zone which contains all the wards and nurse's offices. A ward has two beds and has access to a private toilet and shower.

Administration
The administration zone contains all supporting staffing functions that are better managed centrally, such as administration, records and laundry.

Figure 2.x shows the full program of the hospital. Every room type is assigned a size in units of grid cells, and a quantity based on the population this hospital services. Some quantities are derived from other functions. For example, we assume that for every five wards a nurse's office is required, so this particular inpatient zone requires two nurse's offices.

Other functions can be assigned more freely, such as the number of cafeteria or education rooms. Here the decision-maker has freedom to tailor the hospital to the specific local and urban needs.

Zone	Function	Size	Quantity
General	Lobby	2x3	1
	Reception	1x2	1
	Postal Office	1x2	1
	Pharmacy	1x2	1
	Cafeteria	2x3	1
	Kitchen	2x2	1
	Education Room	2x3	1
	Toilet	1x1	4
Emergency	Waiting Room	1x2	1
	Triage	1x2	1
	Operation Room	2x2	2
	Doctor's Office	2x2	1
	Ambulance Parking	3x3	1
	Toilet	1x1	1
Outpatient	Waiting Room	1x2	8
	Primary Healthcare	2x2	3
	Women's & Children's Healthcare	2x2	2
	Oral Healthcare	2x2	1
	Rehabilitation	2x2	2
	Storage	1x1	8
	Toilet	1x1	6
Inpatient	Nurse's Office	2x2	2
	Ward (two beds)	2x2	7
	Toilet	1x1	7
	Shower	1x1	7
	Storage	1x1	2
Administration	Staff Room	2x2	1
	Administration Office	2x2	1
	Laboratory	2x3	1
	Records Room	1x2	1
	Laundry Room	1x2	1
	Supply Room	1x2	1
	Toilet	1x1	2

Figure 17. Full Program

2.3.2. Organisation of the Program

REL Chart

Now that the full program is determined, it is important to explore the relationships and connections of all functions. We use a REL chart to define the relationship between functions. Since circulation forms an integral part of the hospital configuration, it is adopted as a function of its own.

If a significant relationship exists between two functions, this relationship can be characterized as two types: adjacent and close. An adjacent relationship means that the two functions have to be directly connected in order to function properly. An example are the toilet and shower of a ward, or the waiting room of a consultation room. A close relationship however does not have this rigid criteria: rooms can be close but not adjacent, for example on opposite sides of a corridor or courtyard.

It comes as no surprise that many adjacent relationships exists with circulation as one end of the relationship. Other functions that have a dominant adjacency criterium are waiting rooms and storage units to their respective healthcare units, and shower and toilet modules to wards.

The functions within each zone have (at minimum) a closeness relationship to each other, because each zone needs a certain level of cohesion. There are only two interzonal closeness relationships: nurse's office to kitchen, and nurse's office to laundry room. Meals for the wards are prepared in the kitchen, so minimizing this distance is desirable. The same applies for bringing laundry to and from the inpatient zone to the laundry room.

Adjacency relationships are very binary and therefore relatively easy to enforce. These will form the basis for the rules of the CARETHY+ game, by demanding very specific connections which will result in a logical floor plan.

Closeness however is far more difficult desribe and evaluate (after all, when is a room 'close'?). Enforcing strict closeness in a floor plan (for example, room A has to be within x cells distance of room B), can lead to strange situations where the only solution is actually an overall suboptimal solution and can severely compromise the freedom of the user.

Restricting the solution space beforehand is dangerous, so we find that the closeness relationships lend themselves better for post-game evaluation than to be enforced rules. For every closeness relationship that exists, the distance in the plan can be calculated and the summation of all relationships gives an overall score which can be evaluated and compared to other plans.

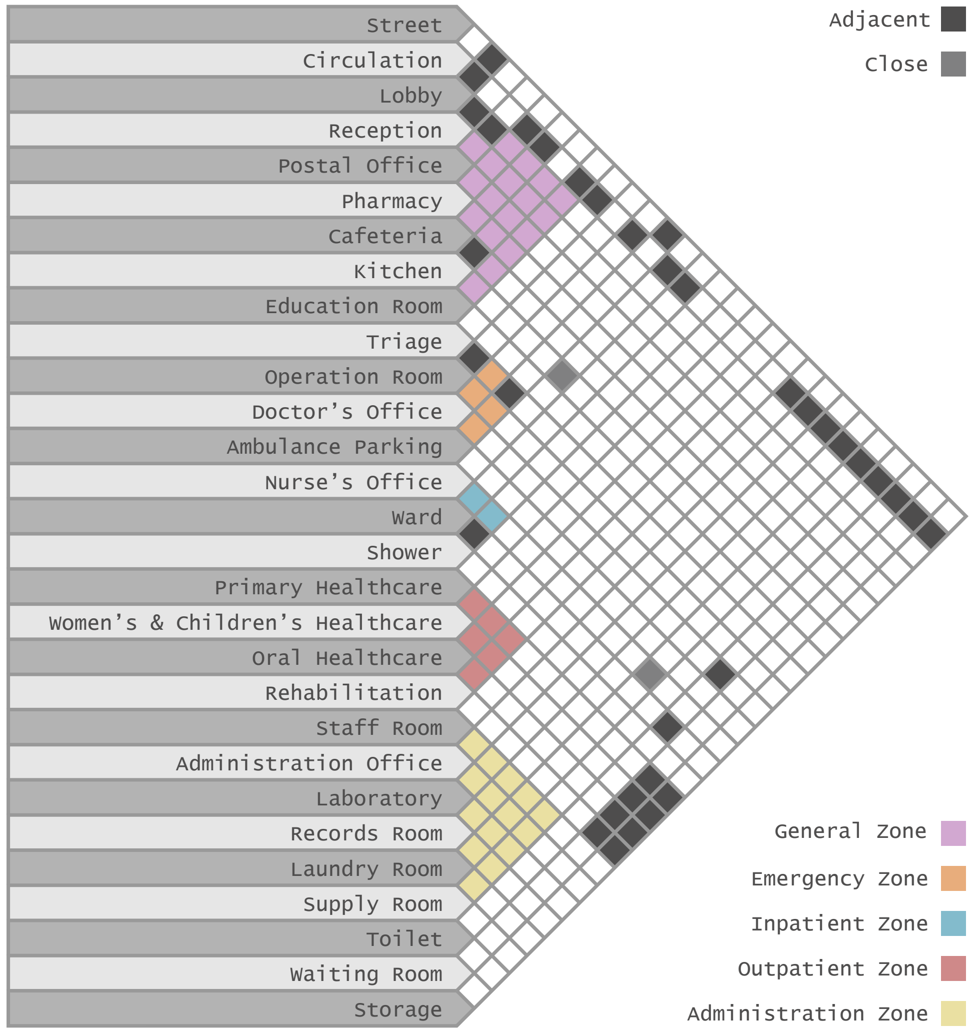


Figure 18. REL chart

Bubble Diagram

Bubble diagrams give insight in the spatial organisation of objects and are are derived from the REL chart. Since the hospital is organised on two levels (on a zonal and room level), a representation of both levels can be made.

Zonal bubble diagram
On a zonal level, we can see that the general zone directly connects to the street. This general zone then forms the circulation core of the hospital, forming the distribution point for the other zones. The emergency zone also requires connection to the street to ensure ambulance access.

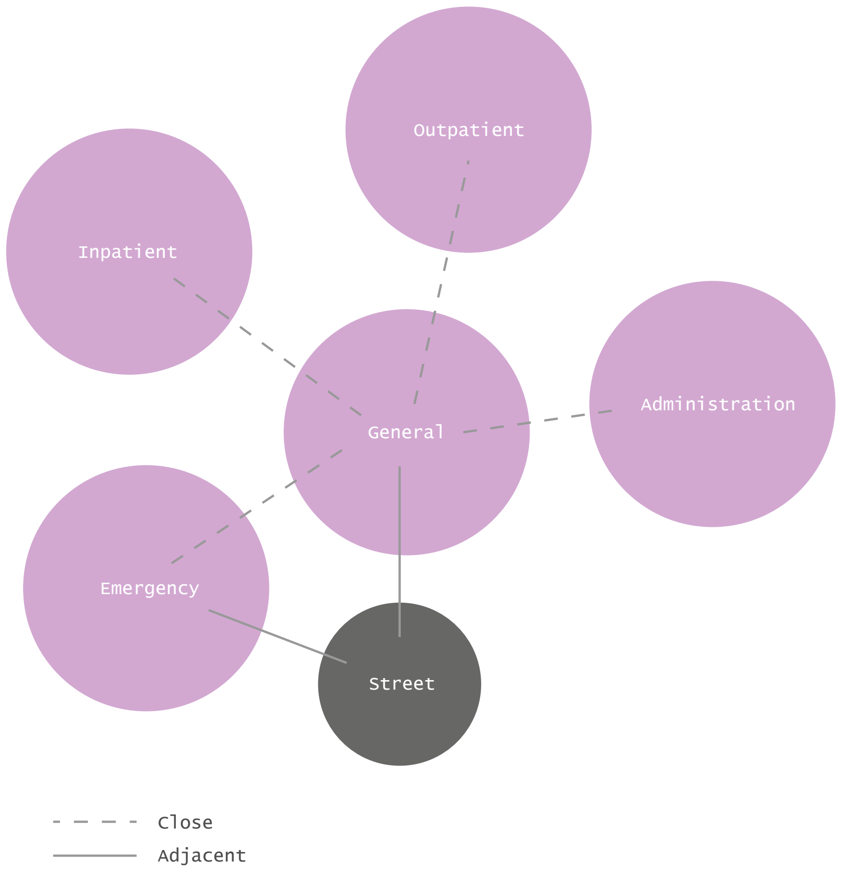


Figure 19. Bubble diagram of zones

Function bubble diagram
A detailed bubble diagram of the whole program illustrates the spatial organisation of the program particurly well. This bubble diagram is directly derived from the adjacencies in the REL chart.

Although closeness relationships are not directly adopted in the bubble diagram, we came up with a solution that vastly improves the cohesion of each zone without restricting the solution space.

As mentioned earlier, circulation forms the heart and veins of a hospital layout. We assigned to each zone their own circulation system to which the respective zonal program configures, and these circulation systems can be interconnected to create a cohesive network on the hospital level.

Overall the bubble diagram gives a good representation of how the hospital should be organised. It is possible to identify the different zones, the circulation network is clearly defined, and the way specific rooms are connected make sense. This bubble diagram can be used as the blueprint to define the ruleset for the CARETHY+ game, but first the organisation of the program has to be evaluated one more time.

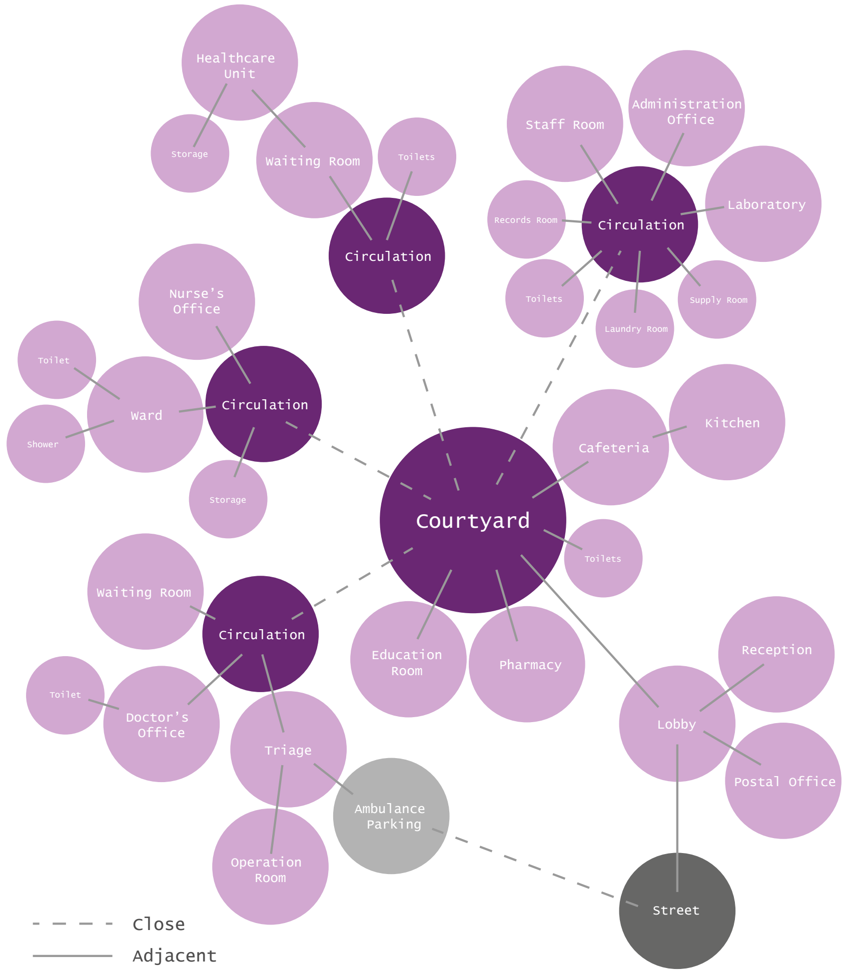


Figure 20. Bubble diagram of functions

+ +

Depth Chart

The REL chart and bubble diagrams give insight in the spatial organisation of the plan, but the spaces through which a user moves through the configuration can best be shown in a depth chart.

The depth chart shows for all rooms their "depth", that is the number of spaces one needs to traverse from the street until the specific room is reached. For example, to reach a ward one has to travel through the lobby, through the main courtyard into the inpatient circulation (three spaces, so level 4). It also shows the path someone has to take within the hospital, from one room to another. Important to note is that the chart does not contain information about physical distance: one space can be very large but still counts as one level to traverse (like circulation).

For some functions, it is important that only a limited number of spaces have to be traversed so that the room is well accessible (emergency functions come to mind). On the other hand, spaces can also be used to separate private and public spaces, like how a waiting area functions as a buffer between the public circulation and private consultation room.

In hospitals, cleanliness is an important consideration. The chart can also provide insight in this regard by showing how rooms with high demands for hygiene are separated from 'dirty' spaces to maintain a high level of hygiene. Hygiene-demanding spaces are operation rooms, consultation rooms and the kitchen, which are all separated from the 'dirty' public circulation or functions like toilets.

Overall the plan looks well-distributed. The access to each zone is well-accessible, the emergency zone is in fact directly accessible from the street. Functions within each zone are kept from the very public main courtyard through their own circulation systems (which can be a simple corridor or a courtyard), making these spaces still public but 'less' public.

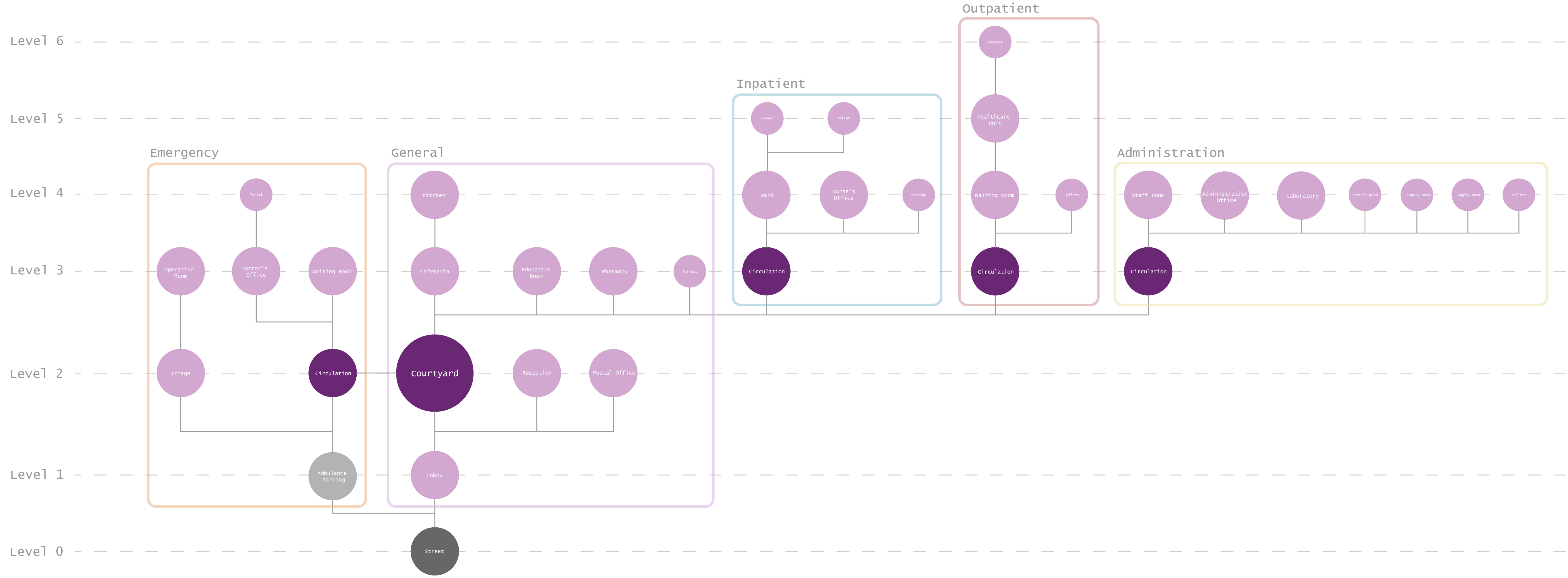


Figure 21. Depth chart of the organisation

2.3.3. The CARETHY+ Game

The Game Rules

Now that the program is well-defined and well-organised, it is time to set up the rules for the game. We define three simple rules with which the player can come up with a logical and structurally safe configuration. These rules apply to both the manual and the digital game, but in the digital game the rules are already baked in to improve the user experience (and because the player does not have to worry about the rules anymore, they can focus more on making the best layout possible).

Rule 1: Complete the build order
The configuration is built zone by zone, and room by room. To provide structure to the player, the build order of rooms is already established. In between building rooms the players is of course allowed to add circulation, courtyards and staircases as they deem necessary.

Rule 2: One rule to rule them all
This is the most important rule and tells the player how the rooms are supposed to be connected to arrive at a logical configuration.

Every room is assigned both a room ID and a connect ID. When the player wants to connect a new room to the existing layout, the room is only allowed to connect when their connect ID matches the ID of the room they want to connect to. This system gives us, the designers, a lot of control over what the player is allowed to do, yet gives the player one simple rule to follow instead of a long list of very specific rules.

The assignment of connect ID’s is directly derived from the adjacency relationships in the REL-chart and bubble diagram. If illogical sequences or connections occur in the configuration, then this can be traced back and changed accordingly in the bubble diagram.

Rule 3: To stack, or not to stack
The player is even allowed to expand the floor plan to a second floor. The second floor can be reached once a staircase module is built. Of course it is only possible to build on top of already placed ground floor rooms, with exception of the 3x2 spatial units. Since these units have a column-free space, building on top of these is disallowed to prevent floating columns, which would otherwise lead to guaranteed structural collapse.

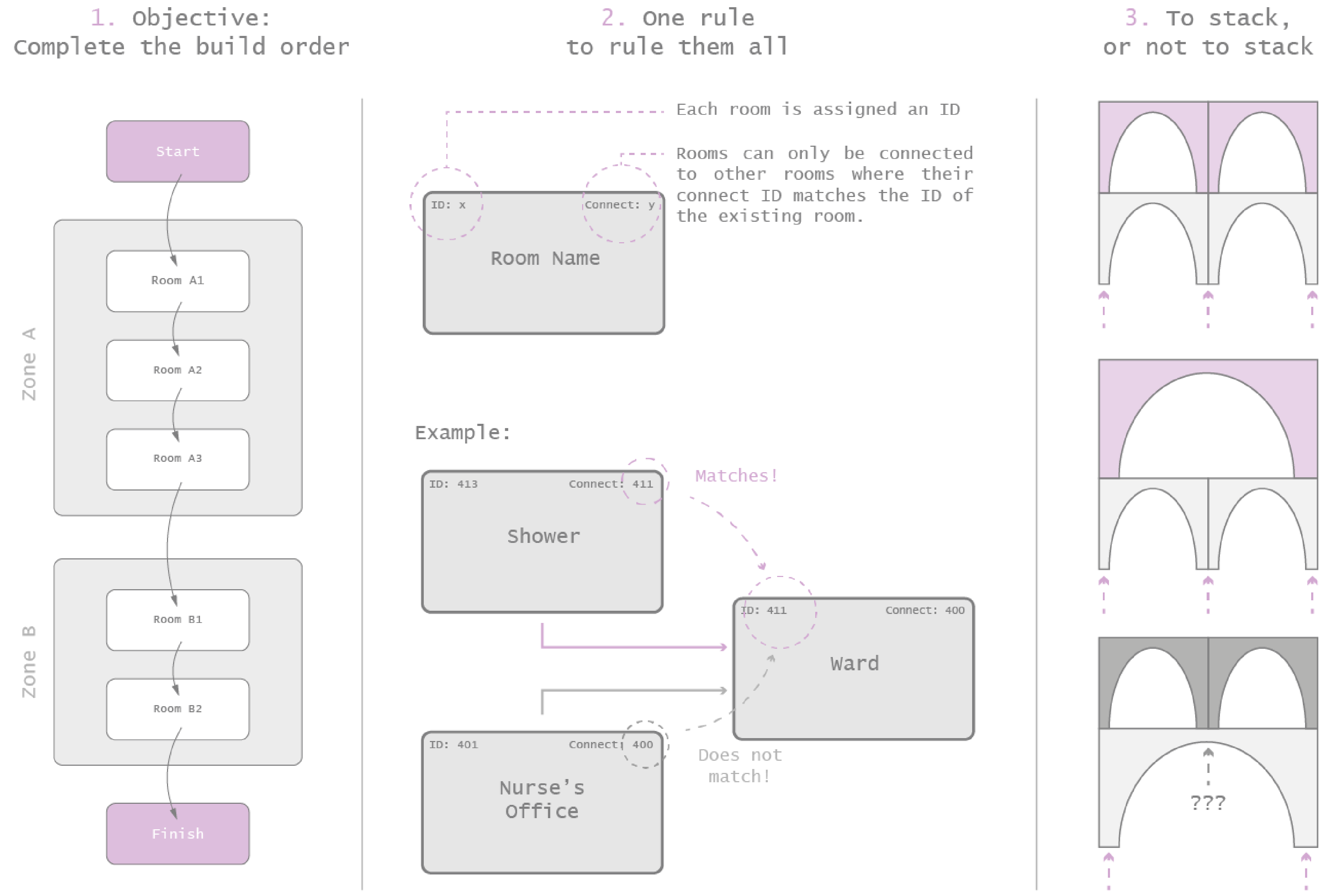


Figure 22. Overview of game rules

Digital workflow

So now that we have defined the full program (the puzzle pieces) and the rules with which these puzzle pieces can be configured, we move to a digital implementation of what so far has been a manual process.

We already discussed the urban analysis and how a Grasshopper script is used to determine a location and serviced population. The script derives the number of healthcare units from the chosen level of service, which are written into and Excel file.

In this Excel file the bulk of the data is stored and managed. Here you find a list of all spaces with their properties (including their ID's and connect ID's) the composition and build order of each zone, and some other data that is required to play the game.

All this data is loaded into class objects in Python using Pandas. Now the CARETHY+ game can be played, after which the resulting floor plan can be exported to a new Excel file.

Finally the exported floor plan can be loaded into Grasshopper for further analysis, for example a solar radiation analysis with Ladybug. This last step has not been deeply developed, but it is very interesting to complete the workflow from urban analysis through configuration to evaluation of the floor plan.

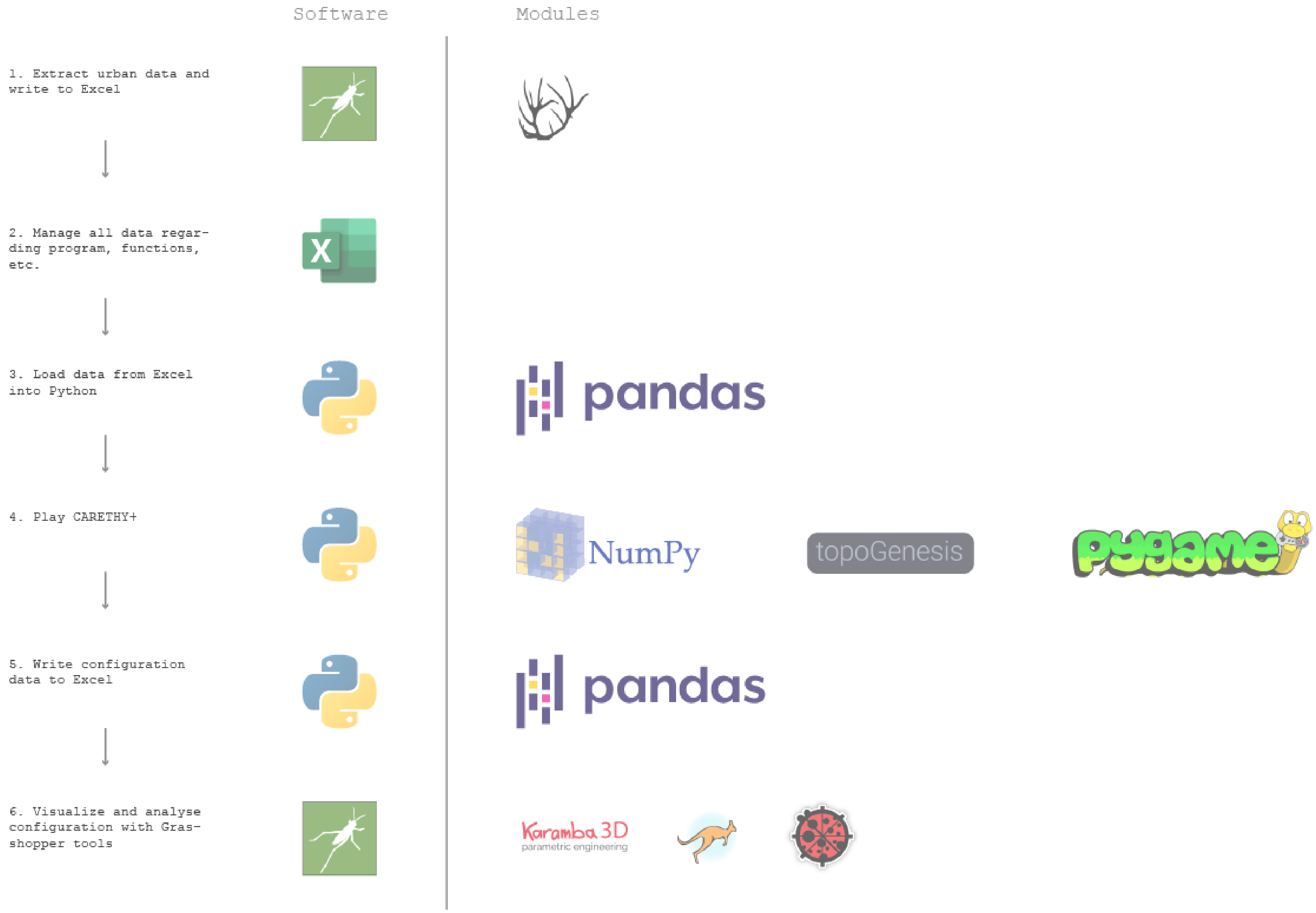


Figure 23. Digital workflow

Data Management

Managing data is usually not the most enticing part of a project, but it plays a crucial part in making sure that the game runs well and that changes can be easily made. We store and manage our data in a Microsoft Excel file which we briefly explain so that when the actual game is explained in the next sections, it is clear where all data comes from.

All spaces have about a dozen properties which all play a part in the digital implementation of the game. These properties are directly loaded into class objects so the game can use them right away. If necessary properties can be adjusted without trouble, so a reception can be made larger or smaller at will as long as a structural module supports the shape (either 1x1, 1x2, 2x2 or 2x3). The same goes for different courtyard and circulation types.

The sheet also gives a good overview of which rooms can connect to which, which also can be altered. We define a distinct circulation type for each zone with its own ID so that, as described in the bubble diagram, zones are given more cohesion. Speaking of zones, the composition and build order of zones can also be managed here. We kept the sheets for zones separated so that management is easier and more flexible.

name	quantity
lobby	1
x44	1
reception	1
postal	1
pharmacy	1
cafeteria	1
kitchen	1
education	1
toilet_general	4

name	tag	legend	type	id	shape_x	shape_y	connection	color_id	floor_id	floor_color_id
lobby	Lobby	G1	room	101	2	3	99	5	999	2
reception	Reception	G2	room	102	1	2	101	6	0	0
postal	Postal Office	G3	room	103	1	2	101	6	0	0
pharmacy	Pharmacy	G4	room	104	1	2	100	6	0	0
cafeteria	Cafeteria	G5	room	105	2	3	100	5	999	2
kitchen	Kitchen	G6	room	106	2	2	105	7	0	0
education	Education	G7	room	107	2	3	100	6	999	2
toilet_general	Toilet	G8	room	108	1	1	100	7	0	0
waiting_emergency	Waiting Room	E1	room	201	1	2	200	17	0	0
x22	Courtyard (2x2)	Courtyard	courtyard	55	4	4	101	8	55	8
x23	Courtyard (2x3)	Courtyard	courtyard	55	4	5	101	8	55	8
x24	Courtyard (2x4)	Courtyard	courtyard	55	4	6	101	8	55	8
circulation_general	Circulation		circulation	100	1	1		1	0	0
circulation_emergency	Circulation		circulation	200	1	1		1	0	0
circulation_outpatient	Circulation		circulation	300	1	1		1	0	0

Figure 24. Excerpt from data management sheets

The CARETHY+ GUI

Below is an image of the CARETHY+ graphical user interface. We will briefly explain key parts of the interface here.

1. Overview of different spatials units that can be built. Click courtyard to rotate through different sizes.

2. Choose direction and orientation of the spatial unit from the selected cell.

3. Build button to build selected spatial unit.

4. Toggle between different sidebars.

5. Toggle between viewing the ground and first floor.

6. Notification display.

7. Take a screenshot!

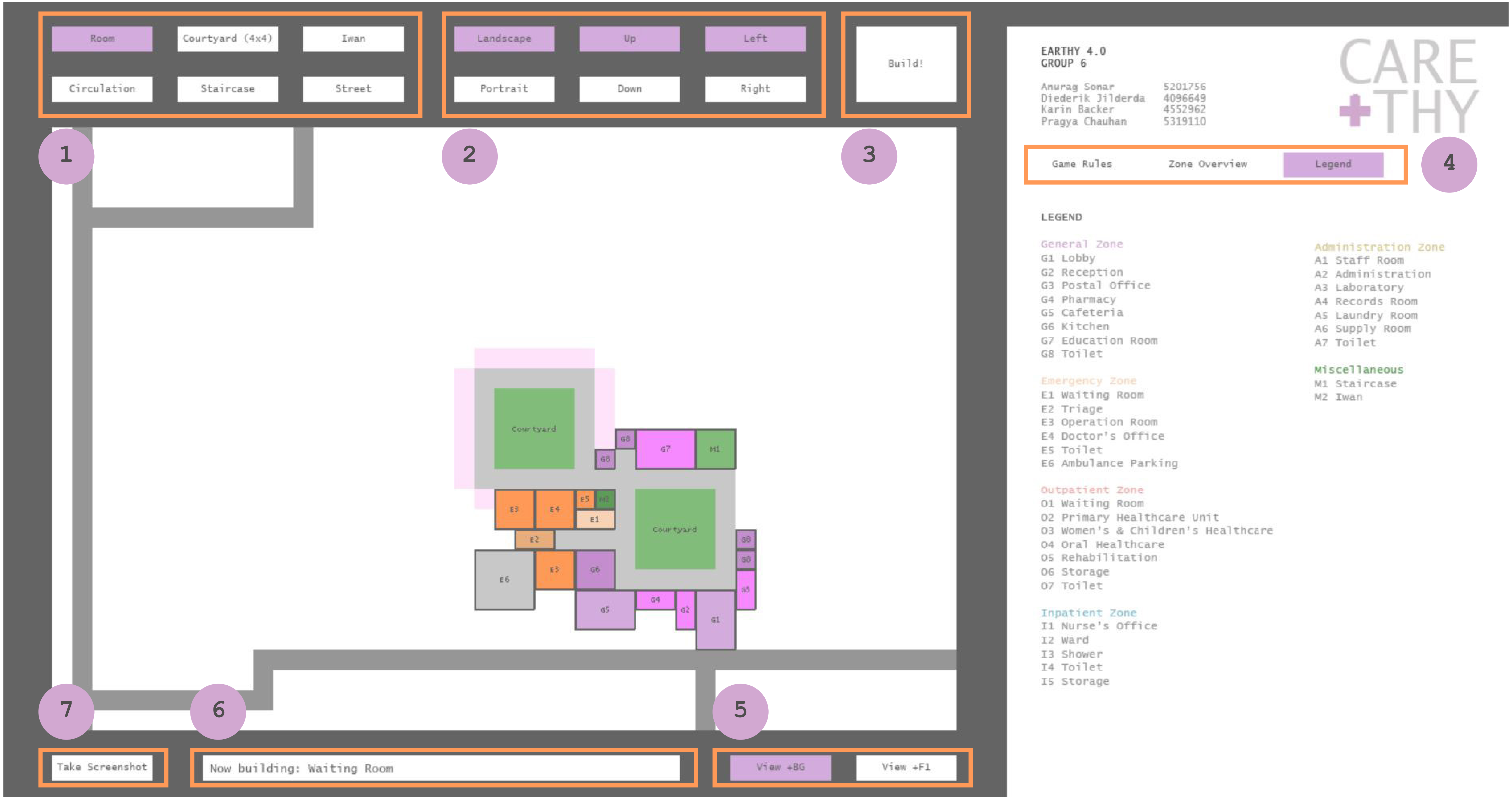


Figure 25. CARETHY+ Interface

Game Process

Figure 26 shows the flowchart of a game from start to finish. We discuss a few processes in further detail, indicated in the flowchart.

1. Initialize game
The first step is to initialize the game. This means setting up all the necessary data regarding space properties, zone composition and the empty grid with local context.

2. Select spatial unit
As can be seen in the GUI, the player has access to different types of spatial units they can build. The selected spatial unit determines which indices in the grid are viable.

Courtyards can be chosen from various sizes. Which version of circulation is added is derived from the active zone. Finally when the player selects room, the first room the build order queue is loaded in. When the room is successfully built, the program deletes this room from the list and the next room takes its place at the top of the queue, until the program is exhausted.

3. Indicate viable indices
The most important rule in the game is about how rooms are connected. Behind the colorful grid seen in the GUI, several arrays work in the background to track among which the room ID's on the grid.

When a spatial unit is selected, the game takes its connect ID and looks for indices that are themselves unoccupied ('0') and are adjacent to a cell that matches the connect ID. These cells are then highlighted to show the player where they are allowed to build their new spatial unit.

Not allowing the player to select any nonviable index acts as a crucial gate to make the system functional. When a player selects any index, the game checks if this index matches any of the viable indices: if not, the selected index is reset and the player has to select a different one. This is a very important safety check.

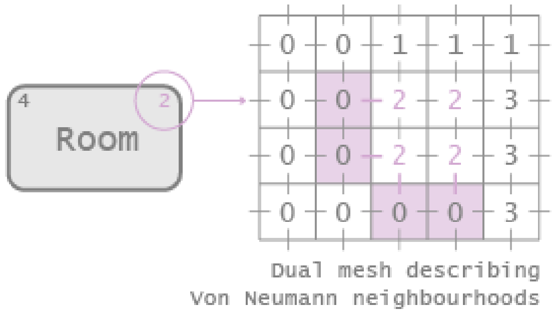


Figure 28. Viability check

4. Collision?
When the build button is clicked, the game attempts to build the spatial unit. However, it first checks if any of the indices that will be occupied by the new room are already occupied. If this is the case, then the build function is unsuccessful and the player has to try again. Otherwise, the room is successfully built and the player can progress to the next unit.

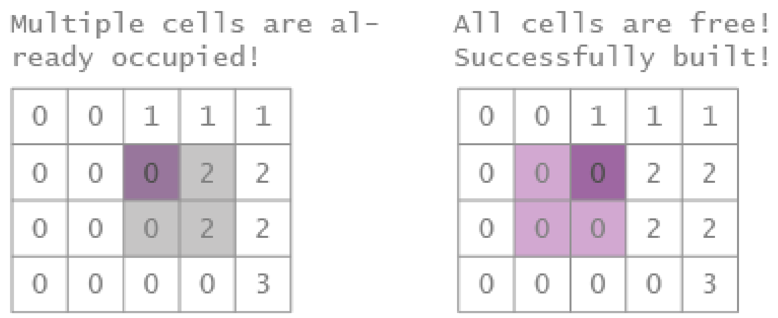


Figure 27. Collision check

5. Zone complete?
After successfully building a room from the build order, this room is deleted from the queue. If there are still rooms left to build, the player is directed back to building a new spatial unit. If the queue is empty, then the game deletes the zone from the zone queue and progresses to the next zone.

6. Hospital complete?
Similar to the build order in a zone, there is a build order of zones. When the last zone is completed, the resulting configuration can be saved, exported to Grasshopper and properly evaluated. The game can then be played again, learning from mistakes in previous iterations.

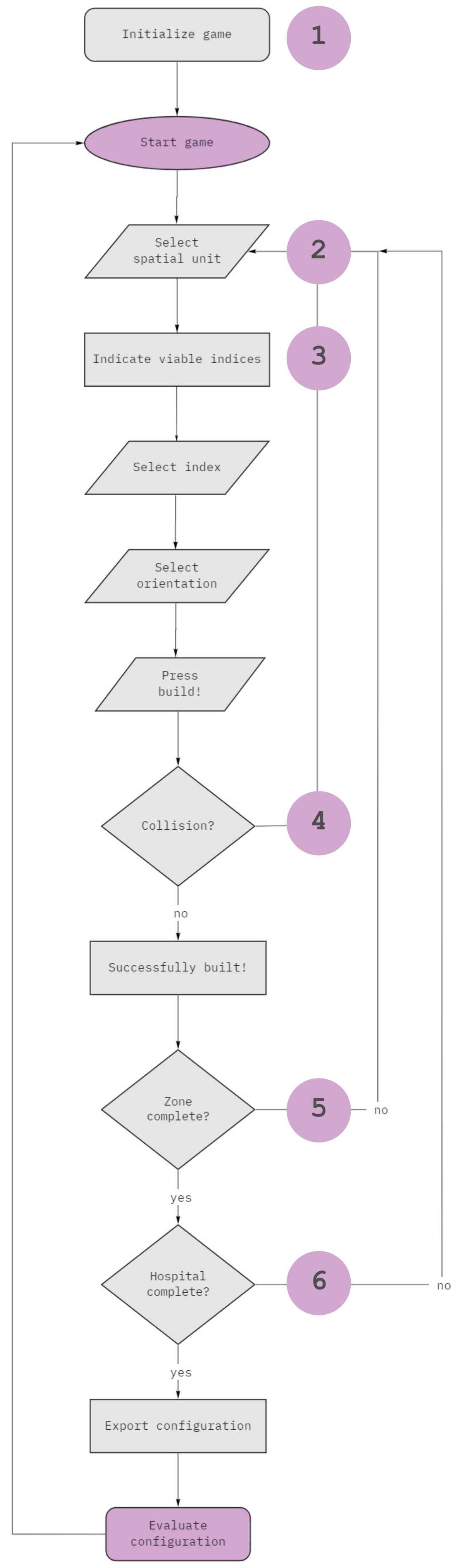
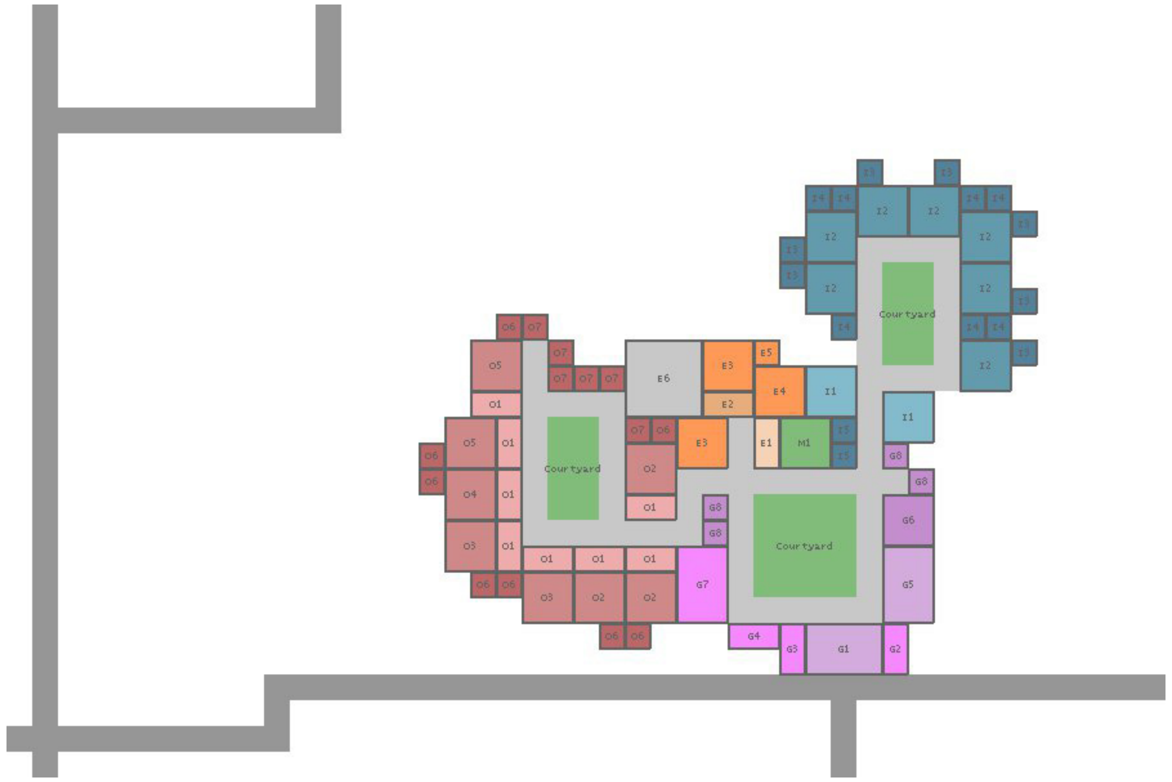


Figure 26. Game flowchart

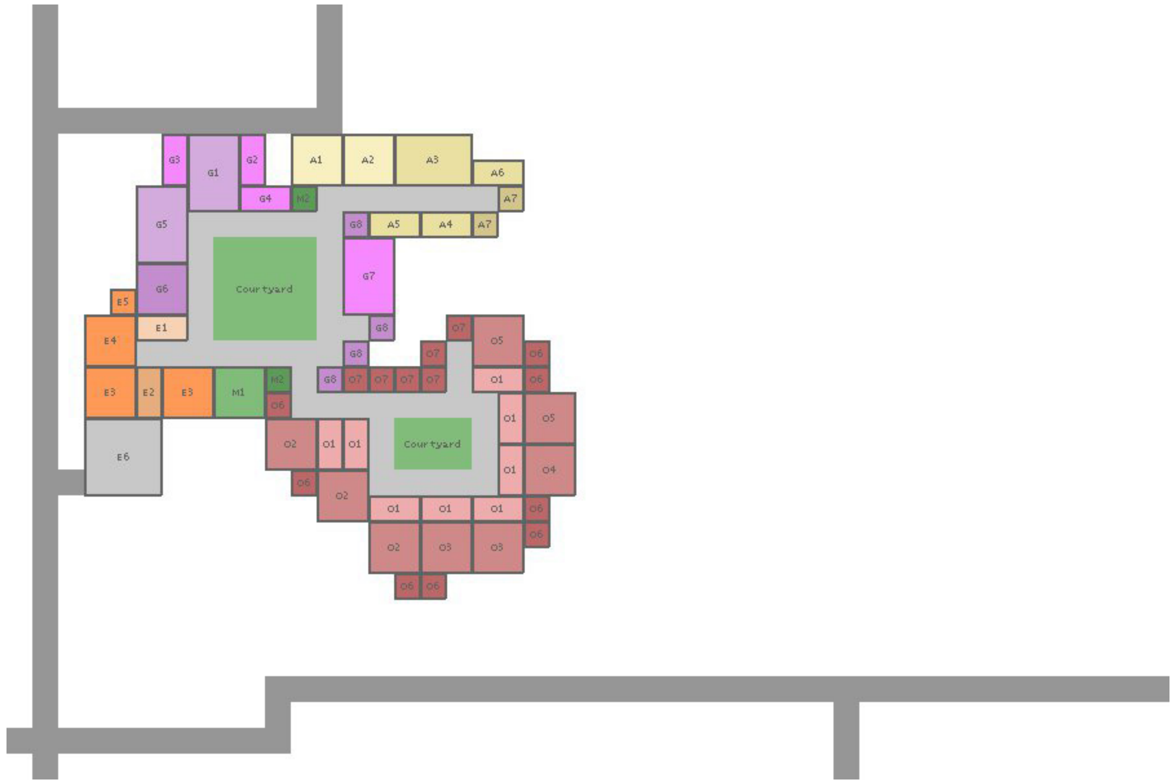
Configuration Options

We played the game a number of times and saved a few floor plans which are shown below.

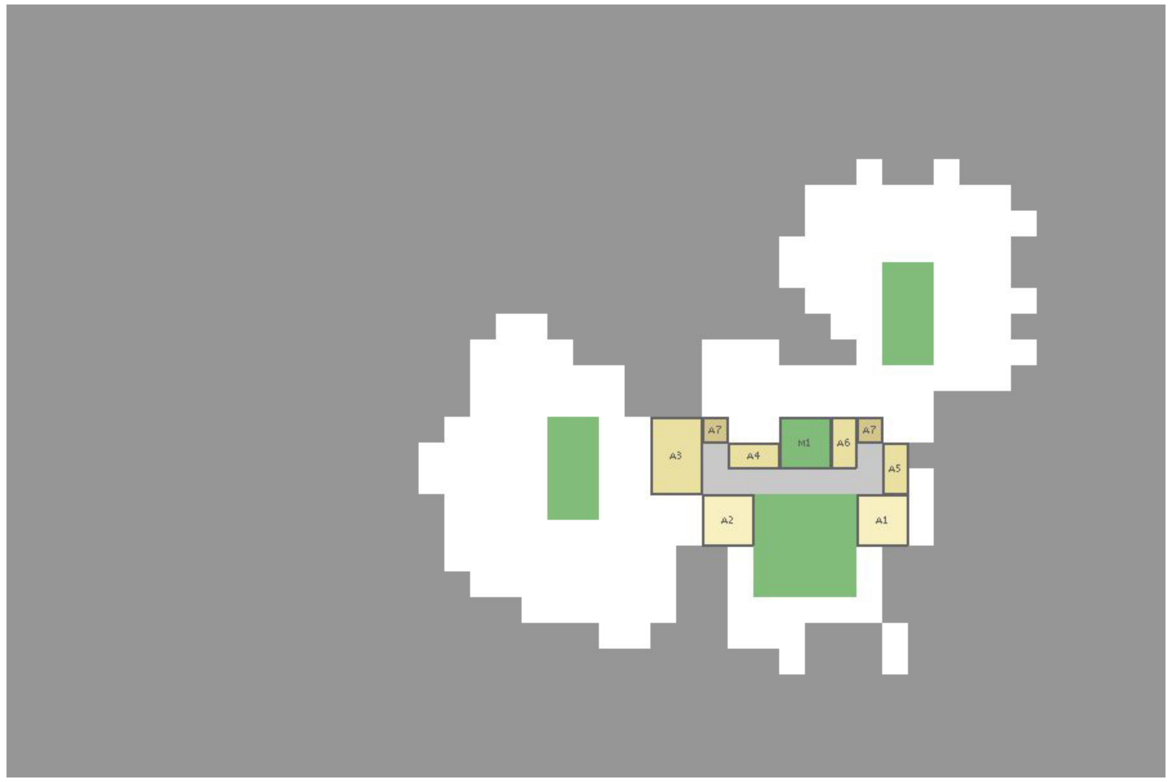
A demonstration of us playing the game is recorded and viewable through the QR-code to the right.



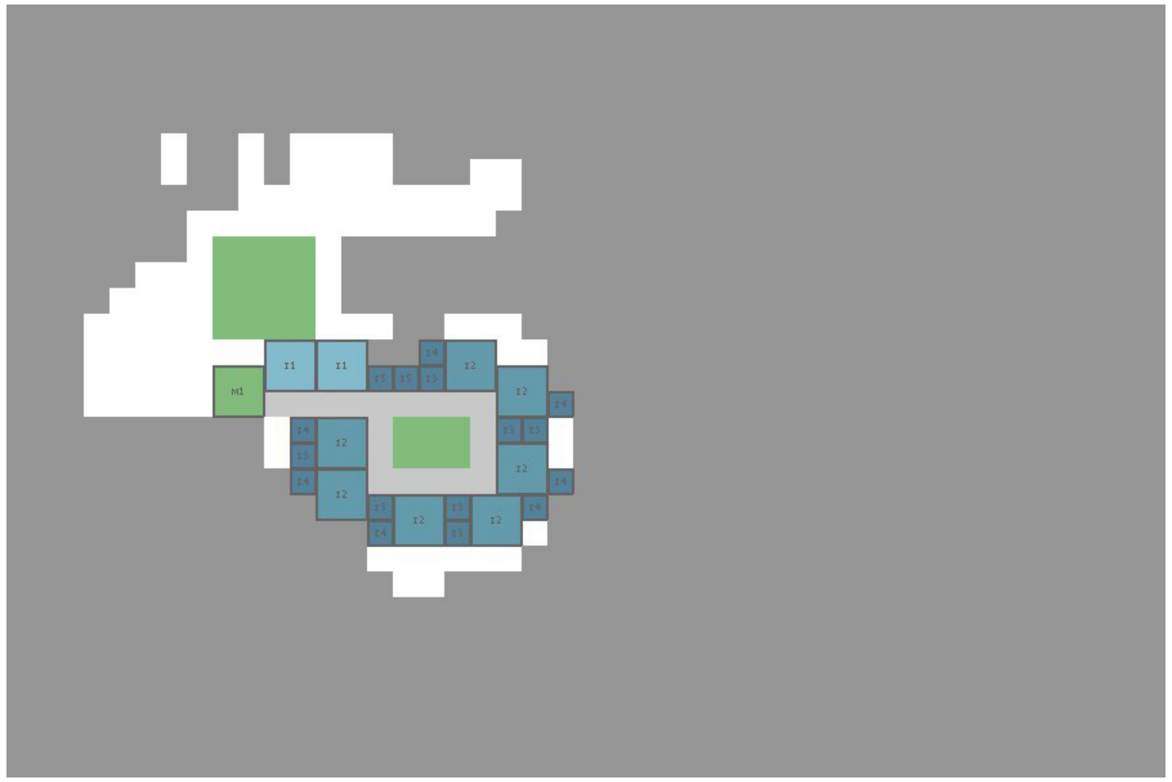
Option 1: Ground Floor



Option 2: Ground Floor



Option 1: First Floor



Option 2: First Floor

Grasshopper Analyses

The final step in the digital workflow is to export the configuration through Excel to a Grasshopper script that can analyse and visualize the floor plan on various topics, such as solar radiation load with Ladybug or structural analysis (Finite Element Analysis) with Karamaba.

Structural analysis will be performed on the individual 1x1 and 3x2 modules, but other analyses fall outside the scope of the project despite being very interesting avenues to explore. For that reason we leave this section as is.

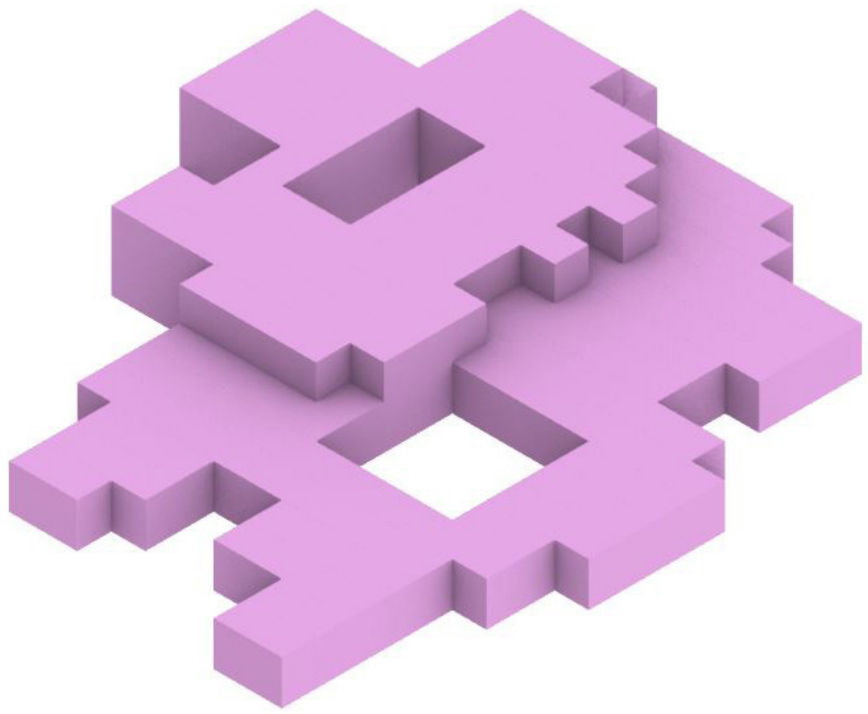


Figure 29. Isometric cube model of configuration

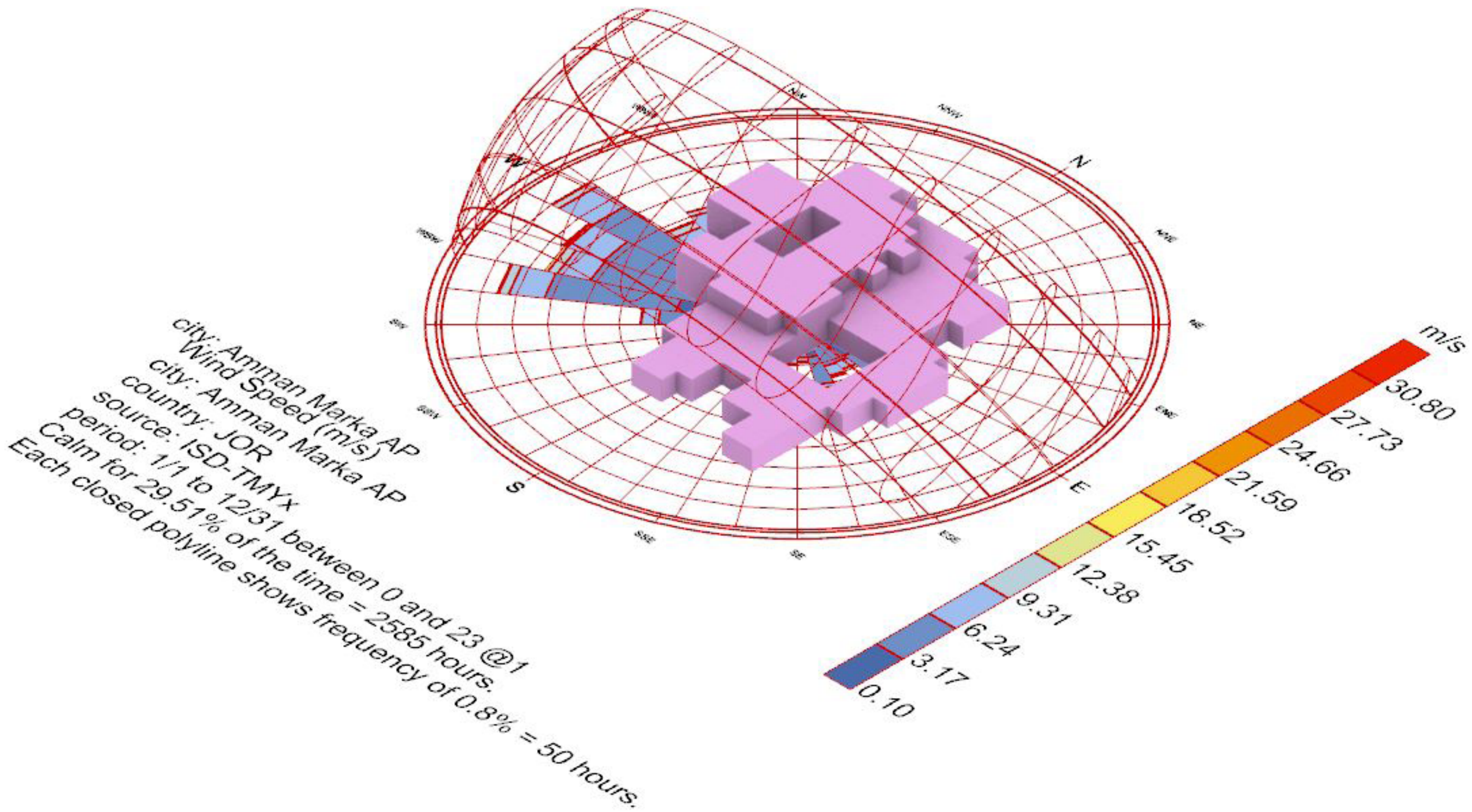


Figure 30. Sun path and wind rose

Final Configuration

We have demonstrated a highly repeatable process to come up with different floor plans, which is the vision for CARETHY+.

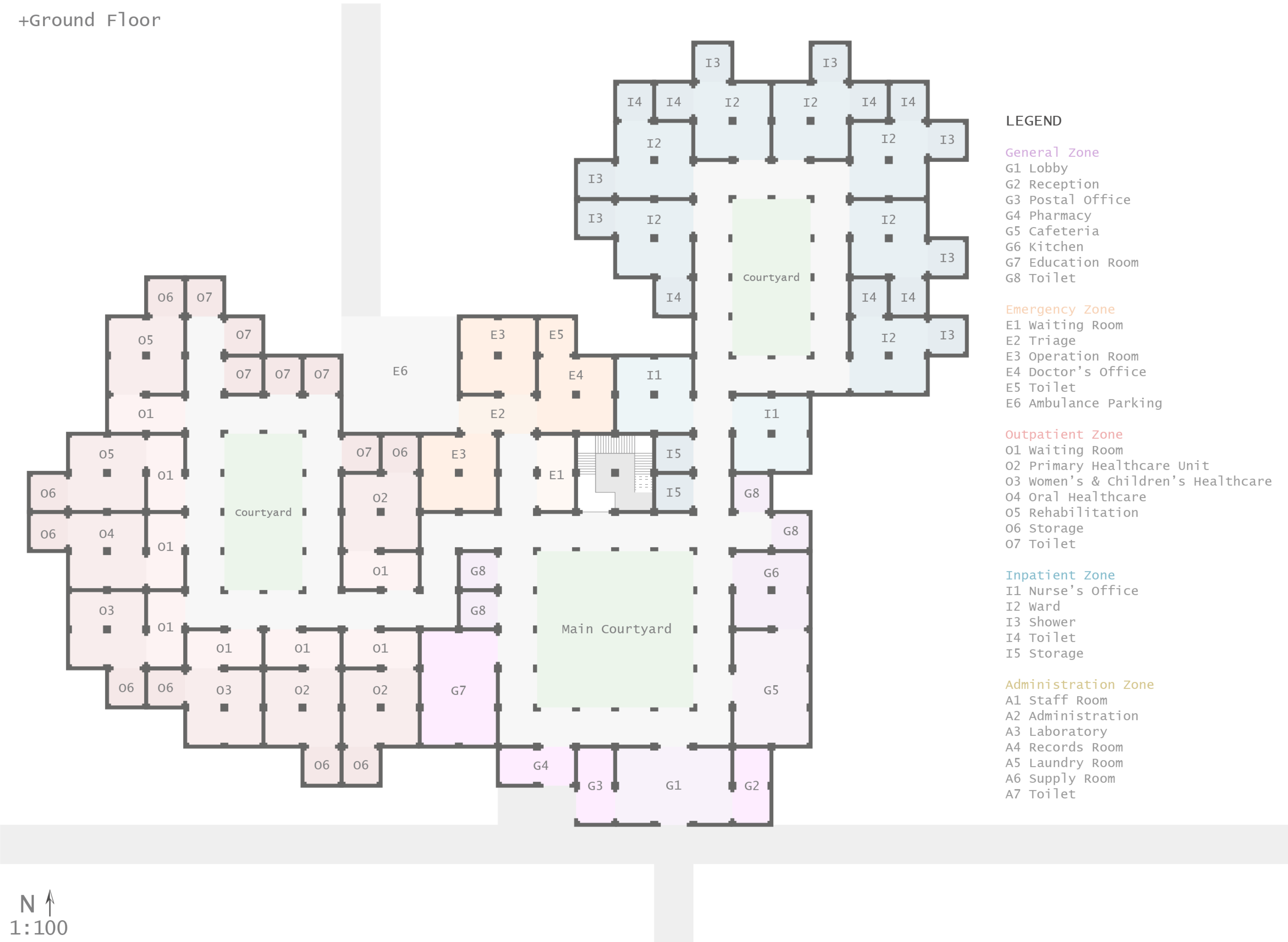
What follows in the report regarding shaping and construction can be applied to any floor plan resulting from the game. That being said, a concrete floor plan is necessary to visualize the shaping and construction processes.

We chose option 1 as the final configuration moving forward. We appreciate its compactness and architectural qualities (administration is better suited to be situated on the first floor than the wards, and the three courtyards make the plan feel more ‘balanced’).

Unfortunately the placement of walls is not part of the computational workflow, and was done manually. In certain places we followed a specific logic (for example, the openings in waiting rooms are staggered so that a direct line of sight inside the doctor’s office from the corridor is minimized), but a proper computational approach is not something we were able to accomplish. We will reflect back on this during the reflection.

More detailed floor plans are shown on the next two pages. These floor plans show the wall thicknesses properly scaled, and the manually placed door openings are added so that the connections between rooms become better defined.

+Ground Floor



LEGEND

General Zone
G1 Lobby
G2 Reception
G3 Postal Office
G4 Pharmacy
G5 Cafeteria
G6 Kitchen
G7 Education Room
G8 Toilet

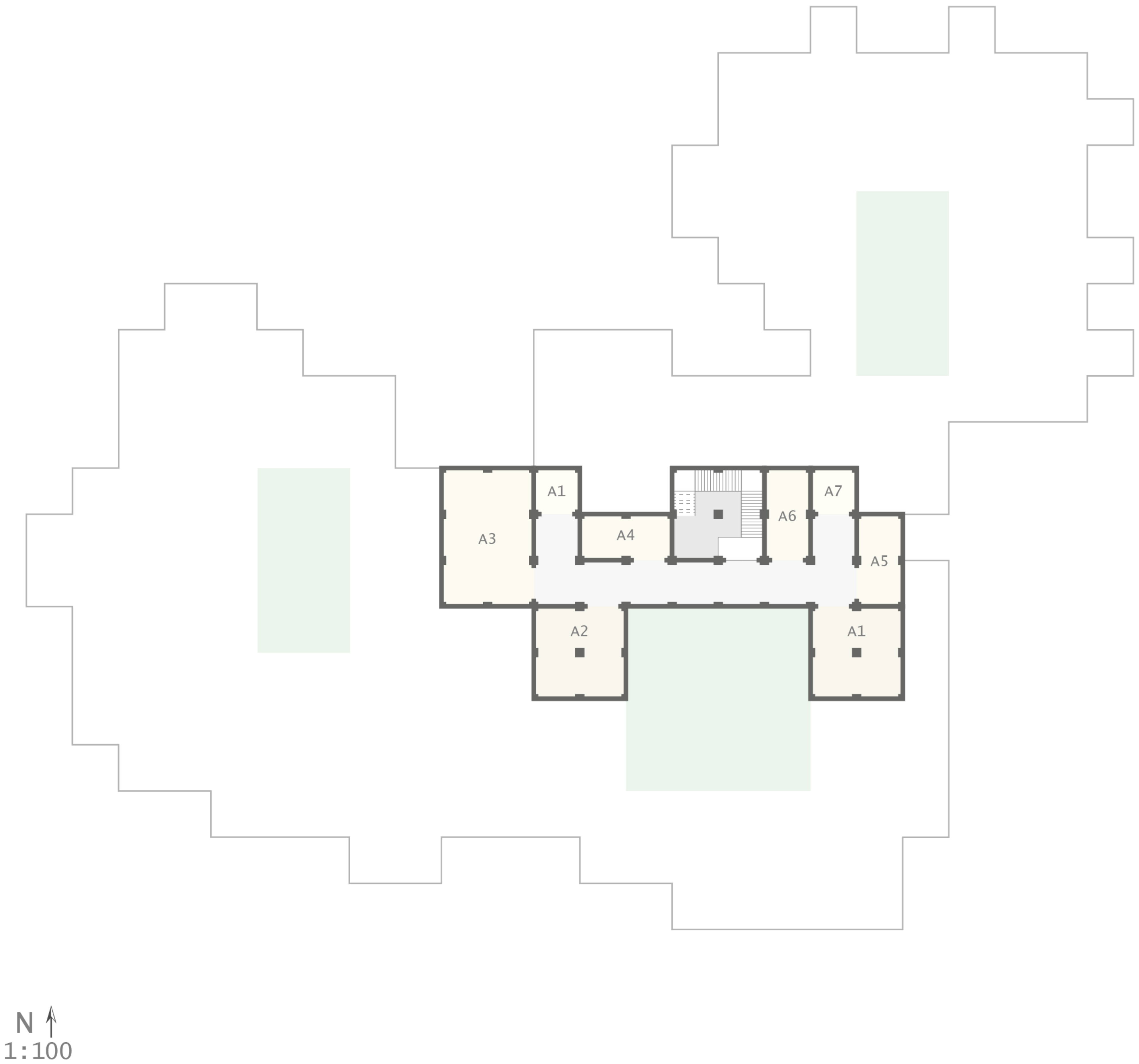
Emergency Zone
E1 Waiting Room
E2 Triage
E3 Operation Room
E4 Doctor's Office
E5 Toilet
E6 Ambulance Parking

Outpatient Zone
O1 Waiting Room
O2 Primary Healthcare Unit
O3 Women's & Children's Healthcare
O4 Oral Healthcare
O5 Rehabilitation
O6 Storage
O7 Toilet

Inpatient Zone
I1 Nurse's Office
I2 Ward
I3 Shower
I4 Toilet
I5 Storage

Administration Zone
A1 Staff Room
A2 Administration
A3 Laboratory
A4 Records Room
A5 Laundry Room
A6 Supply Room
A7 Toilet

+First Floor



LEGEND

General Zone
G1 Lobby
G2 Reception
G3 Postal Office
G4 Pharmacy
G5 Cafeteria
G6 Kitchen
G7 Education Room
G8 Toilet

Emergency Zone
E1 Waiting Room
E2 Triage
E3 Operation Room
E4 Doctor's Office
E5 Toilet
E6 Ambulance Parking

Outpatient Zone
O1 Waiting Room
O2 Primary Healthcare Unit
O3 Women's & Children's Healthcare
O4 Oral Healthcare
O5 Rehabilitation
O6 Storage
O7 Toilet

Inpatient Zone
I1 Nurse's Office
I2 Ward
I3 Shower
I4 Toilet
I5 Storage

Administration Zone
A1 Staff Room
A2 Administration
A3 Laboratory
A4 Records Room
A5 Laundry Room
A6 Supply Room
A7 Toilet

2.3.4. Evaluation

CARETHY+ is a game that is designed to result in logical floor plans. However this does not necessarily mean that the floor plans are also good.

Games like chess and go have deceptively simple rules, yet it is this simple ruleset that gives the player a lot of freedom and these games a lot of depth. Playing chess by the rules means you are playing the game, not that you are playing it well.

Similar to these games, CARETHY+ has a simple ruleset that gives a player large freedom in the way they want to design their environment. This freedom is necessary to anticipate all kinds of contextual restrictions and situations, but that also means that floor plans have to be well-evaluated in order to come up with the right configuration for the right situation.

The game is not meant to be played only once. The first few times the player will probably struggle with gauging and approximating the program until they are familiar with the zones and their compositions. The idea is that after this point, they play the game with a specific evaluation criterion in mind: play the game with as goal to minimize the total circulation length, or improve ventilation, or reduce solar radiation load. This way the player gains more and more a grip on not only the program, but also the quality of the program, and eventually is able to come up with well-balanced configurations.

Making a good set of evaluation tools falls far outside the scope of this project and might be better suited for the scale of a PHD dissertation, but we took the liberty to evaluate our chosen floor plan for circulation length.

Circulation length

Evaluating the circulation length is easily done in Python, and can even be distinguished by zone to refine comparisons with other plans. For the chosen configuration the results are:

General: 21
Emergency: 2
Outpatient: 24
Inpatient: 19
Administration: 9

These values include the riwaqs around the courtyard. The challenge of the game is now to make a new floor plan that requires less circulation space to complete.

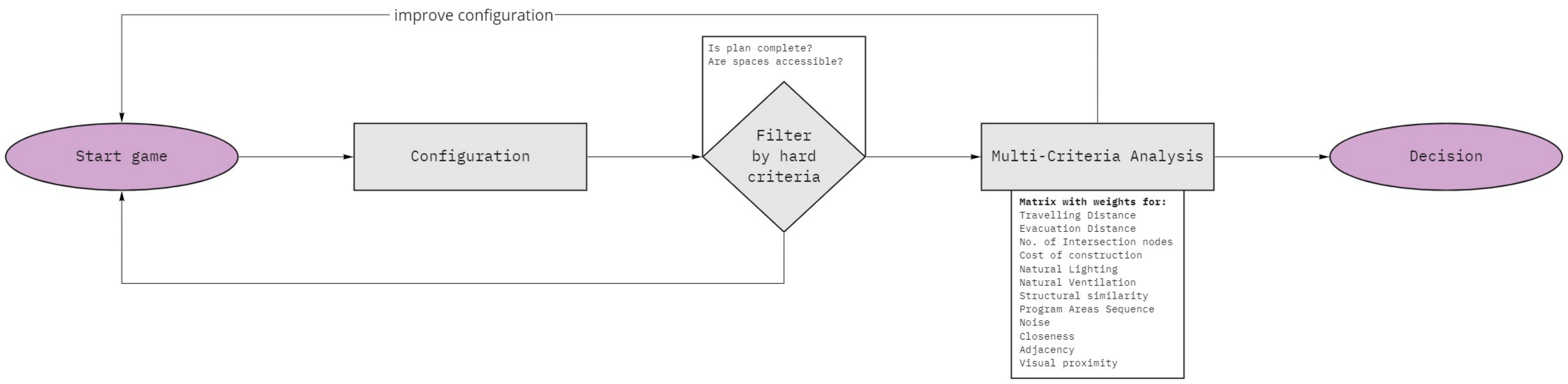


Figure 31. Evaluation flowchart

Comparison to bubble diagram

We would also like to relate our configuration to the bubble diagram to confirm the accessibility of spaces.

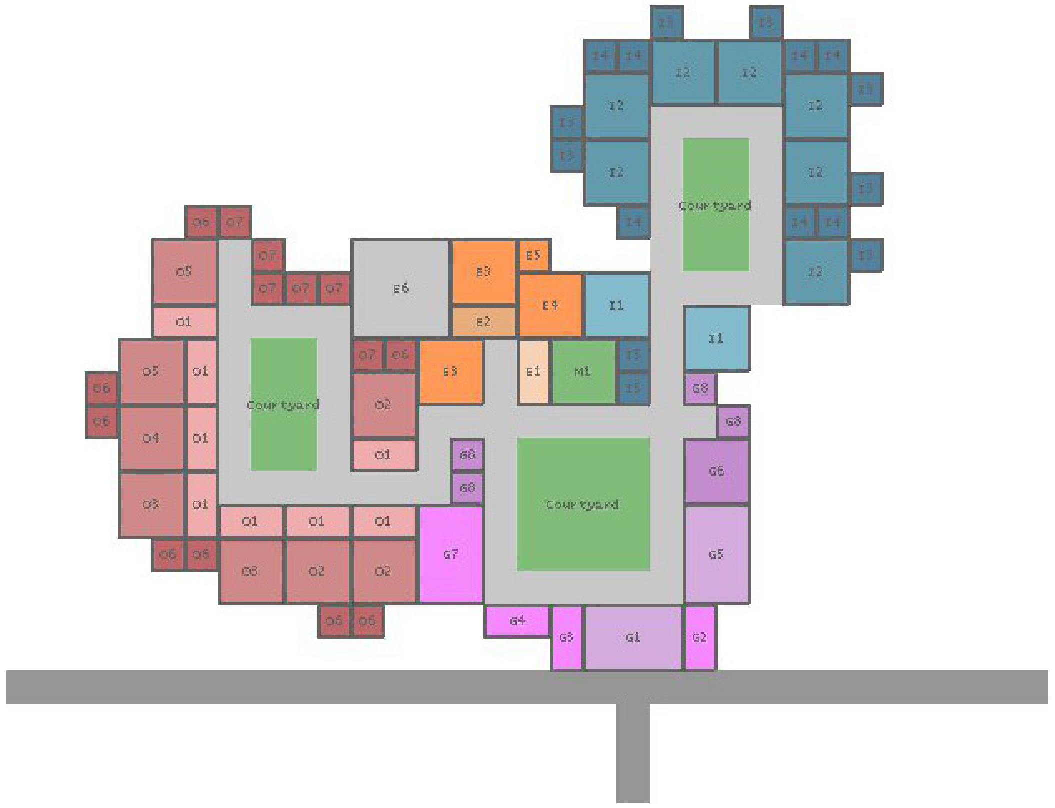
From a first glance, the zones can be clearly identified and distinguished. This is because they are all configured around their own circulation type, which strongly supports the zonal cohesion we intended.

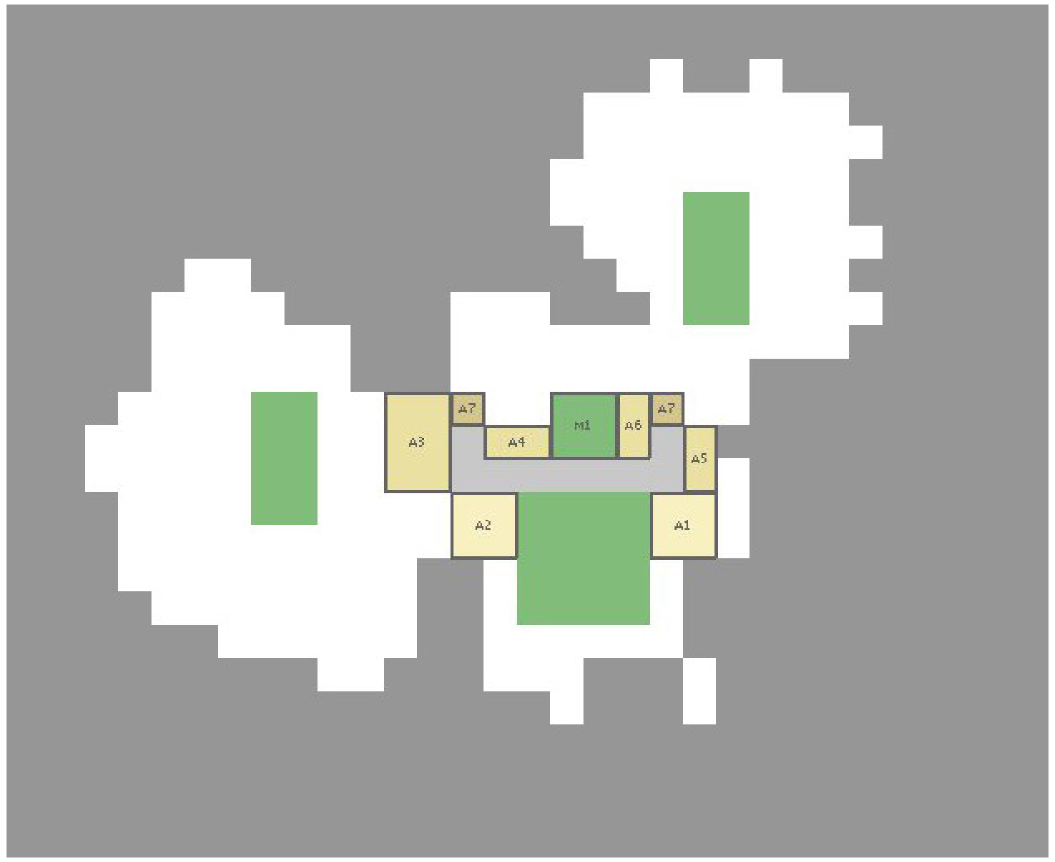
Zooming in on the rooms themselves we can conclude that the configuration matches the bubble diagram exactly, which does not come as a surprise since the rules enforce this.

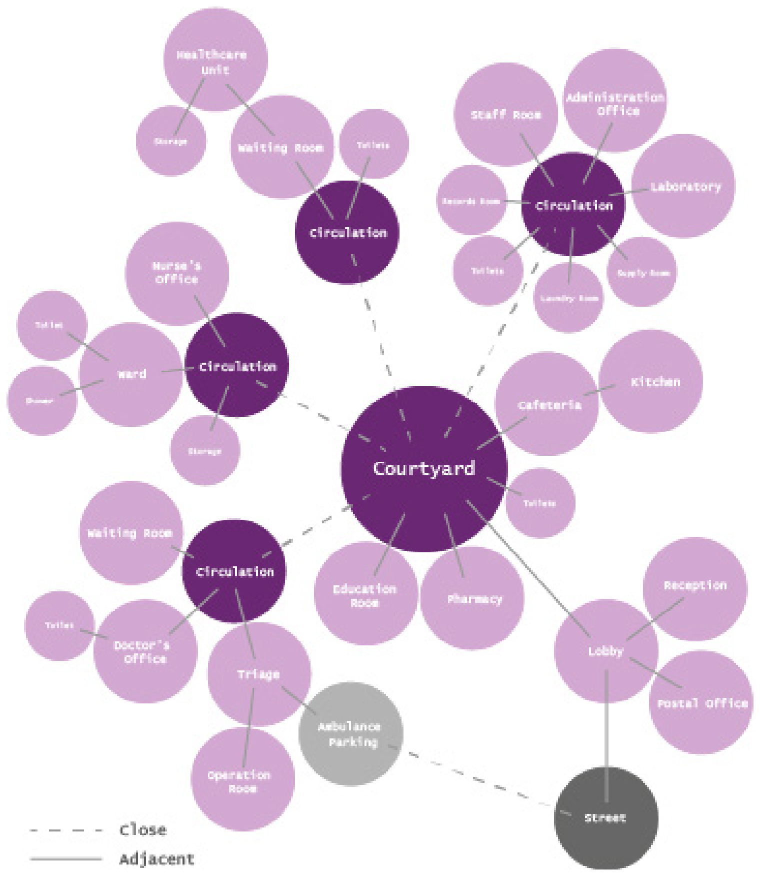
Of course the limitation of the bubble diagram is that it only enforces direct adjacencies. To take the quality of the organisation of the configuration to the next level, closeness should also be taken into account. The clustering within zones already takes care of this to a large extent, but an evaluation tool can be made which calculates the distance between the center points of rooms with a closeness relation. This results in a distance for each closeness relation, which after mass addition results in a total score which should be minimized for optimal configuration.

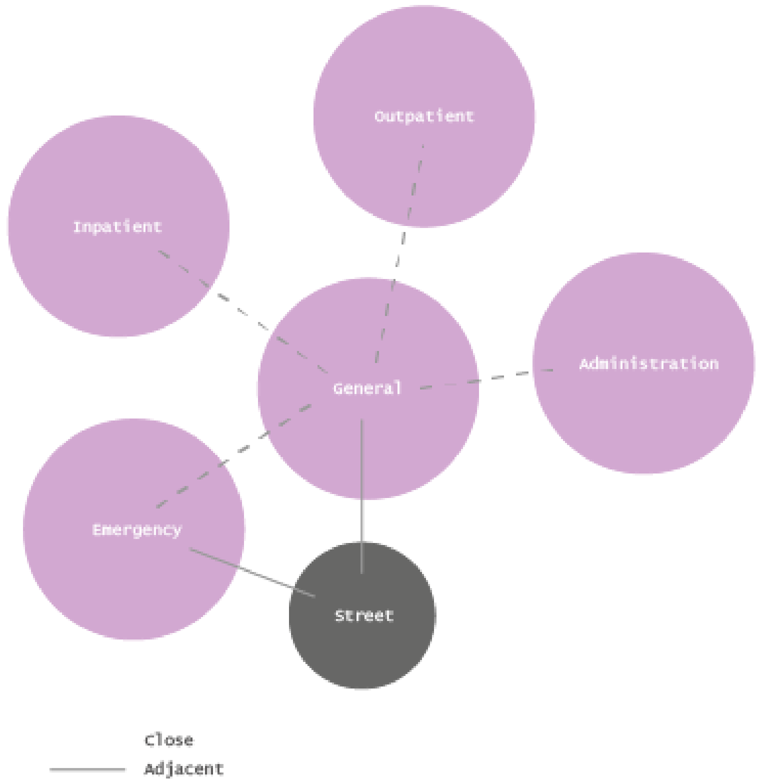
Unfortunately we did not have time to develop this tool, which could have been a create project for an extra group member. We did consider approaches to design this tool however, and the easiest implementation would have been to create a list of rooms with their centerpoint coordinates, and then using the REL-chart relations as a mask to calculate the distance between a room and all rooms it is supposed to be close to.

The vision of CARETHY+ however was to create a playable game that results in a logical floor plan. This tool would have been used in the evaluation process to compare floor plans, so it falls outside the scope of the project.









3. Forming and Shaping

3.1. Vision

The Shaping process and spatial unit development follows a systematic approach. The **tartan grid** is the key component that connects together all the focus points and scales of the process. It also dictates the shaping and the dimensions. At the end the tartan grid is used as a reference guideline for the urban aggregation, spatial unit development and the construction process. This basic grid is 2400*2400mm. All the spatial units are within the multiples of this grid dimension and follow the similar ergonomics, construction methodology and its workability requirements in terms of the opening sizes and the voxel sizes.

The goal for the form finding is to design a modular and easily repeatable spatial unit that can be connected to various similar spatial units to create a flexible and indefinite system. One of the most important aspect of the whole shaping process is also to allow some level of participatory design. This enhances and allows the user's preferences influence the final output. For this reason, some design aspects such as wall openings are left to the user to decide out of a set of varied options, which again follow similar tartan grid and do not affect the general shaping and structuring goals.

3.1.1. Shaping Elements

The final output of the configuration process is a 2D plan, that provides us the information of various adjacencies. The transition from 2D plan to 3D space is achieved through our Shaping elements library. A categorization of all the shaping elements before and after the shaping process is shown below.

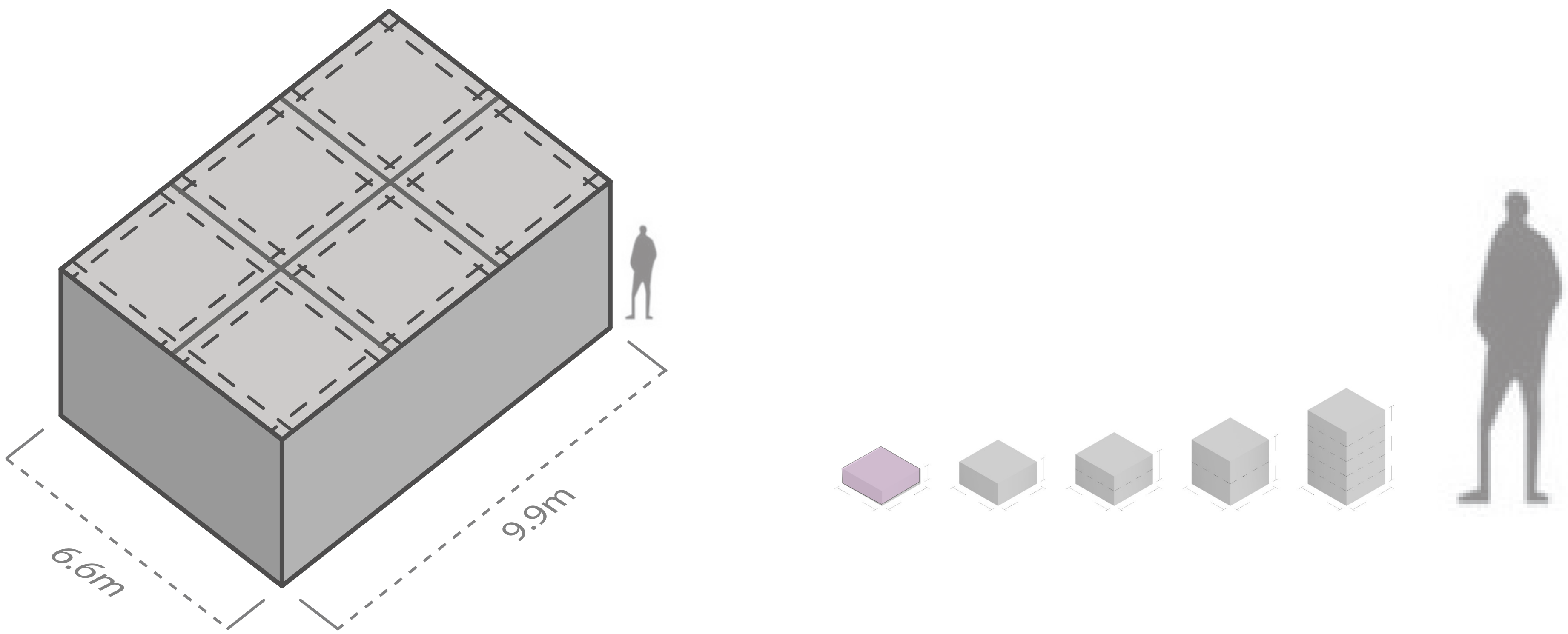


Figure 32. Relations of dimension units and connection to ergonomics, workability and construction

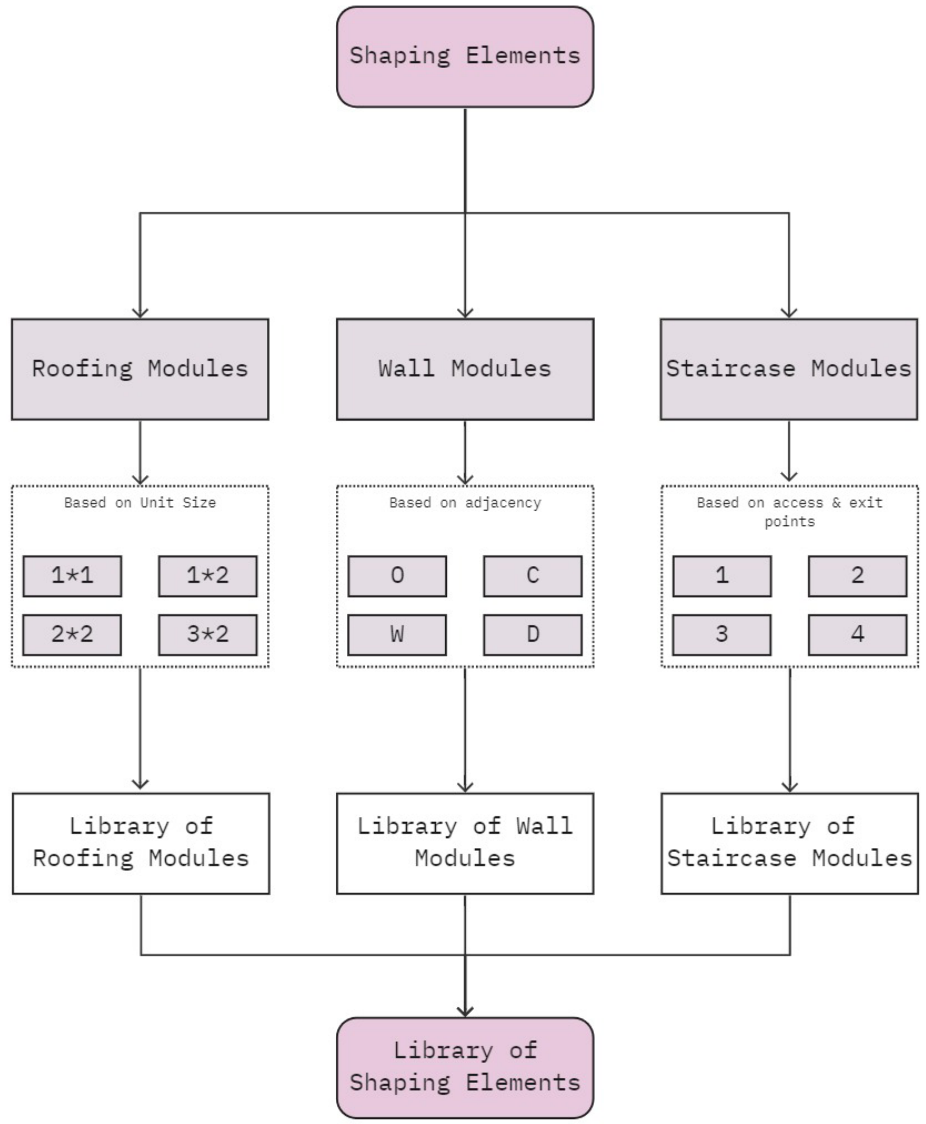


Figure 33. Flow chart of categorisation of elements

3.1.2. Shaping considerations

Based on the shaping vision we identified few important aspects that we need to consider before working on the form-finding process. Following are the considerations,

Structural Modularity

The aim is to design a system that can be easily repeated and allows some degree of choice & freedom to the user to incorporate the wall infills. This meant that each spatial unit need to be self-supported and designed such that any change in the wall opening should not affect its structural stability.

Effective Load distribution

The spatial configuration has the freedom to design on the multiple levels. This raised a very importnt challenege of designing a system that allows effective load distribution. For this we look at gothic architecture and extract some elements such as pointed arches and flying butresses, that later became the basis of our form-finding process.

Flexibility in Wall openings

The program of the healthcare centre is huge and tends to change based on the density it is catering to. This means that there can be various different adjacency cases that demand a level of flexibility in their opening criterias. Thus, the idea is to allow the user to choose these wall openings based on the available modules.

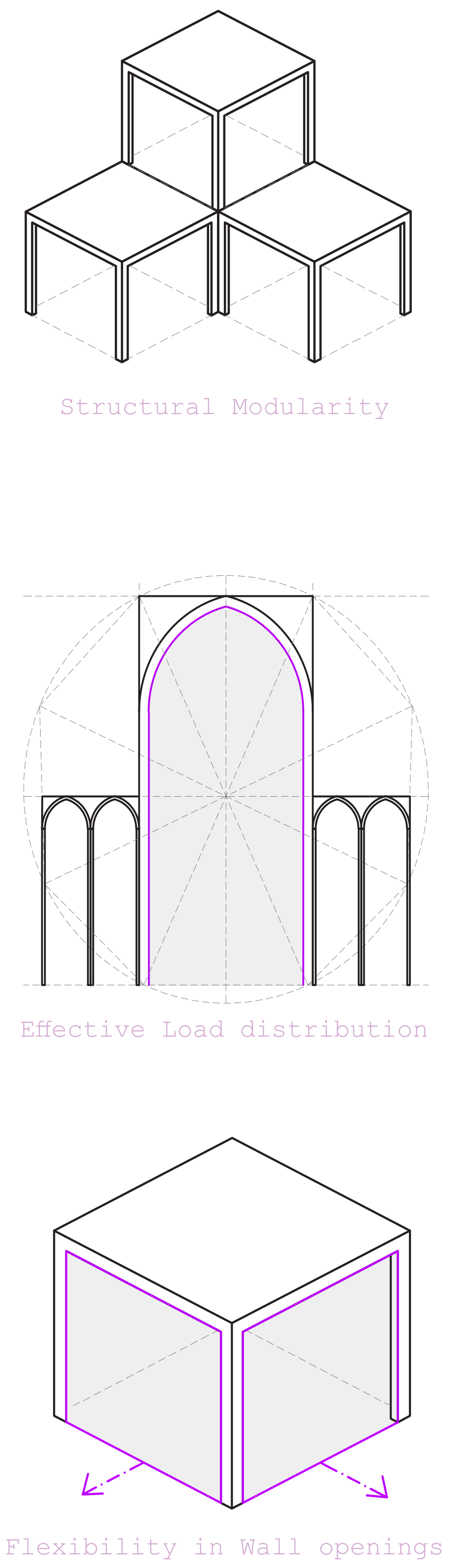


Figure 34. Shaping Considerations

3.1.3. Issues and Simplification

Gothic architecture allows us to translate these considerations in our shaping vision. We identify some issues and try to simplify them for our shaping logic.

The first issue that we notice is that a larger and a smaller unit always share the structural load in order to be more stable. This is mainly due to the fact that the ribs and arches are merged with the structural column, which is not a modular solution, figure 35. Thus, in order to make it modular we split these spatial units and make them self-supported, as shown in figure 36.

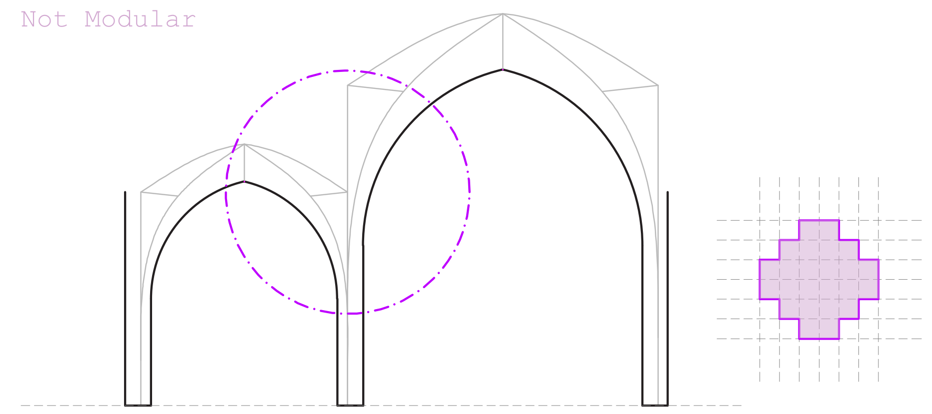


Figure 35. Case of shared load in between 2 spatial units

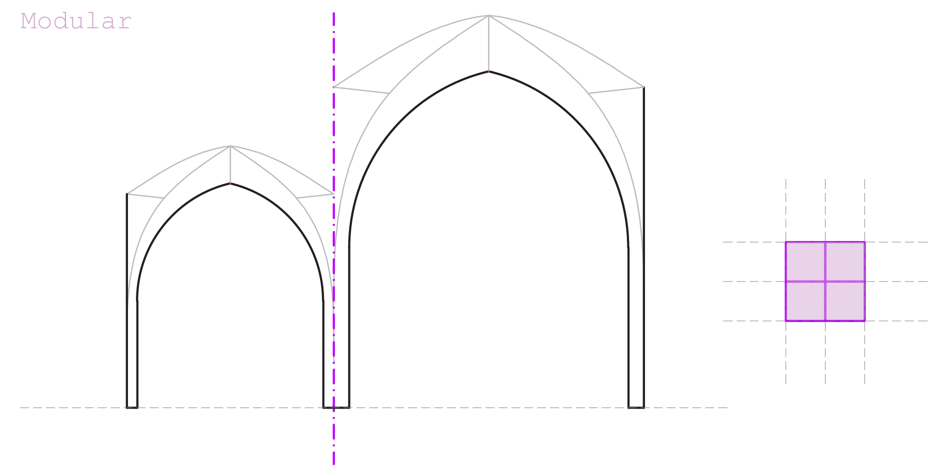


Figure 36. Splitting into self standing spatial units

Further, we observe that the openings in between two different spatial units are mostly not aligne, as seen in figure 37. To simplify this we decide to align the openings within the walls. Since the units are self-standing, the wall infill can allow a modular solution for aligning the openings.

We then propose the opening proportions based on the 2-centre pointed arch logic as shown below. Thus, we can achieve a complete modular spatial unit by setting rules for opening proportions.

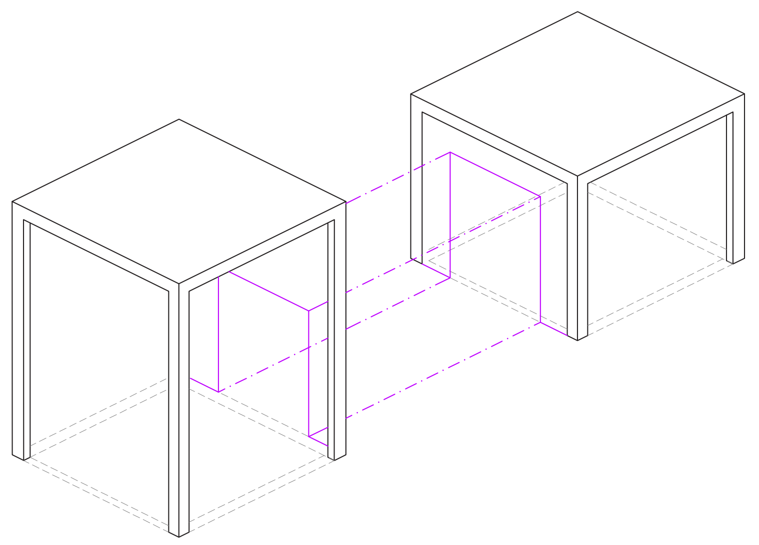


Figure 37. Aligning the openings through wall infills

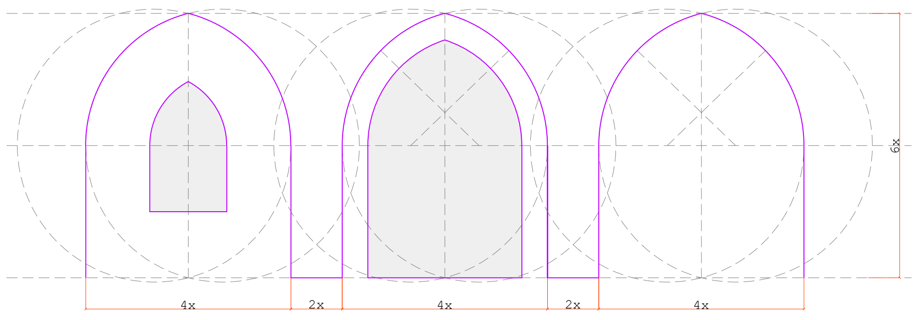


Figure 38. Opening proportions

Introduction of Double-tartan grid

The spatial configuration is based on single tartan grid of 2.4*2.4m. However, for shaping and structuring process we now incorporate another layer of structural grid over the previous tartan grid. Since the spatial units are self standing, the double-tartan grid will incorporate the columns of 0.45m without reducing the effective width of the corridor to be 2.4m.

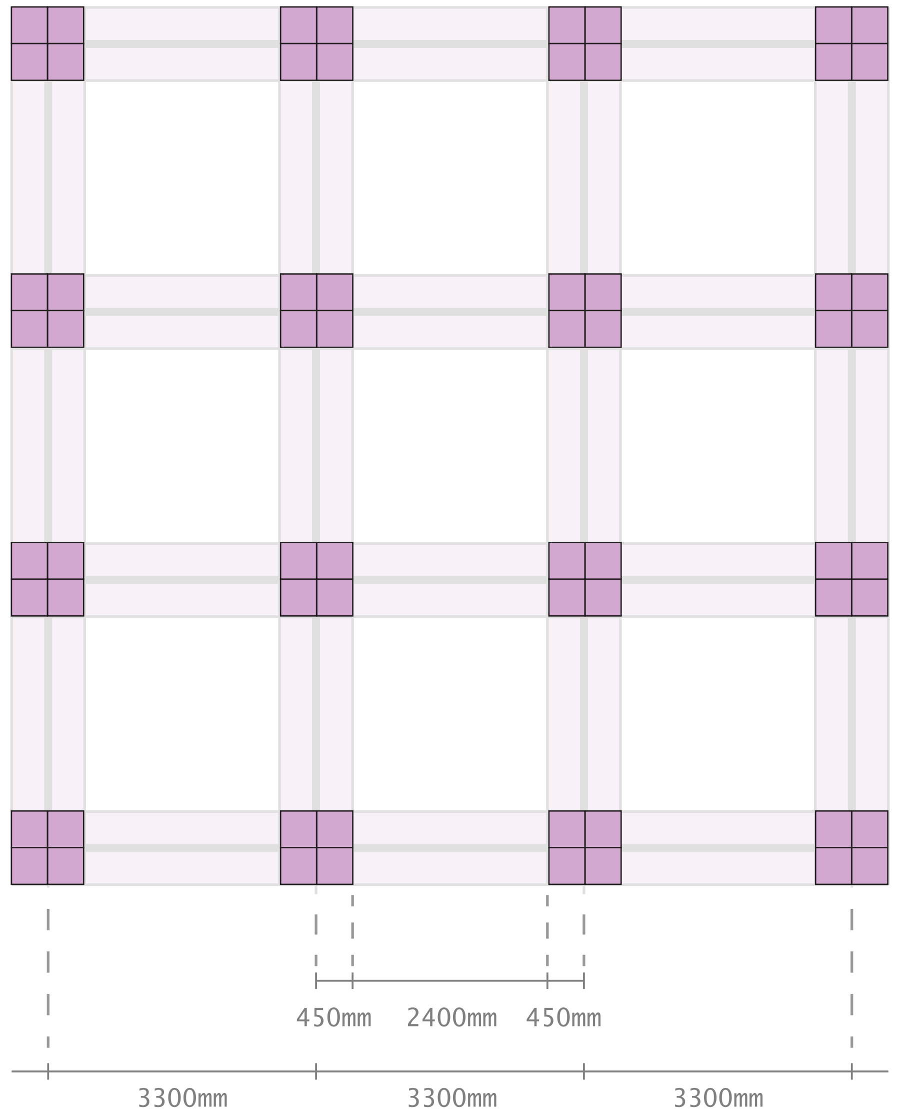


Figure 39. Double Tartan Grid

Simplifying the modules for structuring

During spatial configuration we discussed that there are 4 different types of modules based on the grid size and the functions(see fig. 9). However, to approach the design on first floor we simplified them structurally into only 2 types viz. 1*1 and 3*2. This meant that the 1*1 module is used to shape the 1*2 and 2*2 module with a structural column inside. This simplification is important as it gives us flexibility to design on the first floor

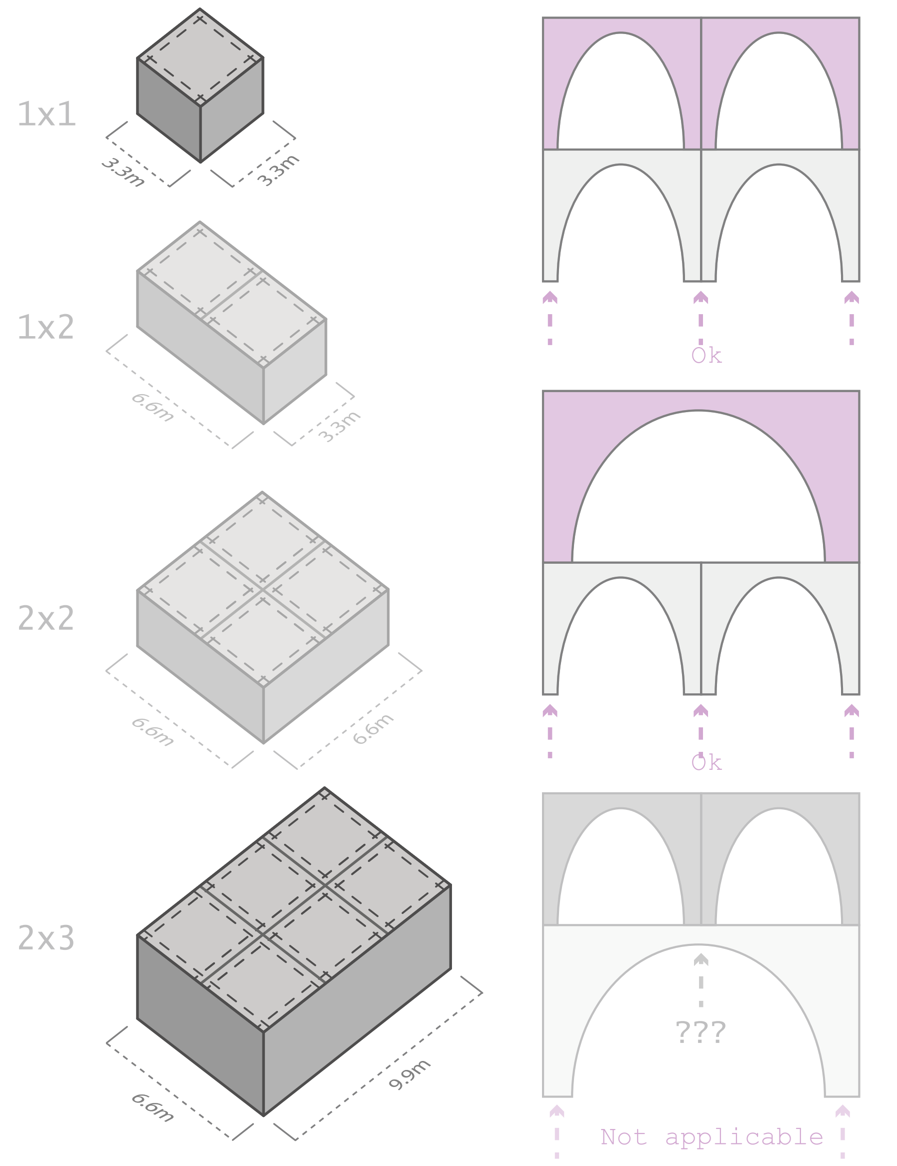


Figure 40. Finalised structural module based on loading

3.2. Shaping Methodology

As we have identified the shaping goals, some basic opening parameters for structuring and the shaping elements requirements, we can now lay down the methodology that we will follow for form finding. The below flowchart is a step by step process that we will follow for achieving the final shell and opening shapes.

Shaping is continuous process that requires to and fro form study and structural verification untill the most functional result is obtained. The final output is manually chosen (one that meets the shaping goals requirements) for further detailing.

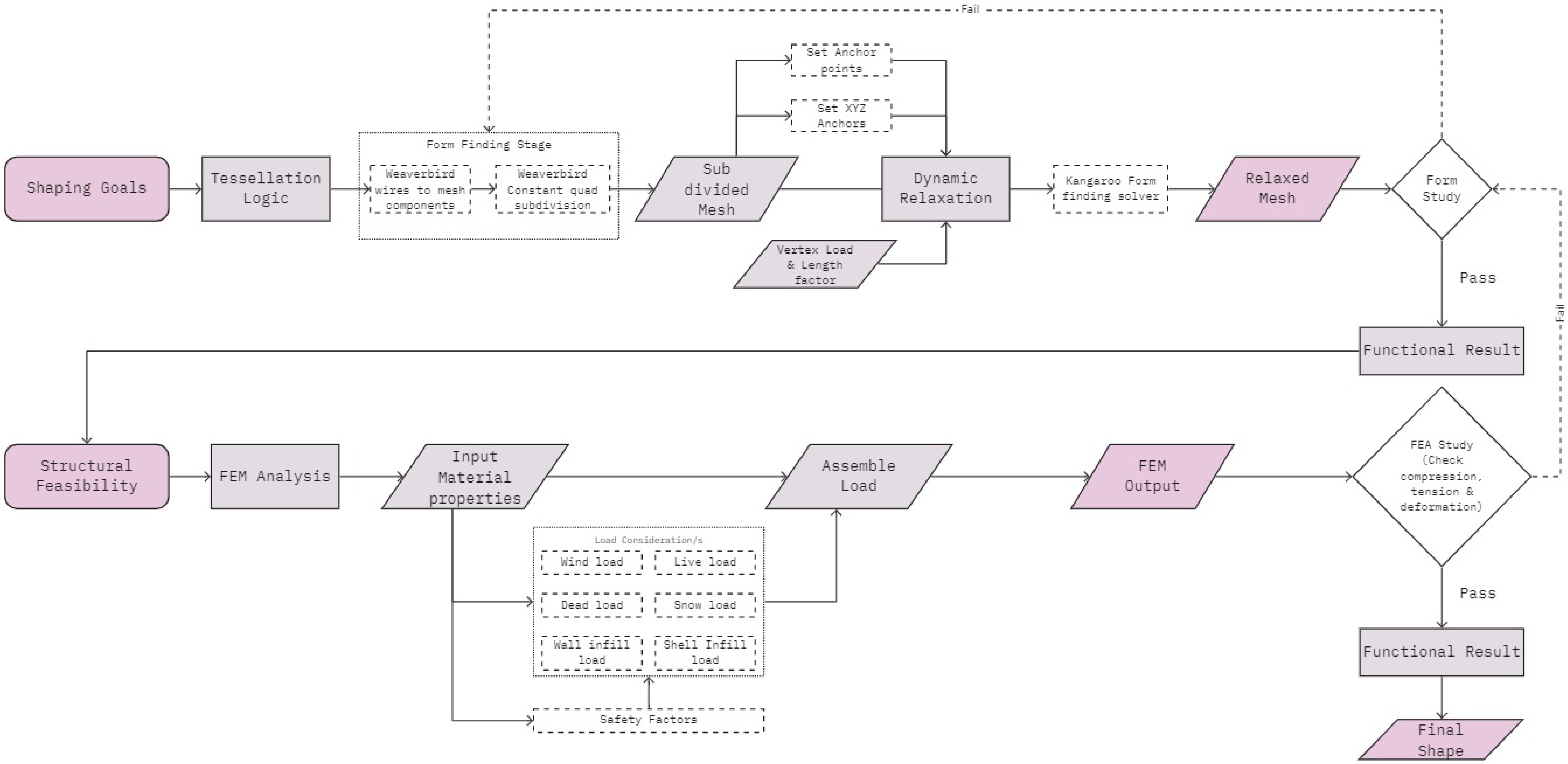


Figure 41. Flowchart shaping methodology and process

3.3. Initial Explorations

3.3.1. Approach towards roof tessellations

Initially we tested the shapes with only 4 anchor points to get better impression of the roof tessellation. However, this approach soon became irrelevant as the shell started to loose its anchorage to the XYZ plane. From the very beginning we ideated these spatial units as sel standing modules, and thus we proceeded further by defining the columns, ribs and the anchor elements so as to achieve realistic results.

The mesh was relaxed using kangaroo plug-in in grasshopper. The idea is to first create a mesh surface using 3D-face manually in rhino. The roof and the ribs are then defined and initially subdivided into quadratic meshes. These 3d mesh surfaces were then joined to form a single mesh. These meshes are then subdivided further into Catmull Clark's constant quad division (using Weaverbird plug-in). A constant quad of level 3 is used so as to analyse the output with similar inputs. Once the mesh is subdivided, vertex load is applied over each vertice with a constant strength and length factor. This forms the basis of relaxing the mesh into most stable form. However, it is important to first set the anchorage points. These are selcted through points in curve function. Further the XYZ planes were anchored.

The output is a relaxed mesh. However, the mesh form is not exactly what we want as it does not consider the height of the structural columns. Thus we redo the mesh now with the 3d face assigned for the columns. This means we create a mesh in X&Y axis. The use of mesh surface instead of the nurbs surface allowed us to visualise the tessellation patterns as the building blocks, which later formed the basis of our detailing.

This process is repeated with different types of sub-divison with slight change in the vertex load and the length factor untill a functional result is obtained.

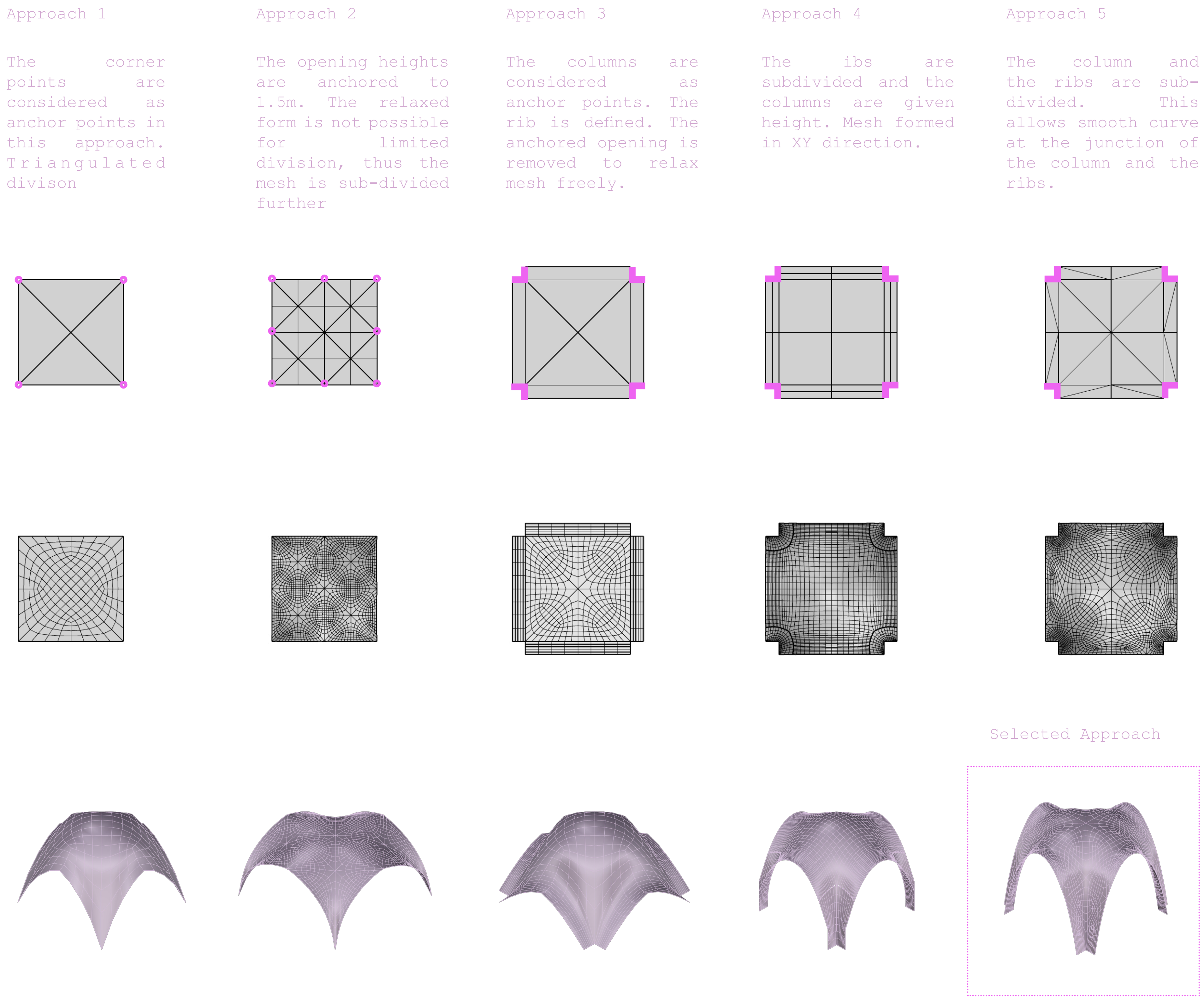


Figure 42. Evolution of Tessellation logic

3.3.2. Explorations for various Module sizes

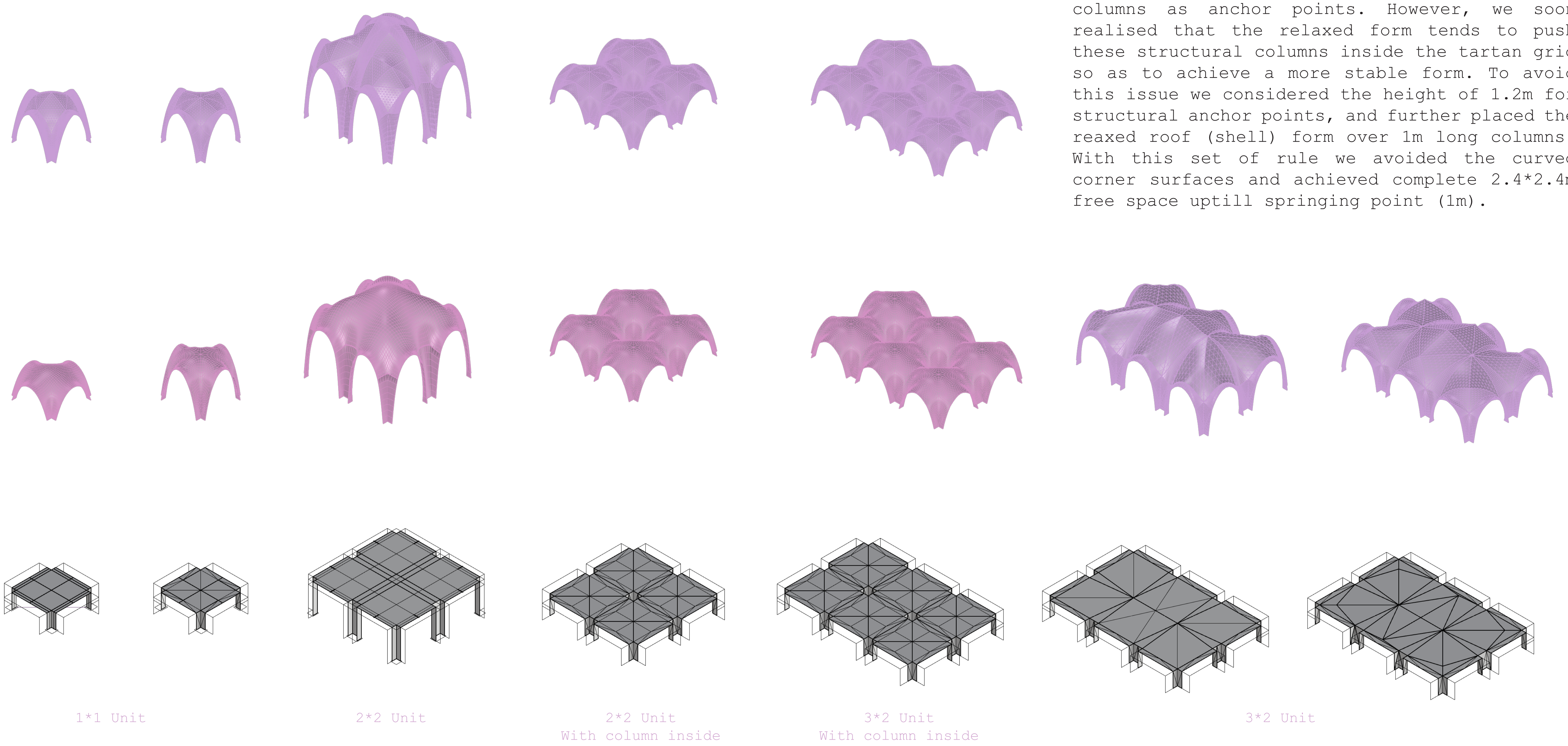


Figure 43. Exploration of Tessellations based on various spatial unit sizes

3.3.3. Conclusions

The spatial units are ideated as self-standing units. Thus the approach towards the roof tessallation was with consideration of structural columns as anchor points. However, we soon realised that the relaxed form tends to push these structural columns inside the tartan grid so as to achieve a more stable form. To avoid this issue we considered the height of 1.2m for structural anchor points, and further placed the reaxed roof (shell) form over 1m long columns. With this set of rule we avoided the curved corner surfaces and achieved complete 2.4*2.4m free space uptill springing point (1m).

3.4. Material Properties

Before the next step of selecting a fitting tessilation, the material properties need to be more clear. To make building blocks made out of raw earth materials, coarse aggregate (gravel) and fine aggregate (sand) are needed to be the structure. Combined with the clay as a binder, these are the dry mixture elements for the blocks. When these dry ingredients are mixed well together, the water can be added to activate the clay and make the final mixture.

The amount of the ingredients needed for the mix will differ since the material properties variate sper excavation site. However for this project, the following percentages are used:

Clay/silt	30%
Fine sand	30%
Coarse	40%
Dry materials	100%
Added water	110%

Due to the fact that the block will be made by hand and that they may have human errors, the full possible maximum compressive stress cannot be used as a standard parameter. Therefore, the factor of 3.0 N/mm^2 is divided by 2. The maximum tensile stress will be 10% of that. Resulting in the next material properties:

$\sigma_{c,max}$ = 1.5 N/mm^2
$\sigma_{t,max}$ = 0.15 N/mm^2
E = 110 N/mm^2
ρ = 15 kN/m^3

In addition to this, it is important to know the building site and to include the following data that applies to the location.

$F_{l,\ floor}$ = 5.0 kN/m^2
$F_{l,\ roof}$ = 2.0 kN/m^2
F_{wind} = 0.2 kN/m^2
F_{snow} = 0.02 kN/m^2

Figure 44. Coarse aggregate

Figure 45. Fine aggregate

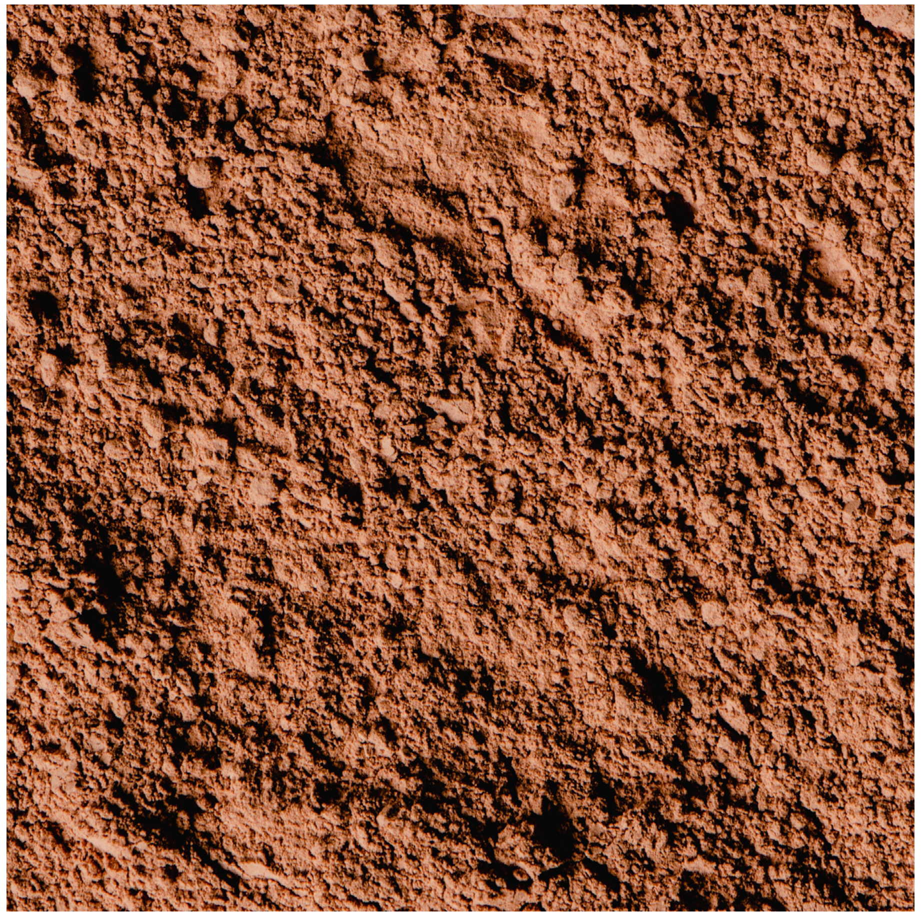


Figure 46. Clay aggregate

Figure 47. Water

3.4.1. Load Considerations

First, hand calculations were made to get a good grip on the forces and the stresses in the modules. The loads are checked for three different module combinations. The first one is the addition of the first floor. This means that the 1x1 module caries another 1x1 module on top of it. The second module is an 1x1 module in the staircase, since the columns are tall and might fail due to buckling. The last one is the 3x2 module. In all of these cases, on of the bottom columns is taken for the analysis.

There are eight different loads on the first module with regards to the base column.

1. The dead load of the upper columns
2. The dead load of the shells
3. The dead load of the sand infill
4. The dead load of the upper walls
5. The live load on the first floor
6. The live load on the roof
7. The snow load
8. The wind load

Since the staircase and the 3x2 never have a floor on top, only load cases 2, 6, 7 and 8 are taking into account for those.

Some values in the calculation are taken from Grasshopper and based on the shell formed by Kangaroo. All the columns are considered to be 45x45mm, the thickness of the shell is 30mm, the maximum compression can be 1.5 N/mm^2 and a maximum tensile of 0.15 N/mm^2. This gives a maximum possible vertical force of 303.75 kN.

To combine all the applied forces together, safety factors and load combinations need to be included in the equation. This will be the following:

$$F_V = SF_{dead}*(F_1+F_2+F_3+F_4) + SF_{live}*\Phi_0*(F_5+F_6+F_7)$$

$$F_H = SF_{live}*\Phi_0*F_8$$

With the equations for all the modules, the following results were obtained and all the values are within the maximum force. The detailed numbers can be found in the Appendix.

First floor module:
F_V = 145.8 kN
F_H = 0.3 kN

First floor module:
F_V = 23.9 kN
F_H = 0.3 kN

3x2 module:
F_V = 40.4 kN
F_H = 0.5 kN

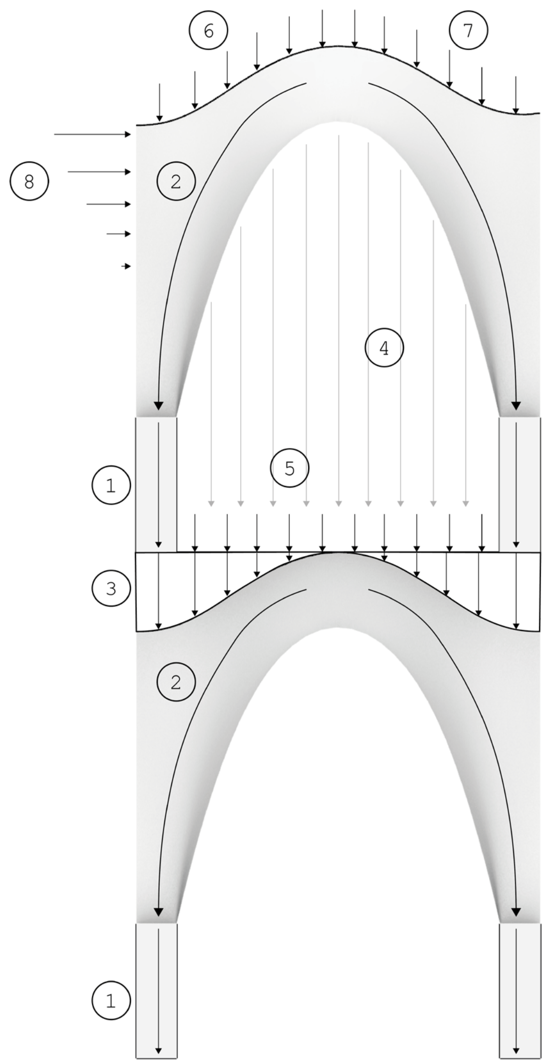


Figure 48. First floor module

Figure 49. Staircase module

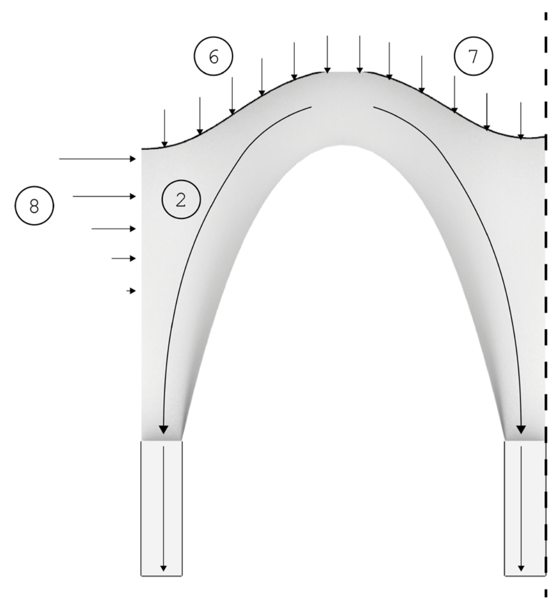


Figure 50. 3x2 module

Stresses

With the obtained forces, the stresses can be calculated using the following equation:

$$\sigma = M/W \pm F/A$$

In which the moment is taken with the V_H and the force is taken from the F_V. Again, all the values are within the maximum of 1.5 N/mm^2 compression stress. The figure shows the stress diagrams for the first floor module.

First floor module:
σ_{max} = 0.74 N/mm^2
σ_{min} = 0.70 N/mm^2

First floor module:
σ_{max} = 0.14 N/mm^2
σ_{min} = 0.10 N/mm^2

3x2 module:
σ_{max} = 0.23 N/mm^2
σ_{min} = 0.17 N/mm^2

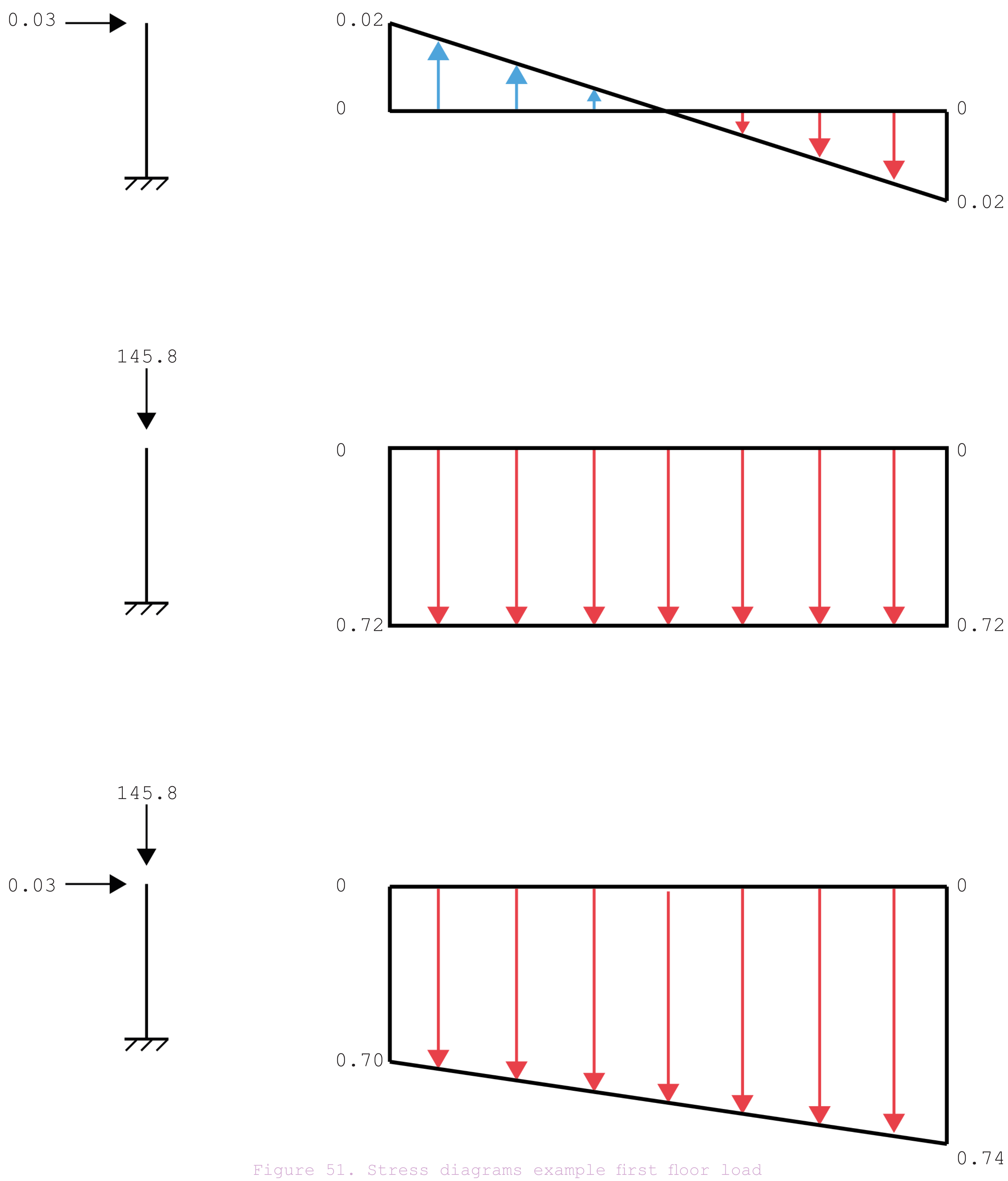


Figure 51. Stress diagrams example first floor load

Buckling

For buckling, it is important to know how what type of supports the structure contains. In this case, the assumption is made for a clamped bottom and freely movable top. This results in a k value of 2, seen in the diagram. The Youngs Modulus for is 110 N/mm^2. The following equation is used:

$$F_{cr} = (\pi^2 * E * I) / (k * L)^2$$

First floor module:
F_{cr} = 927.5 kN
F_{module} = 145.8 kN

First floor module:
F_{cr} = 40.3 kN
F_{module} = 23.9 kN

3x2 module:
F_{cr} = 927.5 kN
F_{module} = 40.4 kN

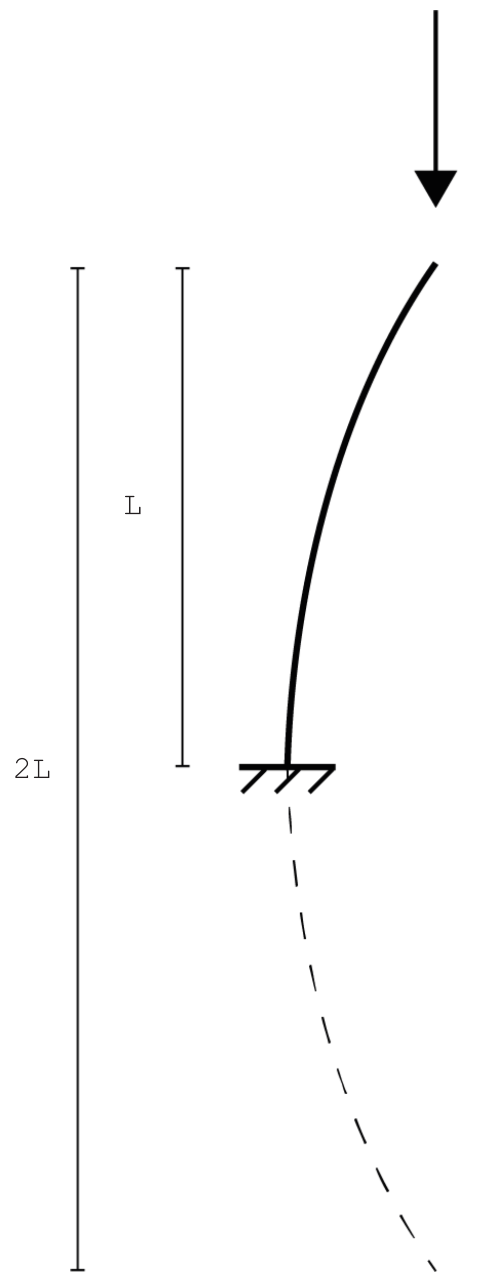
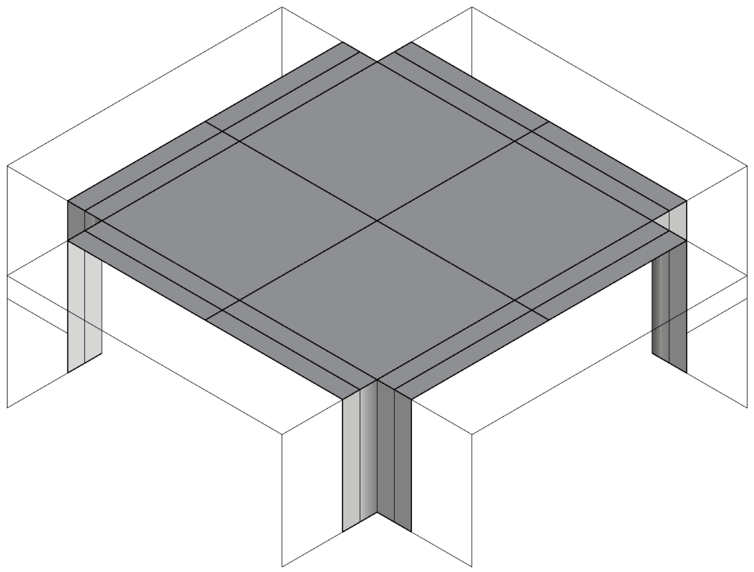
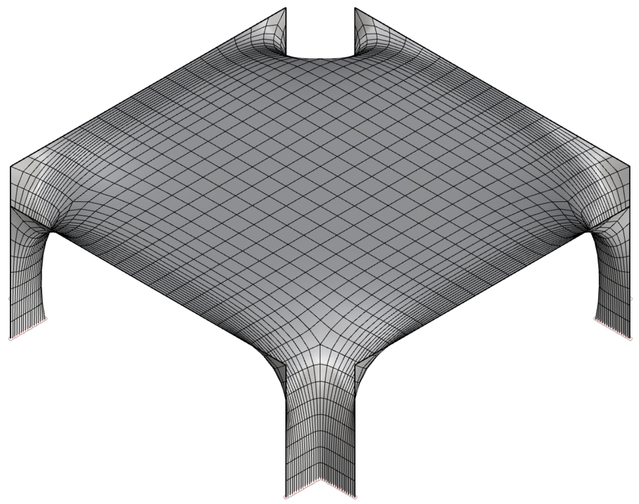


Figure 52. Used k value for buckling

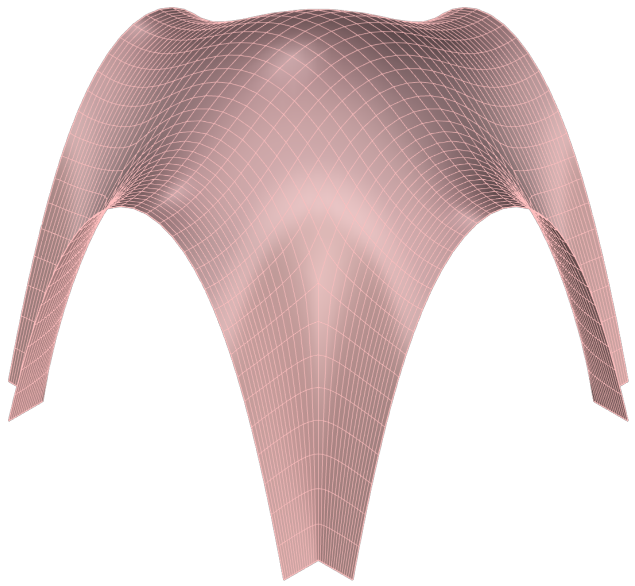
3.5. Final Tessellations



Main Tessellation

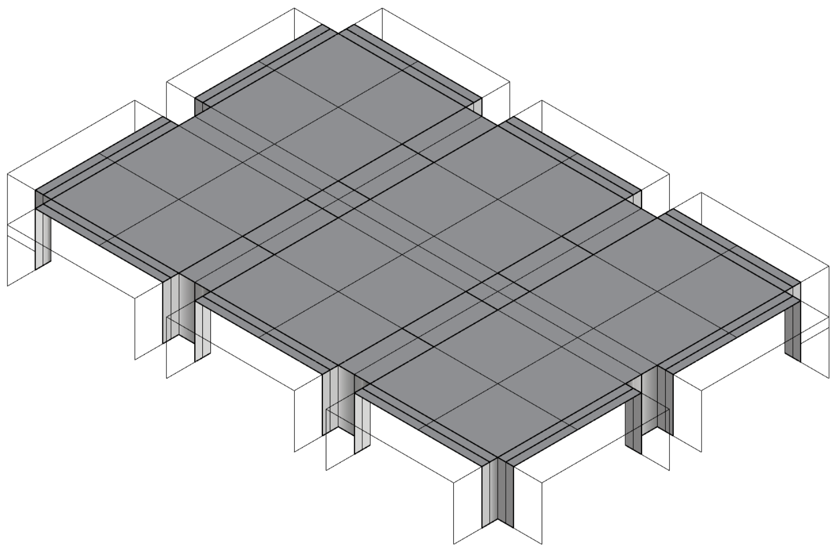


Quad Sub-division
Constant level - 3

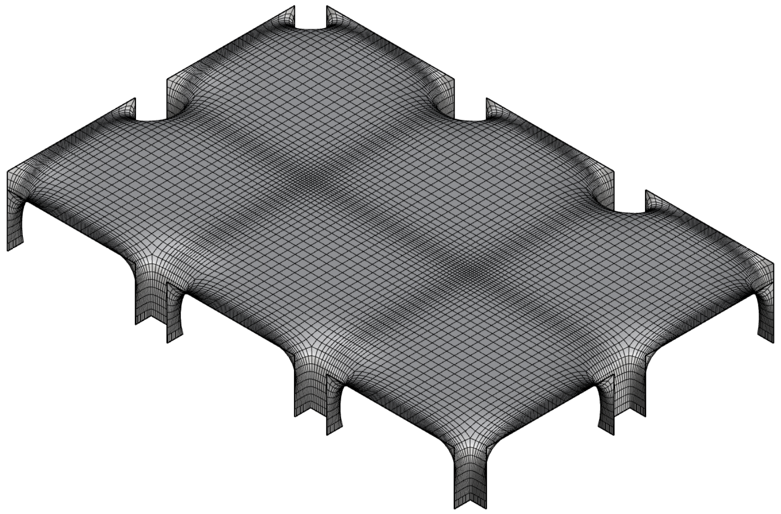


Dynamically Relaxed Mesh
Length factor - 0.5
Strength - 5
Vertex load - 0.05

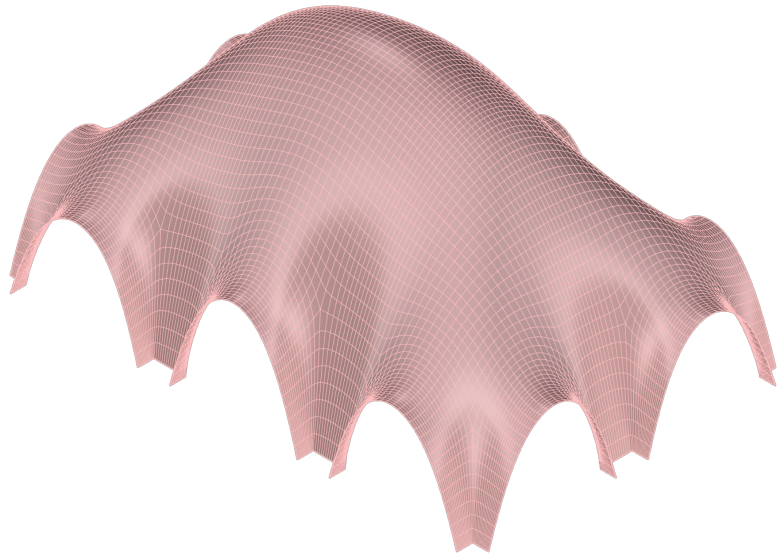
Figure 54. 1*1 Spatial Unit Final Tessellation



Main Tessellation



Quad Sub-division
Constant level - 3



Dynamically Relaxed Mesh
Length factor - 0.5
Strength - 7.5
Vertex load - 0.05

Figure 53. 3*2 Spatial Unit Final Tessellation

3.6. Structural Analysis

3.6.1. Karamba Set-Up

Input_material properties

The structural analysis was carried out on the finalised forms. We chose karamba plug-in within grasshopper for Finite Element Method Analysis.

The material properties discussed in brief previously are considered as inputs. These properties directly relate to the adobe block properties. Special attention is given to the units used in the software, eg. Young's modulus is in KN/cm2 whereas the density is given in KN/m3. The same applies to the output of stresses shown in KN/cm2 whereas the displacement is shown in cm.

Following are the material properties set as input for the FEM analysis,

Material Properties

Family	Earth blocks (adobe)
Young's modulus	11 KN/cm2
Density	15 KN/m3
Shell Thickness	0.3 m
Compressive Strength	-15.0e^-02 KN/cm2
Tensile Strength	1.5e^-02 KN/cm2

Load cases

Two different load cases were considered for the analysis. Since we have a first floor, we considered an additional loading scenario with the shell placed on top of another shell.

Load Case Scenario 1: Without First Floor
A normal mesh is analysed considering following loads (KN/m2),
_Wind Load: 0.2
_Snow Load: 0.02
_Live Load on roof: 2.0
_Live Load on First floor: 2.0
_Self Weight: 1.35

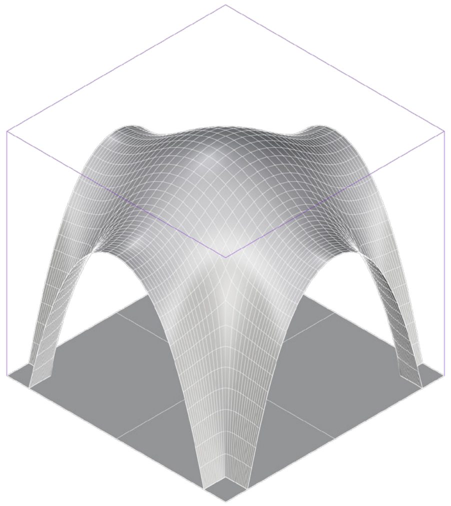


Figure 55. Load Case 1 (Without first floor)

Load Case Scenario 2: With First Floor
For this particular case, we need to consider additional roof infill load. To calculate this we add point load over the mesh surface. This is done by using specific script that first direct all the mesh vertices towards the apex point. Then the intersecting points over the relaxed surface is connected to the apex level in the form of line segments. These line segments are added together over the surface of the relaxed mesh and point load is added over them. This allows us to get the total loading for the roof and floor infill. (see fig. 12)

Further, we need to consider the dead load of the wall infills on the first floor. For this we again use a similar logic, but direct the ines from the base of first floor with the mesh opening segment. (see fig.13). For scripts check appendix.

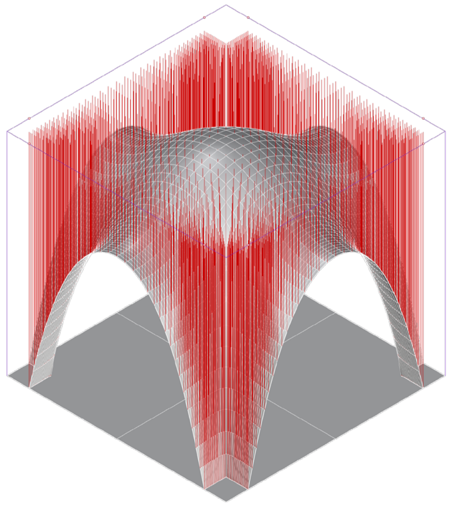


Figure 56. Load Case 2.1 (With floor infill)

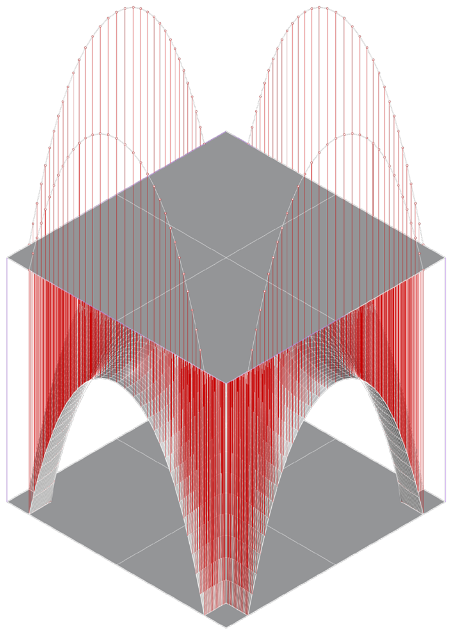


Figure 57. Load Case 2.2 (With Roof & wall infill)

3.6.2. Shell Analysis

Shell Analysis_1*1 Spatial Unit_Load case 1

Unit Size: 3.3m x 3.3m
Number of Supports: 4
Column Size: 0.45m x 0.45m
Apex height: 3.9m
Max tensile: -15 e^-02 kN/cm2
Max compression: +1.5 e^-02 kN/cm2
Max displacement: 1.65cm (span/200)

Discussion of Result:

The maximum compression is -2.28e^-02 kN/cm2 and the maximum tension is +2.49e^-03. This means the shell is within the max allowable stress.

The maximum displacement is 2.32e^-01 (0.85) cm for 100% deformation.

Thus, the shell is stable under load case 1.

*Check appendix for the result sheets.

Figure 61. FEM Analysis_Displacement

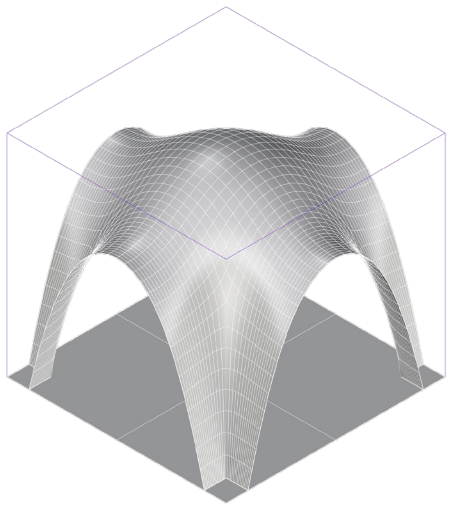


Figure 58. Load Case 1 (Without first floor)

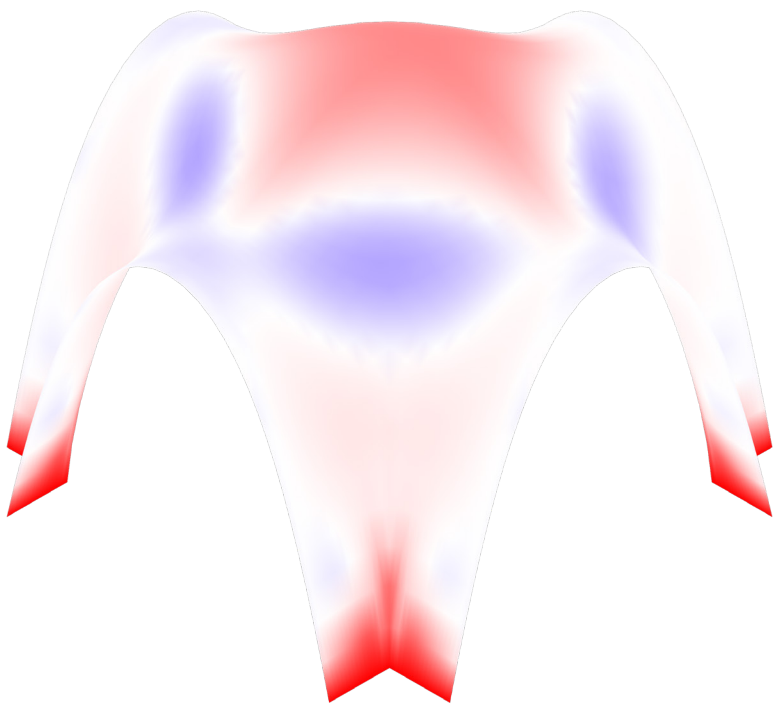


Figure 59. FEM Analysis_Tension

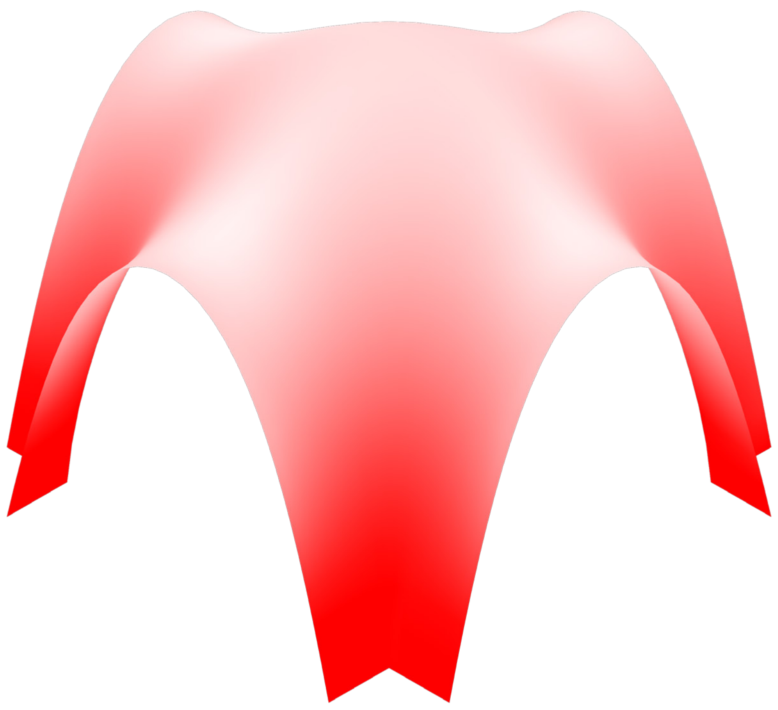


Figure 60. FEM Analysis_Compression

Shell Analysis_1*1 Spatial Unit_Load case 2

Unit Size: 3.3m x 3.3m
Number of Supports: 4
Column Size: 0.45m x 0.45m
Apex height: 3.9m
Max tensile: -15 e^-02 kN/cm2
Max compression: +1.5 e^-02 kN/cm2
Max displacement: 1.65cm (span/200)

Discussion of Result:

The maximum compression is -7.38e^-02 kN/cm2 and the maximum tension is +3.44e^-03. This means the shell is within the max allowable stress.

The maximum displacement is 1.90e^-01 (0.69) cm for 100% deformation.

Thus, the shell is stable under load case 2.

*Check appendix for the result sheets.

Figure 65. FEM Analysis_Displacement

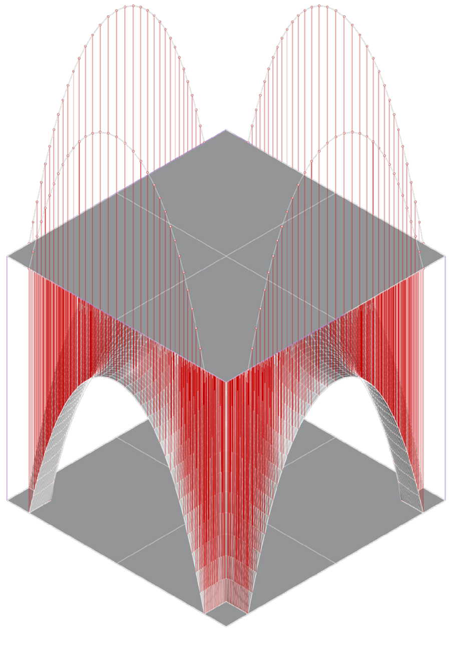


Figure 62. Load Case 2 (With first floor)

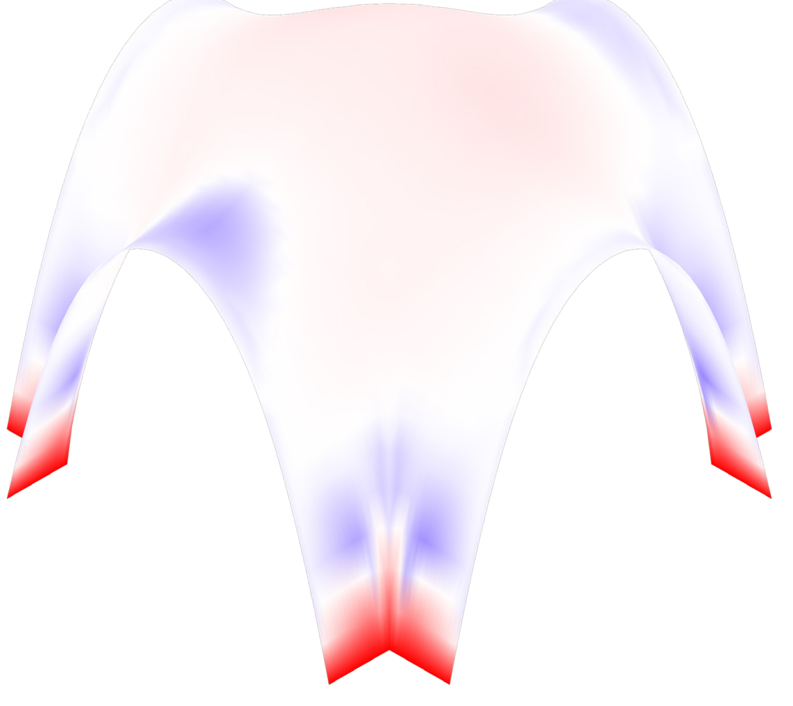


Figure 63. FEM Analysis_Tension

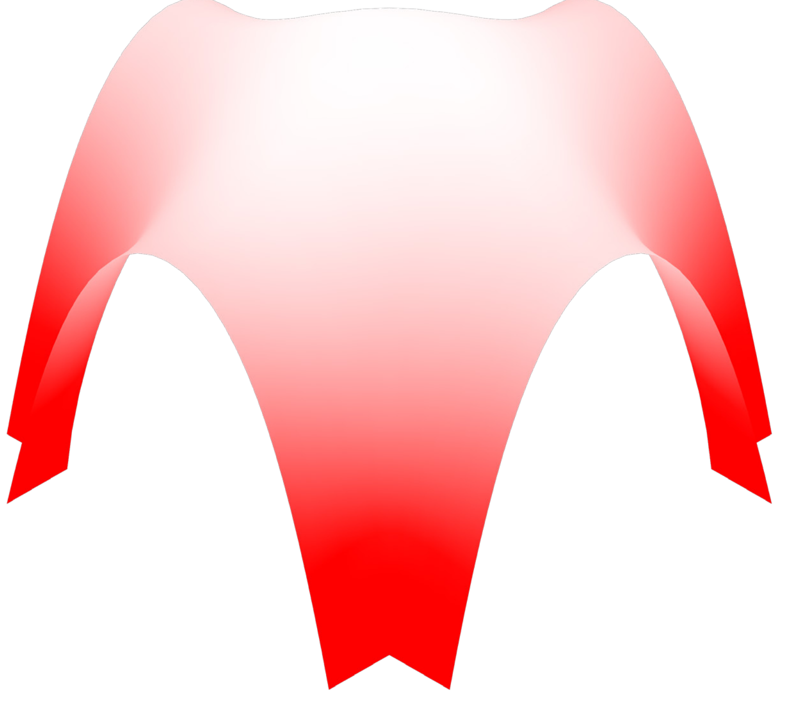


Figure 64. FEM Analysis_Compression

Shell Analysis_3*2 Spatial Unit_Load case 1

Unit Size: 9.9m x 6.6m
Number of Supports: 16
Column Size: 0.45m x 0.45m
Apex height: 6.5m
Max tensile: -15 e^-02 kN/cm2
Max compression: +1.5 e^-02 kN/cm2
Max displacement: 4.95cm (span/200)

Discussion of Result:

The maximum compression is -4.12e^-02 kN/cm2 and the maximum tension is +1.38e^-02. This means the shell is within the max allowable stress.

The maximum displacement is 7.04e^-01 (2.58) cm for 100% deformation. This is still a valid number for best loading scenario.

Thus, the shell is stable under load case 1.

*Check appendix for the result sheets.

Figure 69. FEM Analysis_Displacement

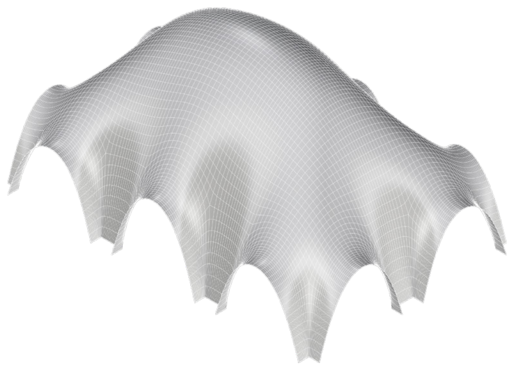


Figure 66. Load Case 1 (Without first floor)

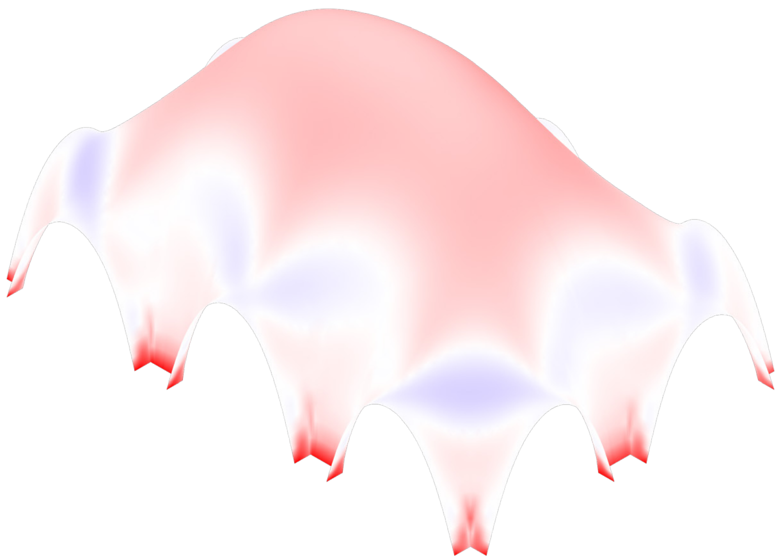


Figure 67. FEM Analysis_Tension

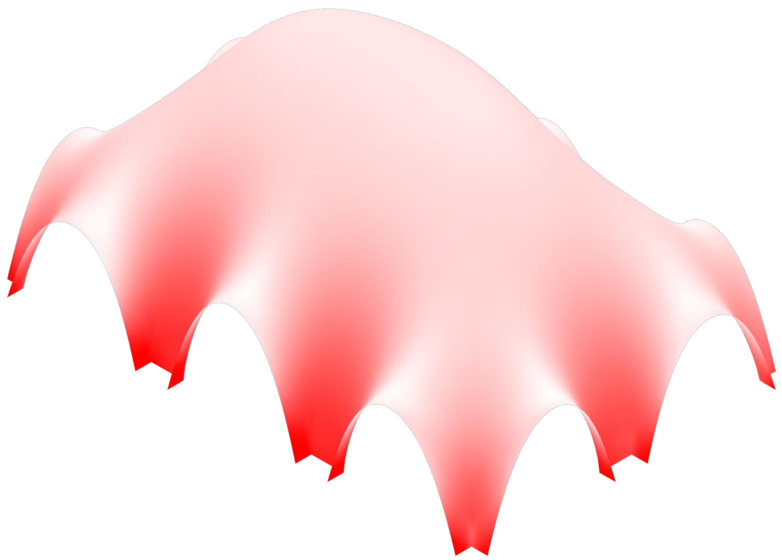


Figure 68. FEM Analysis_Compression

3.7. Staircase

3.7.1. Shaping

A key element to move from the 2D to the 3D scale, are the stair modules. The stairs are being placed in the locations determined during the configuration process. The staircase is determined in a 2*2 module size based on the height it needs to reach.

Considerations:
1. Module size: 2*2 (with column inside)
2. Subdivision: 16 parts
3. Tread: 0.30m
4. Riser: 0.15m
5. Maximum risers in one flight: 10
6. Minimum size of mid-landing: 1*1 subdivision
7. Minimum Height: 3.9m (From the roofing)

With these initial parameters we start to shape the stairs based on the entry and exit points on different levels. These points define the stair shape and its relation to the existing or future configuration.

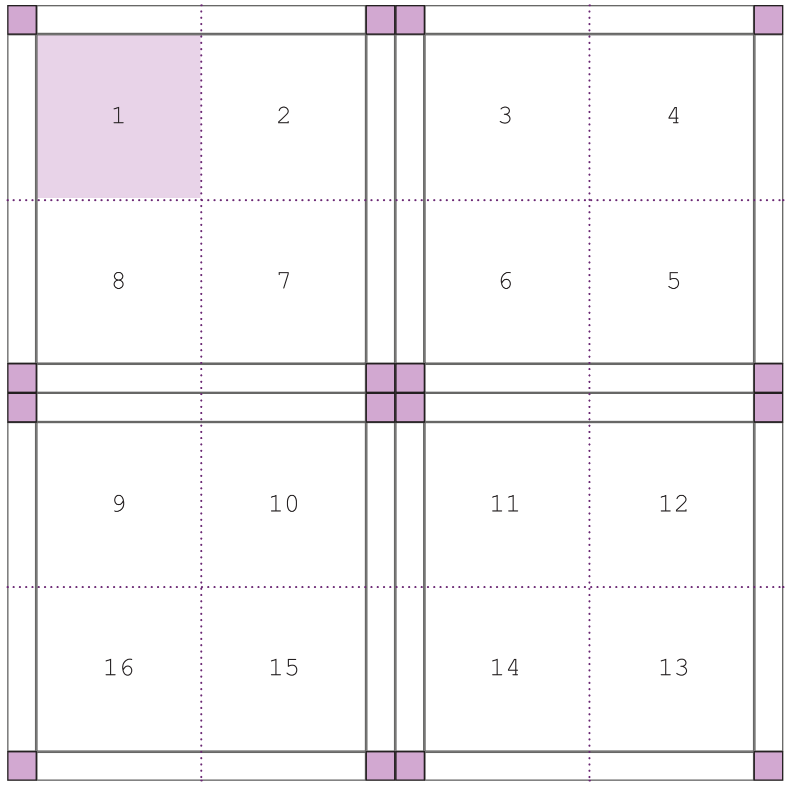


Figure 71. Staircase Module principle

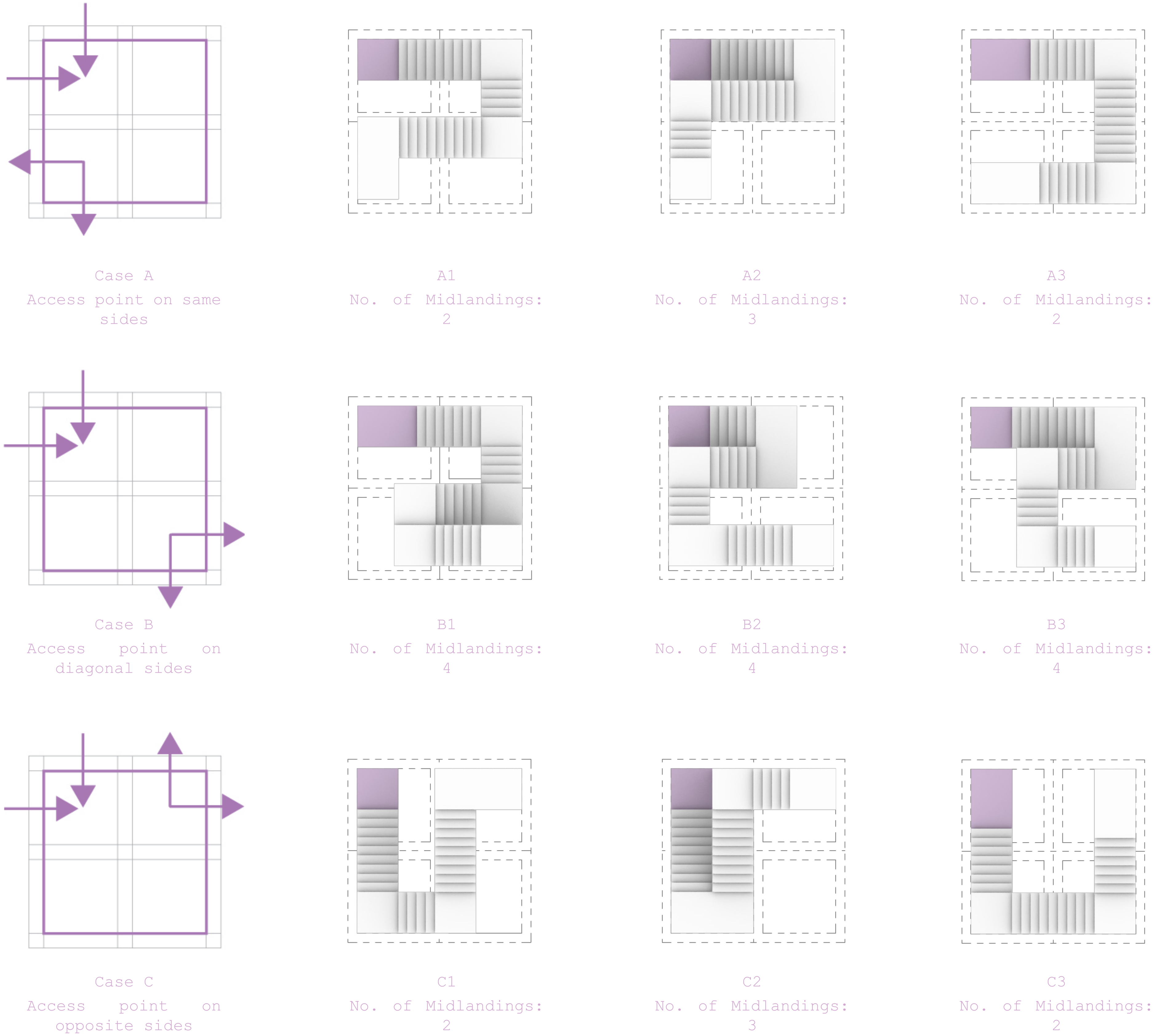
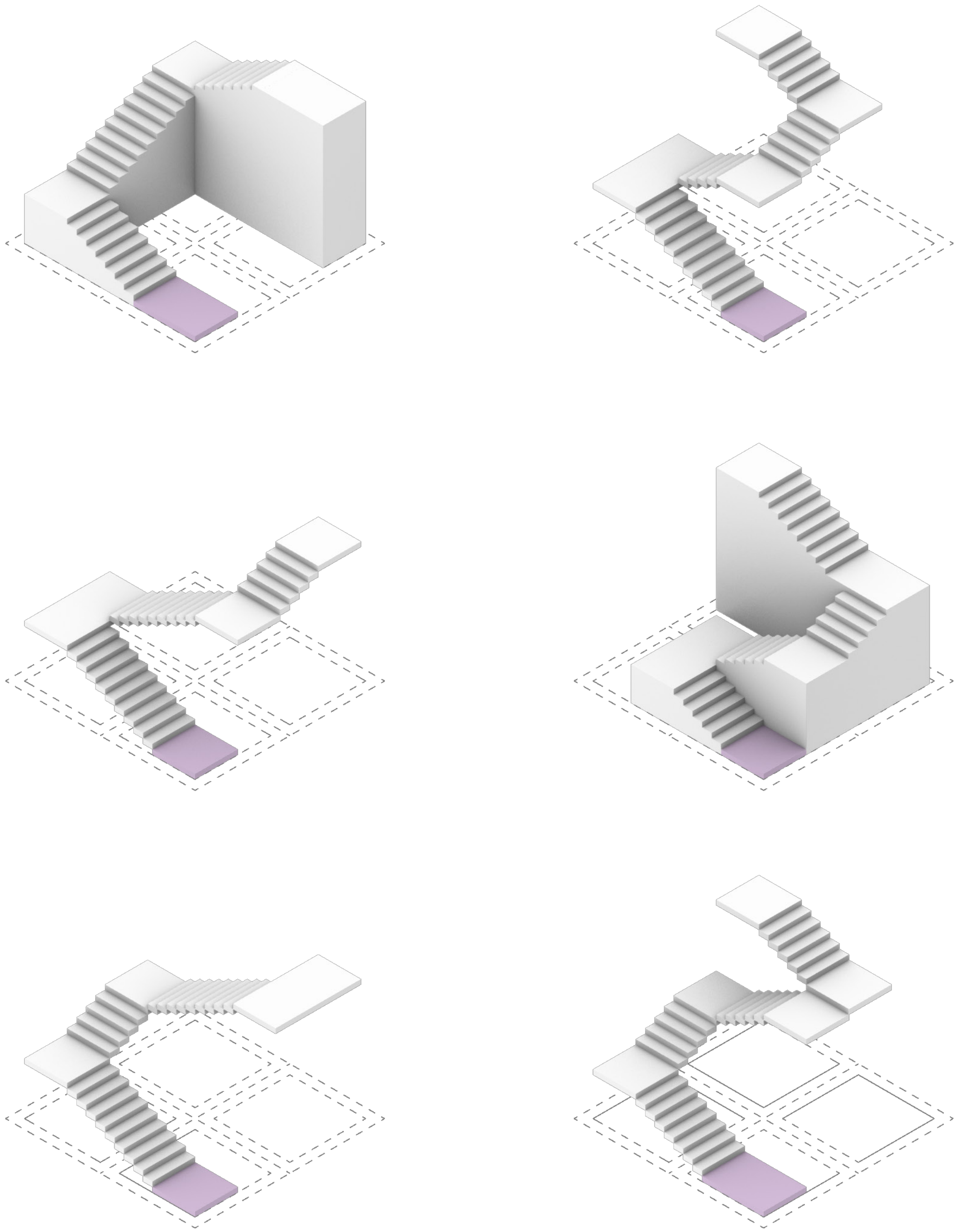


Figure 70. Variations of Stairs based on access points on various levels



Selected Stair Module for structuring

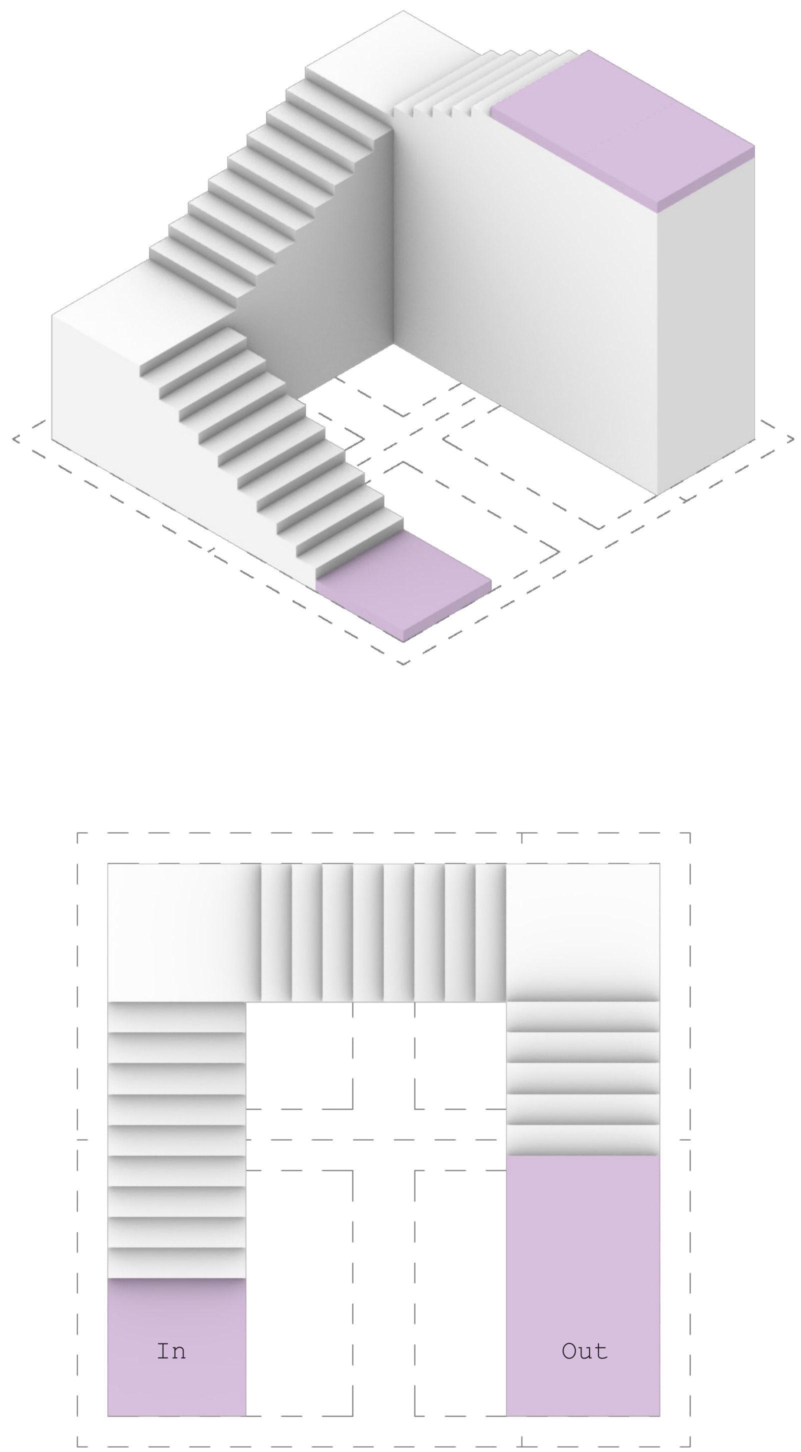


Figure 72. Variations of Stairs based on access points on various levels

Figure 73. Selected option

3.7.2. Structuring

The finalised shell of 1*1 spatial unit is used as the shaping element for staircase forming. The 2 shells are placed below the landing and take the structural support. The staircase flights are filled with sand with walls on the periphery. The structural columns are around the periphery and take care of the vertical forces. These columns are already checked for buckling.

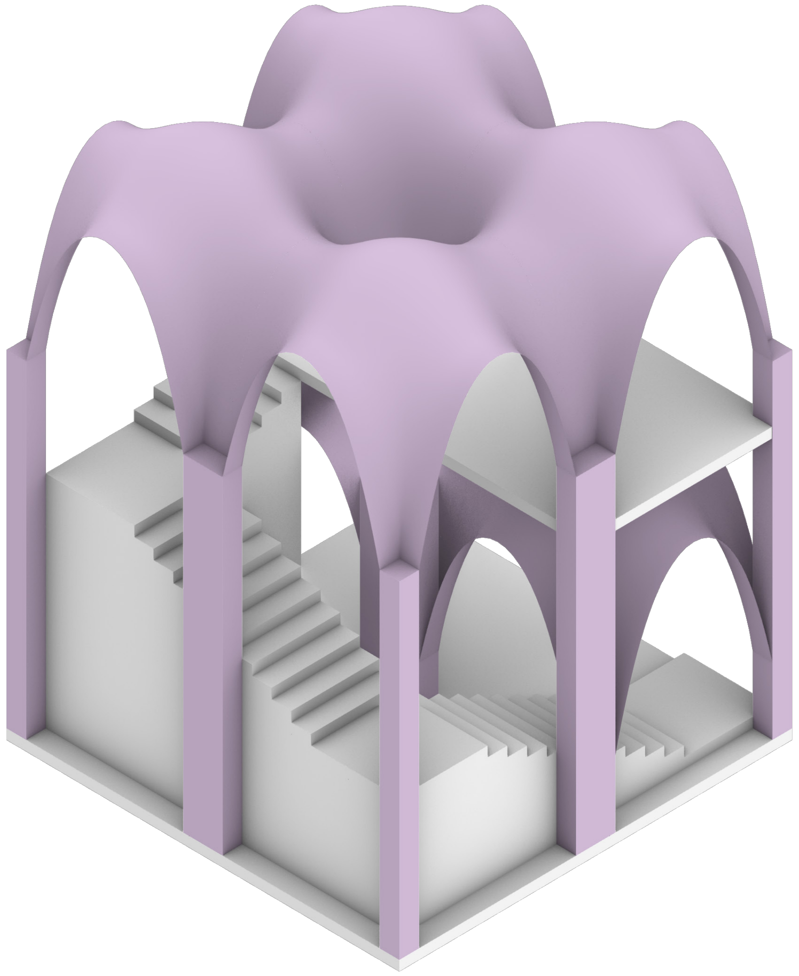


Figure 75. Final staircase module

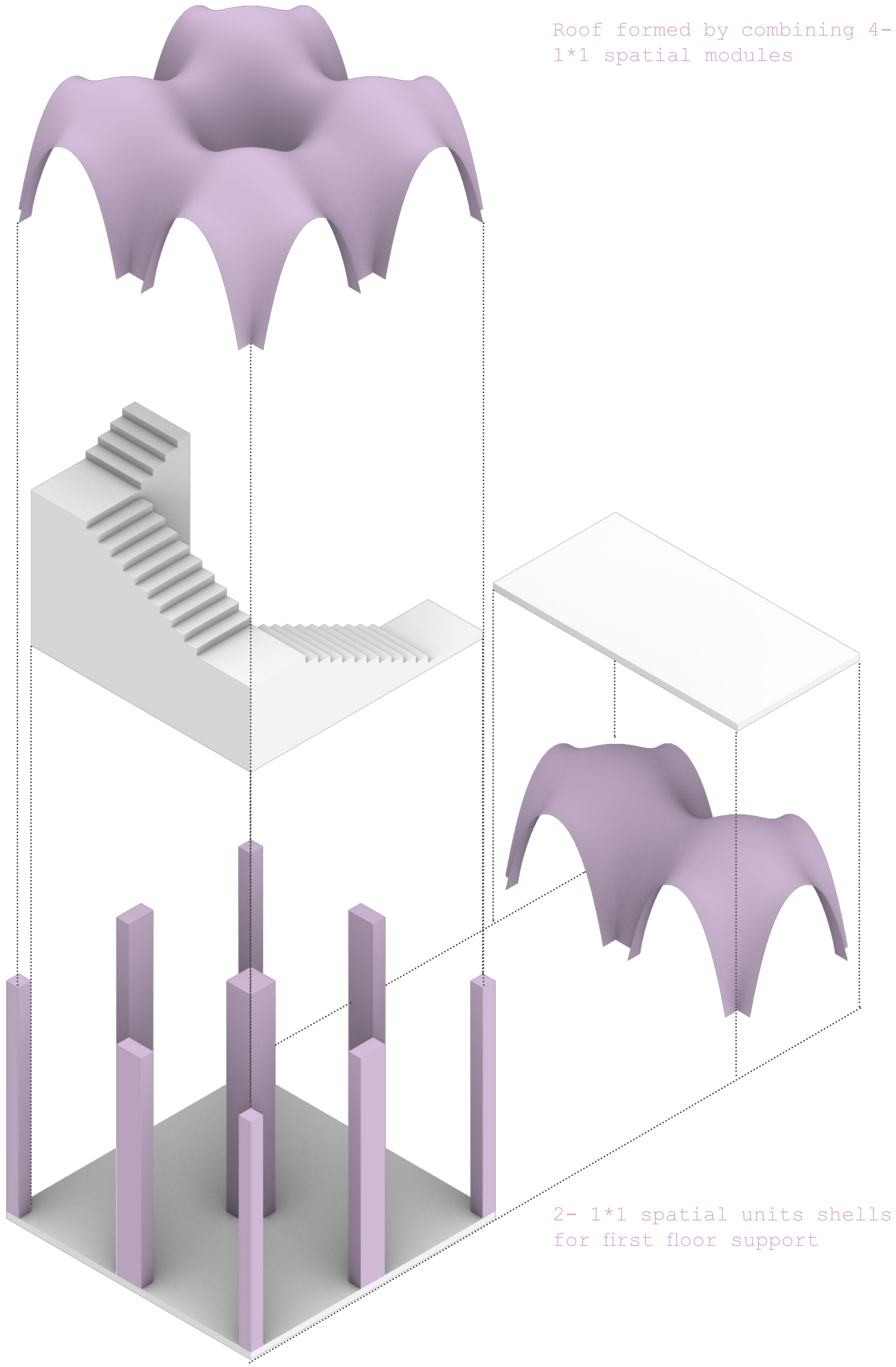


Figure 74. Exploded Axonometric View_Staircase

Figure 76. Optional Roof configuration (Not selected for detailing)

3.8. Wall Modules

Now that the shells have found their most optimal shape, the openings have to be defined. This is done by taking the smallest possible opening size of the meshes. This is on the 3x2 module and is shown in grey. For that opening a closed wall, a wall with a ventilation gap, a window wall and a door wall are designed.

However, to use this wall type on every possible location, an addional layer (purple) is added if the wall infill is placed on a 1x1 module. The wall tpyes W-01, W02, W-03 and W-04 are now done and can be placed where the designer wants to.

There is one other possibily on the wall types that is W-00. This means that there is no wall infill on that place. This is mainly used on places where the opening needs to be the 2.4m width which is the standard dimension for circulation functions in hospitals (corridors).

Figure 77. Wall types

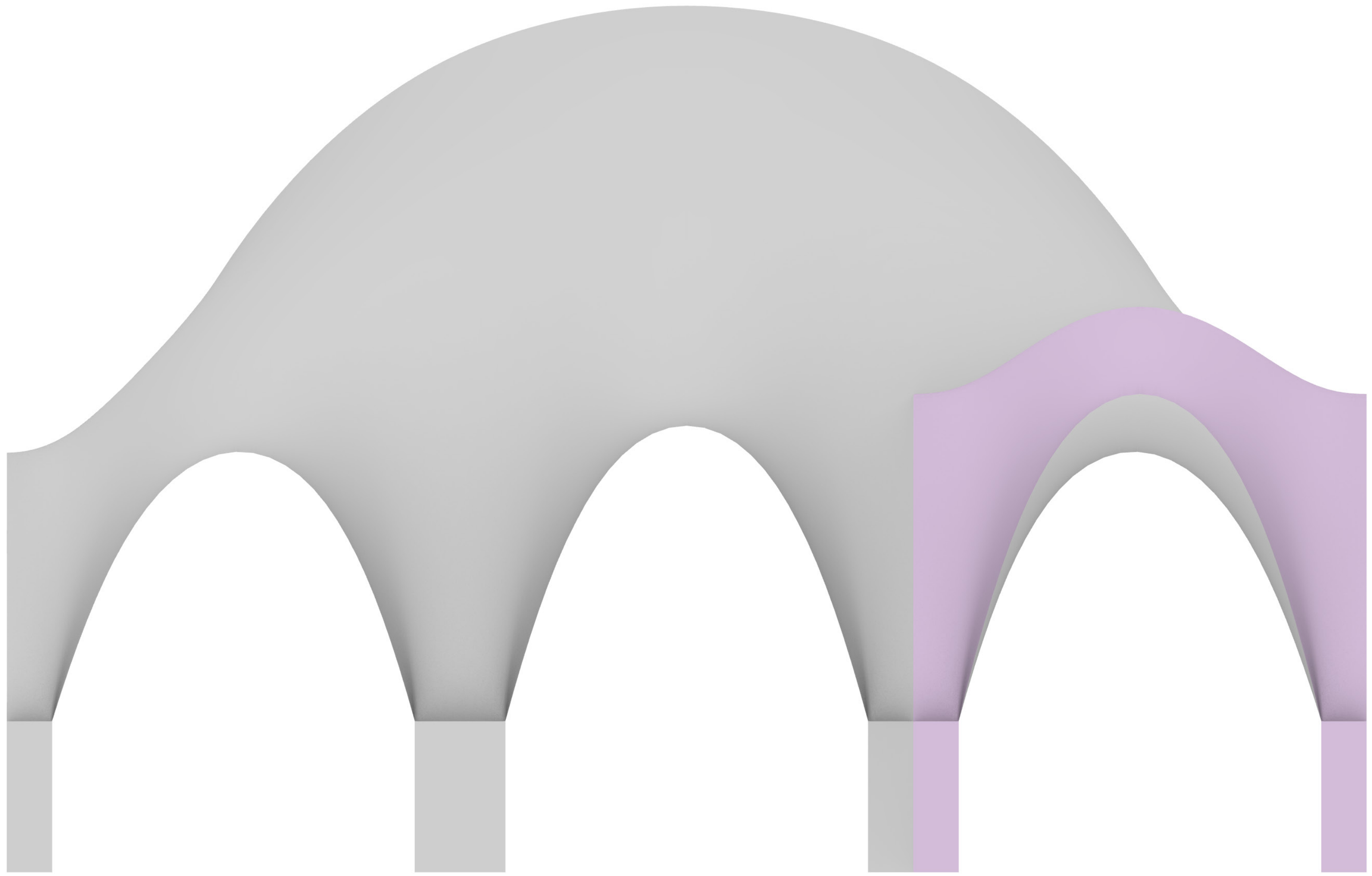


Figure 78. Side view of 3x2 and 1x1 module

4. Construction

4.1. Process

4.1.1. Brick Construction

After the tessilations were done, a way to build the mesh had to be designed. Since the team was inspired by the Gothic architecture, possibilities were tought of to make the sturcture with columns and arches to support the rest of the roof infill.

This was done by cutting the mesh with planes and generate brick like earth blocks. After these arches were made on the mesh, the infill in between could be layed using a simple English brick layering bond.

However, the team quickly felt that this was not the best way to continue, since this technique is perhaps a bit too simple for this course and was not directly interesting to design with computational design. Therefore, a block layering appoach on the next page was tried.

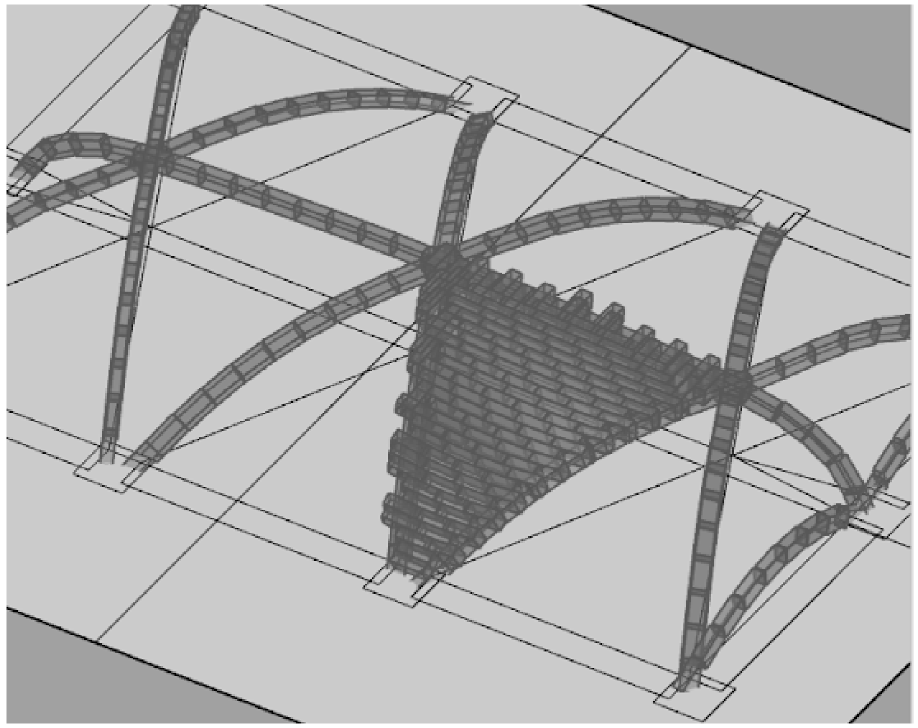


Figure 79. Bricks along the arches and infill

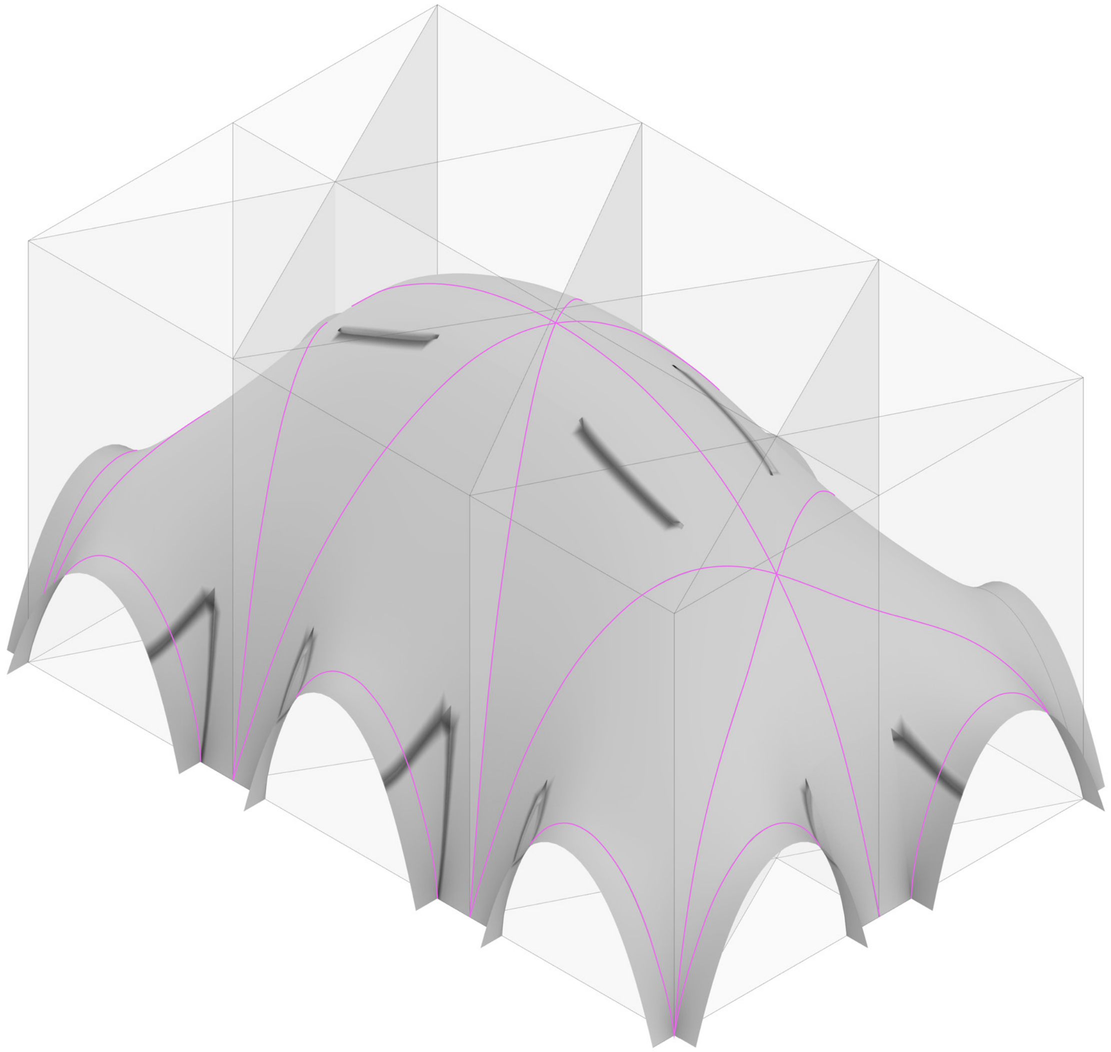


Figure 80. Lines to design arches on

4.1.2. Block Construction

With the material speficied in 3.2.2, the blocks can be made as adobe blocks. To keep the basic block to a minimal amount and the sizes small for a shorter drying time, only two different blocks have to be producted. This is the 300x300x100mm block and the 300x300x150mm block. However, together they can make a total of 5 different block heights. This results in block of 300x300mm area and a height of 100, 150, 200, 300 or 500mm.

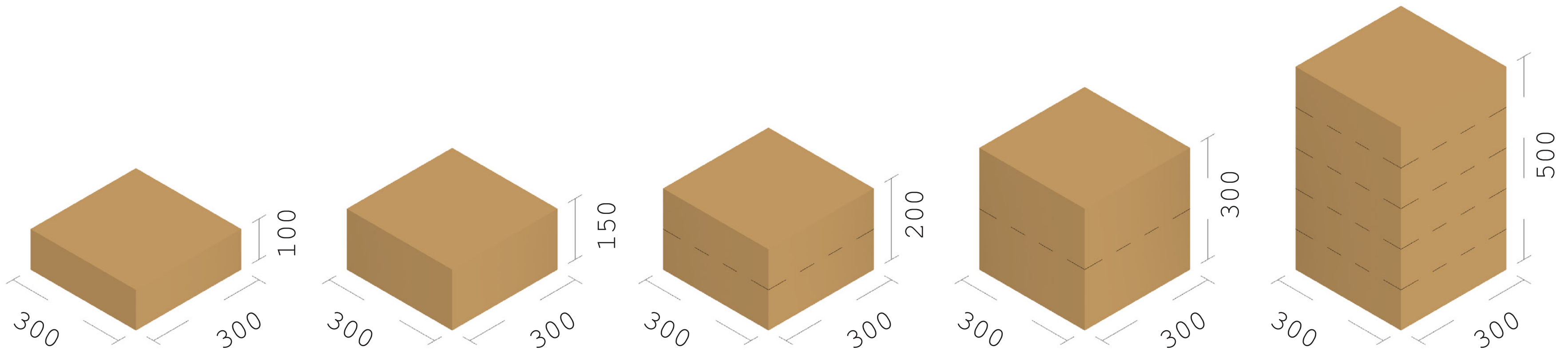


Figure 81. Block sizes in mm

Clustering and Intersection

As a next step, the blocks are stacked on top of each other with an overlap of one quarter, seen in the figure. At the same time, the blocks are placed in the x and y-axis so the total becomes one massive block of blocks. The biggest blocks are in the bottom and the smaller once on the top. It starts with one layer of block of 500mm. Then a layer of 300mm and a layer of 200mm. Next come two layers of 150mm and then a certain amount of layers of 100mm. This should be at least as high as the meshes that were found in the forming and shaping process.

Now that the meshes fit into the big block of blocks, the intersections can be found between the blocks and the meshes. The blocks that intersect are the blocks that are kept and will form the voxelised mesh. Note that it is important to always check if all the voxels are resting on a voxel below. If that is not the case, the mesh is not sufficient and should be reconsidered or the blocks should be stacked in a different order.

This is done for the 1x1 mesh and the 3x2 mesh of which the voxelised elements can be seen on the next page. There, also the columns are added below the elements.

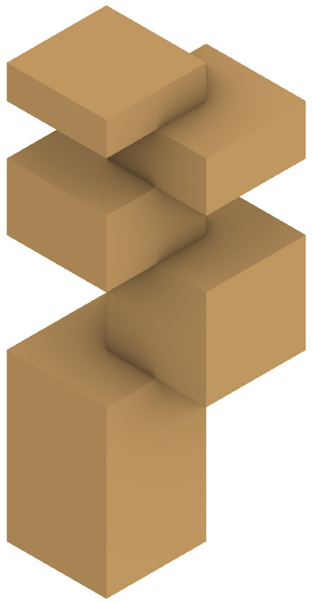


Figure 82. Stacked blocks

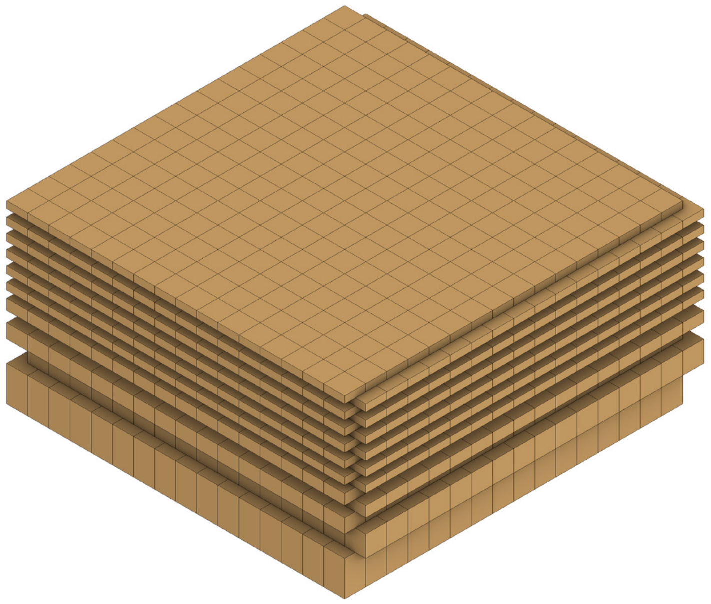


Figure 84. Big block of blocks

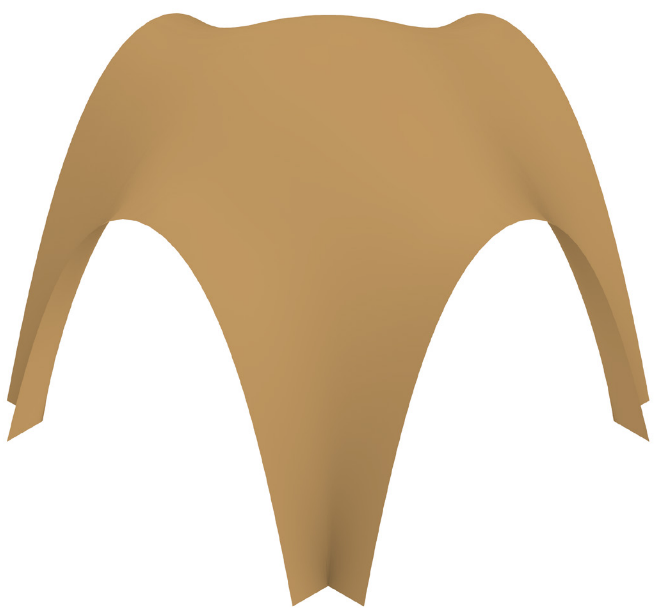


Figure 83. 1x1 mesh

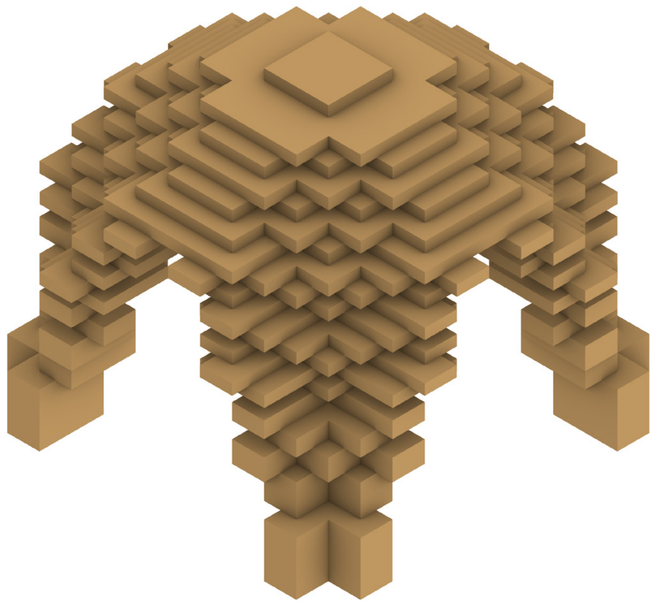


Figure 85. 1x1 voxelised

Staircase and Walls

The staircase module consists of 1x1 modules. Two of those modules are place on the ground floor and will carry the platform for the first floor. Since one of the modules has now a free space on the ground floor, this can be used as an extra storage space. The roof will be supported by those two modules, but also with columns that go down to the ground. (The structural analysis is found in the manual calculation.)The steps of the staircase have a sand filling underneath them for support.

For the walls, type W-01, W-02, W-03 and W-04 are made with the same voxels sizes, corresponding to their layer height. In order to match every wall type in all the modules, the highlighted blocks are added if the wall needs to fit a 1x1 module.

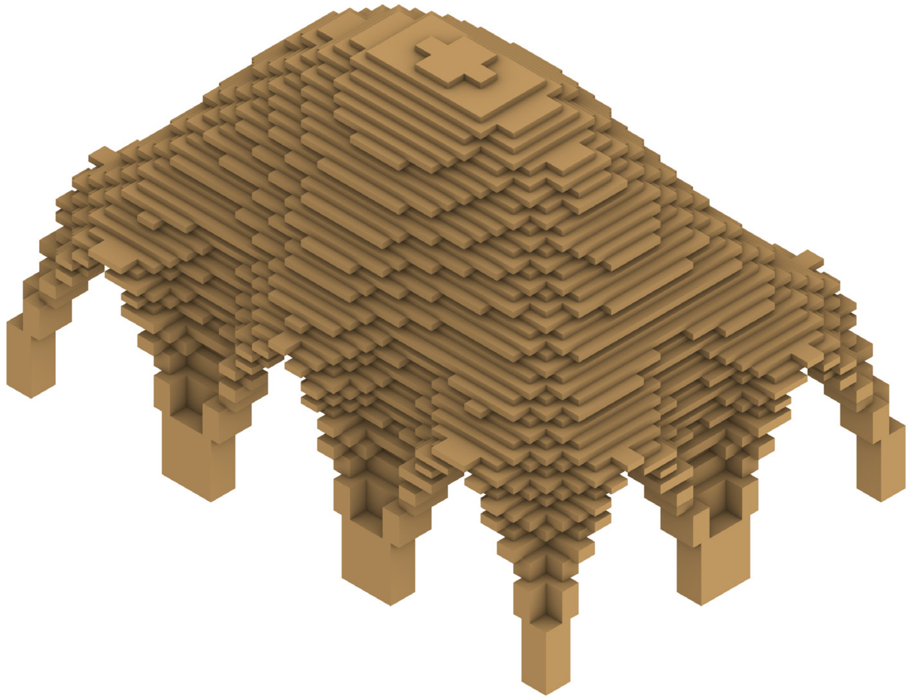


Figure 86. 3x2 module

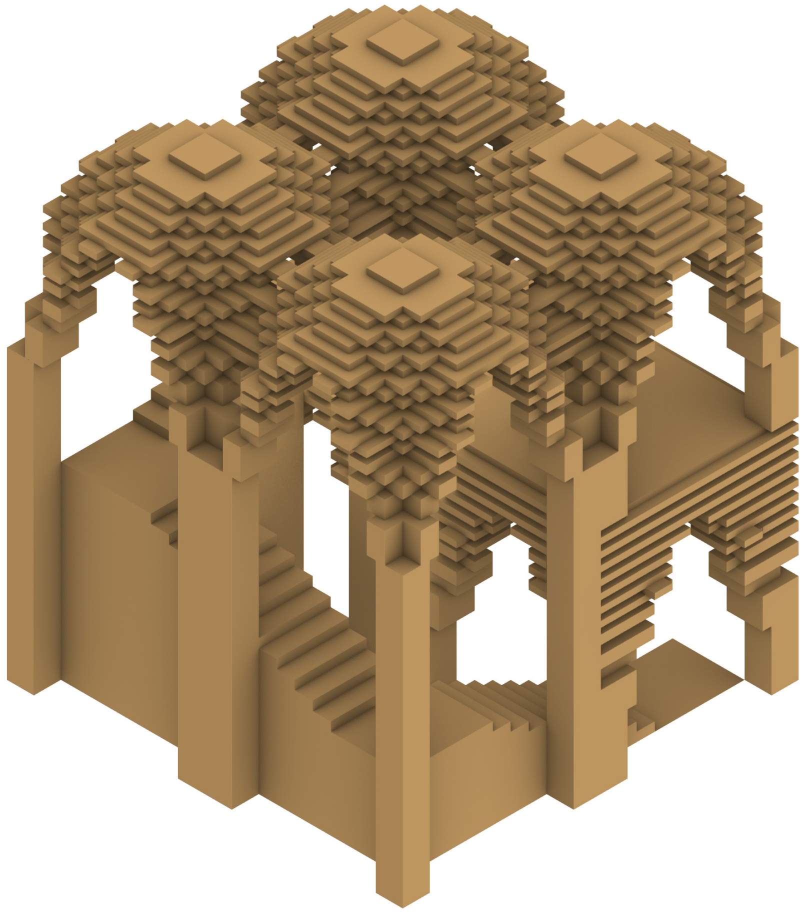


Figure 87. Staircase module

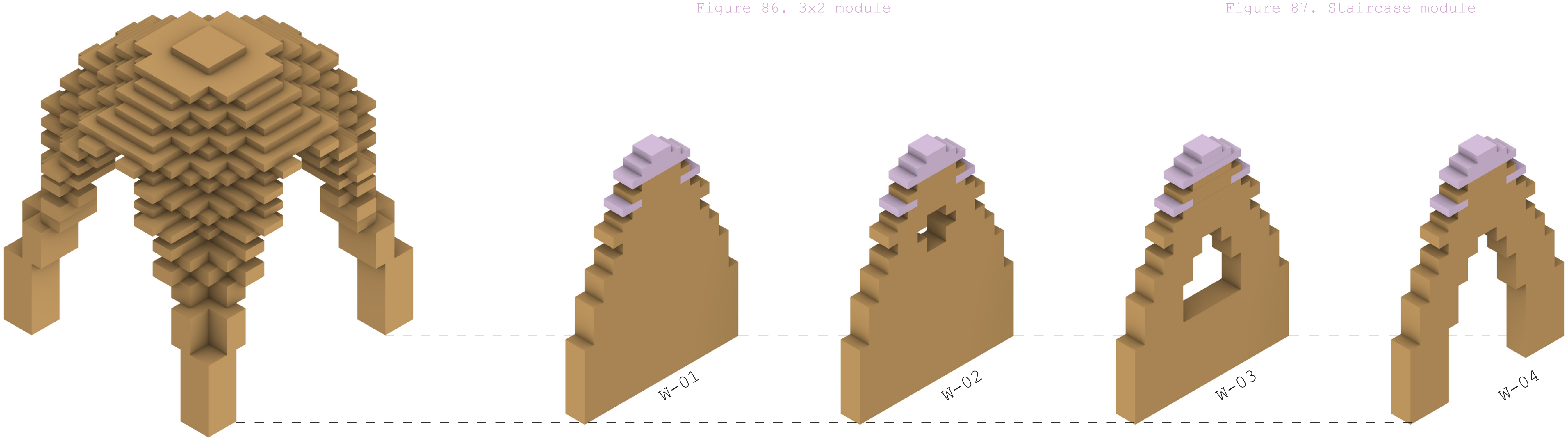


Figure 88. 1x1 module with wall types

Structural Analysis

At first a structural analysis on the mesh was done, but the voxalised mesh will work a bit different. To do a structural analysis on the voxels, the center point of the voxels are taken. The next step is to see which voxels are next to each other and to connect those center points. Based on this truss model, the lines where made into square beams with a cross section of 15x15mm since the voxels overlay one quarter on top of each other.

Sadly, once the model was run by Karamba, the structure did not succeed with the applied forces (maximums of -1.50 and 0.15 N/mm^2). In the elevation, there is a clear area with more stresses. As a next step, twelve blocks were manually added to see if this would help, see the most upper right figure. The structure did improve, but not enough to be a good stable structure. The team thinks this is fixable if the coarse was more ECTS.

The results of the first module are:
Compression = -1.61 N/mm^2
Tensile = 0.86 N/mm^2

The results of the second module are:
Compression = -1.42 N/mm^2
Tensile = 0.61 N/mm^2

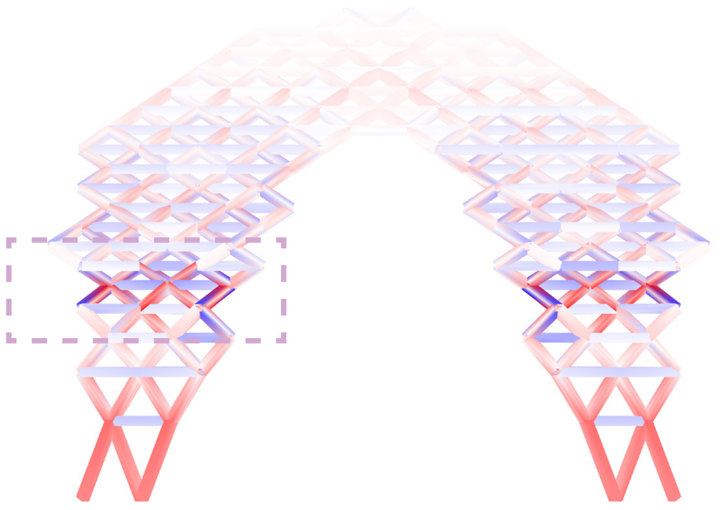


Figure 97. Elevation of stress analysis

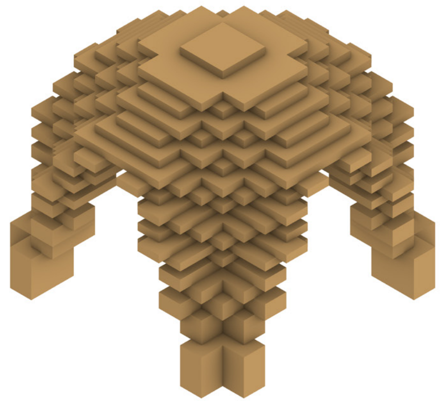


Figure 89. Voxelised 1x1 module

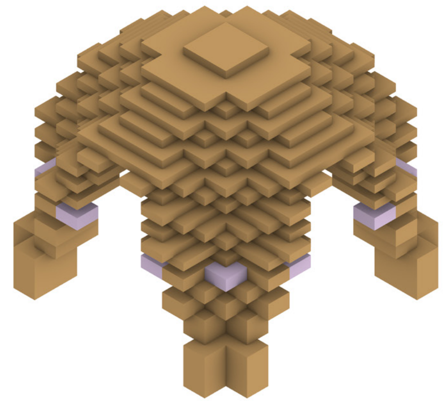


Figure 90. Voxelised 1x1 module with extra voxels

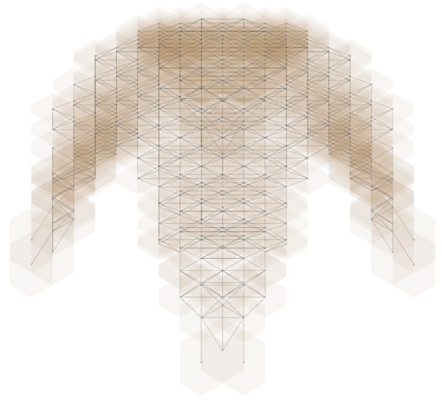


Figure 91. Points and lines of the voxels

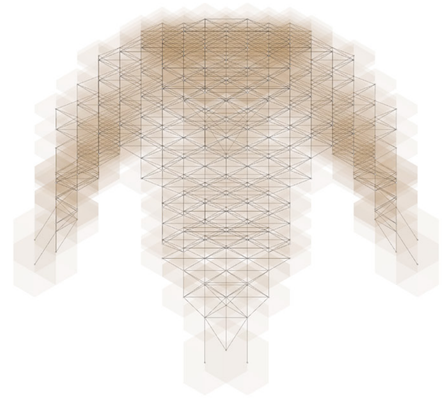


Figure 92. Points and lines with extra voxels

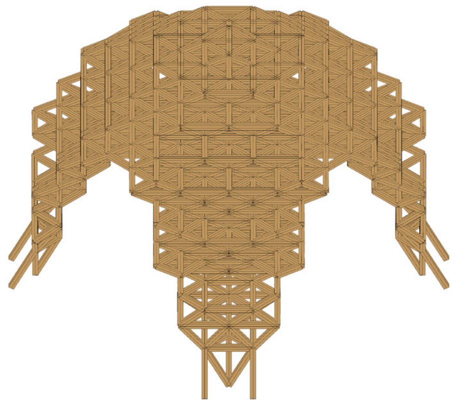


Figure 93. Beams representing the truss model

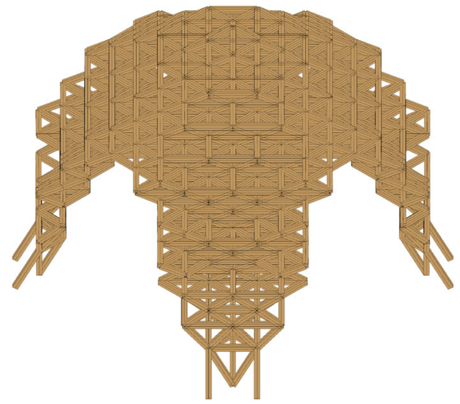


Figure 94. Beams including extra voxels

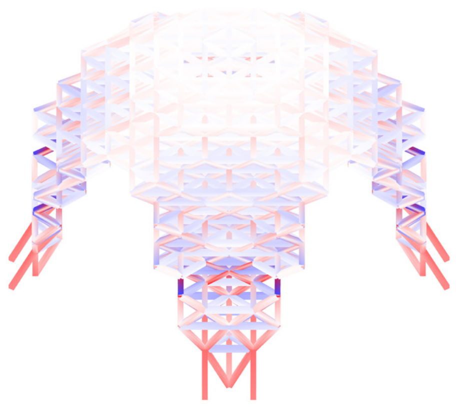


Figure 95. Stress analysis

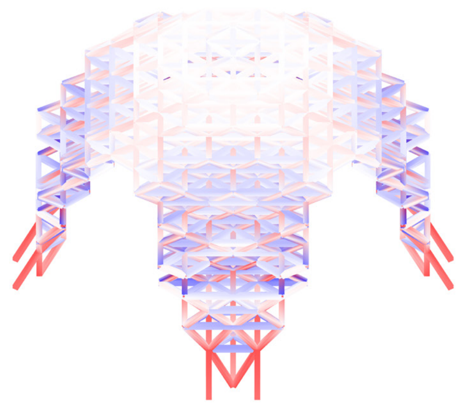


Figure 96. Stress analysis with extra voxels

4.1.3. Construction

Now that the final elements are given, the construction order is considered. The first step is to dug out the location of the funding of the module and to fill it with cement. Once that is dried, the top layer can be made ready and become an outline for the next steps.

Before the set up of the scaffolding can start, it is important to check which walls are used in the module. If wall type W-00 is used, there is also scaffolding needed on the wall opening. Therefore, two types of scaffolding are defined. The type S-01 series for the diagonals and S-02 series in case of a W-00 wall type. In the example wall type W-00, W-01, W-02 and W-04 are used.

After the placement of the scaffolding, the block layering can begin and in this process, the wall infills will be placed together with the rest of the layers. When this is coletely done, the scaffoling can be removed and the final module is left.

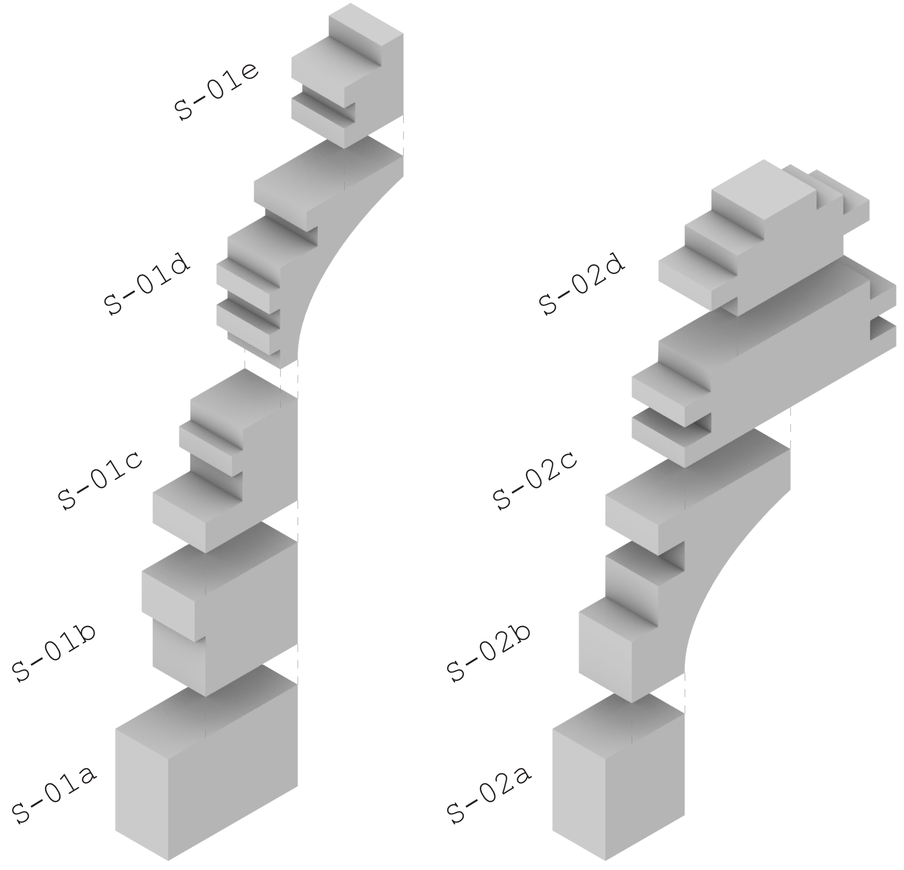


Figure 98. Scaffolding series S-01 and S-02

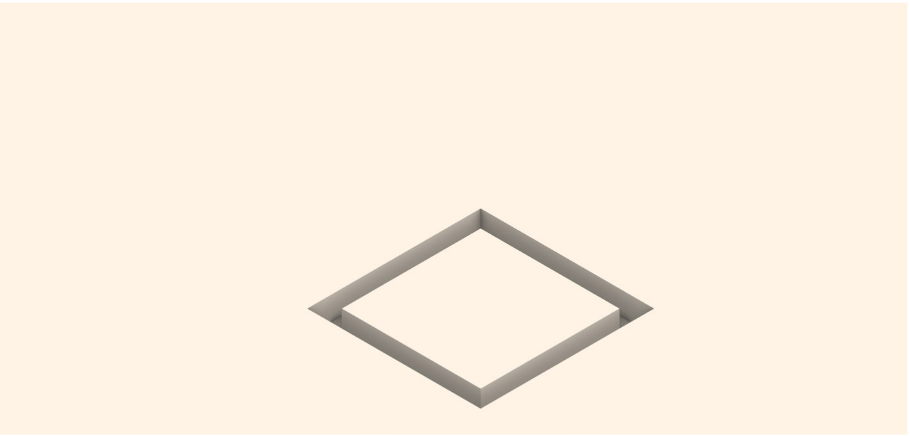


Figure 99. Digging location for foundation

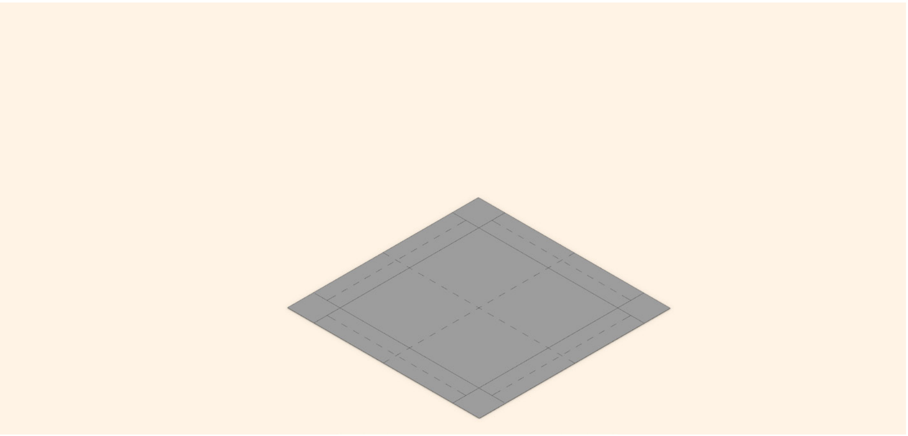


Figure 100. Foundation and smooth surface

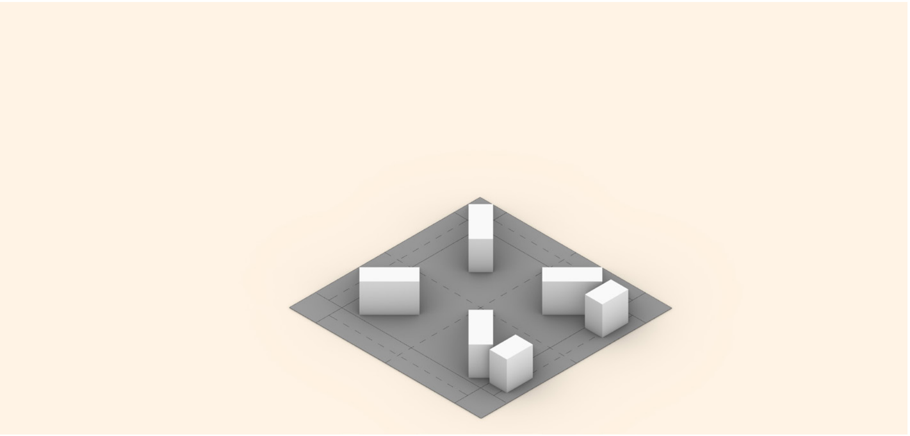


Figure 101. First scaffolding elements

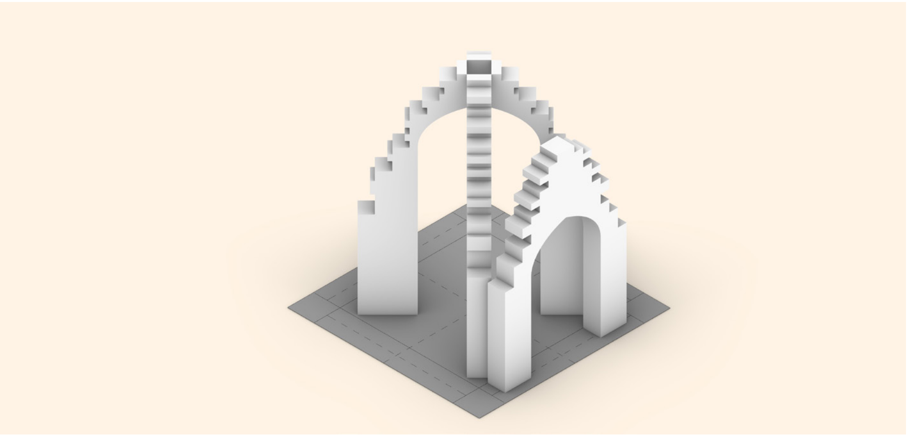


Figure 102. Final scaffolding elements

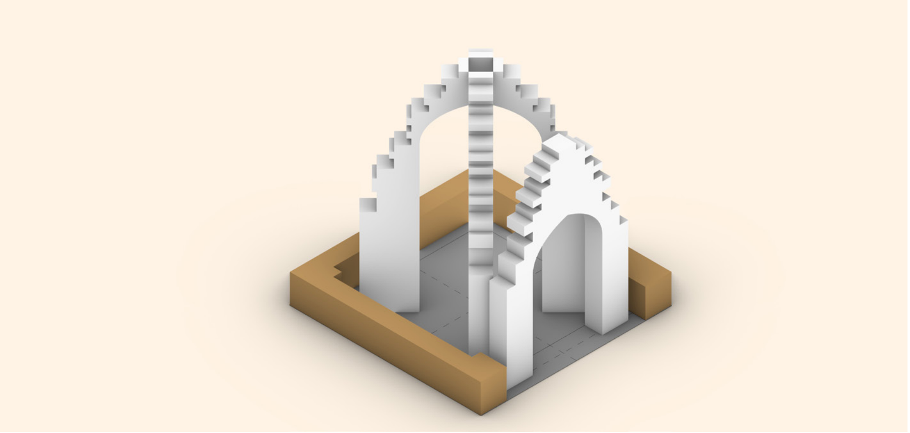


Figure 103. First layer of earth blocks

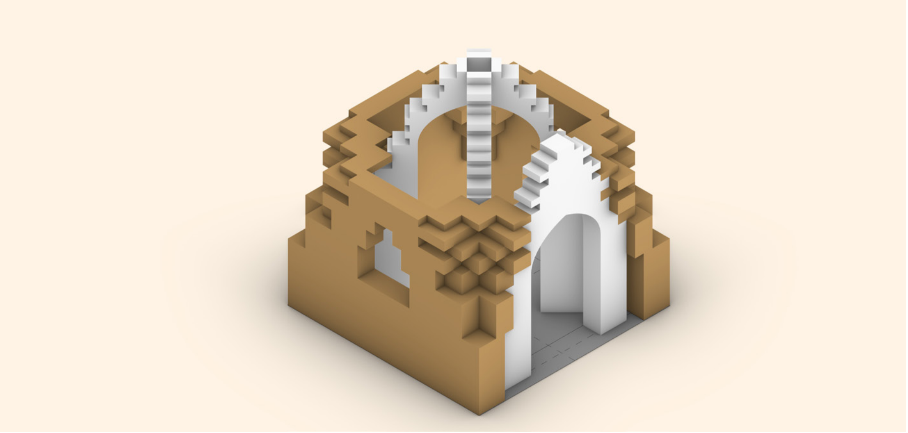


Figure 104. Halfway in the block layering

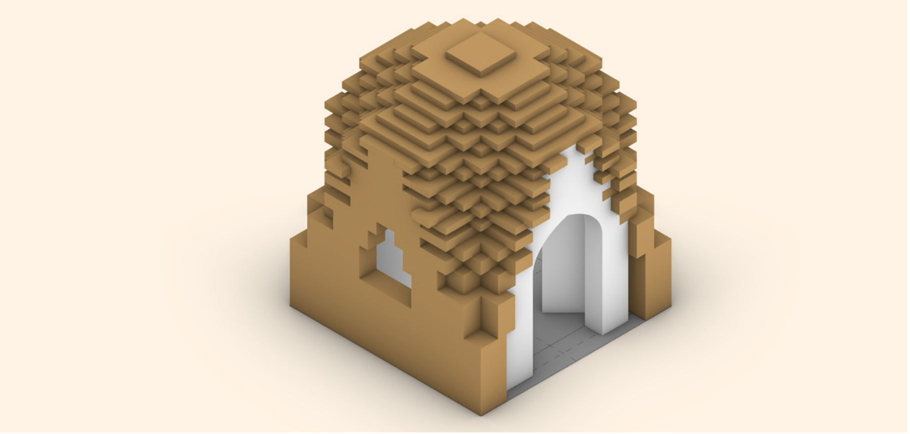


Figure 105. Block layering complete

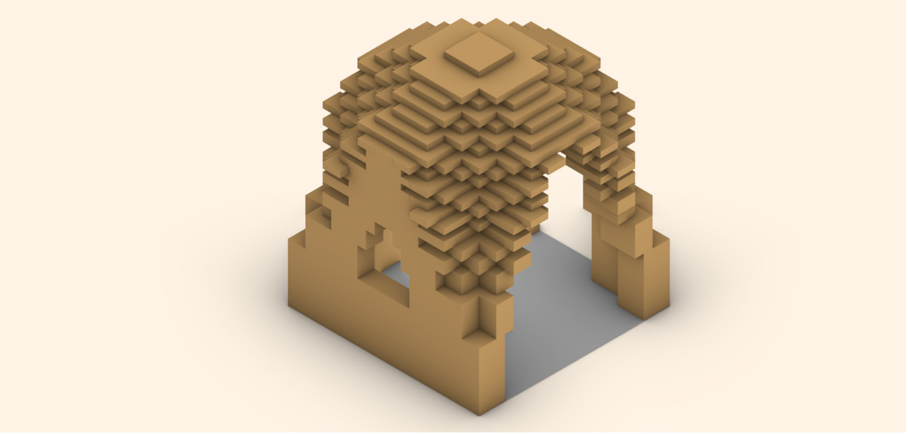


Figure 106. Scaffoling removed, final module

4.1.4. Tools

An overview is given for the tools that are needed to construct the fianl moduel. If a refugee camp has the followings tools, they can teach the people how to build their own hospitals.

One person can make around 300 blocks per day which does include the making of the mixture. The 1x1 module is made with 536 blocks, so with two people enough bricks can be made for 1 1x1 module in 1 day! Small disclaimer: this is taking the 500x300x300 as one element.

Figure 107. Buckets to transport material

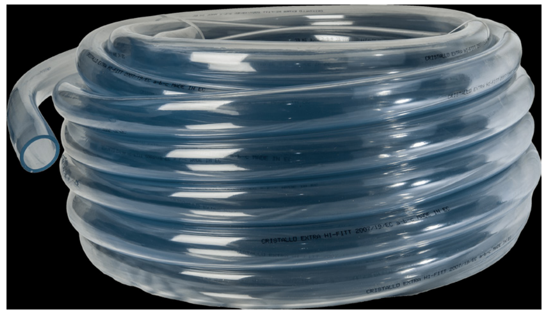


Figure 108. Tubes for water drainage

Figure 109. Robe to measure block placing

Figure 110. Raw materials of earth, gypsum and concrete

Figure 111. Trowel

Figure 112. Level

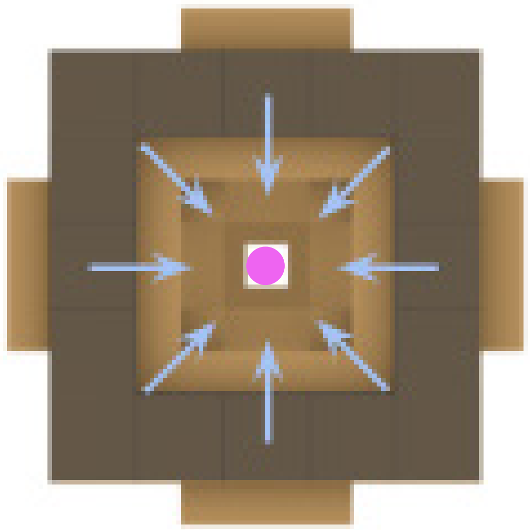
Figure 113. Square

Figure 114. Mould for 100 and 150mm blocks

4.1.5. Detailing

Storm Water Drainage

The rain water is collected in between column junctions. To allow a proper drainage system we propose a small outlet pipe of dia 100mm passing within the column junctions. The pipe needs to be placed during construction of the columns. The pipe is a permenant fixture and cannot be accessed from within, thus proper waterproofing measures are required.



Location of pipe in between square sleeve.

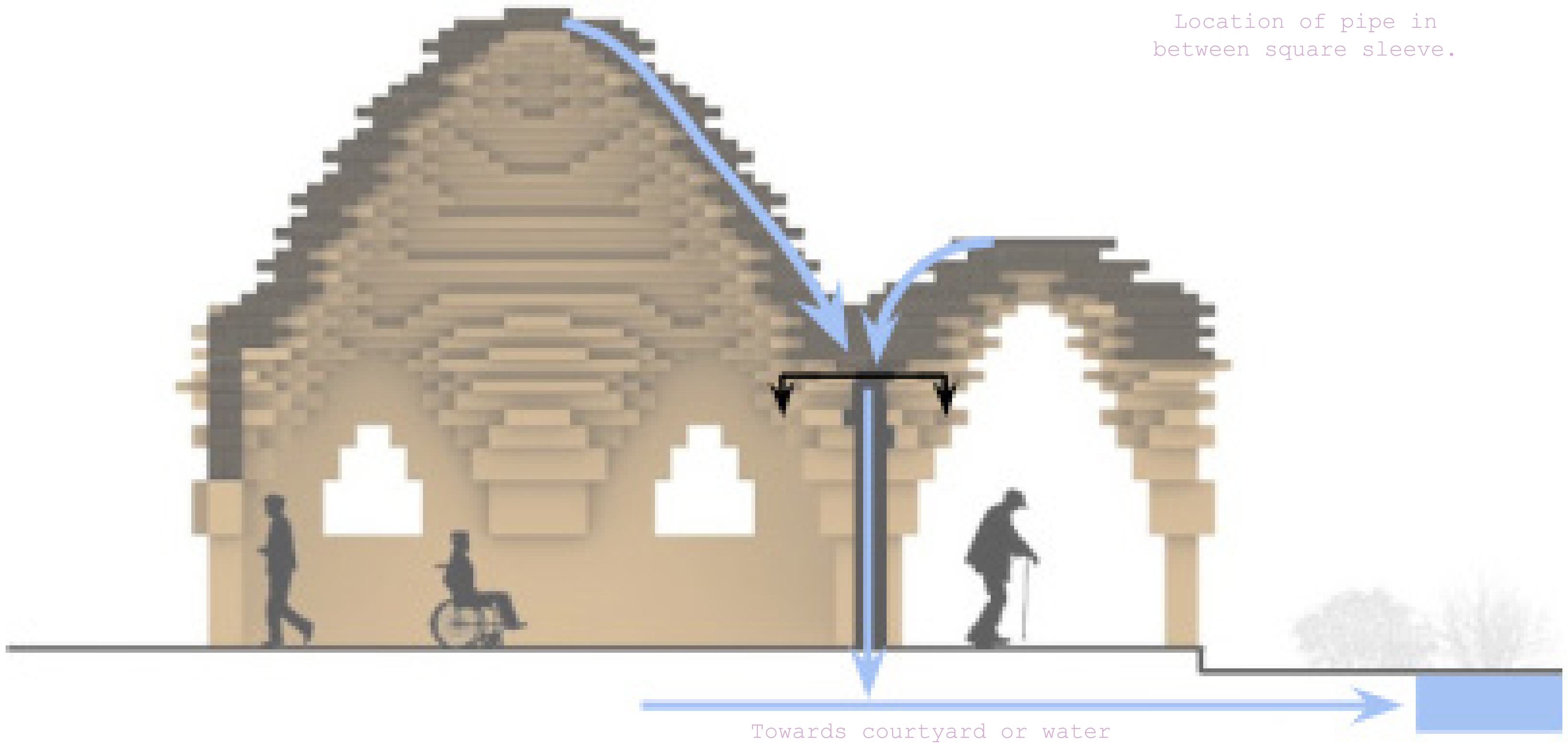


Figure 116. Rain water outlet

Color Palette

For creating an interesting contrast between various zones and functions, we decided to add some color to the adobe blocks. With variations in the material quantity a unique tone of light and dark teracotta shade can be achieved. For more contrast tone, pigment colors can be added in the teracotta mixture.

Figure 115. Color Palette

4.2. Kit of Parts

With the game and the following module parts, the user can make a health clinic or hospital on any given location.

Modules:
1x1 module
3x2 module
Staircase module

Walls:
W-01
W-02
W-03
W-04

Scaffolding elements:

S-01a	S-02a
S-01b	S-02b
S-01c	S-02c
S-01d	S-02d
S-01e	

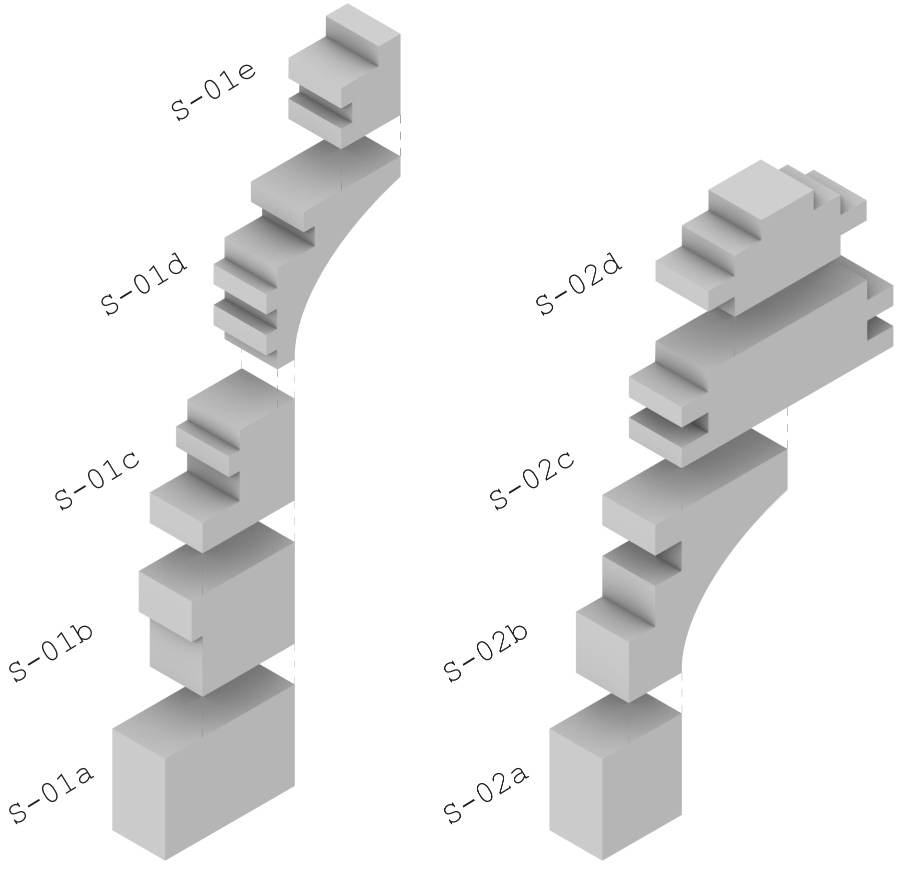


Figure 117. Scaffolding elements

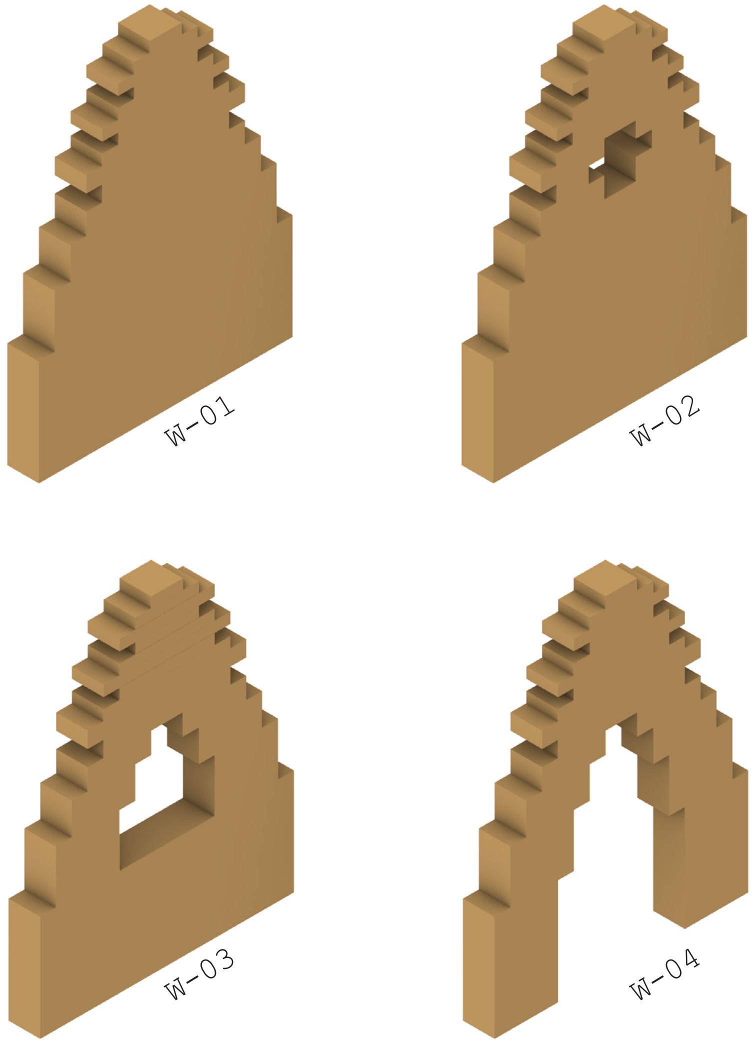


Figure 118. Wall types

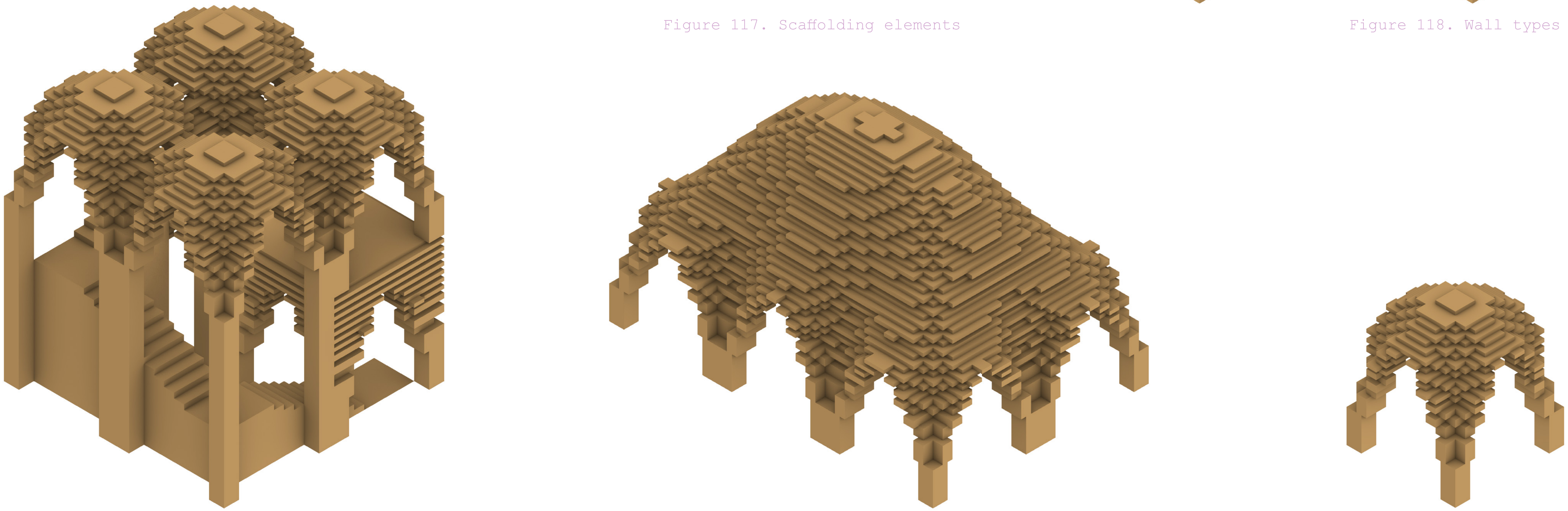
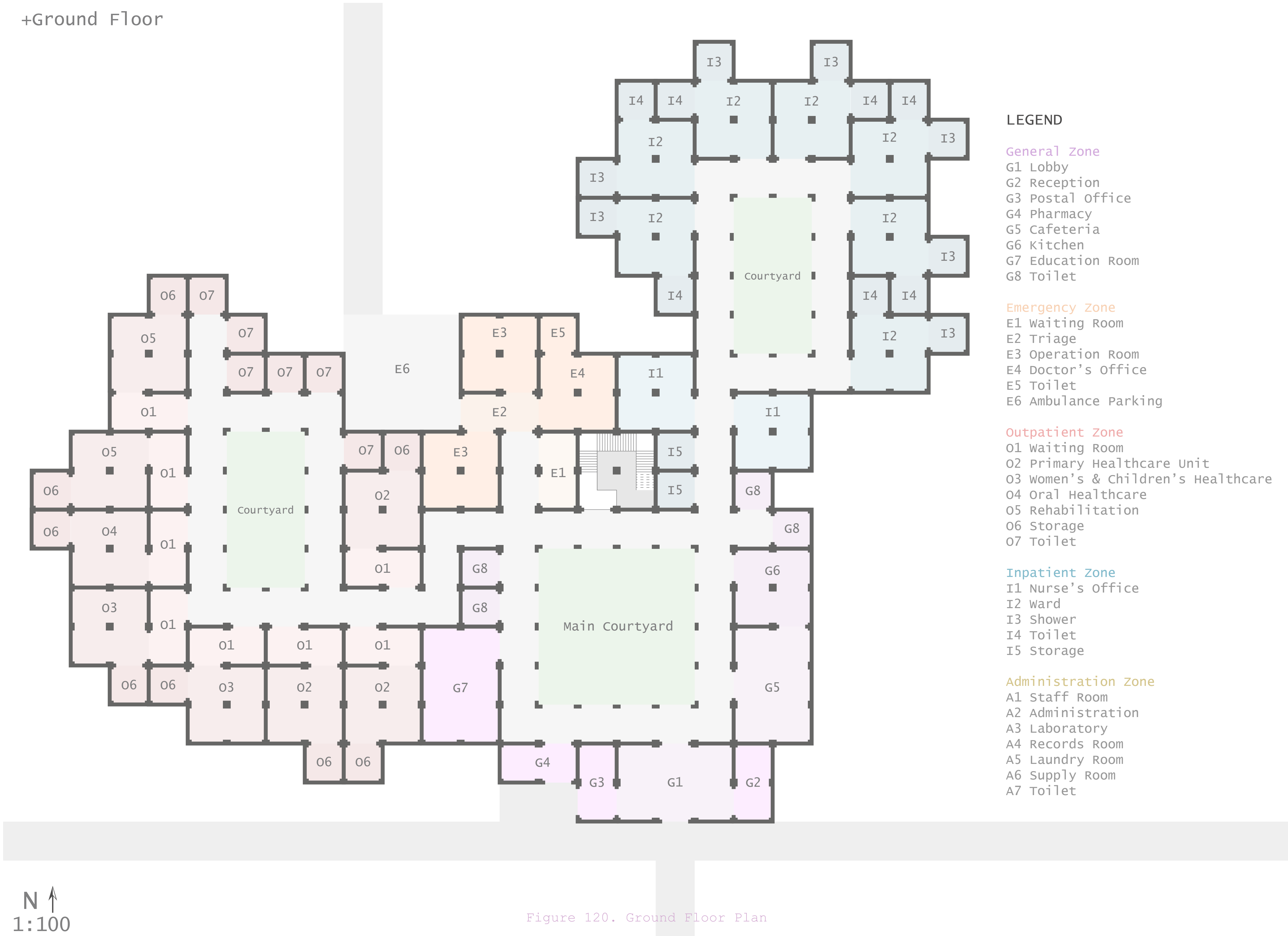


Figure 119. Staircase, 3x2 and 1x1 module

5. Final Products

5.1. Chosen Configuration of Plan

+Ground Floor



LEGEND

General Zone
G1 Lobby
G2 Reception
G3 Postal Office
G4 Pharmacy
G5 Cafeteria
G6 Kitchen
G7 Education Room
G8 Toilet

Emergency Zone
E1 Waiting Room
E2 Triage
E3 Operation Room
E4 Doctor's Office
E5 Toilet
E6 Ambulance Parking

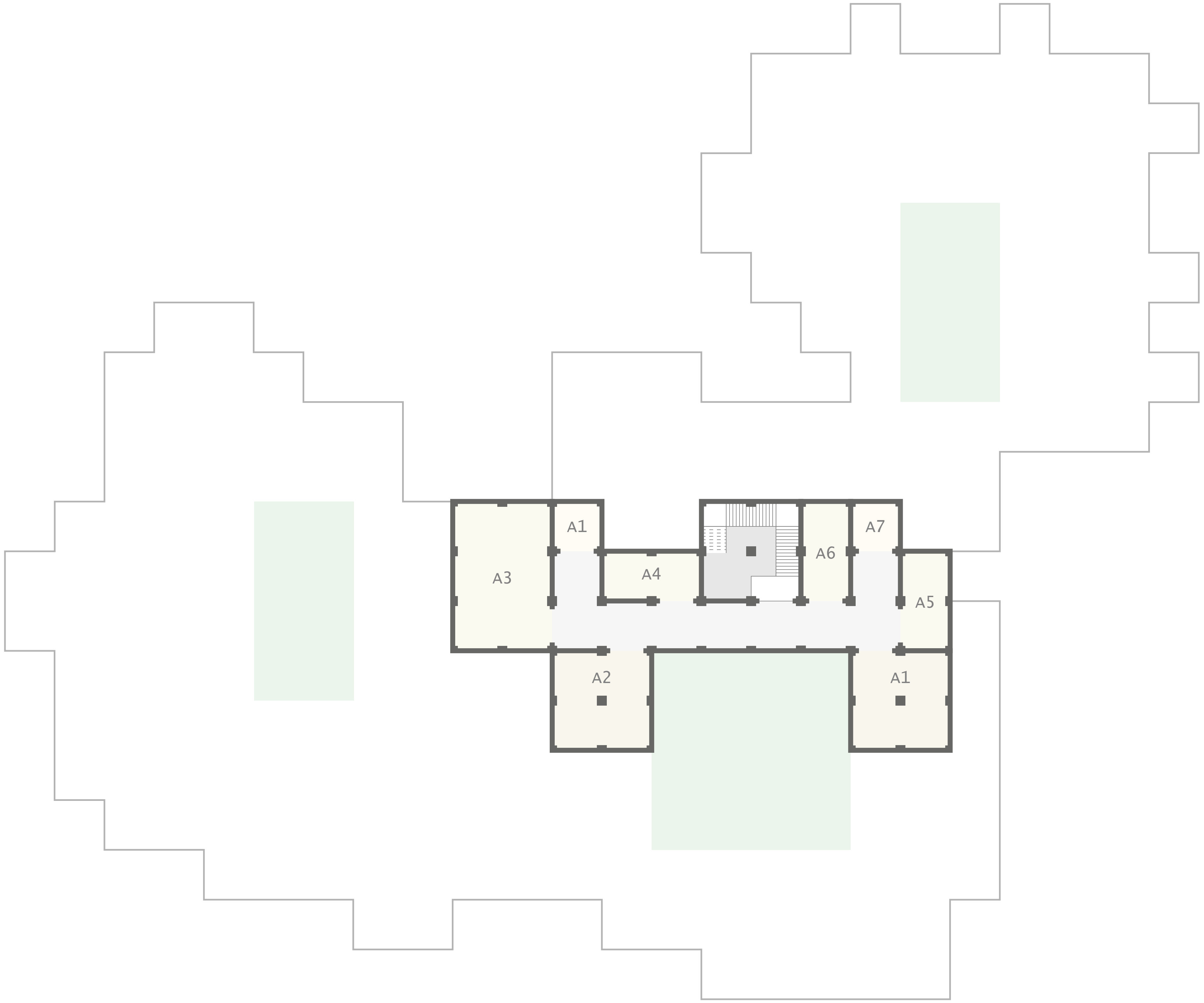
Outpatient Zone
O1 Waiting Room
O2 Primary Healthcare Unit
O3 Women's & Children's Healthcare
O4 Oral Healthcare
O5 Rehabilitation
O6 Storage
O7 Toilet

Inpatient Zone
I1 Nurse's Office
I2 Ward
I3 Shower
I4 Toilet
I5 Storage

Administration Zone
A1 Staff Room
A2 Administration
A3 Laboratory
A4 Records Room
A5 Laundry Room
A6 Supply Room
A7 Toilet

Figure 120. Ground Floor Plan

+First Floor



LEGEND

General Zone
G1 Lobby
G2 Reception
G3 Postal Office
G4 Pharmacy
G5 Cafeteria
G6 Kitchen
G7 Education Room
G8 Toilet

Emergency Zone
E1 Waiting Room
E2 Triage
E3 Operation Room
E4 Doctor's Office
E5 Toilet
E6 Ambulance Parking

Outpatient Zone
O1 Waiting Room
O2 Primary Healthcare Unit
O3 Women's & Children's Healthcare
O4 Oral Healthcare
O5 Rehabilitation
O6 Storage
O7 Toilet

Inpatient Zone
I1 Nurse's Office
I2 Ward
I3 Shower
I4 Toilet
I5 Storage

Administration Zone
A1 Staff Room
A2 Administration
A3 Laboratory
A4 Records Room
A5 Laundry Room
A6 Supply Room
A7 Toilet

N
1:100

Figure 121. First Floor Plan

5.2. Section

Figure 122. Section through the main Courtyard

5.3. Elevation

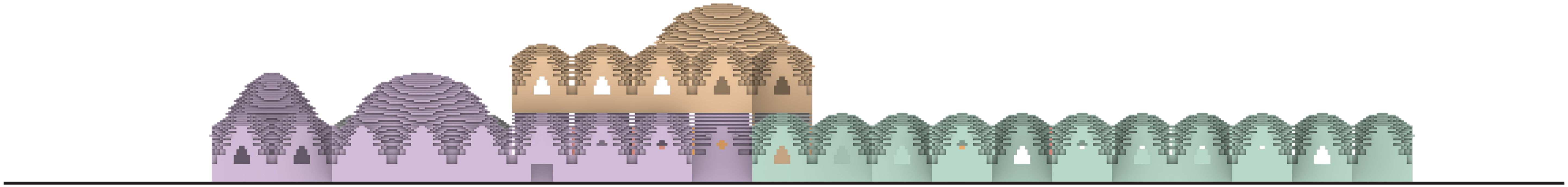


Figure 123. Right Side Elevation

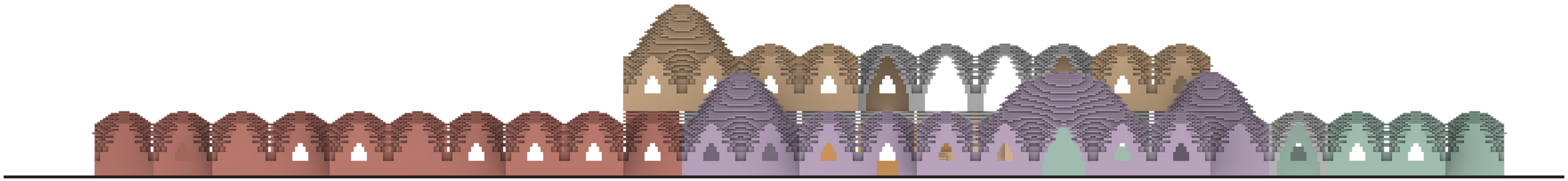


Figure 124. Front Elevation

+ +

5.4. Isometric View

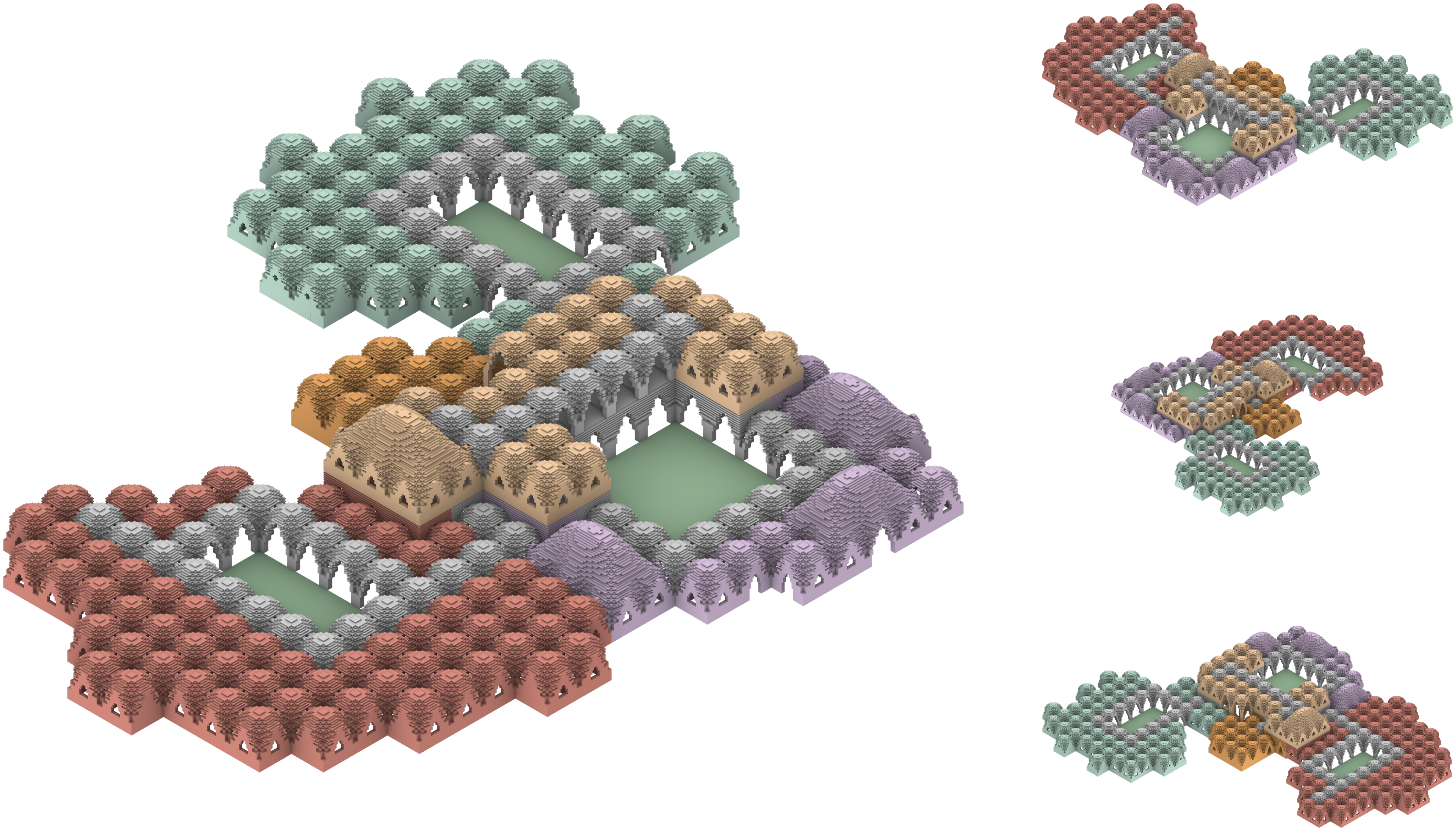


Figure 125. Isometric Views of the Final Configuration

5.5. Impressions

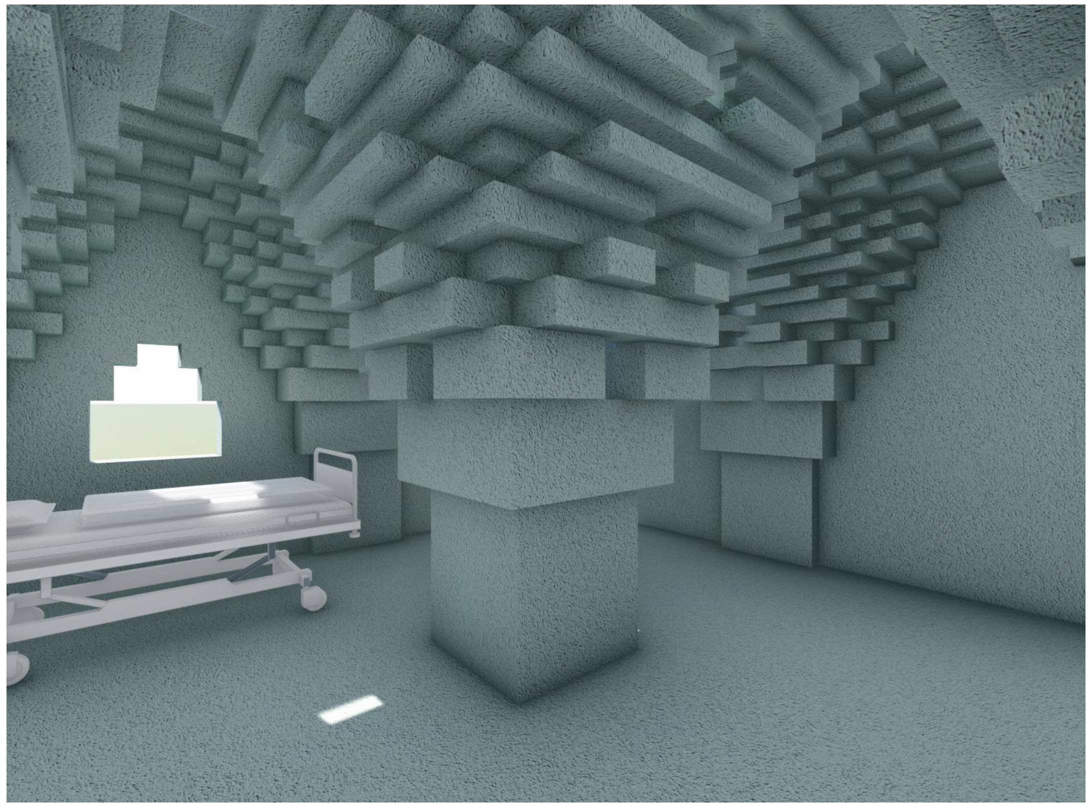
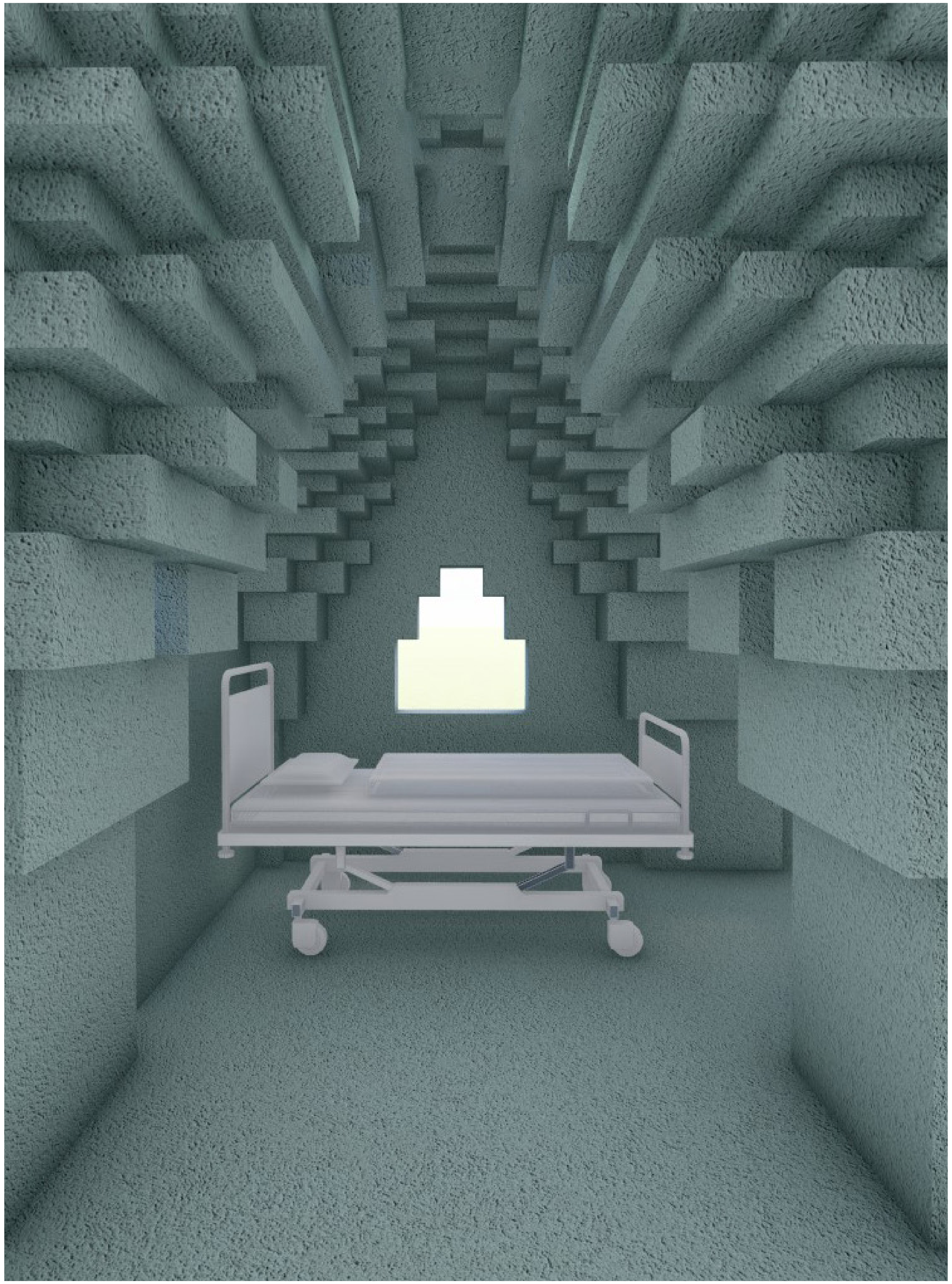


Figure 126. Impression of the Ward



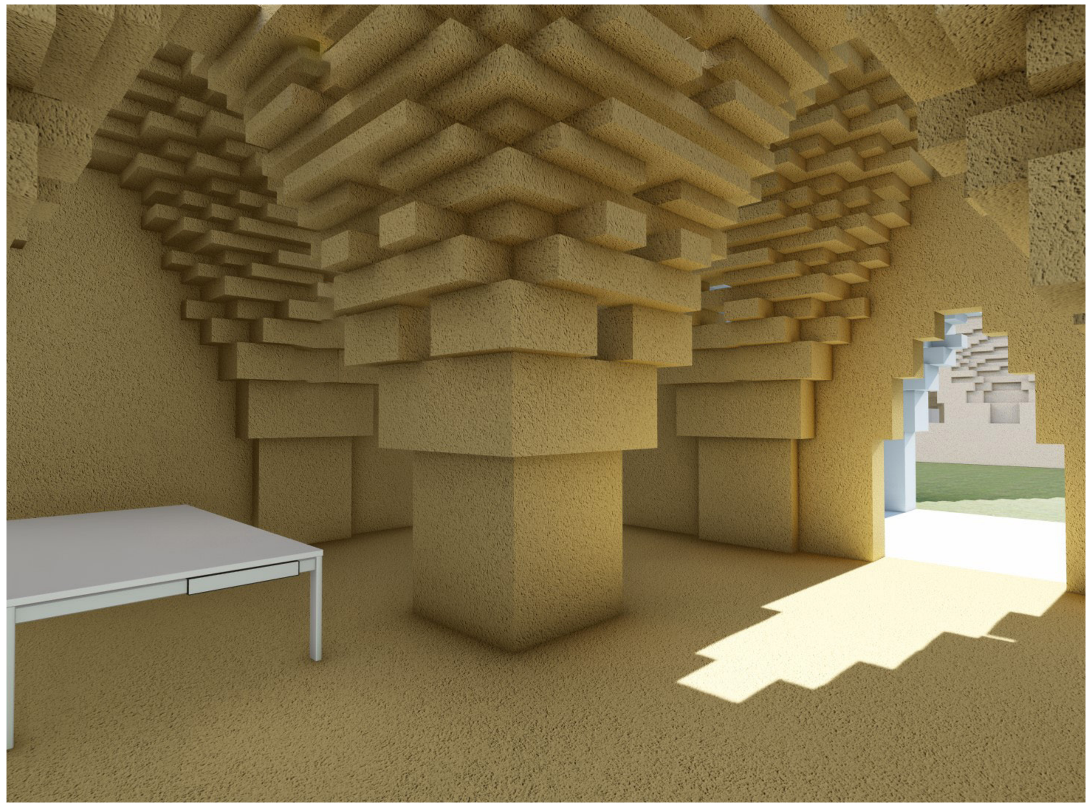


Figure 127. Impression of the Operation Room

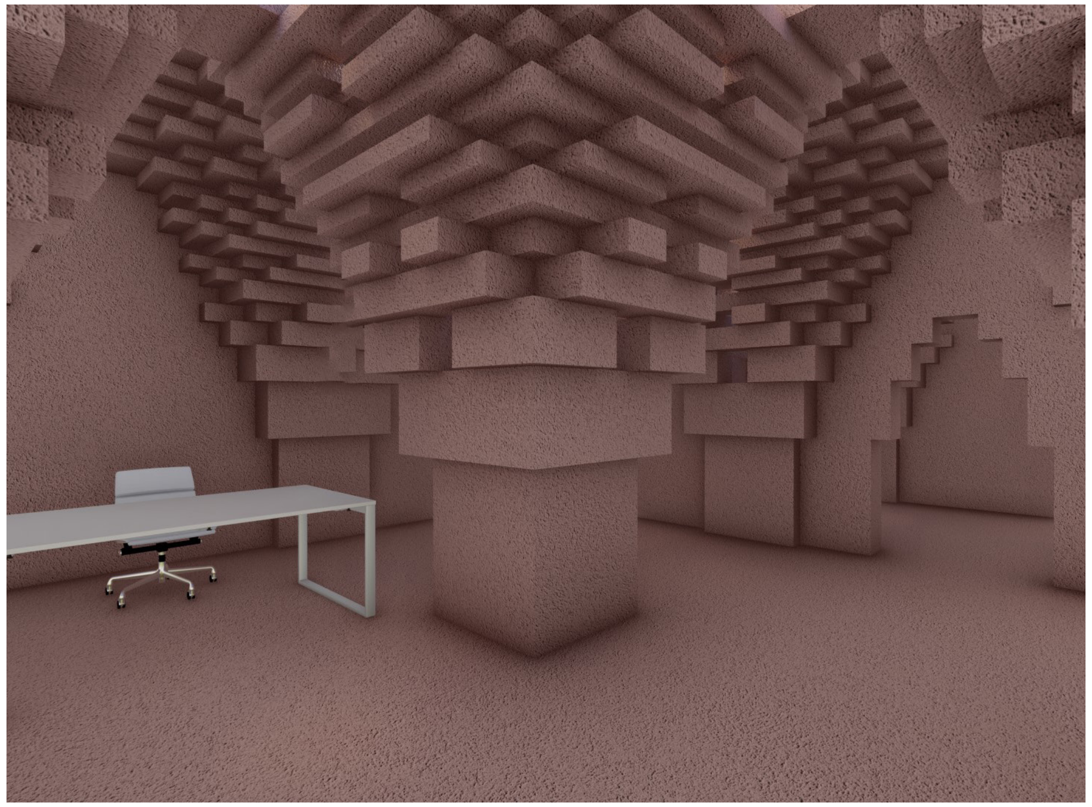


Figure 128. Impression of the Docto'rs Office

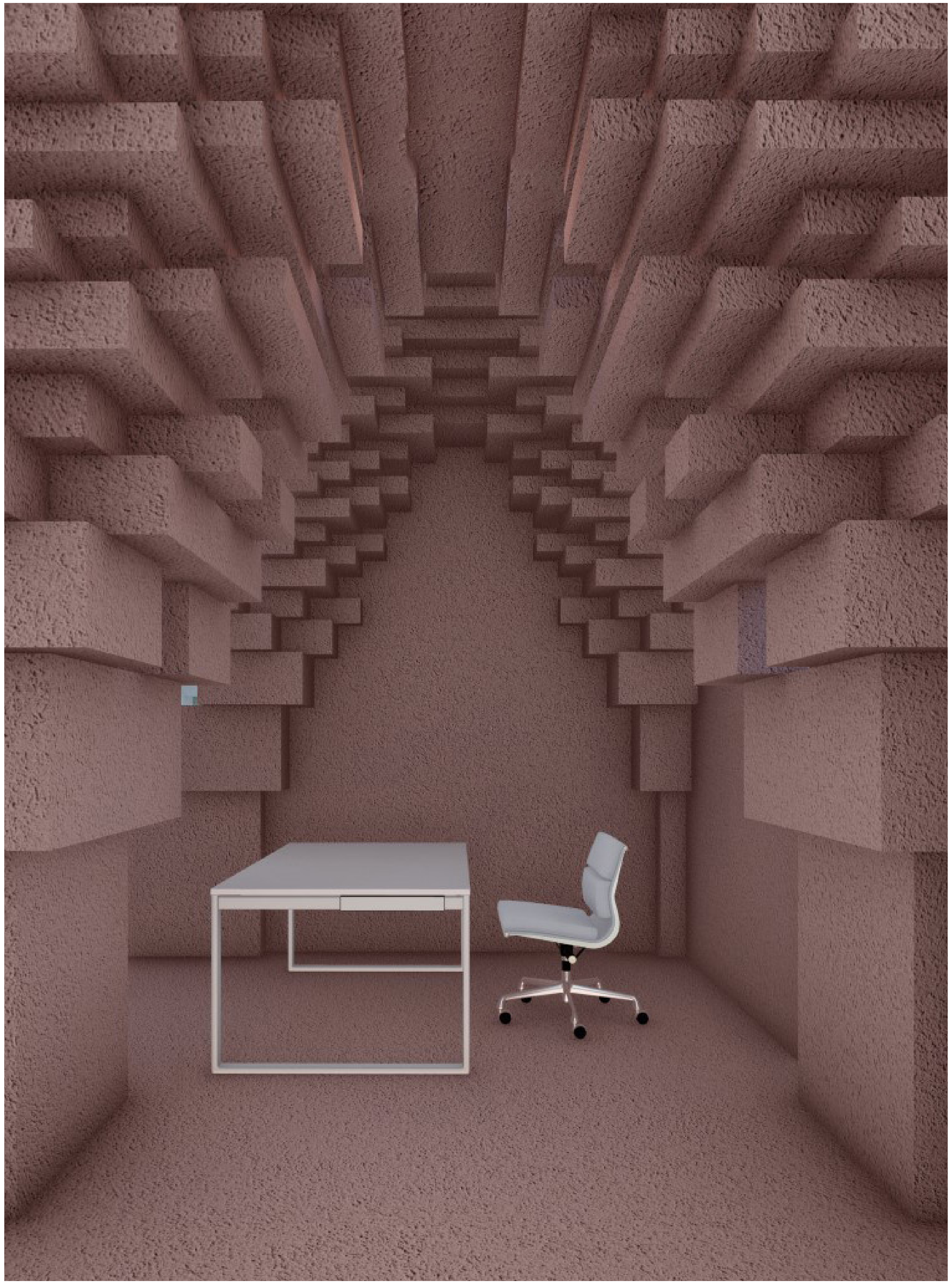


Figure 129. Impression of the Laboratory

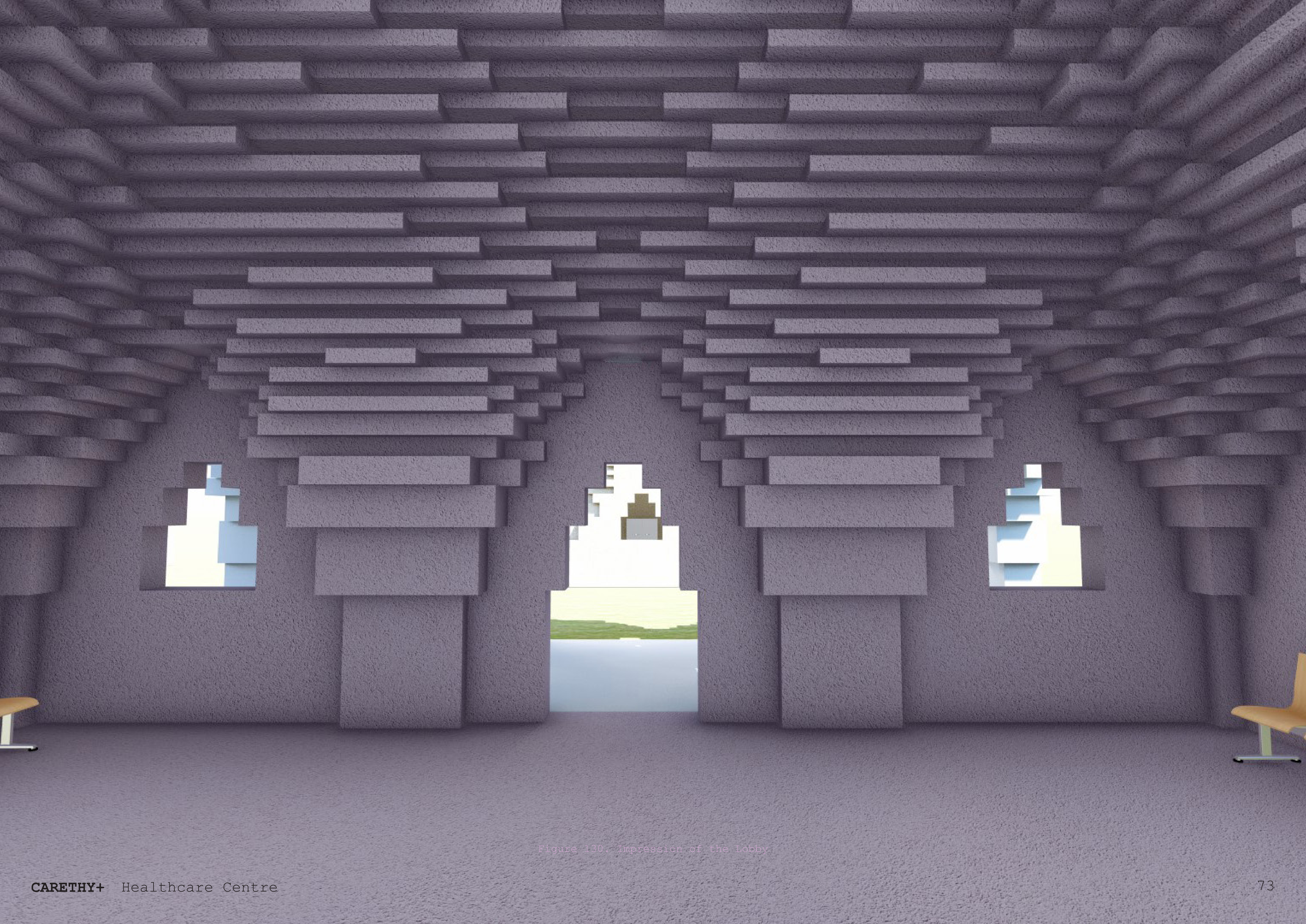


Figure 130. Impression of the Lobby

5.6. Game Interface

Figure 131. Game Interface

5.7. Overview

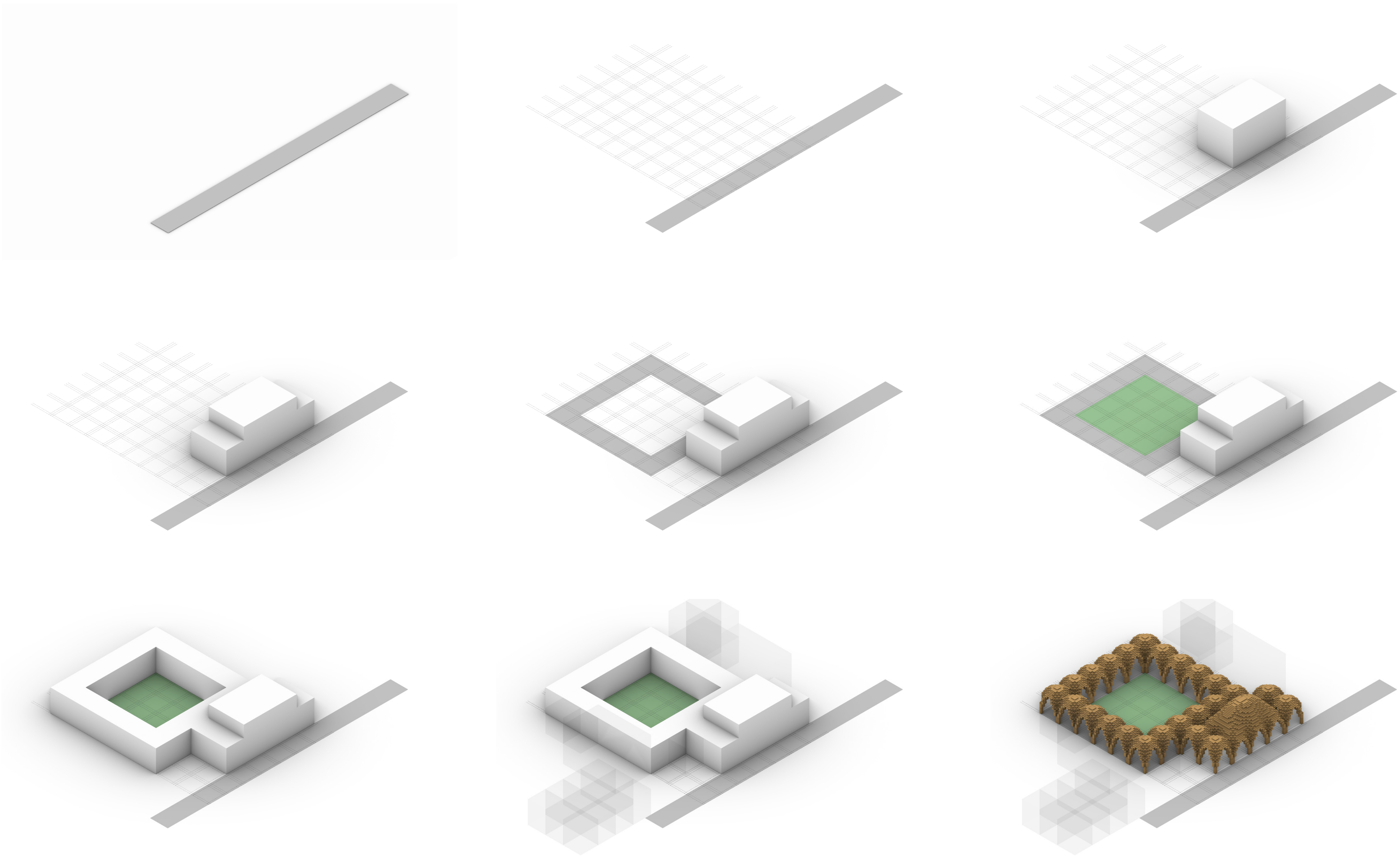
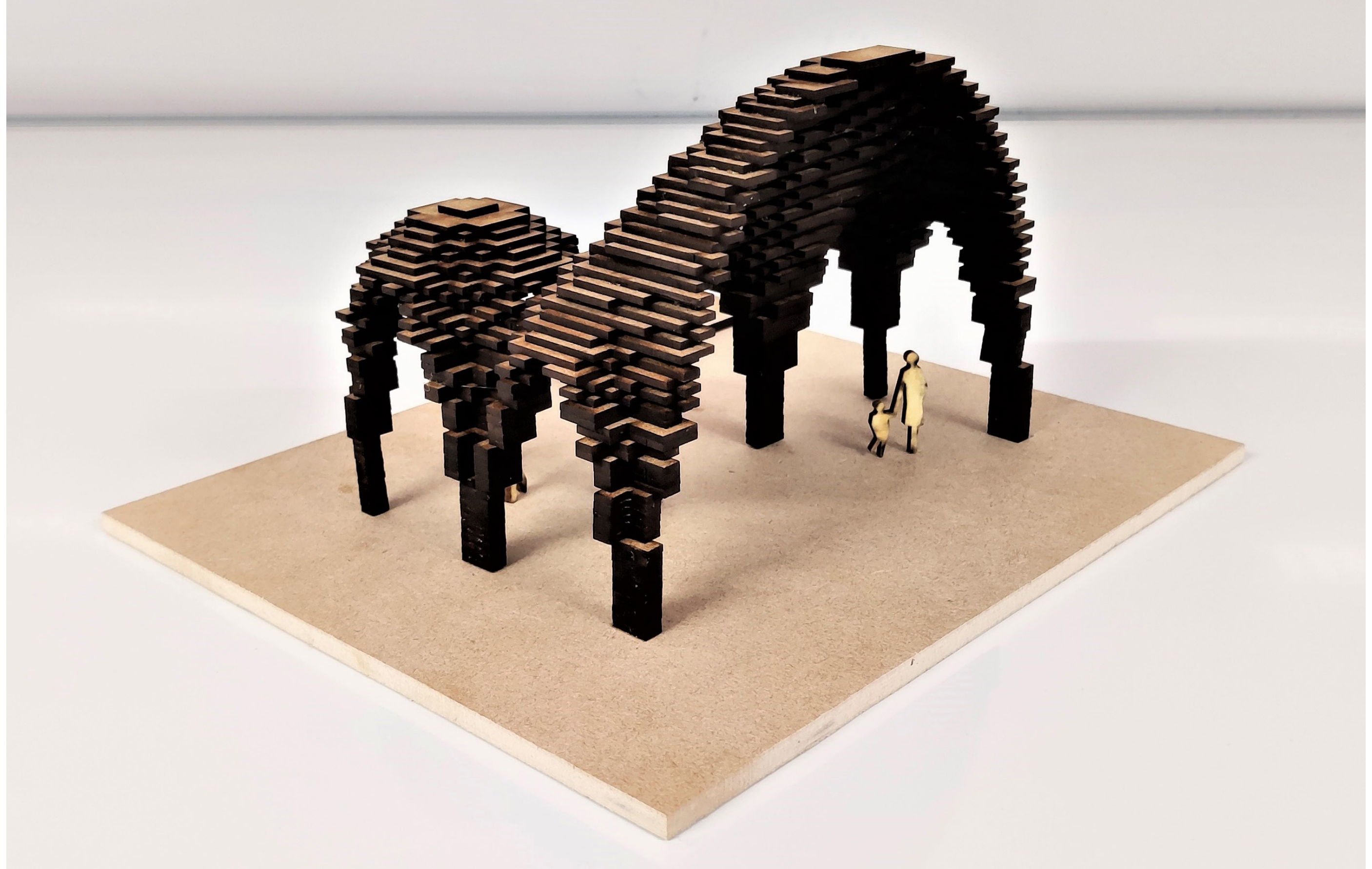


Figure 132. Overview of the Process of the Game

5.8. Physical Model



6. Reflection

6.1. Further Developments in the Project

6.1.1. Configuration

Better Integration of First Floor

The functions located on the first floor should be made more accessible by the use of ramps.

More Evaluation Criteria

Currently in the configuration, we only include ‘least possible circulation tiles’.

Game Control

The user is given high control to place the functions, which might not result in the best design (ventilation, daylight)

Gamifying the whole process

Walls are now only placed manually.
Staircase is configured manually.

6.1.2. Shaping and Forming

Spatial Variation

We wanted to have a columnless 2x2 module, but they were not possible with a first floor.

6.1.3. Construction

Material Optimization

The materials need to be reduced by scooping out extra earth from the blocks.

Scaffoldings

The scaffolding is made for the 1x1 unit only. It is still required to make a unique scaffolding for the 3x2 unit.

6.2. Individual Reflection

6.2.1. Anurag

The idea of generative design has always fascinated me even before enrolling for this course. Through Earthy I was exposed to a variety of different ideas and logical ways of thinking that were not only about the generative design but also about the relationship between earth architecture, it's structure and developing complex geometries, which made the course very interesting.

Initially I was overwhelmed with so many in depth concepts about linear algebra, topology and gamification. I was also a bit nervous to learn python, however, as the course progressed I found my interests in deriving gamification logic and further a bit more in depth derivation of shaping logic and its structural optimization.

The initial workshops about graphs and fields, topogenesis, compass, python were a bit intense and tough to follow. However, I really enjoyed the introduction to these topics and the potential it offers. Although I was more involved in the shaping and structuring process, I hope to use it for my future projects. The later workshops about dynamic relaxation and karamba analysis were really interesting, and helped me to understand the basics of form finding and provided me a good insight on generation of complex geometries.

Being the only group with 4 members, it allowed me to grab various roles and responsibilities. The teamwork was beyond great and a learning experience on its own. Collaborating with fellow classmates with unique personalities and capabilities created a spontaneous balance and a really creative environment in order to push forward.

The tutoring sessions were exceptional and very productive. The in depth discussions about the project ideations not only allowed me to push my limits but it also motivated me to do better.

6.2.2. Diederik

About nine weeks ago we wrote down our expectations of the course. I thought I would learn a lot about programming in Python and gamification, and my experience throughout the course regarding this did not disappoint.

From the midterm on I spent all my time on creating the CARETHY+ game which taught me basic principles of game design and gained me a nice bit of experience with Python. This is exactly what I was looking for in this course, and I am very happy and content with the final product.

Of course there are many things that could be improved in the game, both QoL changes (the constant clicking becomes a bit tedious) and qualitative (I am not pleased with how bigger modules can not be centrally aligned), but overall I am happy with the result. Add to this the great work my groupmates achieved on the shaping and construction, and I can leave the course as a satisfied student!

6.2.3. Karin

At the beginning of this course, I was very excited to finally make a detailed structural analysis on a design project and to make more concrete decisions based on those results. Luckily, I was able to tackle this task within the group dynamics. Moreover, during this course I got a much better grip on the purpose of computational design, since I could never image the extra opportunity. But now, I think the final product could not have been this interesting without computational design. Looking back on the process, I feel like it took us a long time to come up with the perfect tessellation, since Anurag and I constantly went back and forward to check the mesh on the structural analysis and the voxelised mesh. This could have been done much quicker if we just combined the files, but it was worth it. Besides this, I feel that the team was great and we always knew what the other persons were working on. After these 10 intense weeks, I now feel like the extreme Grasshopper expert and I even know a little Python now, who would have thought! It was super fun and take CARETHY!

6.2.4. Pragya

Earthy was an intensive and rewarding course. It pushed me out of my comfort zone and was a very steep learning curve for me. Within the course, the idea of generative design and converting architectural strategies + logic into hard rules appealed to me the most, and was my focus in the project. In my experience, the beginning two weeks of the course had a lot of content and were very overwhelming. It will be useful to proceed stage wise and distribute the workshops/ lectures over the first 6 weeks instead. It would help to study and understand the theoretical and practical aspects together and also provide some kind of deadline for the 3 stages. As a group, I feel that we might have spent much time on deciding the functional program, which was not very critical for the final product. Moreover, the scale of this project has taught me that it's important to limit the scope and in order to not get lost in the possibilities. I am happy that the education for this course was majorly on-campus as it provided an opportunity to learn from peers with the same roles in other groups. In the process, I learnt that the most complex of problems can be resolved through simplification.

Group 6

7. References

[1] https://www.researchgate.net/publication/6795797_How_far_do_nurses_walk/citations

[2] https://healthfacilityguidelines.com/ViewPDF/ViewIndexPDF/iHFG_part_c_space_standards_dimensions#:~:text=handrails%20and%20obstructions-,Patient%20Corridors,widths%20of%202450mm%20are%20recommended.

[3] https://www.theglobaleconomy.com/rankings/physiotherapists_per_1000_people/

8. Appendix

8.1. Rules of the Game

MANUAL RULES OF THE GAME / PROGRAMMING OF THE COMPUTATIONAL GAME

Description of the Game

1. The player is given a combination of functions which are clubbed in their respective zones.

2. These functions are already laid out within tiles of 1x1, 1x2, 2x2, 3x2.

3. The aim is to achieve a functional health care center building of a required scale - which is achieved through configuring the functions by following all the rules of the game.

4. The configuration of functions is done by placing them zone-wise. This follows a priority list which is the following -
   a. *Context (predefined)*
   b. Entrance, Courtyard, General
   c. Emergency Care Unit
   d. Outpatient Department
   e. Inpatient Department
   f. Administrative Block

5. Within these zones, there is a specific 'move order' to place the functions in the order of their importance.

6. The game follows a system of learning, where the configuration can be evaluated after playing it once. This evaluation is based on the following aspects -
   a. Use less corridor space
   b. Improve lighting & ventilation
   c. Improve the hierarchy of sanitation areas
   d. Improve line of sight, smells

   Each time the player plays the game, the configuration can be improved with respect to these aspects.

Configuration of the Game

1. The game begins with the *'List of Functions'* which is developed as per requirements of the district. For this game, the list is predefined. To play for a different set of requirements, place the functions as per the *'Number required in our project case'* in the given list.

Calculation of Amount of Functions

2. Then the total 'built area' or 'occupied modules' is calculated. The main courtyard size is derived from this number. For a built area of [x], the courtyard size is [x/7].

3. Start placing the functions as per the following list of 'Move order' and 'Rules'.

Move Order	**Rules**
0. Context	NA
1. Entrance, Courtyard, General A. Lobby Cluster a. Lobby b. Postal c. Reception B. Courtyard C. Cafeteria a. Kitchen D. Pharmacy, Education Hall E. Toilets	- Lobby is connected to the road and courtyard. - Toilets are connected to the courtyard. - Kitchen is connected to the cafeteria.
2. Emergency Care Unit A. Waiting Area B. Doctor's Office Cluster a. Doctor's Office b. Toilet C. Operation Room D. Triage E. Ambulance Parking F. Toilet G. Connect Ambulance Parking to the Main Road using corridor	- Waiting area is connected to the corridor. - Triage is connected to both the operating rooms and the ambulance parking. - The Doctor's office is connected to the waiting room. - The toilet is connected to the Doctor's office.
3. Outpatient Department A. PHU, OHU, Rehab, WCC Cluster a. Waiting Area b. Doctors Office c. Storage B. Toilets	- Place the functions attached to an extra courtyard. - Place PHU closer to the main corridor, place WCC further away. - Toilets are connected directly to the corridor space.

4. Inpatient Department A. Nurse's Office B. Ward Cluster a. Ward b. Shower, Toilet C. Storage	- Place the functions attached to an extra courtyard. - Storage spaces (common) are connected directly to the corridor space.
5. Administrative Block A. Administrative Office, Staff Room, Laboratory, Laundry, Supply Space, Records Room B. Toilets	- Place the functions attached to a corridor or courtyard. - Toilets are connected directly to the corridor space.

Common Rules

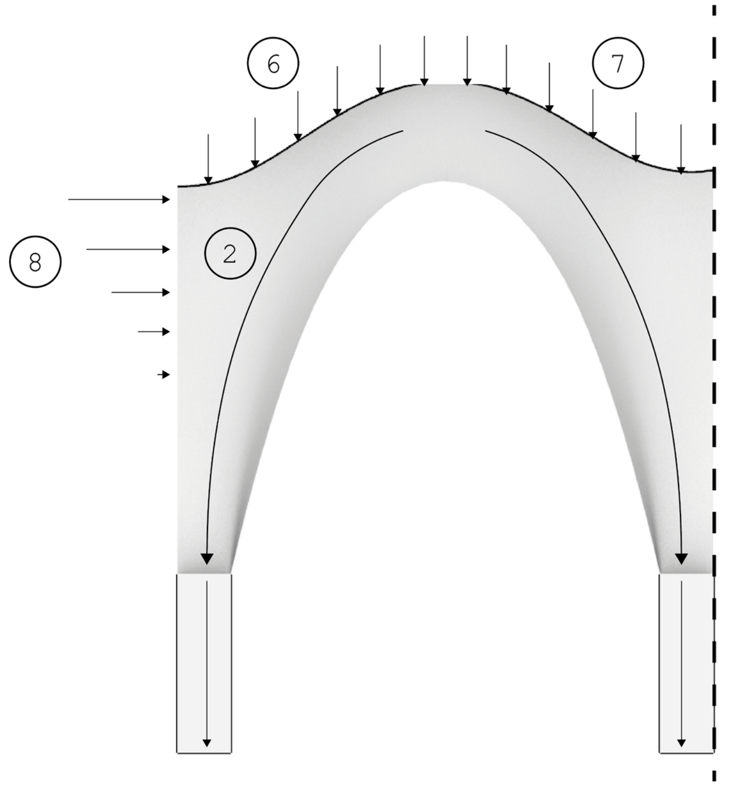
1. The first tiles in the cluster are connected to the corridor.
2. The first tiles in each cluster should have at least one edge connected to the corridor. (A, B, C, D, E are clusters)
3. Storage, Showers and Toilets are end functions, more functions cannot be attached to them.
4. Place the function tiles as close to the riwaq as possible.
5. Constraints
   a. Maximum length of the corridor is 8.
   b. Minimum width of the courtyard is 2 cells.
   c. Minimum size of the courtyard is 4 cells.
   d. Maximum size of the courtyard is 16 cells.
6. If the required space for the function tiles has a configuration exceeding the maximum length of the corridor, make a courtyard.
7. If the required space for the function tiles has a configuration exceeding the maximum size of the courtyard, add another floor.
8. First Floor Rules
   a. The function tiles on another floor should have the same edges as the tiles below.
   b. No function tiles can be placed above the 3x2 modules.
9. Staircase
   a. The staircase tile has to be connected to the corridor with at least two edges connected.

Hints

1. Use vertices of the courtyard to connect new courtyards.
2. Clusters with the same function tiles have the same configuration.

8.2. Structural Calculations

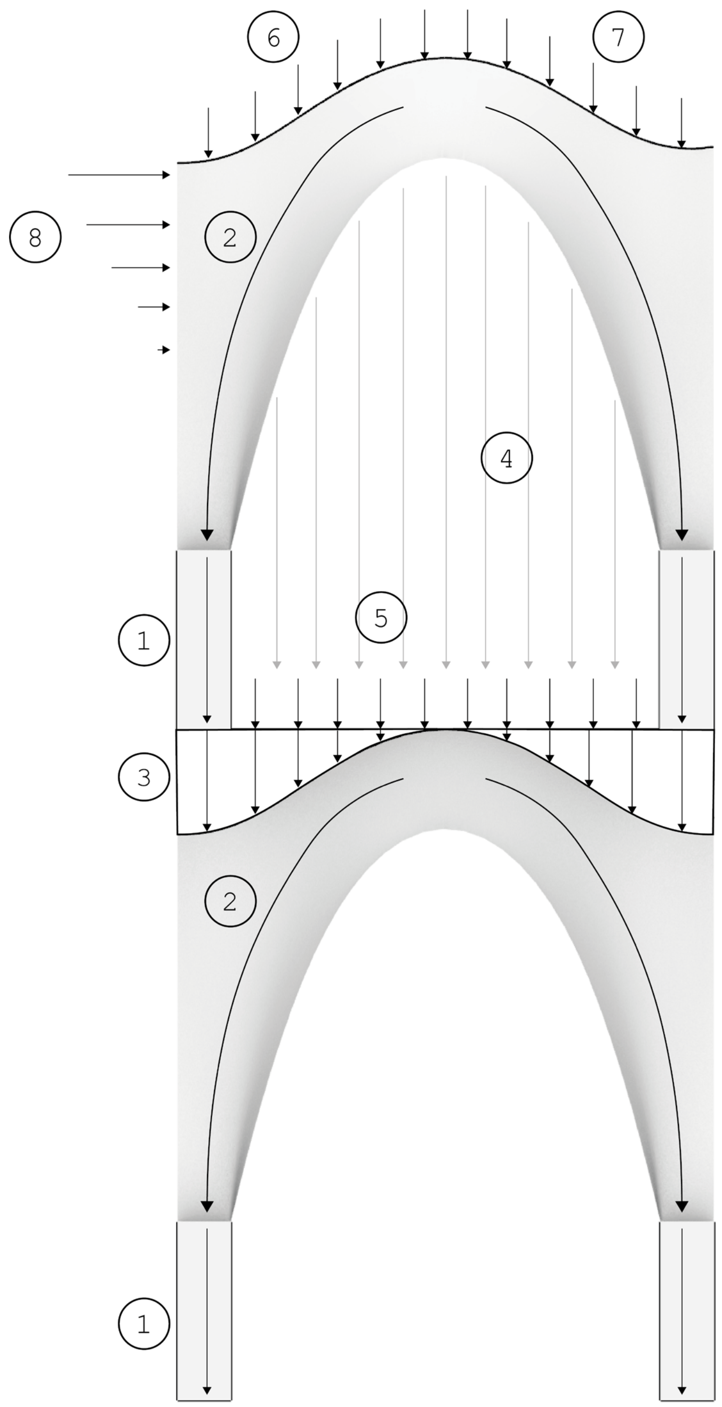
8.2.1. Load Calculations



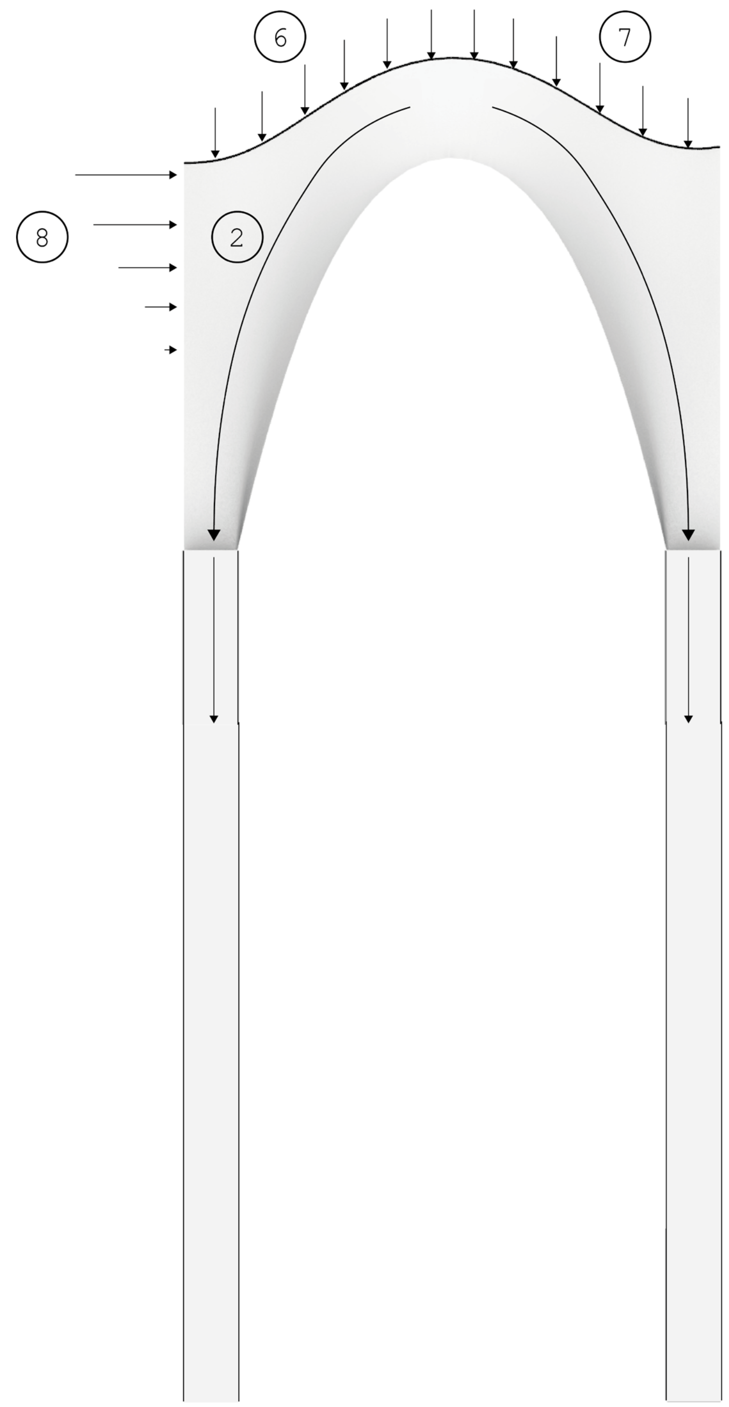
3X2 MODULE	For the columns	Value	Unit	Comments
2	*Shell on column*			
	Area of shell	105,2	m2	Taken from GH
	Thickness of shell	0,3	m	
	Density	15	kN/m3	
	Total mass	473,5	kN	
	Total force on column	29,6	kN	
5	*Live load on roof*			
	Live load	2	kN/m2	
	Area of roof	51,3	m2	
	Total force on column	6,4	kN	
6	*Wind load*			
	Wind load	0,2	kN/m2	
	Side area of shell	52,6	m2	Taken from GH
	Total force on column	0,7	kN	
7	*Snow load*			
	Given factor	0,02	kN/m2	
	Area of roof	9,0	m2	
	Total force on column	0,05	kN	

Total	Safety factor on dead load	1,35	
	Safety factor on live load	1,5	
	Φ0, Snow load	0,05	
	Φ0, Wind load	0,5	
	Φ0, Live load roof	0,05	
	Vertical forces	**40,4**	**kN**
	Horizontal forces	**0,5**	**kN**

FIRST FLOOR LOADS	For the columns	Value	Unit	Comments
1	*Dead load of column*			
	Width	0,45	m	
	Depth	0,45	m	
	Height	1	m	
	Density	15	kN/m3	
	Total force	3,0	kN	
2	*Shell on column*			
	Area of shell	15,5	m2	Taken from GH
	Thickness of shell	0,3	m	
	Density	15	kN/m3	
	Total mass	69,8	kN	
	Total force on column	34,9	kN	
3	*Dead load of sand infill*			
	Volume of infill	11,3	m3	Taken from GH
	Density of infill material	15	kN/m3	
	Mass of infill	169,5	kN	
	Force of infill on column	42,4	kN	
4	*Live load on 1st floor*			
	Live load	5	kN/m2	
	Area of 1st floor	9	m2	
	Total force on column	11,3	kN	
5	*Live load on roof*			
	Live load	2	kN/m2	
	Area of roof	9,0	m2	
	Total force on column	4,5	kN	
6	*Wind load*			
	Wind load	0,2	kN/m2	
	Side area of shell	7,8	m2	Taken from GH
	Total force on column	0,4	kN	
7	*Snow load*			
	Given factor	0,02	kN/m2	
	Area of roof	9,0	m2	
	Total force on column	0,05	kN	
8	*Load of the walls on 1st floor*			
	Area of wall (front view)	3,6	m2	Taken from GH
	Thickness of wall	0,45	m	
	Density	15	kN/m3	
	Total force of wall on column	24,3	kN	



Total	Safety factor on dead load	1,35	
	Safety factor on live load	1,5	
	Φ0, Snow load	0,05	
	Φ0, Wind load	0,5	
	Φ0, Live load roof	0,05	
	Φ0, Live load 1 floor	0,25	
	Vertical forces	**145,8**	**kN**
	Horizontal forces	**0,3**	**kN**



STAIR LOADS	For the columns	Value	Unit	Comments
2	*Shell on column*			
	Area of shell	15,5	m2	Taken from GH
	Thickness of shell	0,3	m	
	Density	15	kN/m3	
	Total mass	69,8	kN	
	Total force on column	17,4	kN	
5	*Live load on roof*			
	Live load	2	kN/m2	
	Area of roof	9,0	m2	
	Total force on column	4,5	kN	
6	*Wind load*			
	Wind load	0,2	kN/m2	
	Side area of shell	7,8	m2	Taken from GH
	Total force on column	0,4	kN	
7	*Snow load*			
	Given factor	0,02	kN/m2	
	Area of roof	9,0	m2	
	Total force on column	0,05	kN	

Total	Safety factor on dead load	1,35	
	Safety factor on live load	1,5	
	Φ0, Snow load	0,05	
	Φ0, Wind load	0,5	
	Φ0, Live load roof	0,05	
	Vertical forces	**23,9**	**kN**
	Horizontal forces	**0,3**	**kN**

8.2.2. Stress Calculations

First Floor Module

Hor. and Vert. stresses combined	Value	Unit
Section modulus	15187500	mm3
Moment	290625	Nmm
Stress	0,02	N/mm2
Area	202500	mm2
Force	145753	N
Stress	0,72	N/mm2
Max compression	**0,74**	**N/mm2**
Min compression	**0,70**	**N/mm2**
Max possible compression	**1,50**	**N/mm2**

Staircase Module

Hor. and Vert. stresses combined	Value	Unit
Section modulus	15187500	mm3
Moment	290625	Nmm
Stress	0,02	N/mm2
Area	202500	mm2
Force	23882	N
Stress	0,12	N/mm2
Max compression	**0,14**	**N/mm2**
Min compression	**0,10**	**N/mm2**
Max possible compression	**1,50**	**N/mm2**

3x2 Module

Hor. and Vert. stresses combined	Value	Unit
Section modulus	15187500	mm3
Moment	493246	Nmm
Stress	0,03	N/mm2
Area	202500	mm2
Force	40437	N
Stress	0,20	N/mm2
Max compression	**0,23**	**N/mm2**
Min compression	**0,17**	**N/mm2**
Max possible compression	**1,50**	**N/mm2**

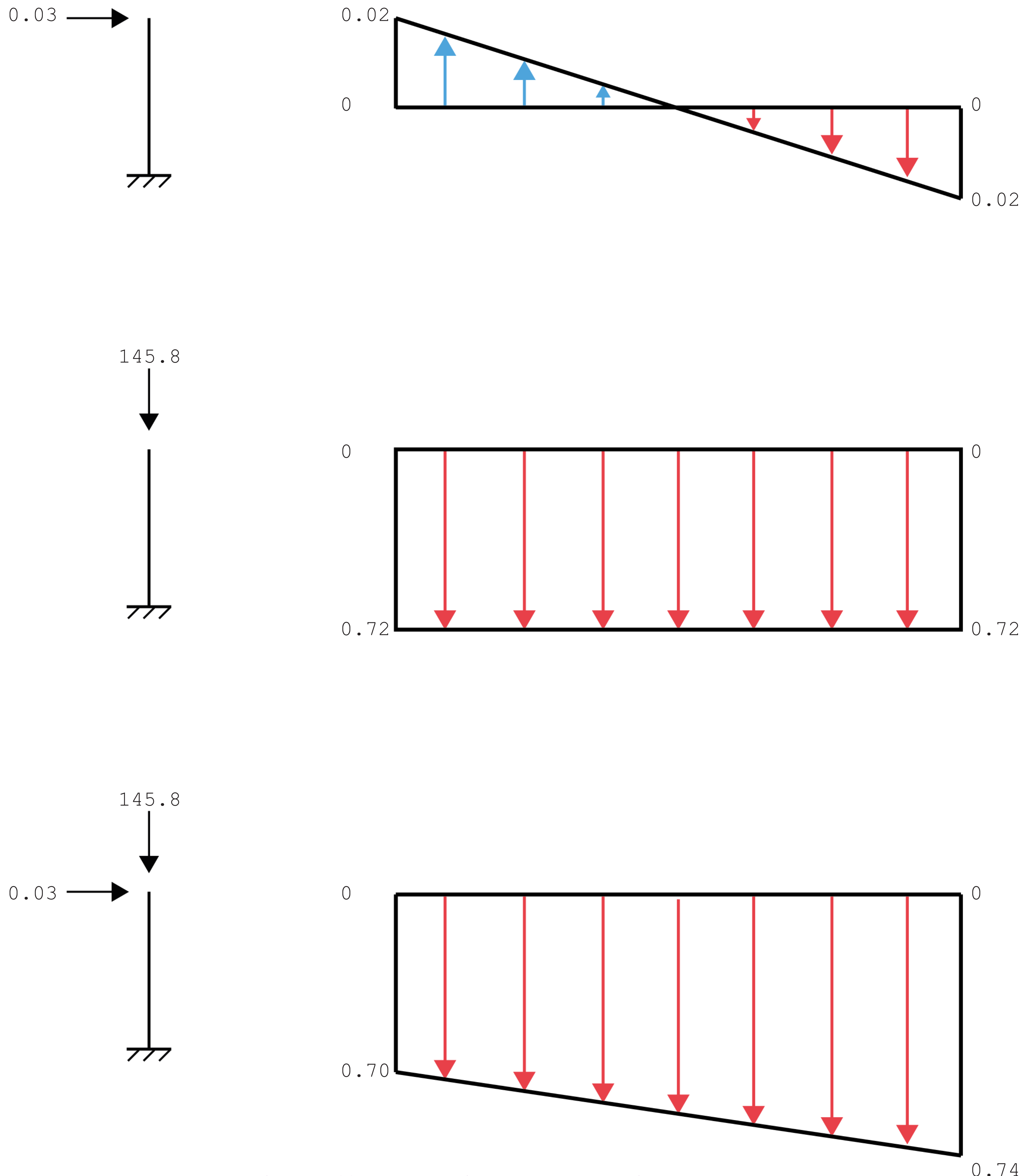


Figure 133. Stress diagrams example first floor load

8.2.3. Buckling

First Floor Module

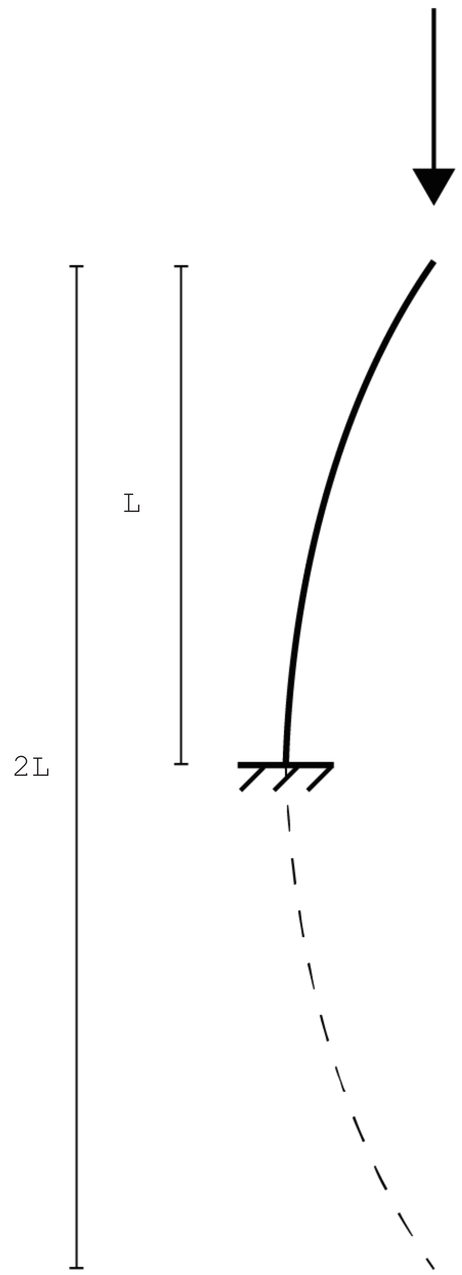
Buckling check column	**Value**	**Unit**
Youngs Modulus	110	N/mm2
Moment of inertia	3417187500	mm4
Height	1000	mm2
k	2	
Max force	927473	N
Max force	**927,5**	**kN**
Current Max force	**145,8**	**kN**

Staircase Module

Buckling check column	**Value**	**Unit**
Youngs Modulus	110	N/mm2
Moment of inertia	3417187500	mm4
Height	4800	mm
k	2	
Max force	40255	N
Max force	**40,3**	**kN**
Current Max force	**23,9**	**kN**

3x2 Module

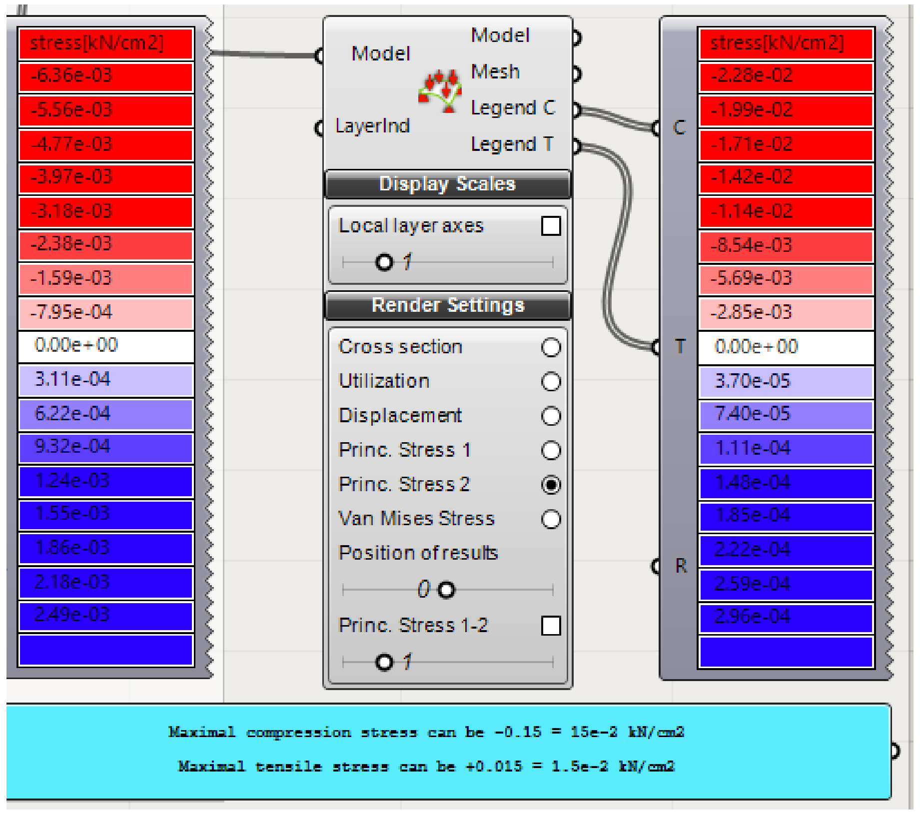
Buckling check column	**Value**	**Unit**
Youngs Modulus	110	N/mm2
Moment of inertia	3417187500	mm4
Height	1000	mm
k	2	
Max force	927473	N
Max force	**927,5**	**kN**
Current Max force	**40,4**	**kN**



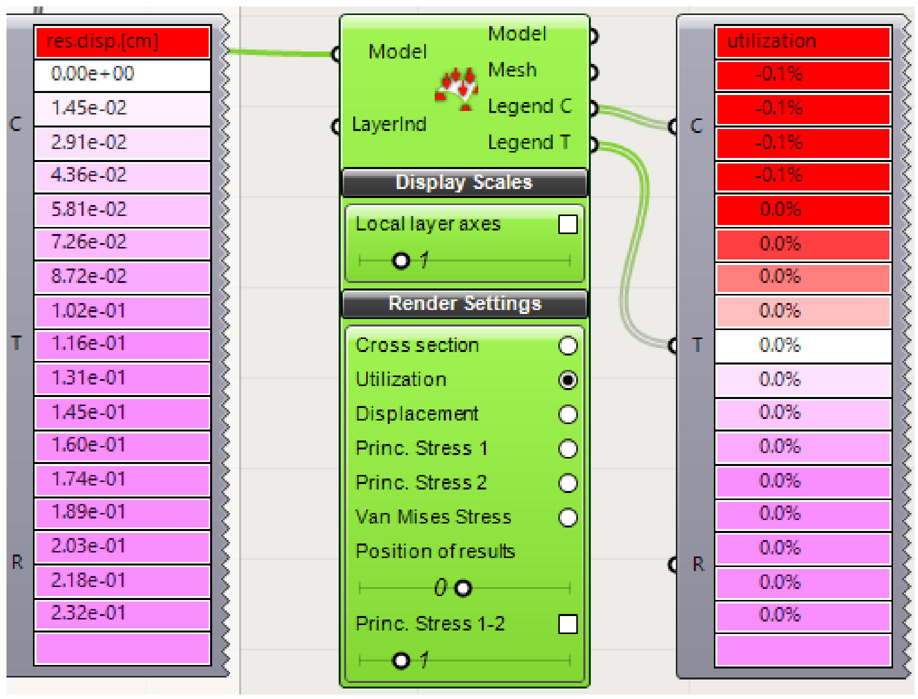
8.2.4. Analysis Result for Load Cases

Structural Results: 1*1_Load Case 1

Princ. Stress 1 Princ. Stress 2

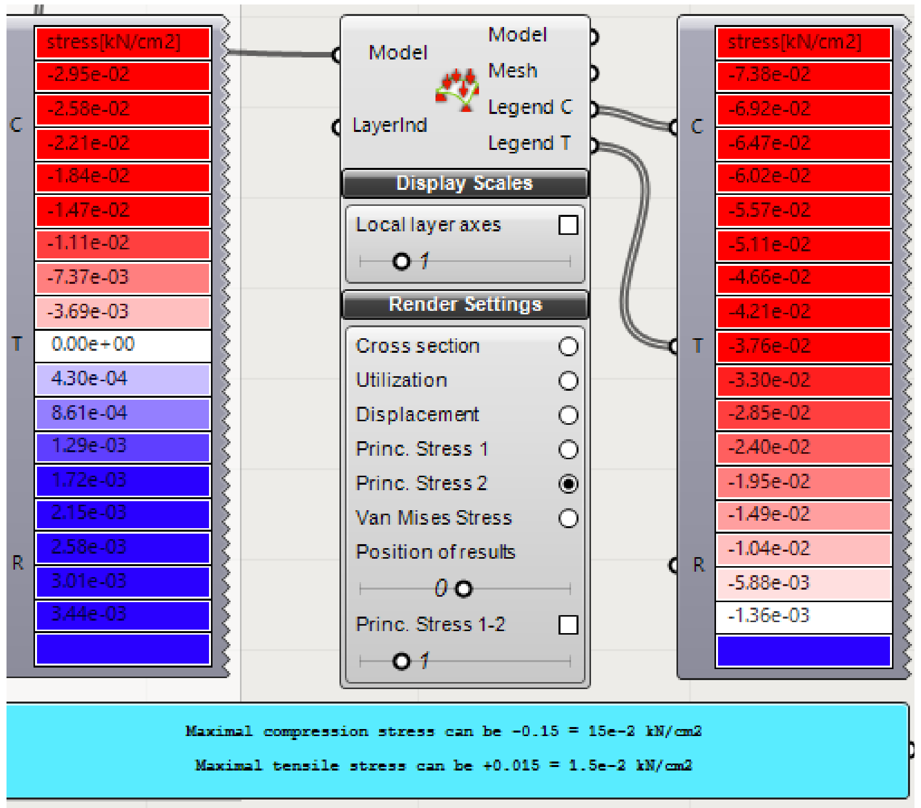


Displacement Utilization

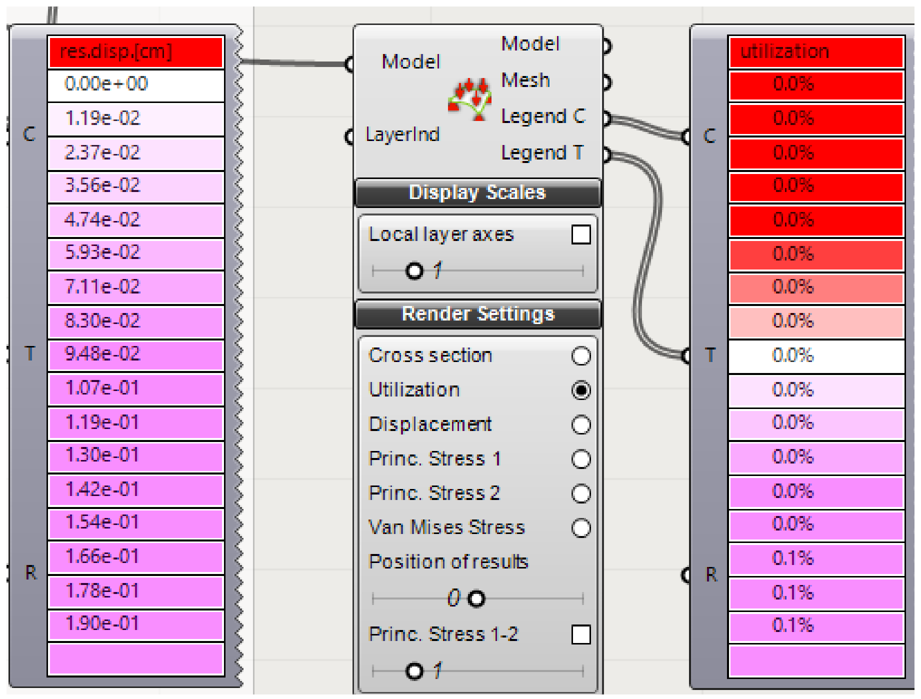


Structural Results: 1*1_Load Case 2

Princ. Stress 1 Princ. Stress 2

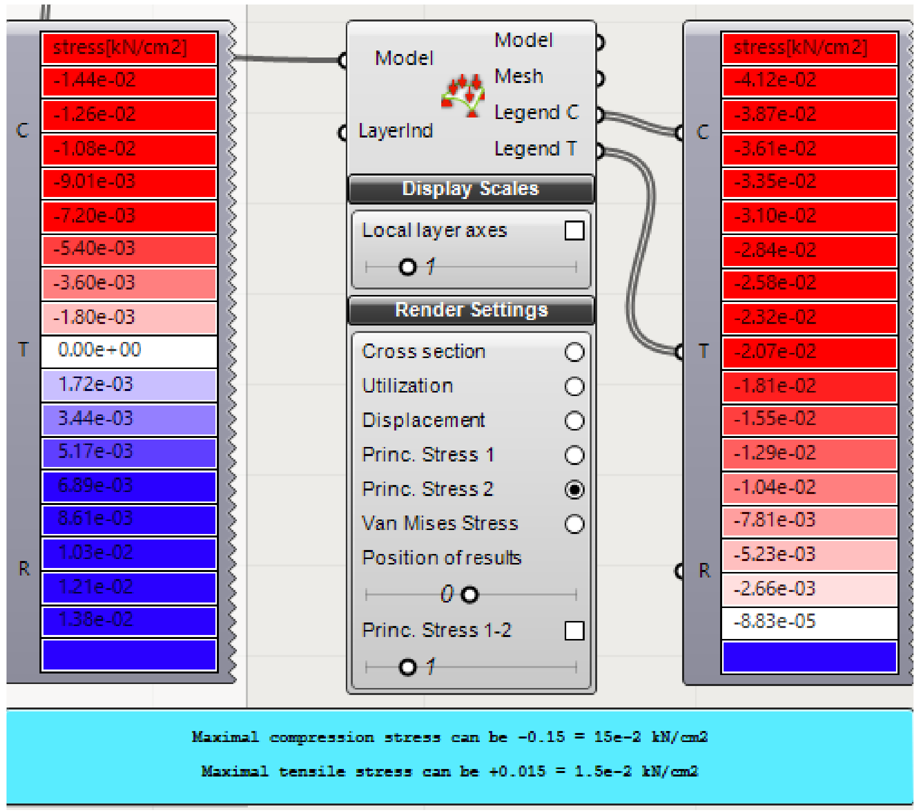


Displacement Utilization

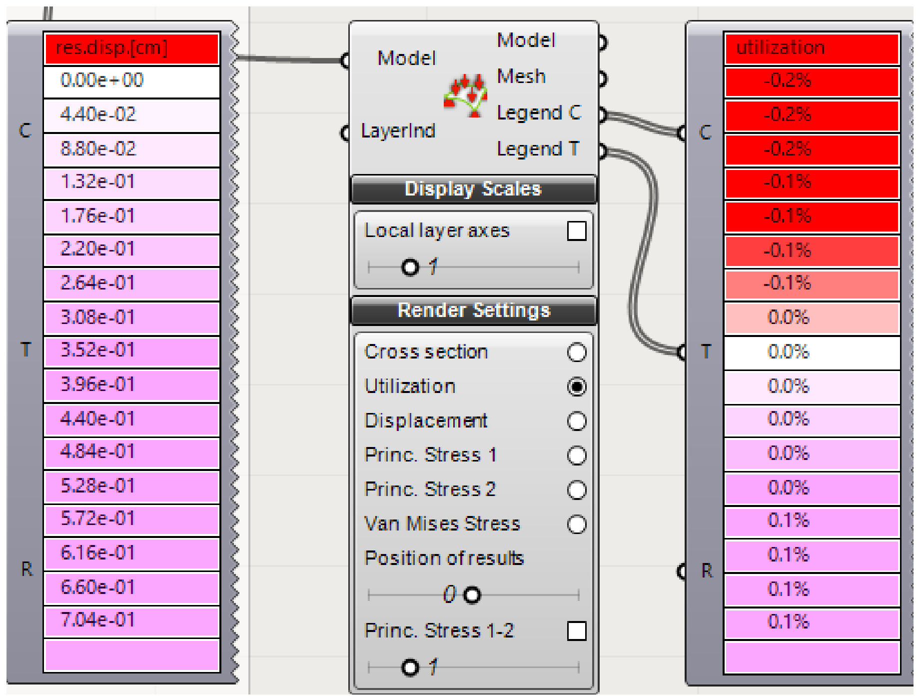


Structural Results: 3*2_Load Case 1

Princ. Stress 1 Princ. Stress 2



Displacement Utilization



CARE
+THY