Shifa' Mosaics

Health + Wellness Center

Alex Coelho - Juan Cruz - Georgina Giassia - Thomas Lindemann – Tim Schumann

Table of Content

1.1 | Site

The location of this project is the Zaatari refugee camp in Jordan. This sets already some boundaries regarding the design. The climate is very hot and dry (desert), and building materials are rare and have to be carried with huge effort over long distances. Therefore, taking earth as a building material is a reasonable approach.

Figure 1: Camp Overview (source: maps.google.com)

1.2 | Design Problem

In the refugee camp, we found three main problem areas, we wanted to tackle in our design:

The Social Factor

Lack of social space and the extreme psychological situation to live in an artificial city impact social issues.

The Physical Health Factor

Due to war injuries and people in the camp suffer from physical issues and need intense physical treatments.

The Health Factor

injuries and mental health problems are common in a medical camp. The COVID-19 pandemic added another challenge to that.

lack of social space

difficulty to have a normal social life

physical suffering

lack of water hygiene

injuries and infections

mental health problems

1.2 | Design Vision

Out of the three main problem areas we want to create a unique building typology that contains the areas „community centre ", „sports centre + thermal bath" and „health centre ".

The focus is therefore to add to the health and pleasure of the camp inhabitants in a general way. The strength of the project is the collective of all these various functions to one whole complex.

This is also reflected in the name „Shifa's Mosaic". Shifa' is the Arabic expression for healing and is often used as a name for women that are supposed to heal. Mosaics refers to the variety of the building functions and its rectangular, but eclectic arrangements. The building, its arrangement, as same as mosaics are working referring to the theory of emergence: "The sum of the parts is greater than the pieces themselves".

Reduce Mental Stress

tightening society:
community center

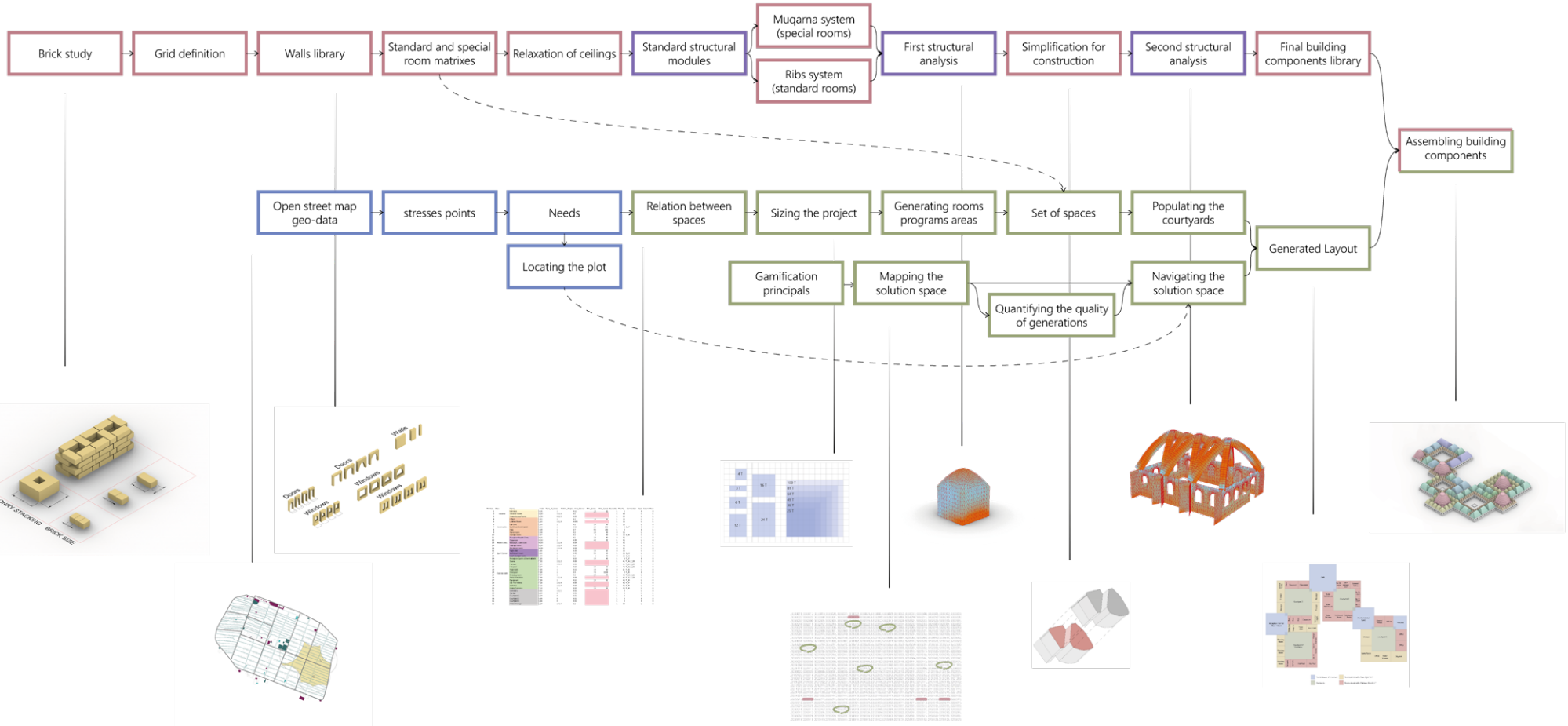
Reduce Physical Pain

increase wellbeing:
Sports Center +
Thermal Bath

Prevent Illness

prevention + education:
Health Center

1.3 | Design Process



2.1 I Camp Scale

Al Zaatari Refugee camp in Mafraq, north of Jordan, is currently the second-largest camp in the world. Moreover, represent the fourth largest city in the country. With an area of 530 hectares, this non-permanent settlement has gradually transformed into a town with 80,000 Syrians. Despite the local government's efforts to plan a small-medium scale grid, the rapid increment of Syrian immigrants into the country required adjusting the urban design. Thus, what started with the placement of two sectors, rapidly was transformed into 12 districts.

The rapid urban growth showed a higher concentration of units per area in districts number 1 and 2 (initial stages of the camp), with a population of 2.817 and 3,458 habitats: these two areas evidence agglomeration and lack of organization. And therefore, the need for exponential urban growth. Consequently, the development of the following sectors of the camp showed an improvement in the urban grid and the placement of the residential units.

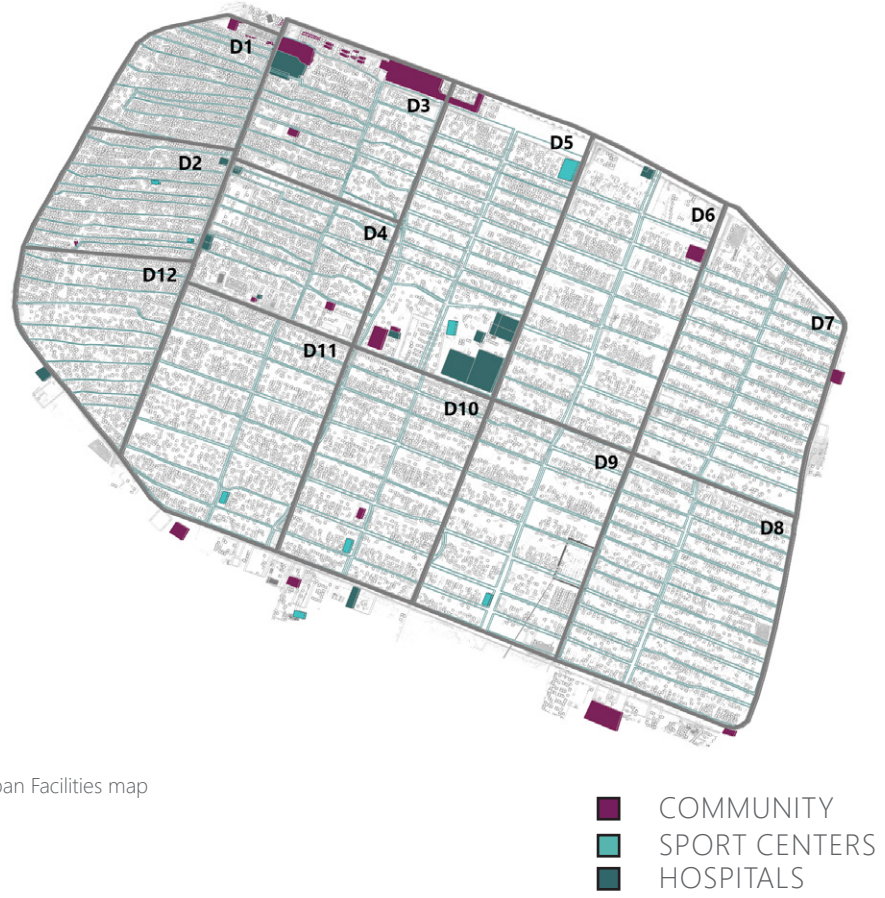


Figure 2: Urban Facilities map

2.2 I Methodology

The urban analysis process uses two primary sources of data recollection. First, an excel file that describes the population density per district. This document allows identifying overpopulation areas and infrastructure deficits concerning inhabitant numbers. Second, using Geo-location data from https://www.openstreetmap.org/, information is recollected throughout selecting an urban plot and the later export of this information into an OSM file.

The group created a script that provides mapping information into a rhino file from satellite data using grasshopper and Urbano Plugin. Thus, identifying the current camp state allows the classification of relevant components focused on streets, blocks, housing, hospitals, communal centers, and sports facilities. The primary purpose of this process aims to recognize areas in a deficit of specific activities to choose the most suitable project location.

Figure 3: Urban Plot from open street maps

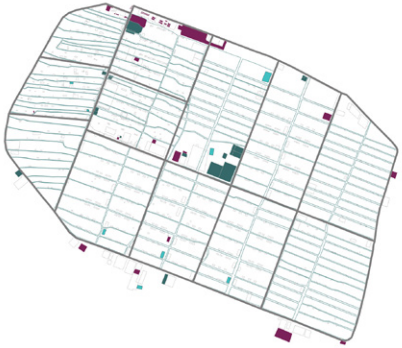
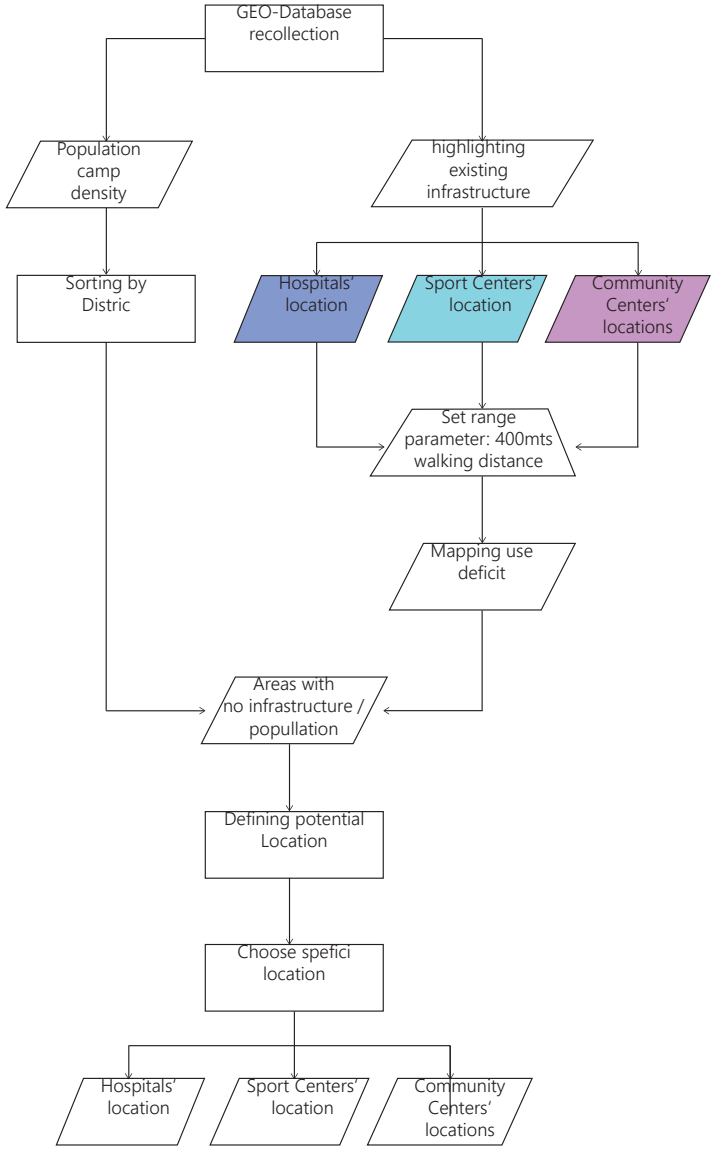


Figure 4: Identification of camp Facilities

Tools



2.3 I Urban Analysis

As previously explained, the methodology applies population density information and geo-reference mapping information into a grasshopper script. Moreover, it focuses on three main categories; Hospital, communal centers, and sports facilities. Due to project objectives, other facilities are discarded in this analysis, as the project aims to cover specific functionalities. The study assigned to each uses a range radius of 400 meters. Therefore, a walking distance parameter from the center of each point is set, stressing access points available for various districts. After giving a radius parameter to each coordinate or, in this case, buildings." The remaining areas that are not covered show possible places for project locations. Following the repetition of this process for each category, the script produces a combination of ranges. And finally, the analysis shows the areas of the city where camp facilities are absent of services.

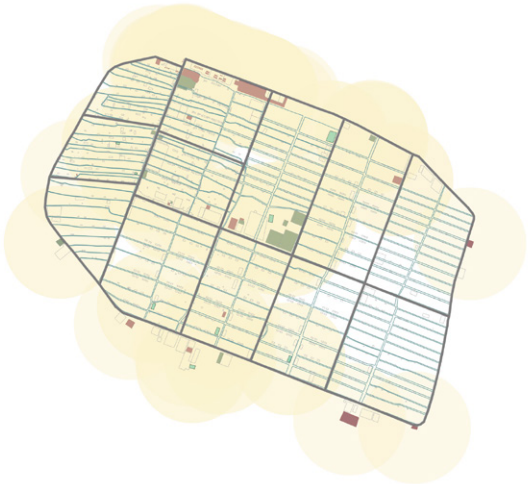


Figure 11: Remain Areas in a total absence of facilities

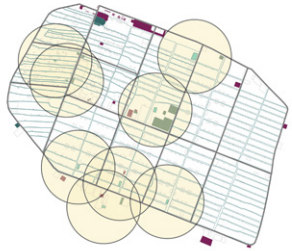


Figure 5: Sport Facilities - 400 mts range

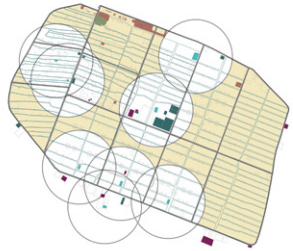


Figure 6: Lacking areas of Sport Facilities

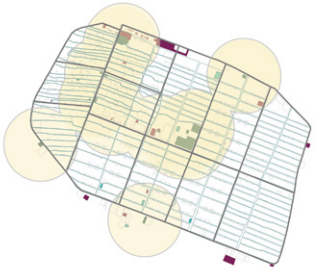


Figure 7: Hospital Centers - 400 mts range

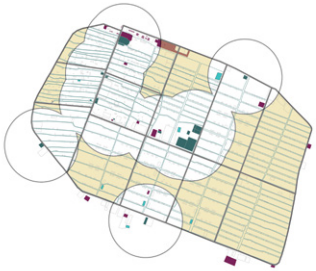


Figure 8: Lacking areas of Hospital services

Figure 9: Community Centers 400 mts range

Figure 10: Lacking areas of Community Centers

2.4 I Location Generation

Finally, after the script has identified the camp areas in complete lack of services. The process is repeated, setting tree cores of 400 meters range. The purpose of this step aims to identify an actual project plot and prove that the project could potentially cover the demand. However, this stage required the evaluation of the existing urban grid and the current conditions of the camp. Therefore, after approximating the possible location, the group decided to place the building in a specific domain where the intervention is minimum, which means that this analog process tries to seek empty areas where the placement of the project will not interfere with the existing social and urban grid.

Notice that the project placement has set an area parameter minimum of 45 x 45 meters and a maximum of 80 x 80 meters. Consequently, this decision pretends to work with an existing structure of blocks and housing composition. Despite the fact, the project is an entirely generative design focus; For this specific case, the group has chosen this particular location as it suits the constraints and conditions of the urban analysis. .

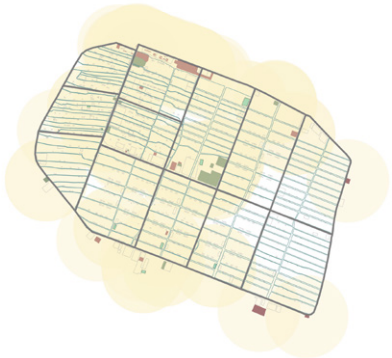


Figure 12: Intersection of several uses range

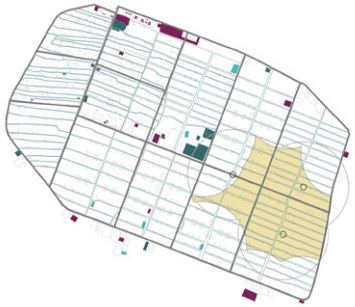


Figure 13: Remain Areas in a total absence of facilities

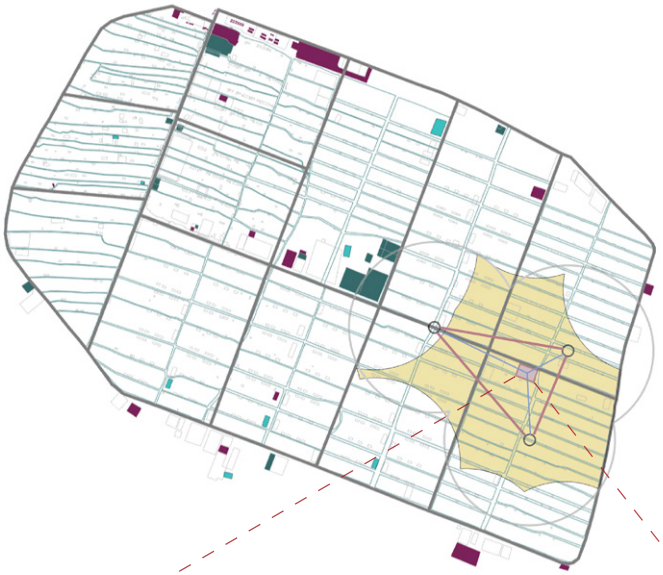
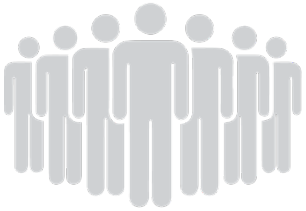


Figure 14: Project Plot

2.5 | Capacity Calculation

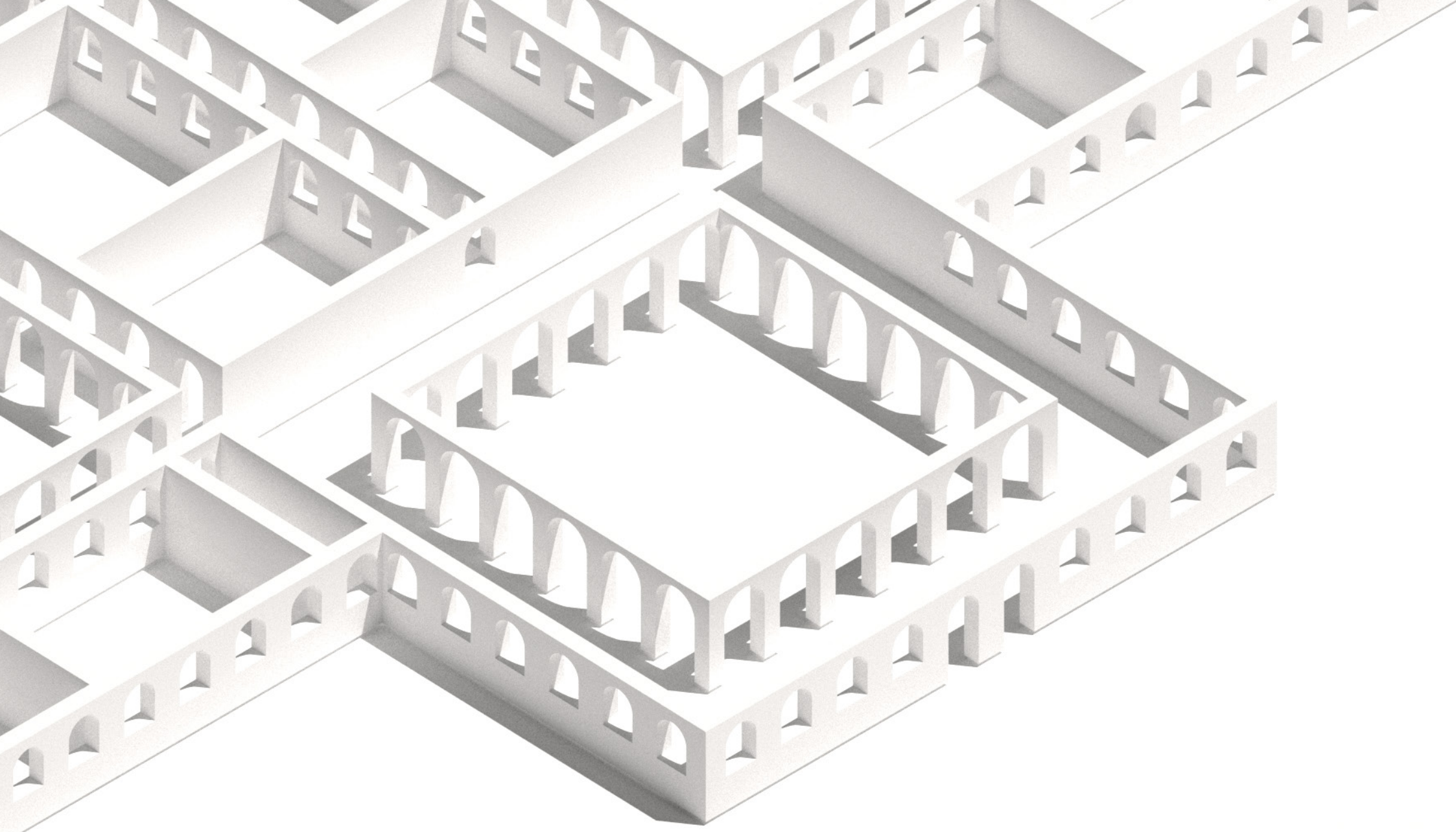
TARGET POPULATION

First step at identifying the target population for the new facility is to define the ratio between users and the existing facilities for each of the functions analysed before - sports, community and health. These numbers will be the capacity threshold.
From the urban analysis, the number of needed facilities in order to ensure full coverage of the camp (meaning that every person would be within a 400m radius circle from a centre of any given function) is obtained and, by combining with the existing ones, gives us the ideal total number of centres to be installed in the camp.
When dividing the total number of inhabitants by the quantities obtained above the ideal maximum of users per centre is defined, by assuming that, once all centres are installed, there would be an equal distribution of the population among all facilities. Since the number of total facilities is different per function, the target populations will also be unique.
On the other hand, it is estimated the amount of time each user would spend in a facility and the amount of visits per day/week or month.
By crossing all the data above, a total number of simultaneously users is estimated per function and for the total building.
This is the fundamental output from the urban analysis to start the configuring process and define the building program and sizes.



78,086

	Existing Facilities	Approx Rate	Needed Facilities	Total Facilities	Equal Populational Distribution	Target Population
	qty	users per centre	qty	qty	inhabitants	inhabitants
	8	9,760	7	15	5,205	5,205
	17	4,590	9	26	3,003	3,003
	9	8,670	7	16	4,880	4,880



3.1.1 I Participation

The design process for the Shifa' Mosaics allows the participation of three players: the architect, the camp manager and the refugees. They influence the design based on their skill level, responsibility and weight.

The Architect

The Architect can influence the typology of the building, he can transform for example a health-oriented centre into a more sports-oriented centre. Therefore he can create and delete functions and change adjacencies between individual rooms.

Camp Manager

The camp manager has a less powerful influence on the design. He can influence the factor, how much space is assigned to a specific function per person. Furthermore, he can set, if a function is necessary for the project or not. Therefore, he has a scaling tool for the centre and can adjust the size manually regarding specific needs.

Refugees

The refugees influence the size of the building. Regarding in what part of the camp they are located, and how intensely they are using the functional areas of sport, health and community, the size of the building will become smaller or larger. The Influence of the refugees is expressed in the Sizing as an outcome of the urban analysis.

The specific design parameters influenceable by the players are displayed in the graphic at the right.

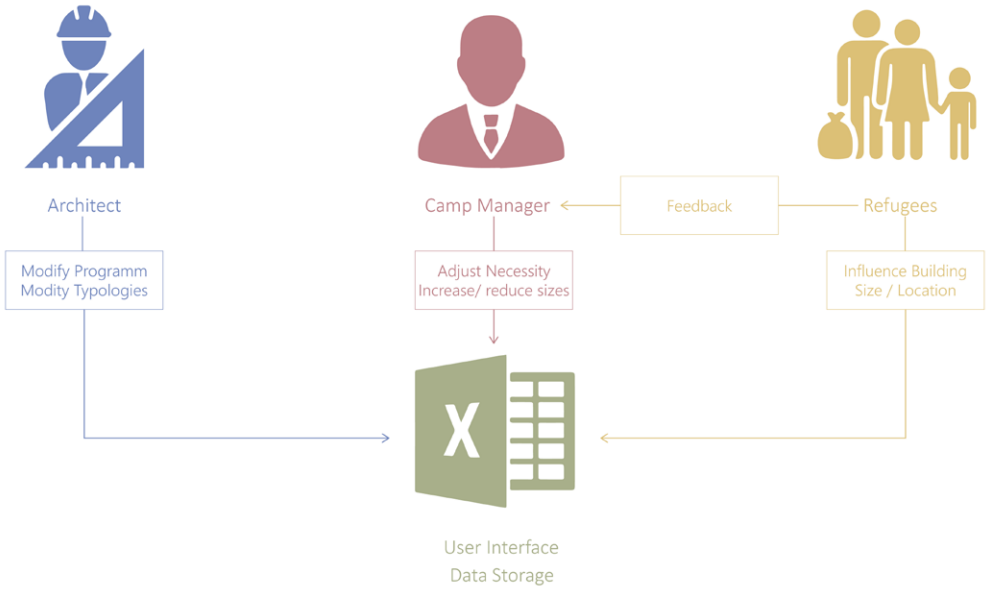


Figure 15: Players

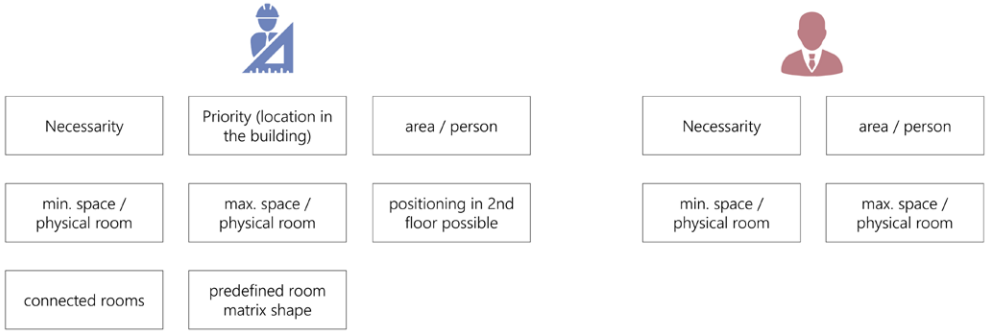


Figure 16: Modification Possibillities of each player

3.1.2 I Excel User Interface

Excel was chosen as the main platform for data input. That choice was convenient in the development of the automation of our project, as same as the possible application at the camp location.

The Excel file is embedded in the digital workflow. Code elements in Python will first read the excel, convert functions into physical room spaces, and write them back into the file. Then, a second python script accesses the file, reads the created room spaces and generated the building layout.

	A	B	C	D	F	G	H	I	J	K
1	Number	Class	Name	Index	Matrix_Shape	Area_Person	Min_Space	Max_Space	Necessity	Priority
2	1	General	Entrance	U_01		0.12	15	50	1	2
3	2		General Toilets	U_02	2, 3	0.07			1	99
4	3		Water Access Points	U_03	1, 1	0.03			1	99
5	4	Community	Office	C_01		0.2	10	30	1	11
6	5		Utilities Room	C_02	1, 3	0.005			1	11
7	6		DayCare	C_03		0.2	10	70	1	11
8	7		Multi-functional space	C_04		0.50	20	150	1	3
9	8		Café	C_05		0.7	50	250	1	20
10	9		Game room	C_06		0.2	10	70	0	11
11	10		Storage room	C_07		0.2	2	50	0	11
12	11	Health Clinic	Reception Health Clinic	H_01		0.1	10	40	0	31
13	12		Classroom	H_02		0.35	40	50	1	31
14	13		Massage / care room	H_03	2, 4	0.09			0	31
15	14		Therapy room	H_04	2, 4	0.09			0	31
16	15		Procedure room	H_05	2, 4	0.09			0	31
17	16	Sport Center	Yoga class	S_01		0.25	20	70	0	21
18	17		Multisport room	S_02		0.45	50	150	1	21
19	18		Sport storage room	S_03		0.2	2	50	0	21
20	19	Thermal Bath	Reception Sports &ThermalBath	T_01		0.15	15	70	1	21
21	20		Sauna	T_02	2, 2	0.06			1	21
22	21		Hamam	T_03	2, 2	0.06			1	21
23	22		Hot pool	T_04		0.15	30	9999	0	21
24	23		Tepid bath	T_05		0.10	20	9999	0	21
25	24		Cold pool	T_06		0.4	70	9999	1	21
26	25		Dressing room	T_07		0.2	20	30	1	21
27	26		Temp Transition	T_08	1, 4	0.08			0	21
28	27		Equipment	T_09		0.12	20	60	0	21
29	28		S & ThB Toilets	T_10	3, 4	0,05			0	21
30	29		Showers	T_11	2, 3	0.03			1	99
31	30		Water Technics	T_12		0.15	10	30	1	21
32	31	Special Spaces	Sport hall (court)	X_01	10, 17	1.5			0	21
33	32		Corridors	X_02	1, 1	0.01			1	0
34	33		Garden	X_03		0.01			1	0
35	34		Courtyard 1	X_04		0.01			1	1
36	35		Courtyard 2	X_05		0.01			1	10
37	36		Courtyard 3	X_06		0.01			1	20
38	37		Water Storage + Stairs	U_04	4, 4	0.2			1	99
39										

Figure 17: Excel Input Mask

3.1.3 I REL-Chart

The room program is quite complex, as we consider the four building typologies „Community Center“, „Sports Center“, Health Center“ and „Thermal Bath“. Therefore, we identified 36 functional rooms, that our building can provide.

Simplification

To be able to generate a layout, we had to simplify the relation between individual rooms. The rooms assigned to one functional cluster are arranged in a courtyard. That has the advantage, that all rooms of one functional cluster have the same accessibility, the disadvantage is, that a procedural row of rooms, for example in the thermal bath, could not be considered. This is the room to improve in a further prospect of the project regarding the configuring phase.

Furthermore, we relinquished the use of a second floor. The configuring algorithm was able to divide the room program into two stories, but in the shaping phase, we focused on the gothic vaulting and therefore did not elaborate the structure of a second floor.

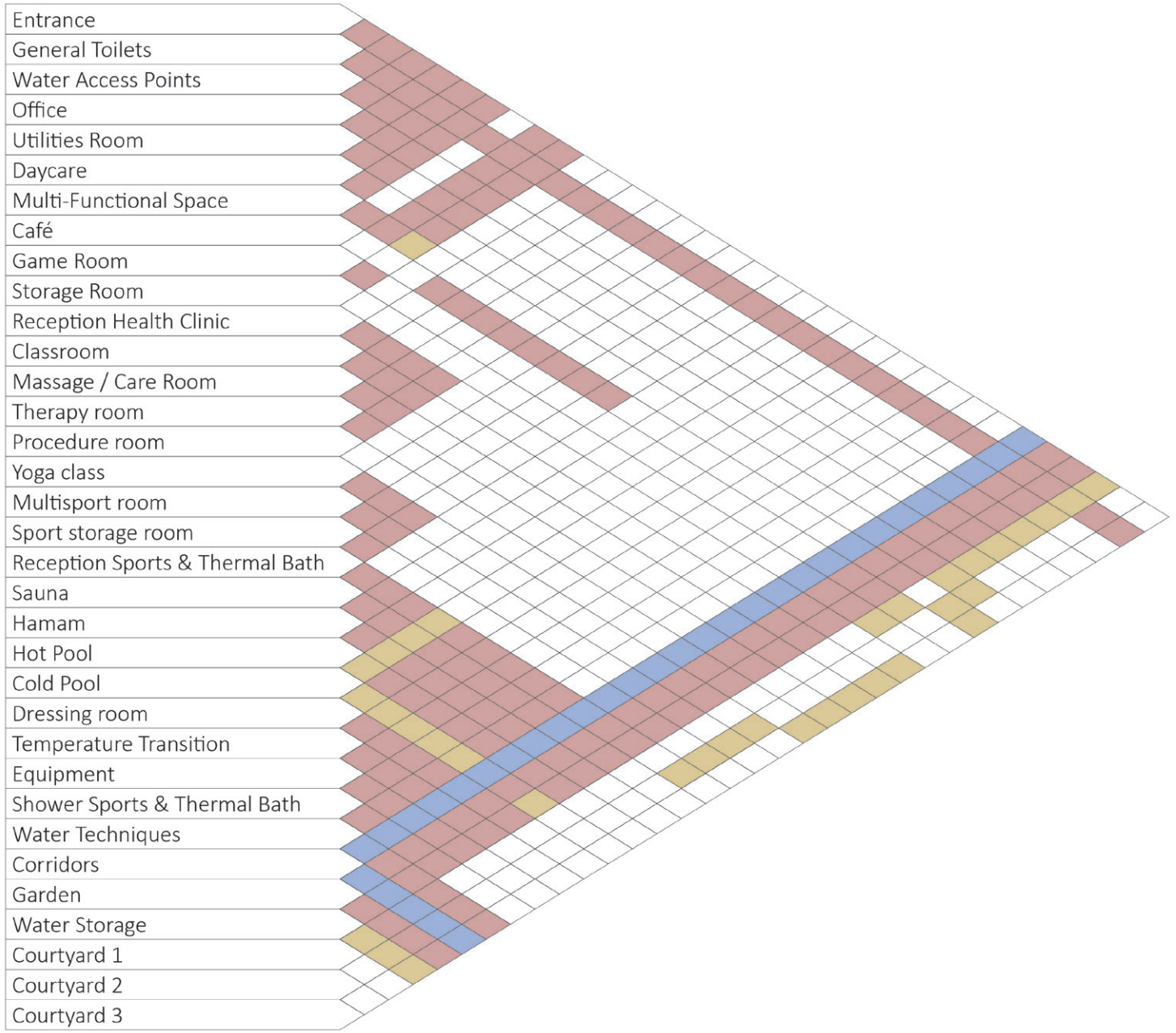


Figure 18: REL Chart

3.1.4 I Bubble Chart

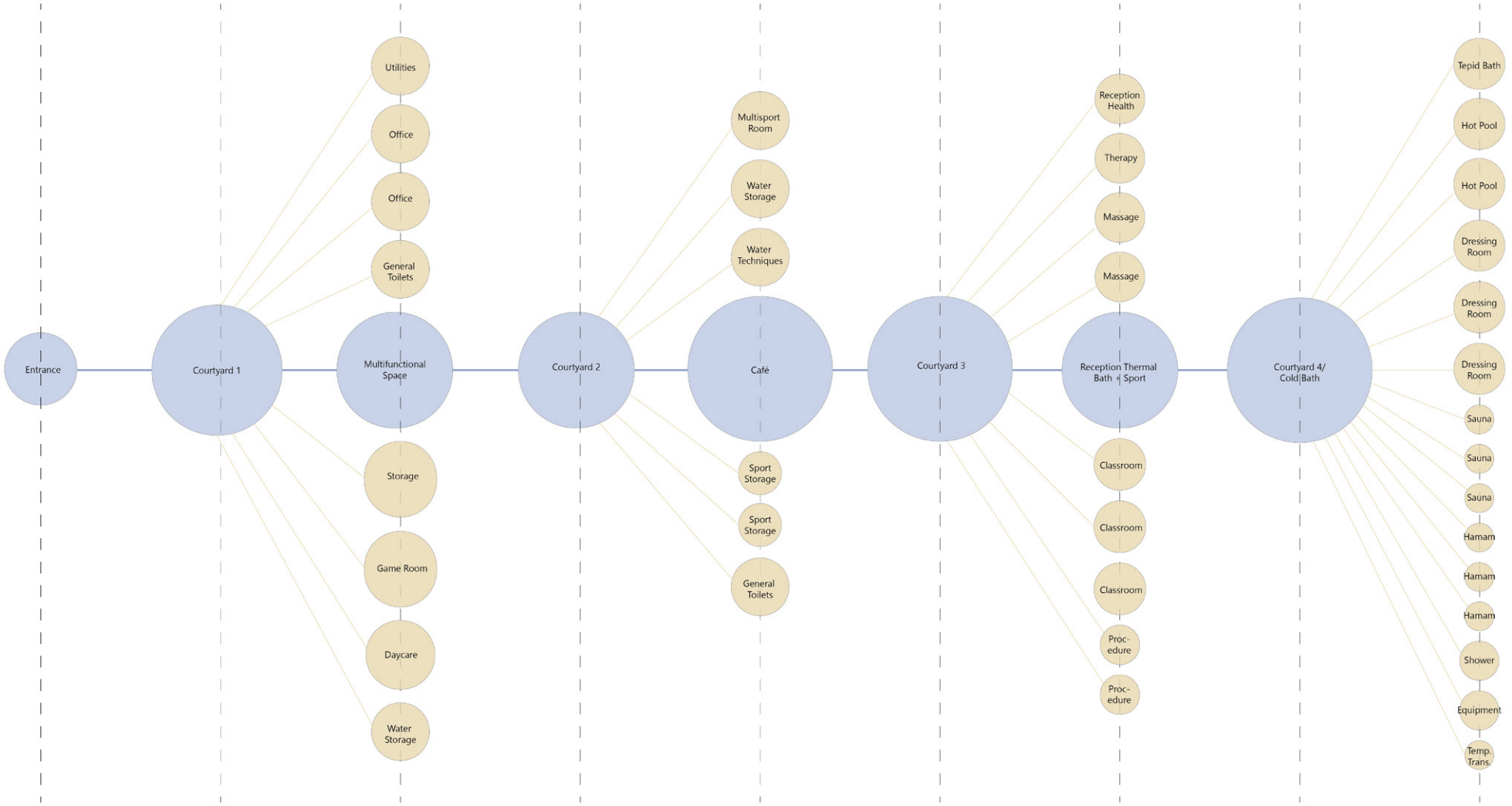


Figure 19: Bubble Chart Diagram

3.2.1 I General logic of gamification

GENERAL LOGIC

The principal of gamification chosen for our project is directly linked with our relational chart. Creating courtyards that follow each other and are connected together via a community space open to all. Each courtyard represent a cluster defined by the urban necessities. The decision of such rule is due to the fact that the Shifa' Mosaic project can be implemented in different location of the camp where the necessities can drastically change the number, size and content of each courtyards. Therefore this first gamification rule is sensible to our design vision and follow the program, as shown the figure below.

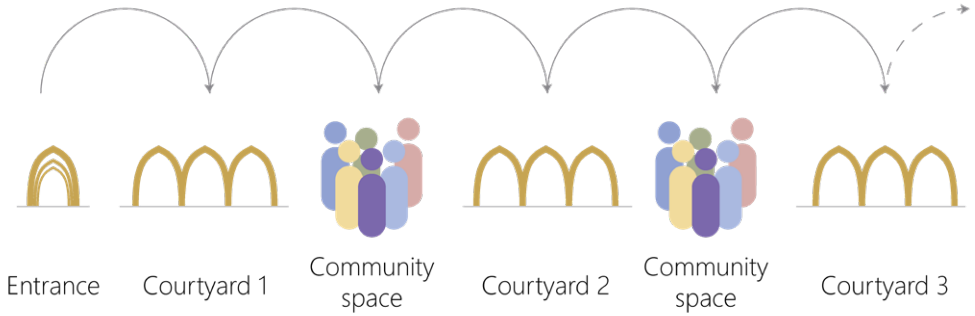


Figure 20: Space relation and sequence

This step is implemented using a specific column, in the common spreadsheet, to assign functions to a certain cluster or step in the general array. The column used is called "Priority". This column allow to group functions when they share a common value and define the different steps. The logic of numbering is:

0: Functions that will created internally in the python script
from 1 to 10: List, in order, of the spaces that will follow each other
11, 22, 33,...: Functions connected respectively to courtyard 1, 2, 3, etc...
99: Is used for the functions that need to be dispatched through the courtyards

SPATIALISATION

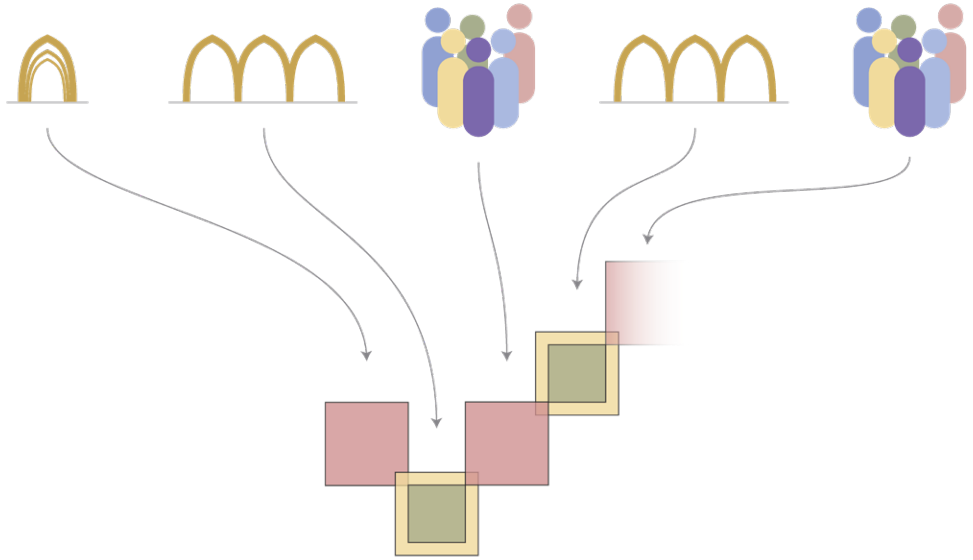


Figure 21: Space relation and spatialisation logic

The big steps themselves, which sizing is described in the previous chapter, connect between them from corner to corner. This connection type was inspired after the work of the Columbian architect named Rogelio slamona who uses the courtyards as connecting elements. The second reason of this spatialisation is that it can allow for more available area around each courtyard, as it can be seen in the next figure.

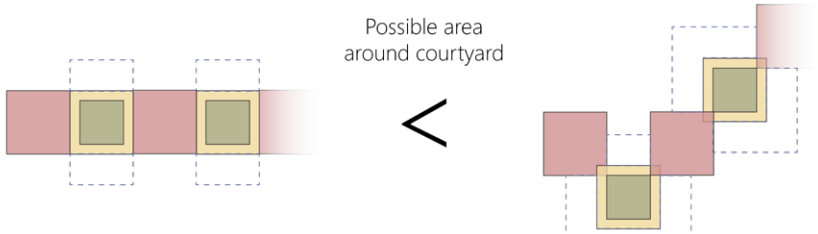


Figure 22: Choice of spatialisation method

MAPPING THE SOLUTION SPACE

After defining the logic of placement and the order of functions, mapping the solution space can start. Placing rectangle after rectangle, we have 3 options at every step, the amount of possible generations is exponential following the amount of steps in the chain. In our final design we had 7 steps, therefore $3^7 = 2187$ options. A small sample can be seen in the next figure.
Calculating and testing all of the options is not necessary as most of them will naturally step on top of each other.

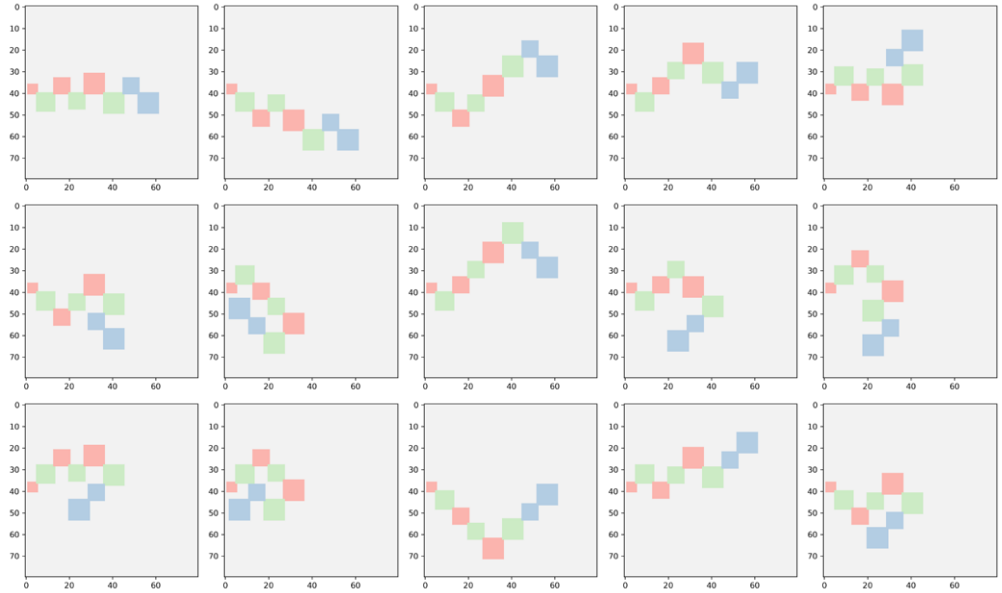


Figure 23: Sample of possible iteration, starting from an edge

For that reason a preliminar work has been done to map the first valid steps in order the reduce the overall amount of different generations. This preliminar work is done manually and up to a certain number of steps. The manually made tree can be seen in the next page. More steps can be added and will therefore be calculated in the python script, in the part"STEP 2: GENERATION>Mapping solutions: ID of Generation".

The amount of generation validated by hand changes quite drastically if the generation starts from a corner or an edge. After manual reduction of the first steps the amount of generation to test reduced from 2187 to 1024 options if started from an edge and 304 for a corner.

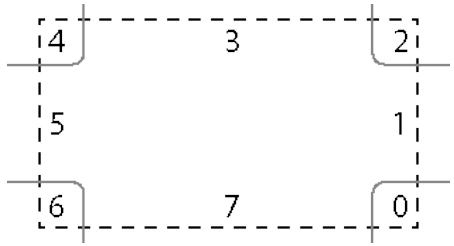


Figure 24: Entrance key value

In order to compute the different options, each directions taken after the starting point needed to be serialised from direction to IDs.

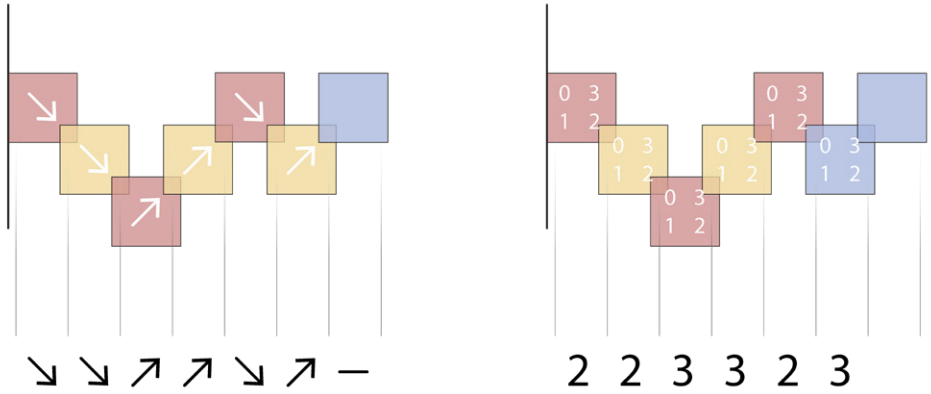


Figure 25: From direction to ID

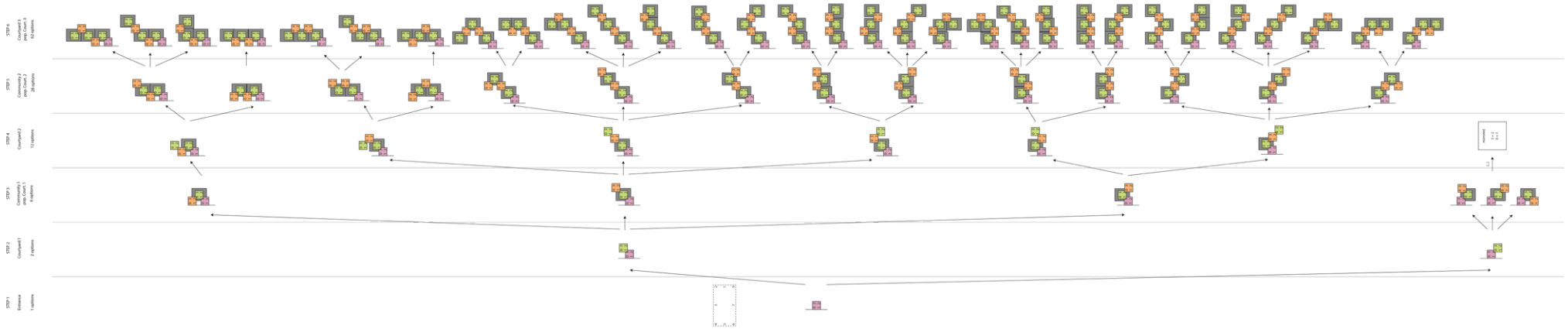


Figure 26: Manually validated generations - starting on a edge

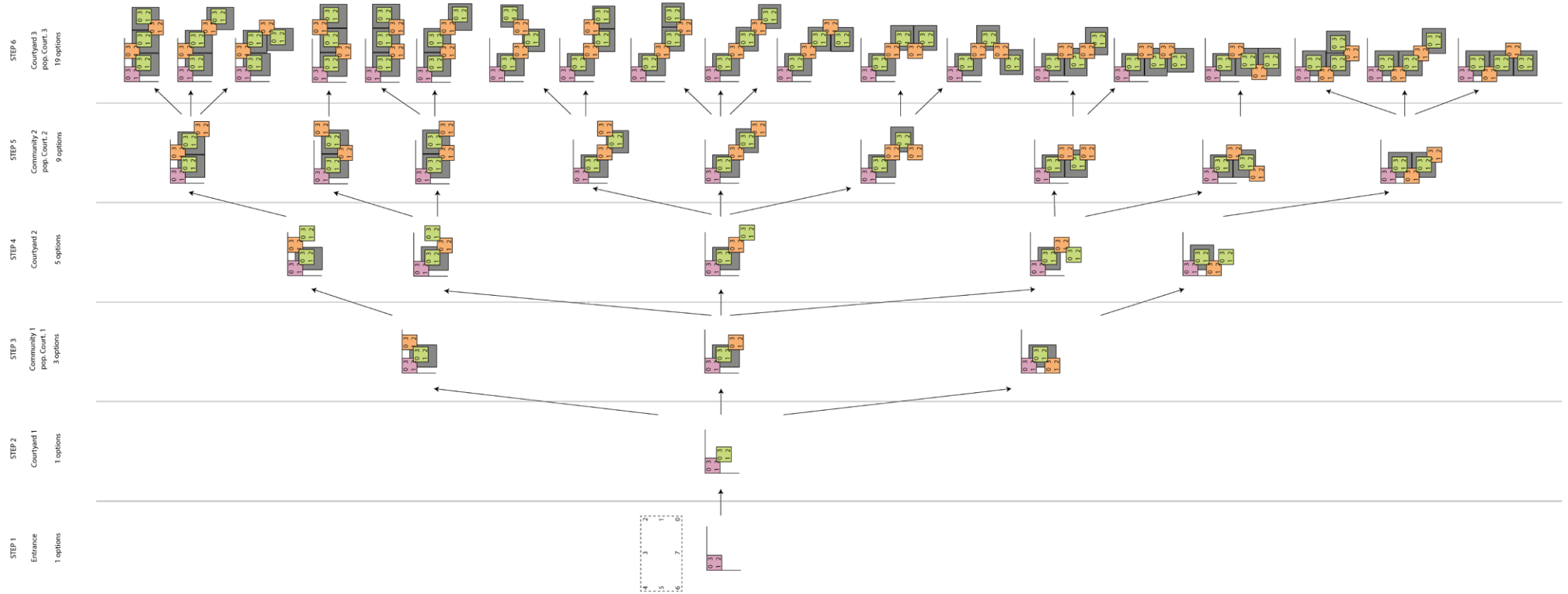


Figure 27: Manually validated generations - starting on a corner

3.2.2 I Quantify quality of generation

During the generation, several output can be computed. The script can either stop whenever it find a generation that fit into the plot or retrieve all the sucessfull iterations. Both can be tested in the script acessible on the github page of the project on the part "STEP 1". In the first output, sorting the different ID is crucial in other to receive the most interresting combination before the less interresting ones. On the other hand, because of the amount of IDs, receiving all valid generation could be overwhelming if the choosing process is manual. Automating a grading system being a whole chapter to develop on it own, we summerised our action by applying one process to sort the "in average best ones" from the "in average bad ones" and choose a small enough plot to receive a small amount of valid outputs. The parameters we used to sort the IDs is the variety of the directions present in the ID. Referring to the figure below, beside a specific need, the generation with more variety has a more interresting configuration.

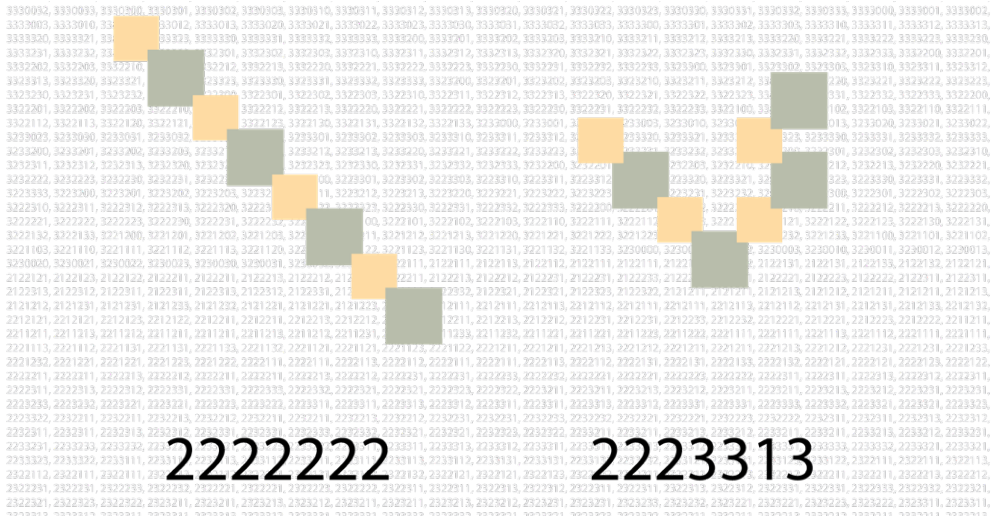


Figure 28: Sorting ID's by variety

Sorting the ID by variety is only one of the many steps that could be done. But due timeframe of the course, we can only leave the steps for further development. The two outputs option we currently have are by themselves incomplete, on one hand we receive one layout but not way to be sure we could have found a better one afterwards, and on the other side a manual process is needed before solving the adjacencies.

The optimum output would be to receive all the valid iterations sorted by quality. Defining such grading system fit perfectly our ID system, as each ID have more information inbedded in them than just the variety of the directions taken. Some examples are the sequences, repetition of patterns.
On the other side, applying stencils on each generated layout searching for specific relations could also be implement.
The compactness factor can be done producing the Nolly plan coupled with a distance diagram and counting the amount of direct neighbors.

Aside of producing a scoring system, design criteria should be sistematised as input to define with importance each parameters takes into defining the best score.

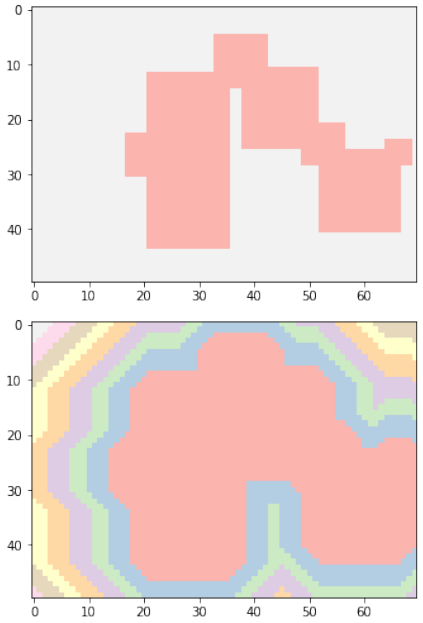


Figure 29: Nolly plan & gradient plan

3.2.3 I Sizing Courtyards

TRANSFORM A FUNCTIONAL PROGRAMM INTO SPACE - THE ROOM DEFINTION ALGORITHM

The first part of the configuring algorithm transforms the functional rooms into spatial room matrizes. Therefore, the area / person column of the room functions from excel is multiplied by the maximum building capacity of each sector. This way, the total area of a functional room in squaremeter is defined. To transform this area now into a room, a unit transformation from squaremeter to tatami units is performend (see below)

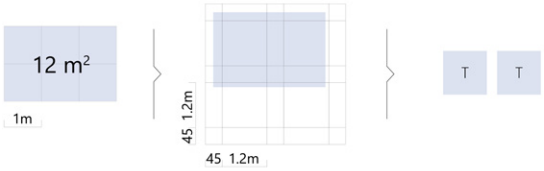


Figure 30: Unit Conversion

With the transition to the tatami-unit, the size of the tartan grid is concidered. This way, the area in tatami-modules is closely related to the previously calculated space. Sizing of Tatami-Grid and Tartan-Grid is elaborated in the section „Shaping".

The output of the Room Definition Algorithm is a new datatype of rooms, that is exported into Excel. Each room function has now an assigned matrix, a position in the building, and a quantity.

data model rooms input

room number	room name	room type	necessity	priority (location in the building)	area / person	min. space / physical room	max. space / physical room	fixed matrix (if yes, no min. and max. space)	positioning in 2nd floor possible
example									
2	general toilets	1	1	99	0.1	min. space / physical room	max. space / physical room	(3, 2)	1

data model rooms (converted)

room number	room name	room type	priority (location in the building)	matrix	number of rooms
example					
2	general toilets	1	99	(3, 2)	3

Figure 31: Data Models Rooms

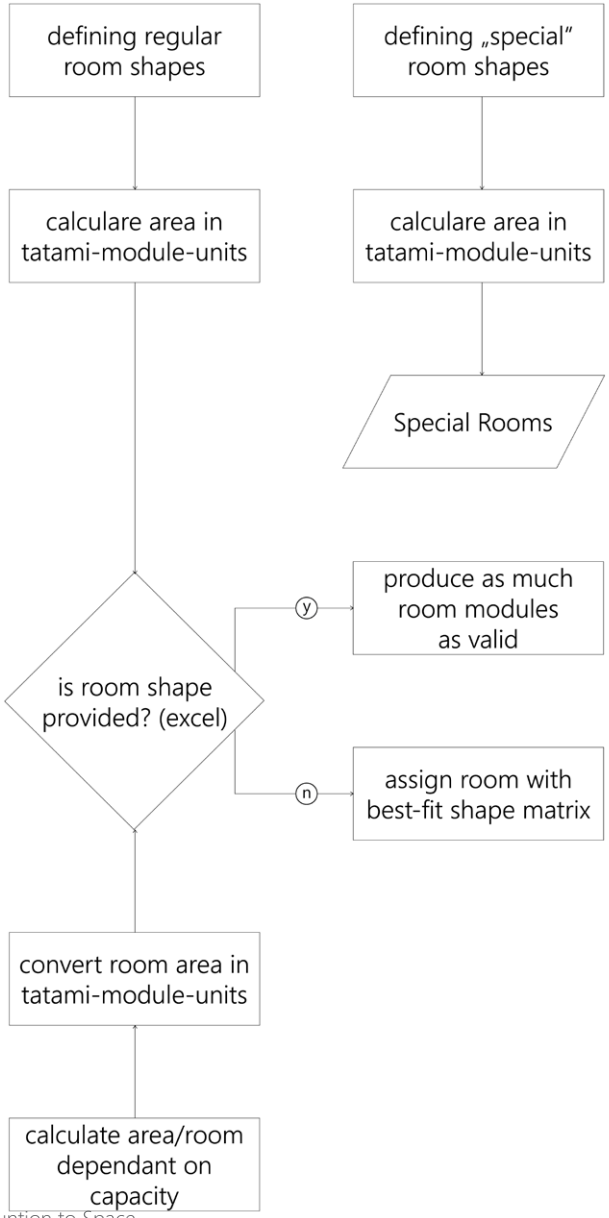


Figure 32: Flowchart Funtion to Space

After converting the area into tatami units, room matrices are generated. There are two methods to achieve that. The first method is applied to rooms that have a manual input for a room matrix. In that case, the area in tatami steps will be divided through the area of the predefined shape to get the number of rooms. The second method allows the assignment of one of the six regular room shapes to a room function. In that case, the room shape closest value of the area in tatami-steps is assigned to the function room.

With these two methods, the user input is transformed into a spatial set of rooms.

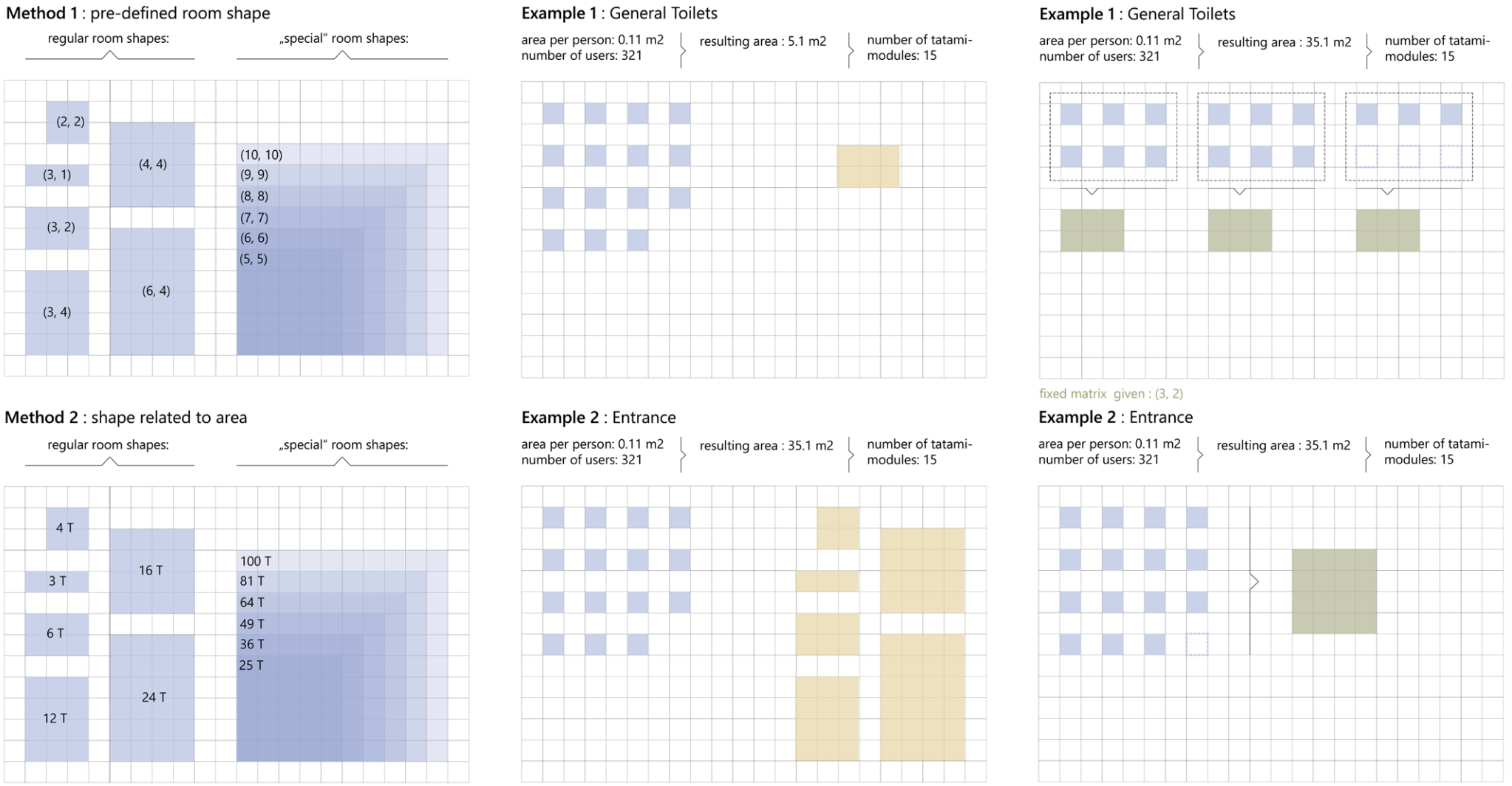


Figure 33: Room Sizing logic

The next step in the configuring phase is the sizing of the courtyard. For that, some rules are defined:
1. A courtyard is always squared
2. A courtyard has four „Slots" (edges) and four „Corners". The Corners and the slots that are used for populating rooms are selected in the gamification process (next chapter).
3. The size of the courtyard is calculated as followed:

balance factor = sum of all un-selected corners and edges * 8

courtyard widtht, lenght = (lenght of each assigned matrix + width of each assigned matrix + balance factor) / 8

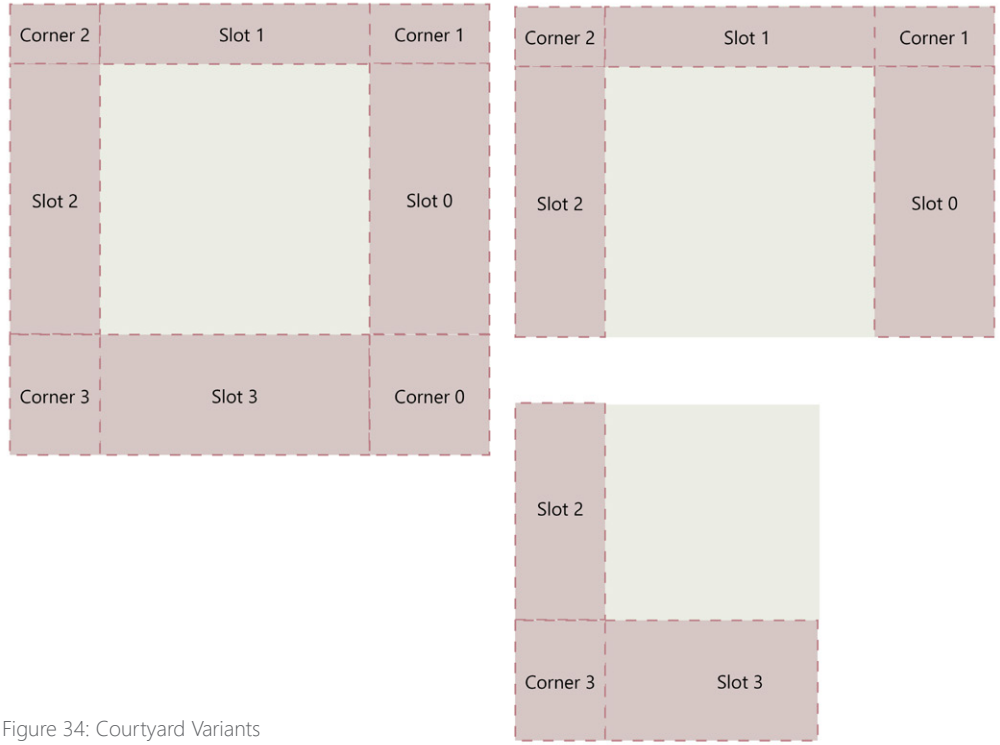


Figure 34: Courtyard Variants

SIZING THE POPULATION AREA

Once a slot/corner configuration is selected and the width and length of the courtyard are fixed, the width of each individual slot is distinguished. For that, each of the four slots is a width from 2, 3 or 4 assigned. Then the area is calculated. This step is repeated for all possible combinations, in the end, the combination out of the 495 options is selected, that is equal or slightly larger than the sum of the area of the assigned rooms.

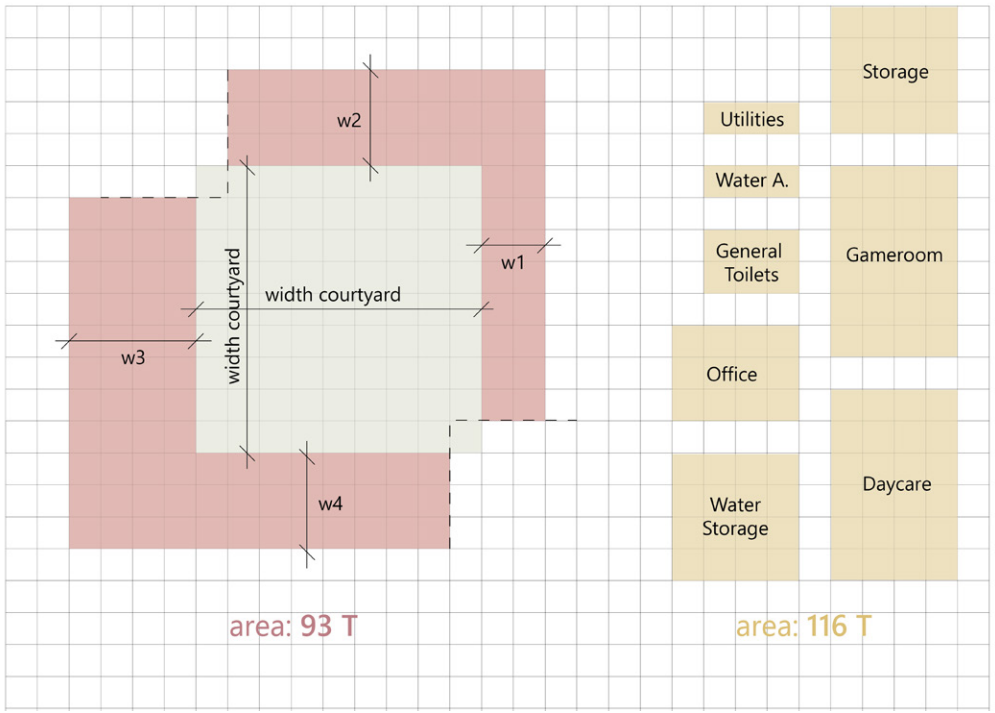


Figure 35: Computing the Population Area size

PREPARING POPULATION

Finally, the width of all four slots can be fixed.

In the next step, the corners will be assigned to the slot with a larger width. This has the advantage, that it is less likely that small rooms are placed in the corners which could cause complications with the accessibility.

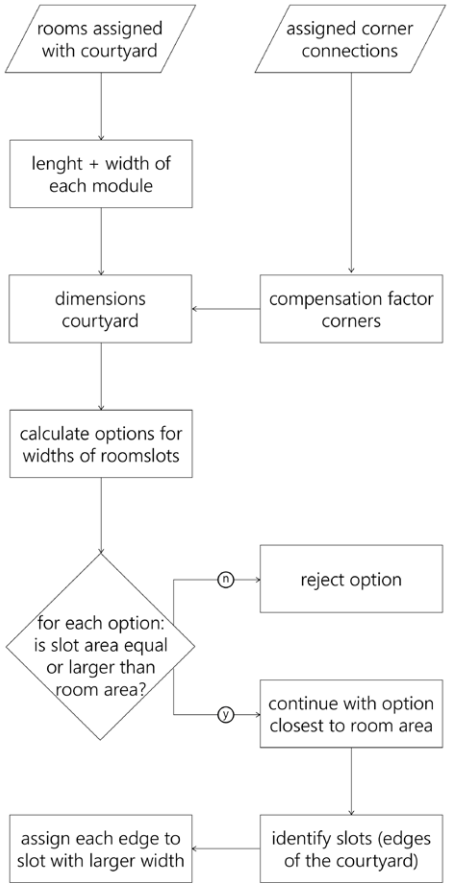


Figure 36: Flowchart Sizing Courtyards

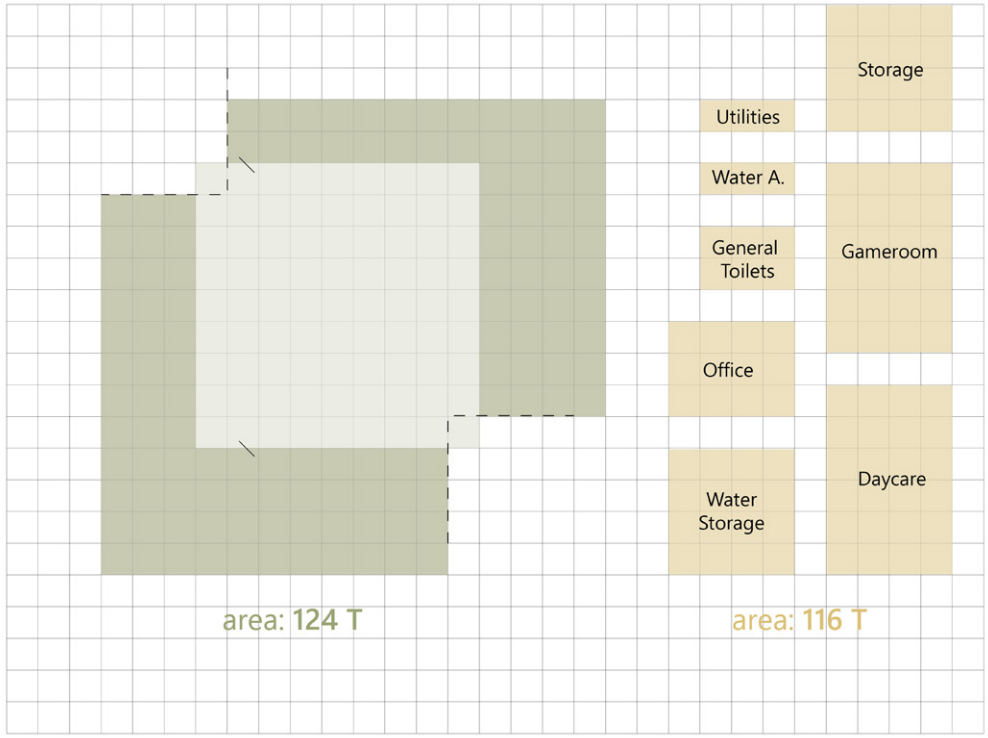


Figure 37: Valid Configuration

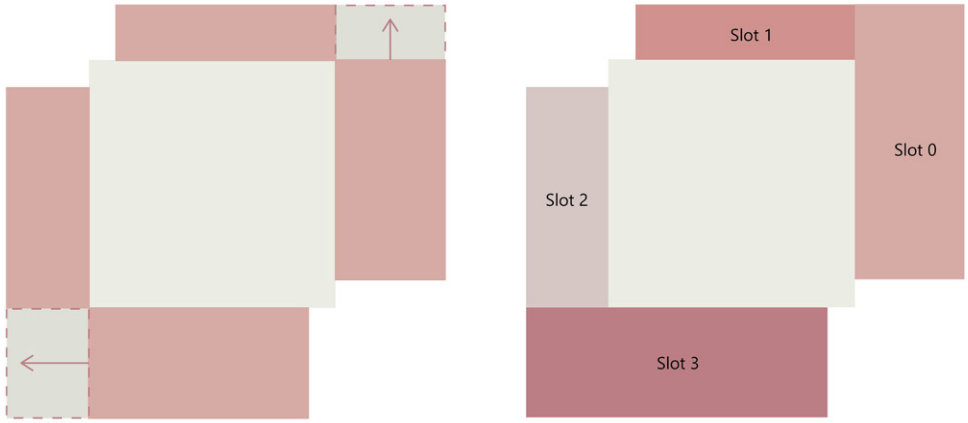


Figure 38: Corner Assignment

3.2.3 I Navigating the solution space

INPUT PARAMETERS

Before generating the layout, several inputs are needed. These several inputs have been explained in depth previously in their dedicated parts and they could be summerised and connected to the python script in that way. Choosing a plot at the urban level define the width and length of the plot inputed a x and y for the Array used in the generation. The location of the entrance within the plot is defined manually and is inputed as a key value. The formulation of the program is stored in dictionnary containing the following elements for each functions:

Repository = {index of function: [Name of function, shape of function , Type_of_ space, priority, Amount of function]}

Plot size from the urban Analysis

Location of the entrance on the plot

Formulation of the program

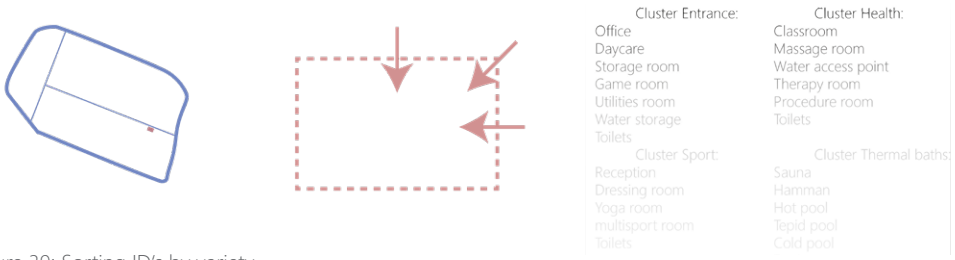


Figure 39: Sorting ID's by variety

Another feature that could be implemented as a next step is the design goals of the specific location. This add-on is directly linked to the rating system talked previously in the chapter: Mapping Solution.
In that next step, the entrance of the plot could then be integrated as one of the goals, testing one or several ones in parrallel.

MAIN STEPS OF THE SCRIPT

As it can be seen in the Vscode file. The first and second input were made manual as we focused more into a automated formulation of the program by bridging Excel to python. Similar input method could be achieve on the plot size and entrance location, therefore centralising every information on one spreadsheet.

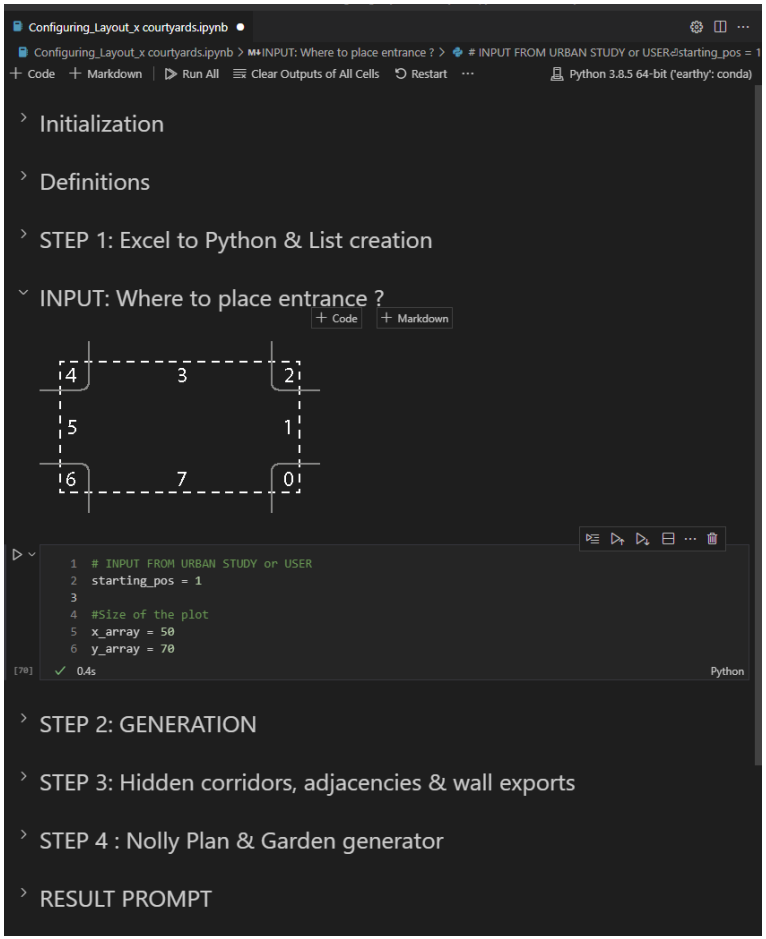


Figure 40: Parts of VsCode script

The generation part of the configuring is located in this STEP 2. It is composed of three subparts. The first one was explain in the previous chapter. The second one relate to the first two inputs of the the previous figure, size of the plot and location fo entrance. The output is a starting point and placement of the entrance, first item of the chain.

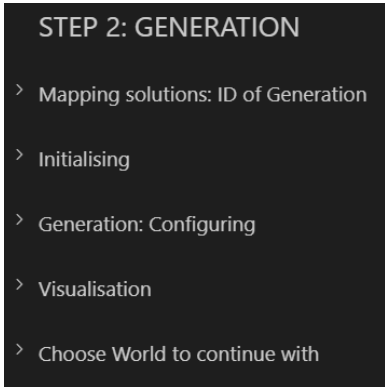


Figure 41: main sub-parts of STEP 2 of configuring

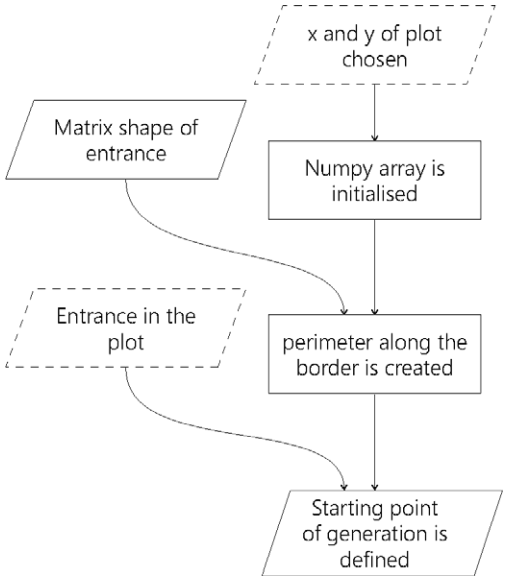


Figure 42: Flowchart "Initialising"

Once the first item of the chain is placed, the following ones can be placed following a different ID at each iteration. Let's remind that the list of ID used is related to entrance on the plot. Everytime the ID result in a sucessfull generation, the ID in question is stored and an image is captured for further use. Similarly the array itself could be stored to automate further operation of selection as next steps.

The next step is the Navigation it self, which consist of loops of loops selecting, checking and placing functions following each instruction given by each ID, and saving the sucessfull ones. The process is the following:

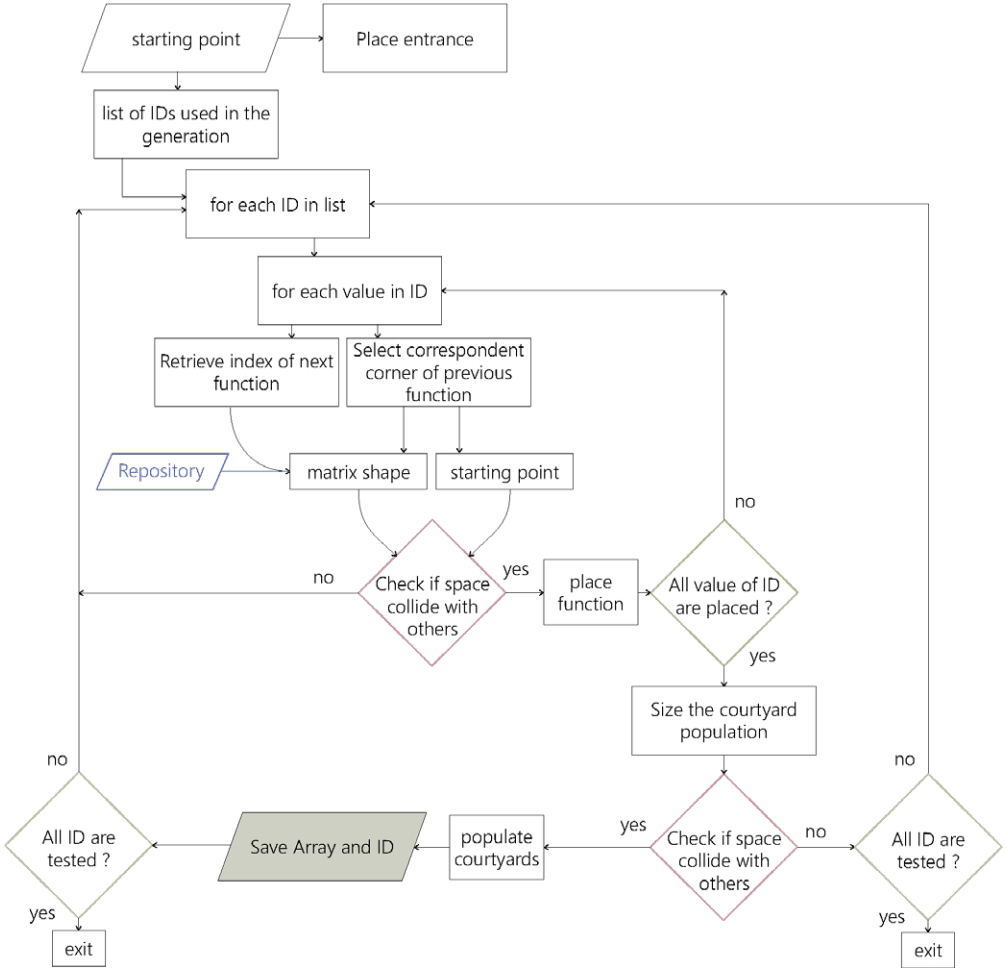


Figure 43: Flowchart "Generation"

The final arrays are visualised thanks to the Matplotlab library (cf script to know how).Once all the generation are produced and saved, a manual selection is done. But a stencil could run through each array and search for specific relation and define certain qualities for each generation. A scoring system would be implemented to find the closest match to the design goals of the wanted design.

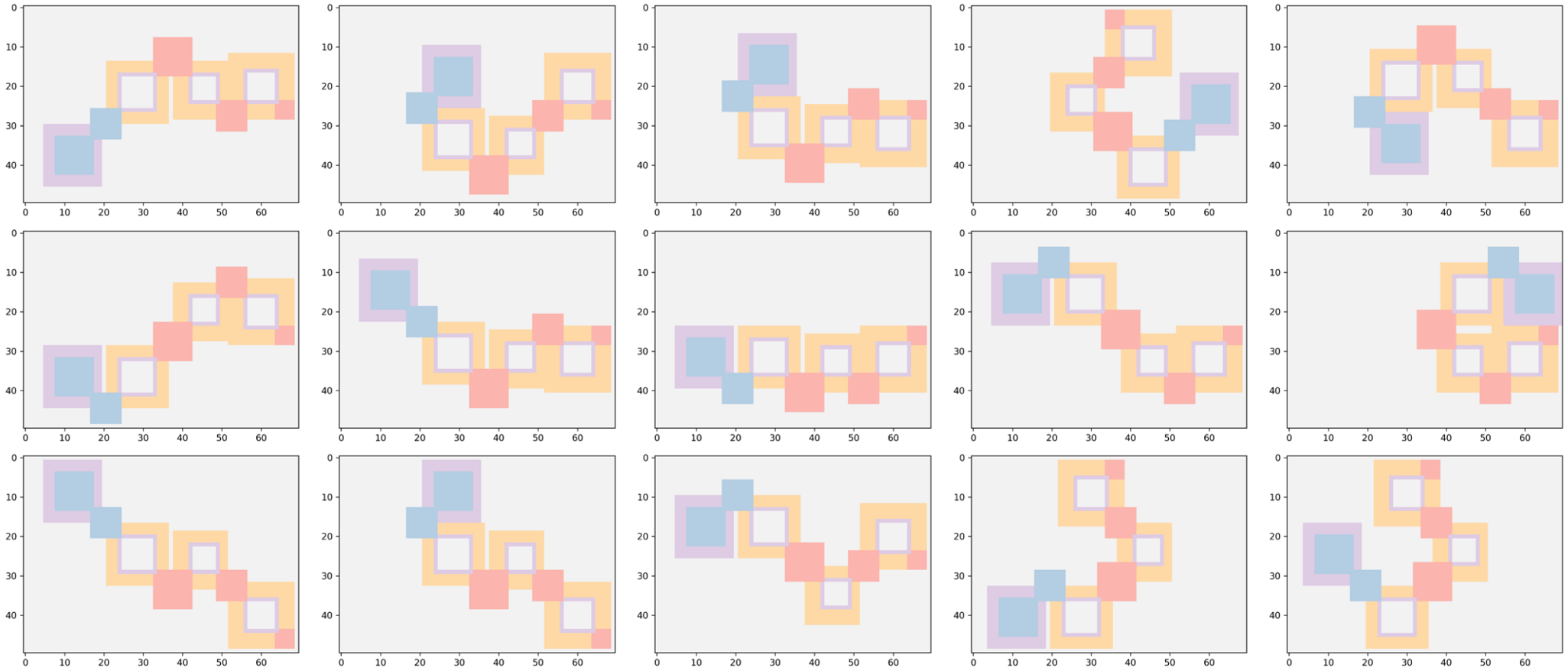


Figure 44: Sample of sucessull generation

3.3.1 I Populating Courtyards

Now, the courtyard and the populatable area is defined. In the next step, specific rooms will be assigned to a specific slot of the courtyard.

Two methods are applied to reach that goal. The first method „Clean Algorithm" tries to fill a slot with room matrices until it is filled and there is no overlapping between matrices. Furthermore, this problem is solved mathematically instead of geometrically.

To achieve that, the room matrices that are assigned to a courtyard are assigned to „queues". Queues are storing the information of the width of a room matrix, and leave therefore only the length as a factor. Rooms are assigned to queues for the three possible slot widths 2, 3 and 4. Since there are rooms that are valid for two queues (e.g. the room matrix (4, 3)), there are special transition queues.

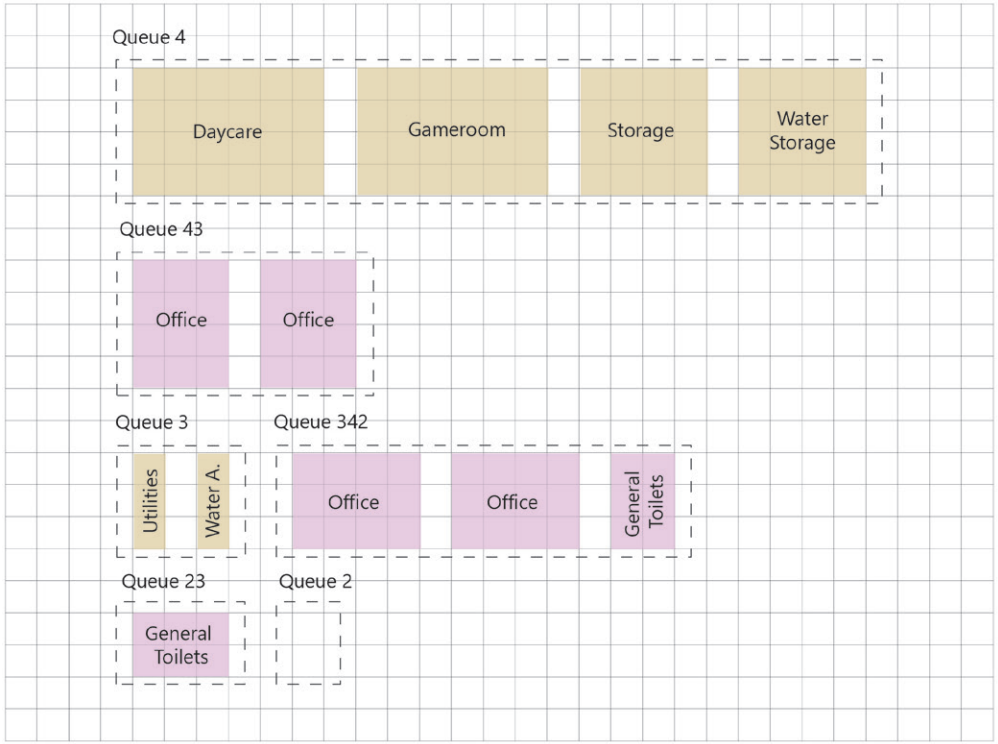


Figure 45: Queues

Having sorted all room elements into queues, the Knapsack-Problem applies. The Knapsack Problem describes a combinatorial problem, for that the combination of a set of items is equal or as close as possible to a fixed size „Knapsack". In this case, the Knapsack is the length of a slot, and the set of items is the assigned queue that equals the width of the slot. Now, the perfect combination of values to fill the slot is computed.

SIMPLIFICATION

The Knapsack problem in Python is solved as a Naive Recursive Solution. To optimize runtime, this problem should be converted into a Dynamic Programming Solution.

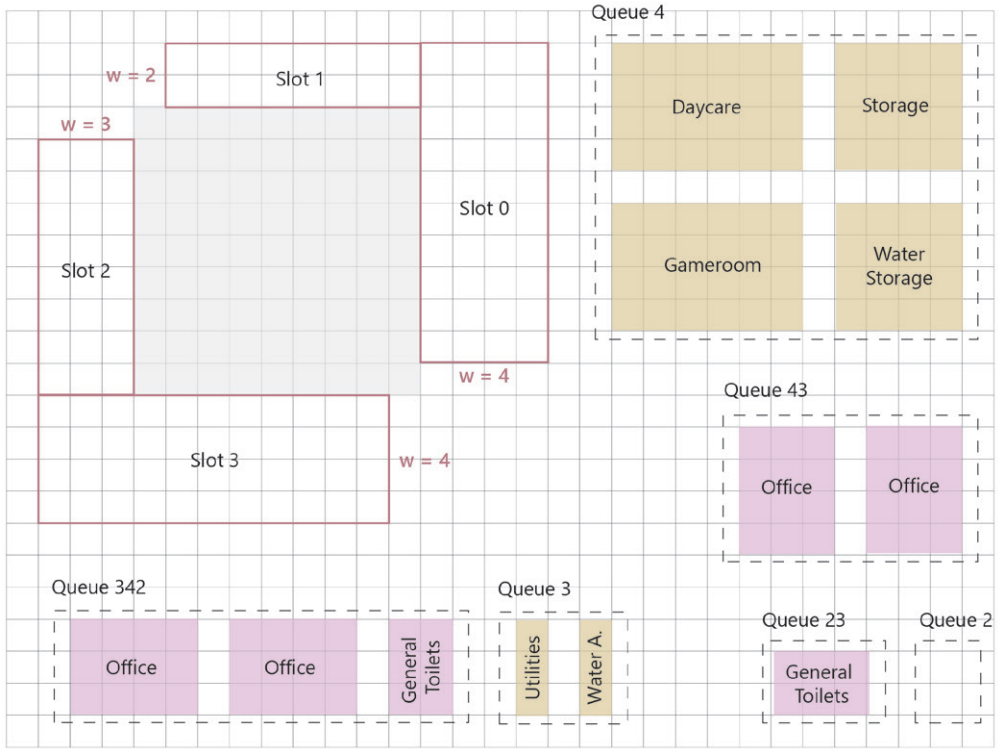


Figure 46: Input for Knapsack-Problem

As soon as a valid arrangement of rooms is found, these rooms are deleted from the queue and the algorithm is applied to the next slot. Since the width of the slots is independent of the actual length of queues, it is unlikely that all slots can be filled with this method (see example below: two out of 4 slots are filled).

To be able to assign all rooms to the courtyard, the second method is applied: the „Chainsaw Algorithm". The Chainsaw Algorithm does not destroy the modular order of the room matrices, it even guarantees the clean transition between configuring and shaping phase. The Chainsaw Algorithm „cuts" the slots that are still empty into smaller pieces out of the set of the initial regular shapes.

Following Conditions are taken into account:

1. The widths and lengths of the assigned matrices have to fill the slot completely
2. The total number of matrices must be equal to or larger than the remaining rooms in the assignment queues
3. The assigned matrices must be as large as possible

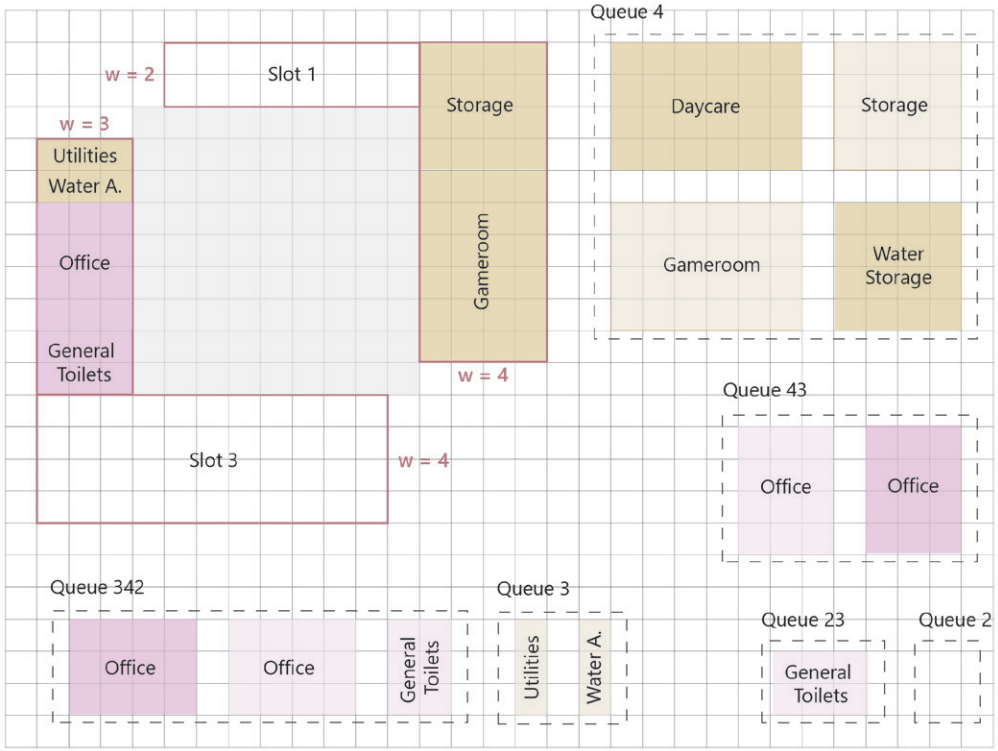


Figure 47: Clean Algorithm

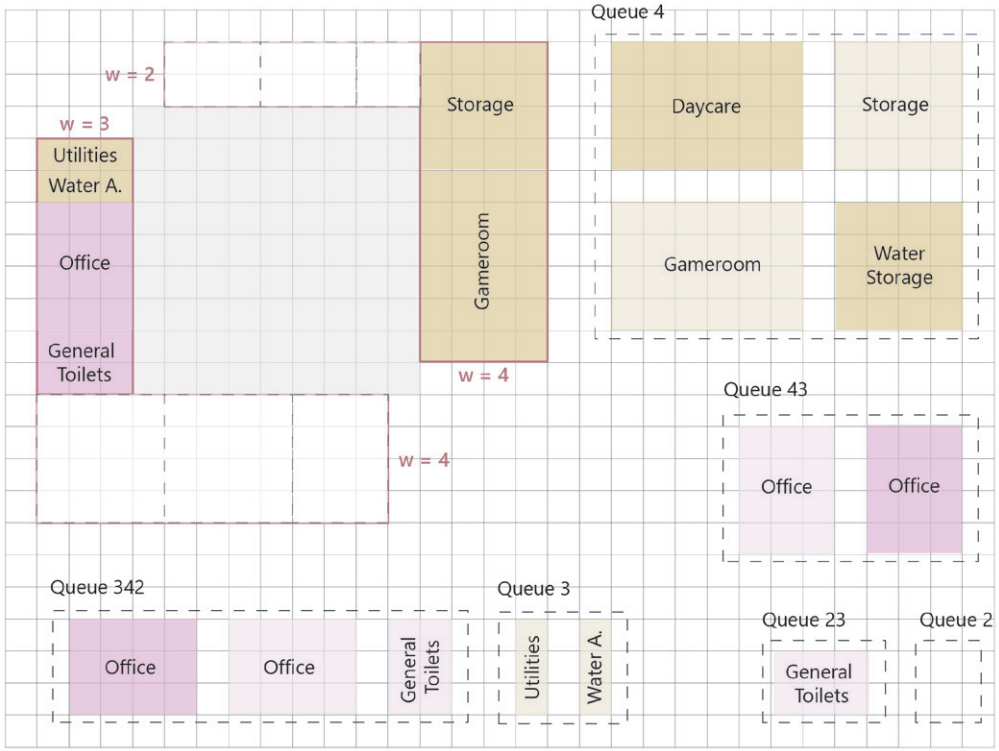
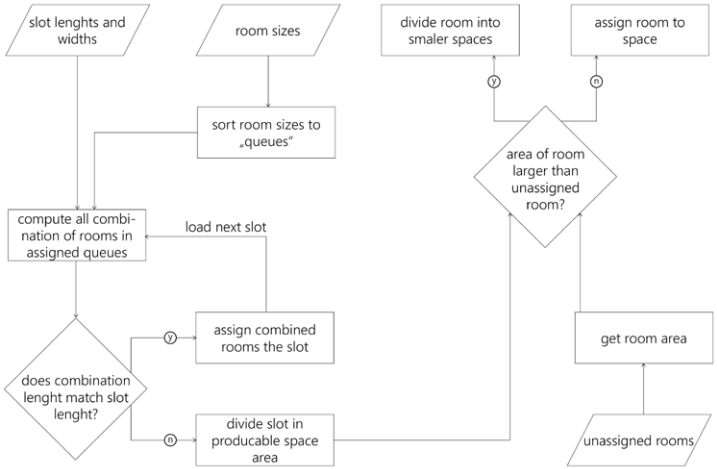


Figure 48: Chainsaw Algorithm

Now, the remaining rooms in the queues are assigned to the newly created room matrices. Rooms with the same matrix are assigned with each other, rooms that are smaller than the created matrices are „upgraded“ to the larger version, and rooms that are larger than any created room are divided into smaller rooms that have in total at least the same area as the initial room.

Finally, all rooms are assigned, and the information is stored in a dictionary, containing: number of slots, the width of slots, order and number of assigned rooms, length and width of the courtyard, and the starting point of each slot.



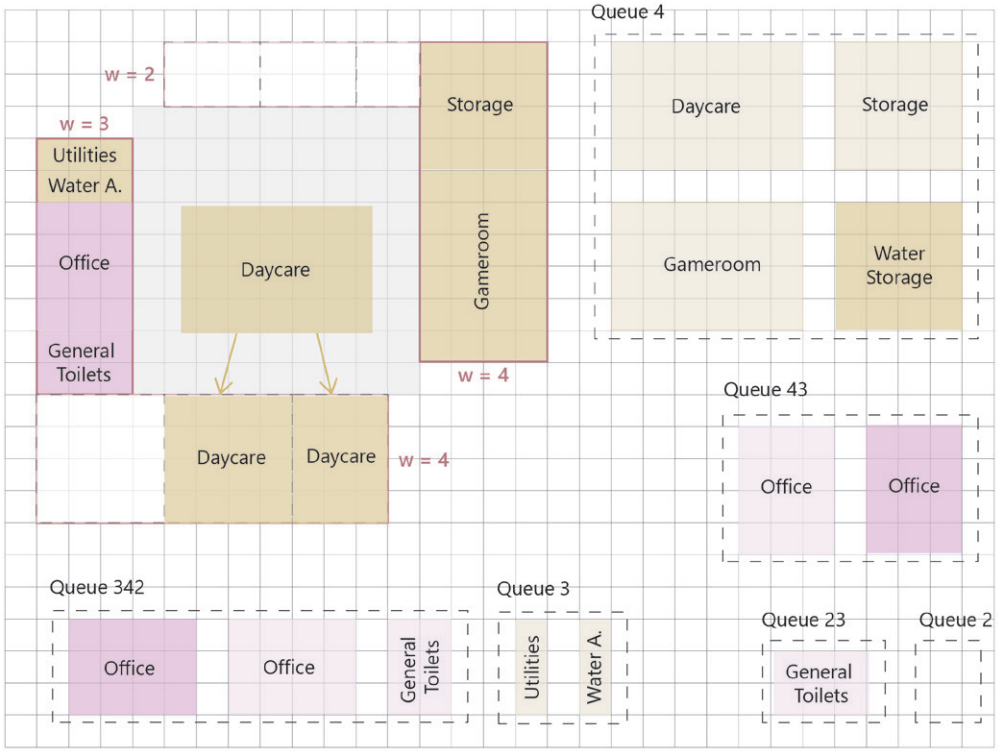


Figure 49: Divide larger rooms

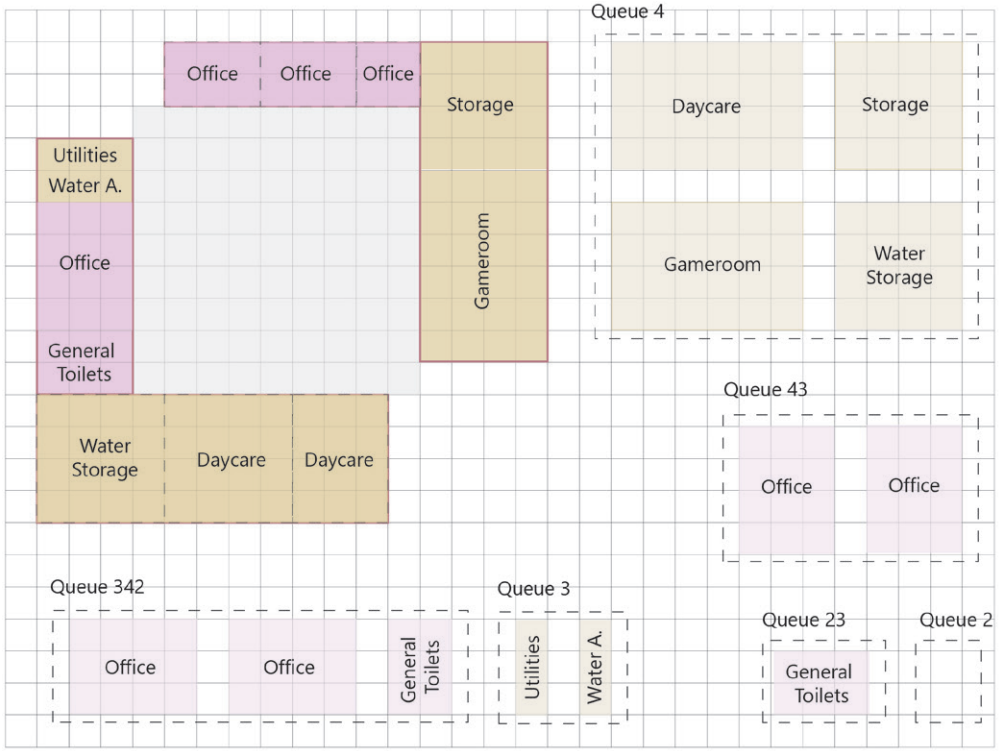


Figure 50: Final Layout

3.3.2 I Final Functional Floorplan

By applying the Population Algorithms on all courtyards, the whole layout evolves. Almost half of the slots were able to be solved with the „Clean Algorithm“, the other slots were assigned with the „Chainsaw Algorithm“.

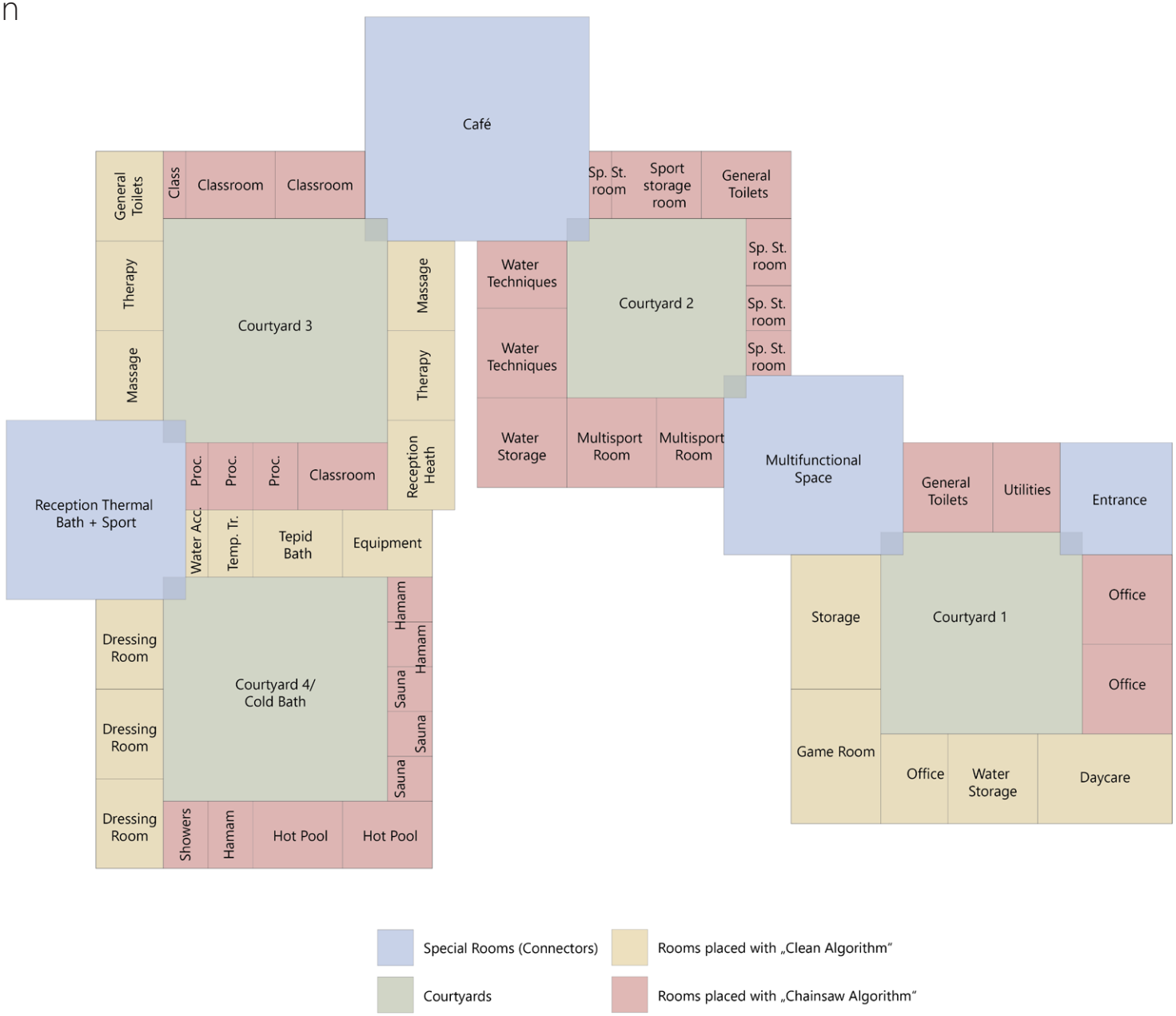


Figure 51: Final Building Layout

3.3.3 I Hidden Corridors

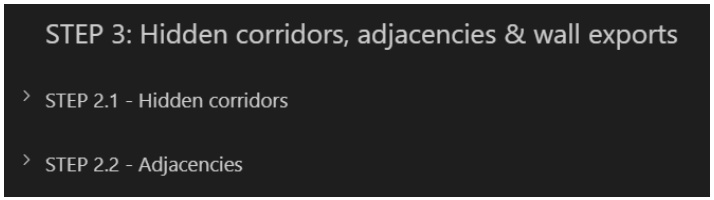


Figure 52: main sub-parts of STEP 3 of configuring

Once every functions is placed on the array, a copy is made and two processes are followed in order to find the hidden corridors. Those processes are explained int eh flowchart and stepped diagram below. It work by aplying a stencil on the array and searched the situations where "a" is a corridor and "b" is another function than "x". In this unique case the cell "x" become a hidden corridor. The hidden corridors help define where to place doors when the adjacencies will be solved.

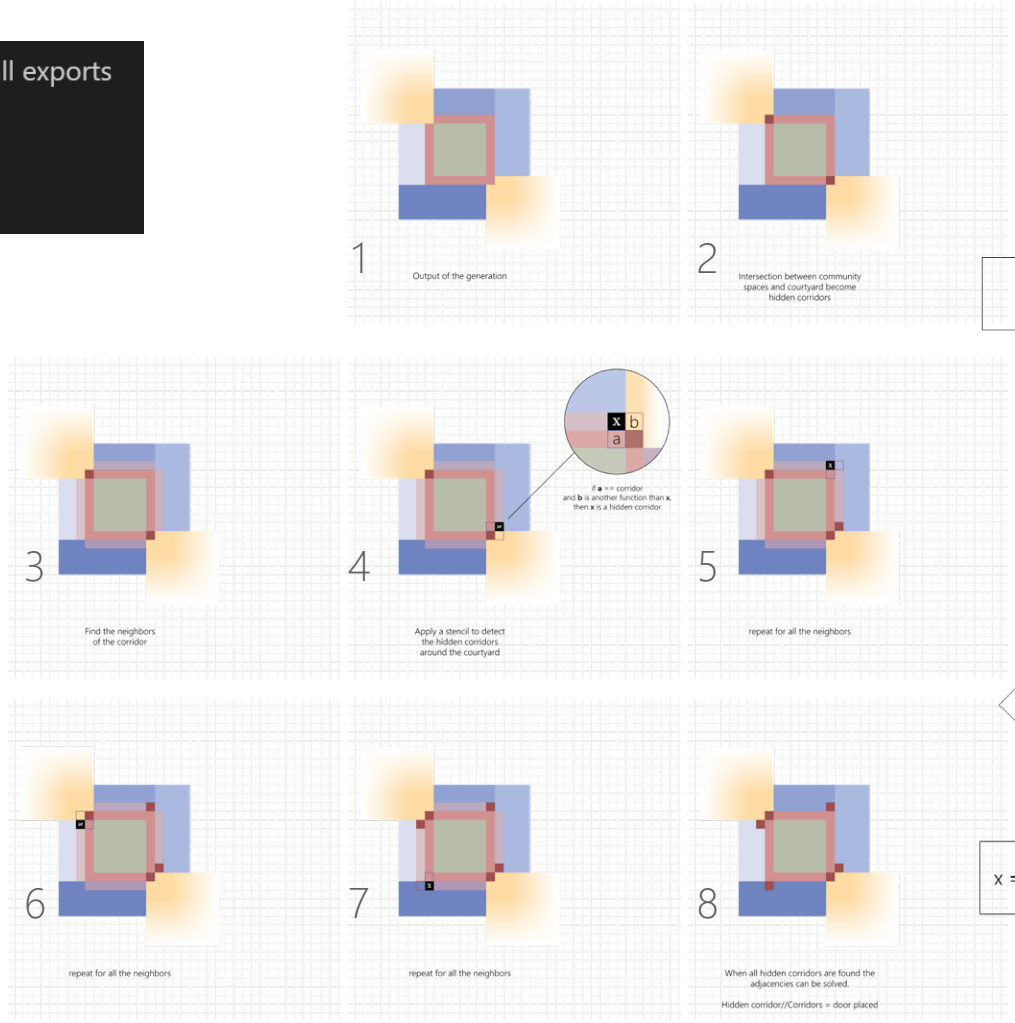


Figure 54: Stepped apllication of stencil

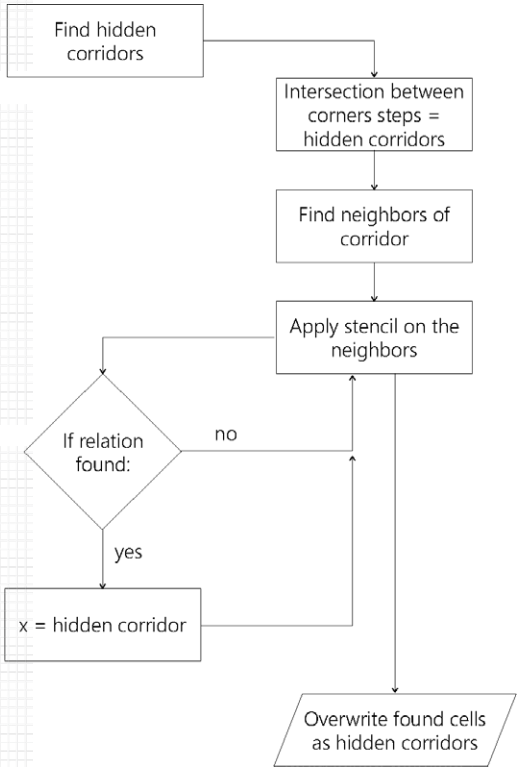


Figure 53: flowchart "Finding hidden corridors"

3.3.4 I Adjacencies

Defining the adjacencies has as goal only to help tell us which type of wall or opening should be put between two cells. The data output should be in such a way usable and implementable with another visualisation software. In our case we decided to export the arrays as csv files and translate value to wall library with grashopper.
Adjacencies could be done by analising the relation of a cell with its four neighbors and do the following for every cell. Being able to only read one value at the time per array, we would need to export 4 arrays and compile them grashopper. Instead, our chosen methods was perform only two copies of arrays, one with the vertically placed walls and the other with horizontally placed ones. Pairs are computed follow the stepped gradient array which is the distance map starting from the top left cell.

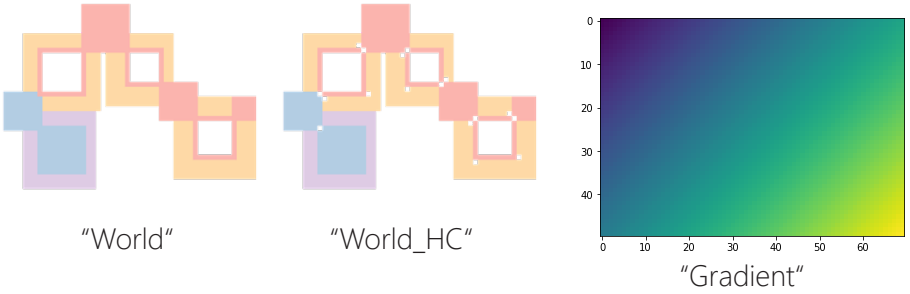


Figure 55: Arrays used to solve the adjacencies

At every step, as it can be seen in the next figure, the cells a and b are cross compared. This allow to join the 4 neighbors relation and 2 neighbors relation. Adding the hidden corridors, marked as hid_c, in the if statement allowed search for specific relation in both arrays, the functional one and its copy with the hidden corridors.

note: The gradient array can also help in animated the visualisation of the walls by steps. cf grashopper script "wall placement".

```
# Value used in if statement
out = max_dist
gard = max_dist-1
hid_c = max_dist+1
cor = 0


if a==b and a_world==b_world:
    # print(„OPEN")
    wall_type = 0
elif ((a==hid_c and b==cor) or (b==hid_c and a==cor)) and a_world==b_world:
    wall_type = 0
elif ((a== hid_c and b!=cor) or (b==hid_c and a!=cor)) and a_world==b_world:
    wall_type = 0
elif (b == gard and a == cor) or (a == gard and b == cor):
    # print(„RIWAQ")
    wall_type = 1
elif (b == hid_c and a == cor) or (a == hid_c and b == cor):
    # print(„DOOR")
    wall_type = 2
elif ((a==hid_c and (b!=cor and b!=hid_c)) or (b==hid_c and (a!=cor and a!=hid_c))) and (a_world==cor or b_world==cor):
    # print(„DOOR")
    wall_type = 2
elif a == out or b == out or a == cor or b == cor:
    # print(„WINDOW")
    wall_type = 3
else:
    # print(„WALL")
    wall_type = 4
```

Figure 56: Extract of the adjacencies script: Applying wall type

Ultimately, we end up as output two arrays, one for the horizontal walls and another one with the vertical ones. These arrays can be then used quite directly on grashopper following the script “placing walls“

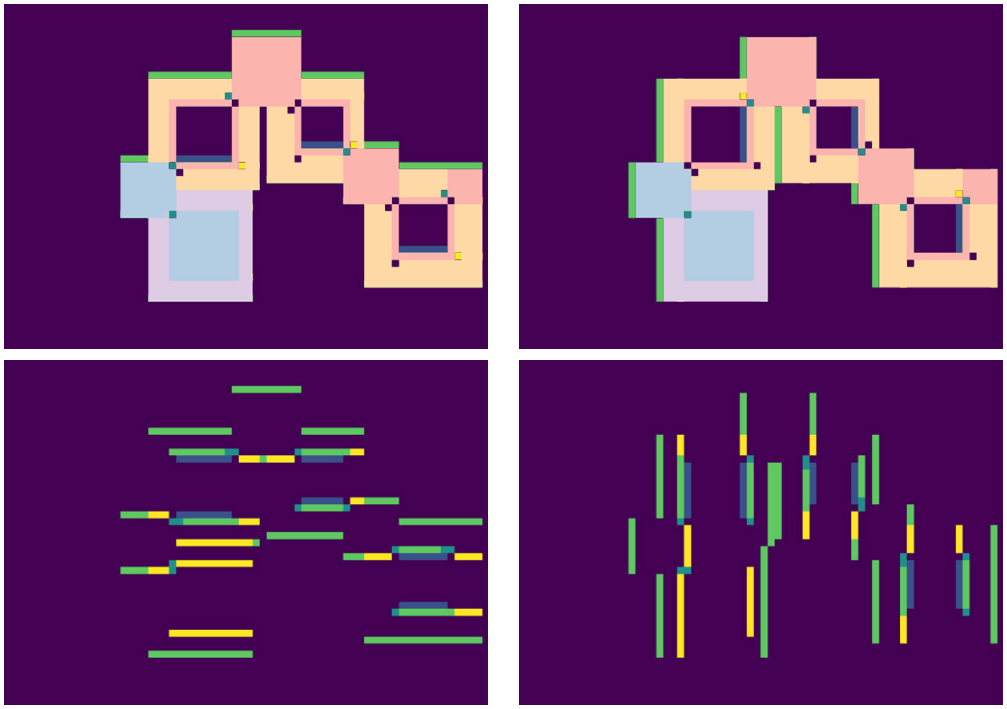


Figure 57: Output of adjcencies

As we are creating all the wall modules with a tartan grid, a both arrays need to be combined in order to derive the existance or not of a tartan column. The rule is:

if a cell has a value different than 0 in both array, then a tartan column is placed in it diagonal
if a cell has a value different than 0 in one array, then a tartan column is placed on its rigth or bottom respectively

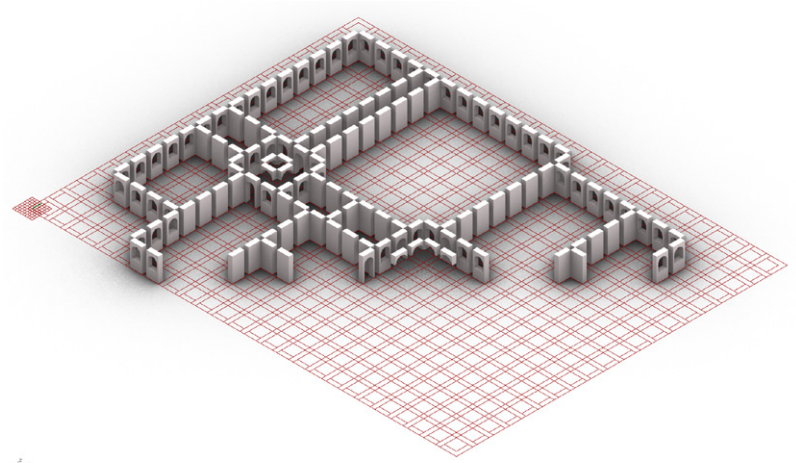


Figure 58: Output of adjacencies transcipted with wall types

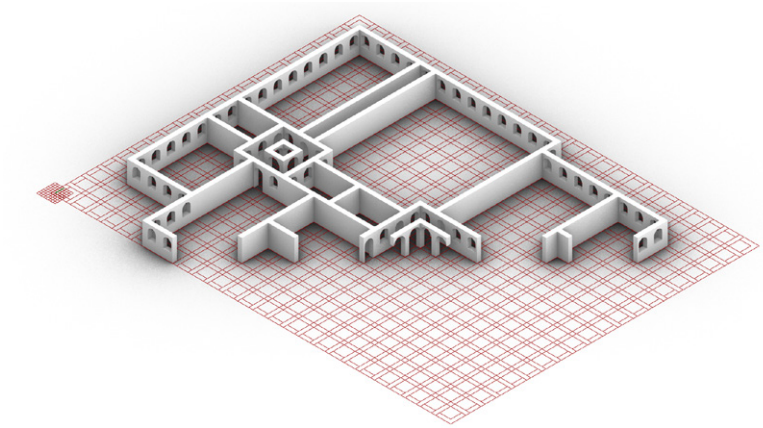


Figure 59: Visualisation after applying the tartan column

3.4 I Reflection

BREAK THE CHAIN OF COURTYARDS

When we first started for the gamifiaction rules we wanted to allow the configuration flow freely from corners to corners. Especially when the configuration starts from an edge, side of a site, expending on both side of the entrance is more than welcome. After the implemention of the ID, it made the configuring run in chain. This could be improved by adding some flexibility on the decision making of the script coupled with a another understanding the function relation. If we understand a community space and courtyard as a pairs. We can can start placing them as such, and improving the flexibility of the breaking point, some Y layout can be achieved. This new implementation denaturalise the notion of ID making it not related to the unique translation of the direction taken during the configuring. This idea could have need been written down without doing the part as explained.

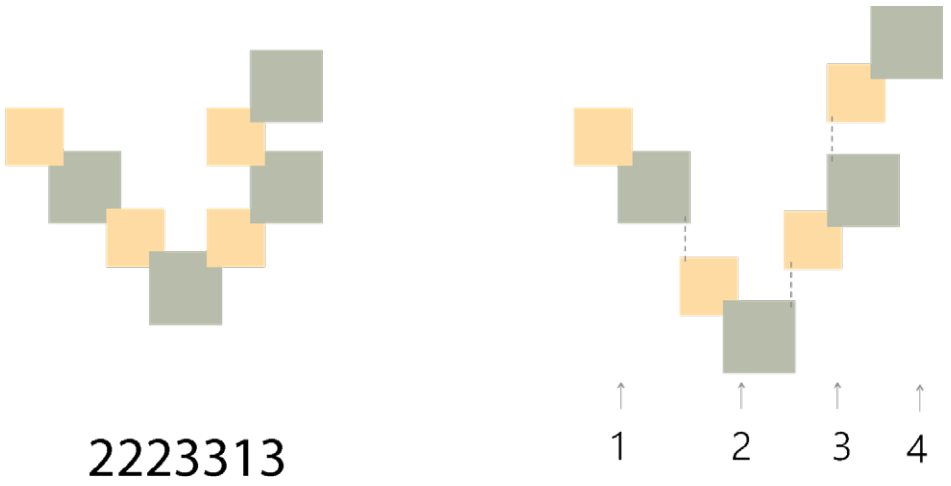


Figure 60: Cluster ID into pairs

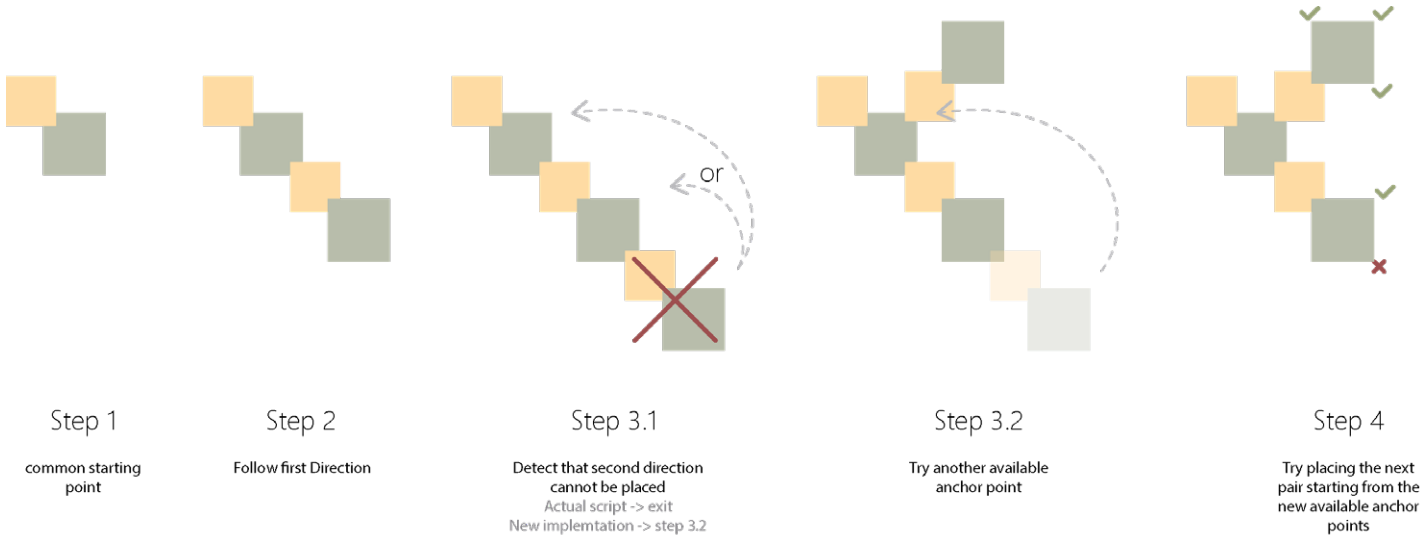


Figure 61: Flexibilty moment in script

L SYSTEM IMPL

PARTICIPATION

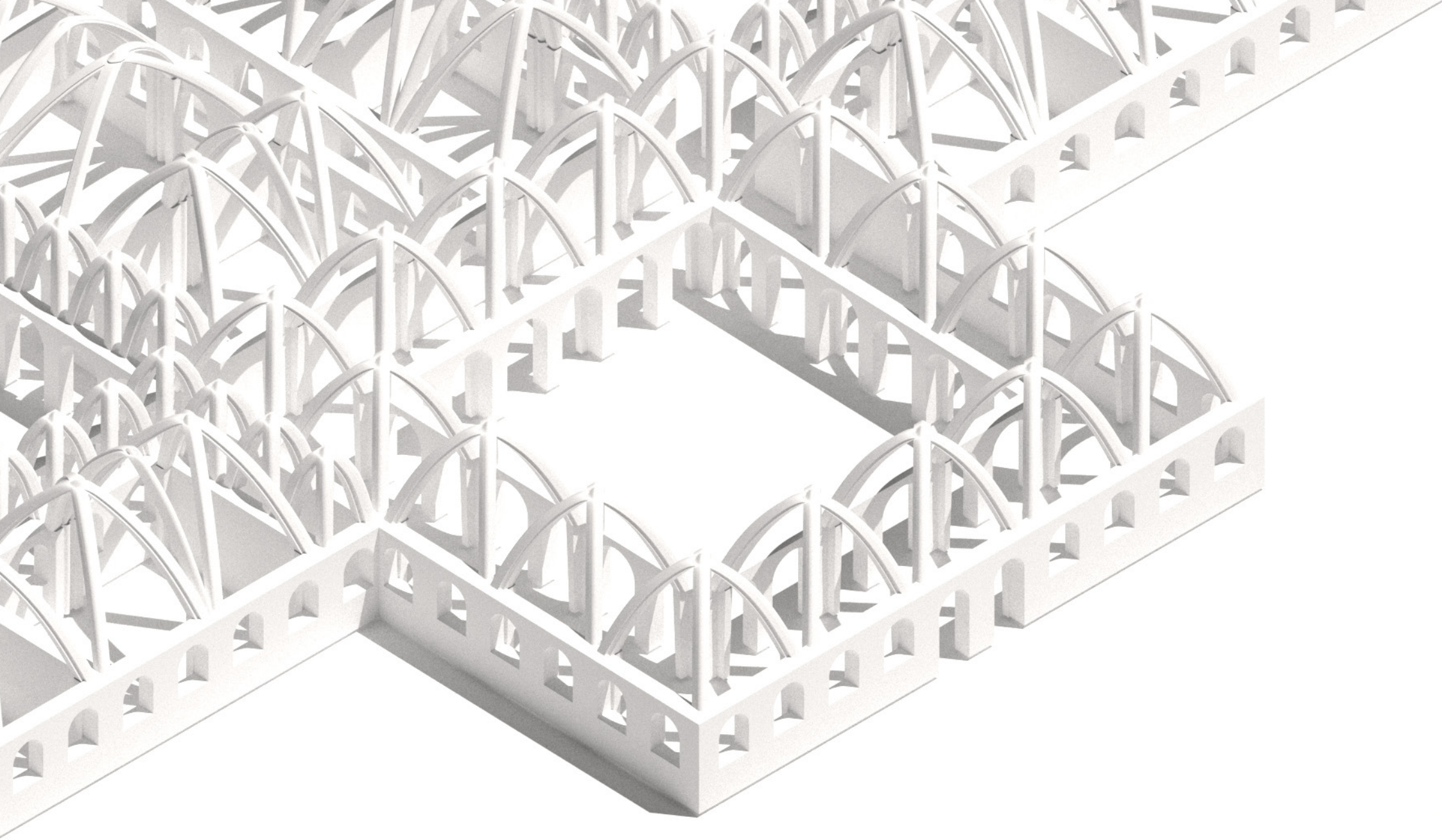
The Excel file as an user interface and data storage platform was a huge success during the elaboration of the project, since the quick modification, safing, and parallel work was possible. Regarding the applicatio in the camp, a more advanced User Interface would be needed. For example, the input mask for the architect and the camp manager must be different. A web-database with encrypted users that have different wrinting and reading rights could be a solution for further improvement.

SIZING COURTYARDS

In the Python script, only the variant to connect courtyards though corners is possible, the slots are always served. Therefore, L-shaped or U-shaped courtyards are not producable, this could be handled with a further elaboration of the project.

POPULATING COURTYARDS

The Population of the courtyards turned out to be very difficult in execution. The Python Script for this part has to be used carefully, since not all results are reliable. Reaching reliabillitiy for all possible cases could not be reached in the sope of this project. Furthermore, there is the potential to implement a further gamification, for example through a „Tower of Hanoi"-Logic with a rating system to find the variant with the least steps taken. Furthermore, the logic of the „Knapsack-Problem" could be solved in Dynamic Programming.



4.1 I Process

The Shaping phase of the project started in parallel with configuring and structuring. The first aim was to find an interesting shape for the roofs, according to their requirements (entrances, windows etc), even if these hadn't been defined at that time. Specifically, for our project that has many spaces with different functions, all acting as focal points for the people in the camp, as for example the Multi-functional space or the thermal bath, it was evident that a special design was needed.

Thus, the most rational step together with the configuring part was to distinguish the rooms that had bigger size as "special" and demanded a more unique approach, than the smaller ones that were the "main" rooms with higher repetition in the floorplan (Fig 32). Also, this decision was intuitively taken as for construction purposes, it was realized that the vaults would be able to cover until a certain span.

During the form-finding process, in regards of the main room tessellations, many experiments were studied forming the illustrated catalogue in Fig 61 & 62. These relaxed meshes are of a size of one module 1.20 x 1.20 m, but their edges are extended to meet half of the tatami grid, since they were designed to be assigned into the code and eventually form the whole ceiling plan of the building. For this reason, the mesh area that would meet the column of the Wall Library Fig 76, was used as support in the Kangaroo solver. After this step, in order to assess which tessellation we would choose for the building, constraints from the structuring and construction processes were applied. Since, the earth vaults should have only compression, many of the shapes were discarded either because they wouldn't transfer the vertical forces to the ground without tension, or they had intricate geometries that weren't constructable. The approximation of the latter ones, even though they deemed inappropriate for the main rooms, they were used as a starting point for the investigation of the Muqarnas' vaults.

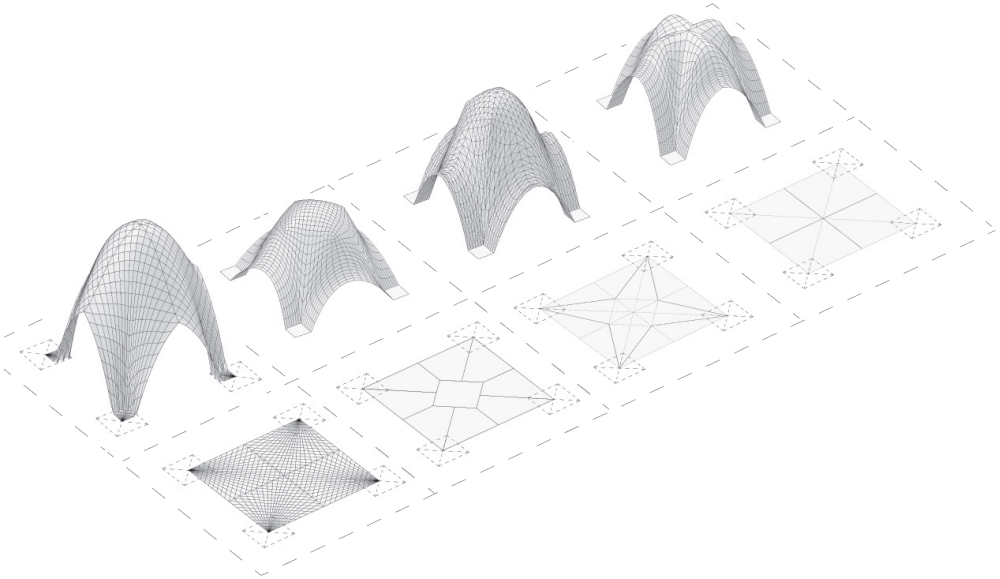


Figure 62: Diagram with tessellation experiments in the 1.20 x 1.20 module (part 01)

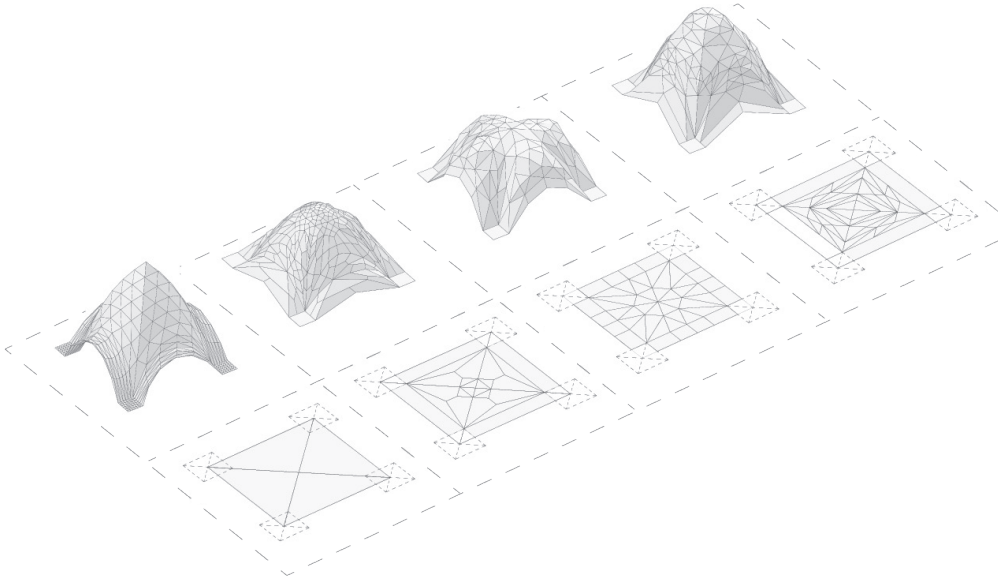


Figure 63: Diagram with tessellation experiments in the 1.20 x 1.20 module (part 02)

Before putting our effort into developping a structural and constructive system, a screening has been done in order to compare the key functionalities of each strategies. By listing all the different options and assessing them, we were able to choose which method to follow. These three following options are a mix between our personal interest and a trend of study, developped during Earthy 4.0.

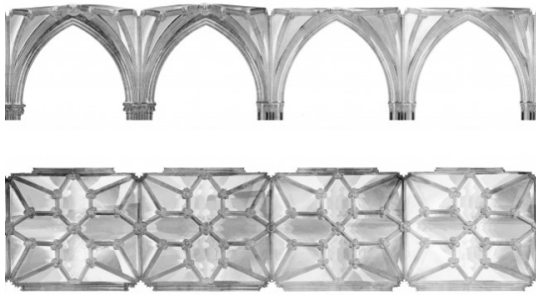


Figure 64: Ribs & filling

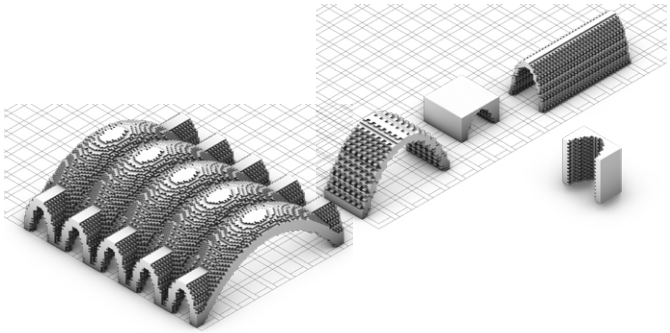


Figure 65: Voxelisation

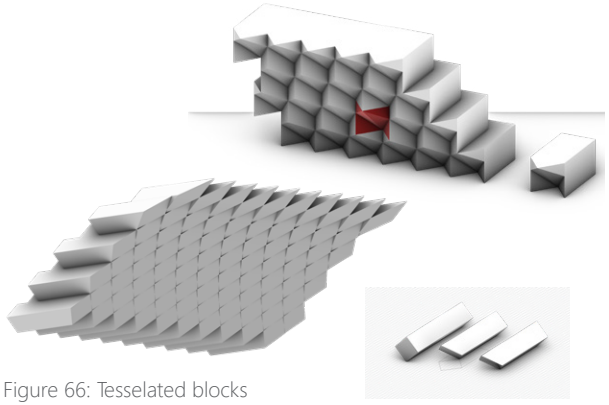


Figure 66: Tesselated blocks

Ribs & Filling:
Ribbed arches with custom cut construction blocks would do the structural suppport while a brick filling would close the vault. The connection between block is done perpendicular to the line of thrust

Voxelisation or polyhedronization:
Assume the vault to be a monolith and process a horizontal layering of blocks aproximating the guide relaxed mesh.

Tesselated blocks:
Blocks formed by the vertical projection of the 2D tesselation on the guide relaxed mesh. The blocks a then extruded horizontaly or perpendicular to the line of trhust forming concentric rings.

Our design require several repetition of the same matrix shape, for that reason we need to find a solution that provide a easy construction. On another side, constructing with earthblocks require the constructive system to be able to get as close to a compressive like structure. Finally, the fabrication need to be easily handled with common molding techniques. None of the option provide a optimum system, but providing some change to certain elements, and applying them where they fit the best in our project is the most sensefull option. Based on the criteria cited, a comparaison table was layed down, and the ribs & filling system was chosen to be applied on common modules. And the tesselated blocks to special rooms as concentric dome, following the muqarnas constructions.

	Construction	Approximation of relaxed shape	Fabrication	Force distribution, by bricks orientation	Choice
Ribs & filling	- demands skilled workers (make ribs as puzzle pieces) (develop telescopic guide + formwork)	+	+	+	Common modules
Voxelisation	+	-	+	-	discarded
Tesselated blocks	+ Only 1 way that these bricks are assembled	+	- (Reduce brick shape variation by design)	+	Special Rooms: Cold bath Café Multispace ...

Figure 67: Table of comparaison between ceiling strategies

4.2.1 I Relate meso and micro scale

Sizing the grid need to be done in parrallel with several other parameters. On the one hand, the walls on the grid need to be constructed with multiple of the bricks. Put together, the bricks need to form a gentle landing and rising stair. And in the height level, the second floor height (height of the grid) needs to allow a multiple of the grid size. To play with those 4 variables: Brick size, grid size, stair landing, rising and length, and height of second floor, a parametric excel sheet was made to find the commom multiplier between the factors. The input is simply the size of the brick which define the size of the rising and landing that then calculate the matching factor between several option of corridor widths (x and y grid size) and height of second floor (z of the grid size). In our project, the second floor was in the end not implemented, but based on our chosen 120cm grid (tatami unit), we could have have a second floor at 3,4m, 4,2m or 5m height.

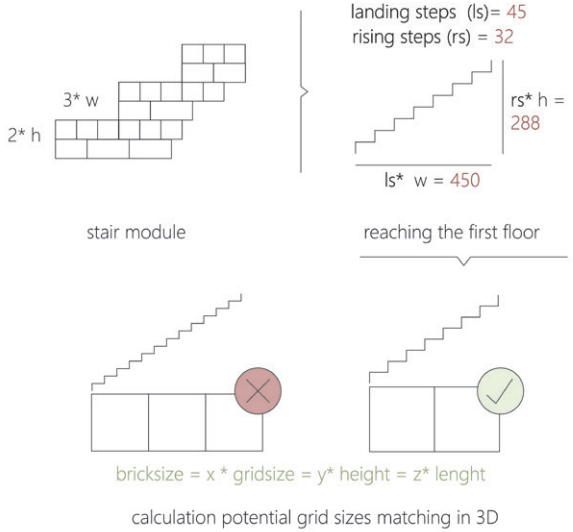


Figure 68: Defining the grid size, and second floor based on the brick

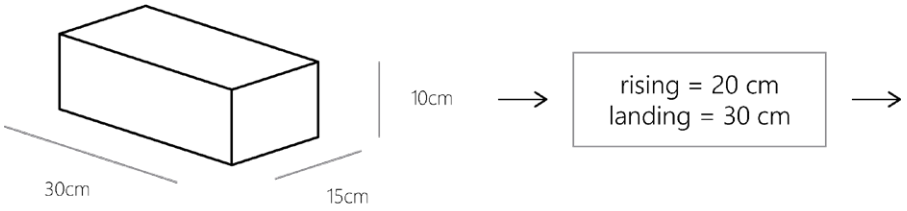


Figure 69: Strategy for neighbooring

Match between three Dimensions		Corridor widths (cm)						
Height 2nd floor (cm)	Stairs length (cm)	110	120	130	140	150	160	170
320	450	4.090909	3.75	3.461538	3.214286	3	2.8125	2.647059
340	480	4.363636	4	3.692308	3.428571	3.2	3	2.823529
360	510	4.636364	4.25	3.923077	3.642857	3.4	3.1875	3
380	540	4.909091	4.5	4.153846	3.857143	3.6	3.375	3.176471
400	570	5.181818	4.75	4.384615	4.071429	3.8	3.5625	3.352941
420	600	5.454545	5	4.615385	4.285714	4	3.75	3.529412
440	630	5.727273	5.25	4.846154	4.5	4.2	3.9375	3.705882
460	660	6	5.5	5.076923	4.714286	4.4	4.125	3.882353
480	690	6.272727	5.75	5.307692	4.928571	4.6	4.3125	4.058824
500	720	6.545455	6	5.538462	5.142857	4.8	4.5	4.235294
520	750	6.818182	6.25	5.769231	5.357143	5	4.6875	4.411765
540	780	7.090909	6.5	6	5.571429	5.2	4.875	4.588235
560	810	7.363636	6.75	6.230769	5.785714	5.4	5.0625	4.764706

4.2.2 I Tartan Grid

The grid conception reflects the analysis of the brick size and wall composition of the construction process. Thus, the dimension of this network is projected based on the block size and its stacking system. In this case, a standardized width 150 mm x 300 mm length x 100 mm height Compress earthen block. The rationalization of this composition from the early stages will allow increasing the precision of the design into the construction stages.

The construction system aimed to produce a simple stacking logic. Therefore, using one typology block, the wall is generated from the variation of block orientation. However, it is essential to remark that this grid does not regulate the generative design concept strictly. In other words, the tartan grid dimensions follow construction parameters for this specific case. Nevertheless, the construction design could potentially implement an alternative solution with an inferior or superior wall width.

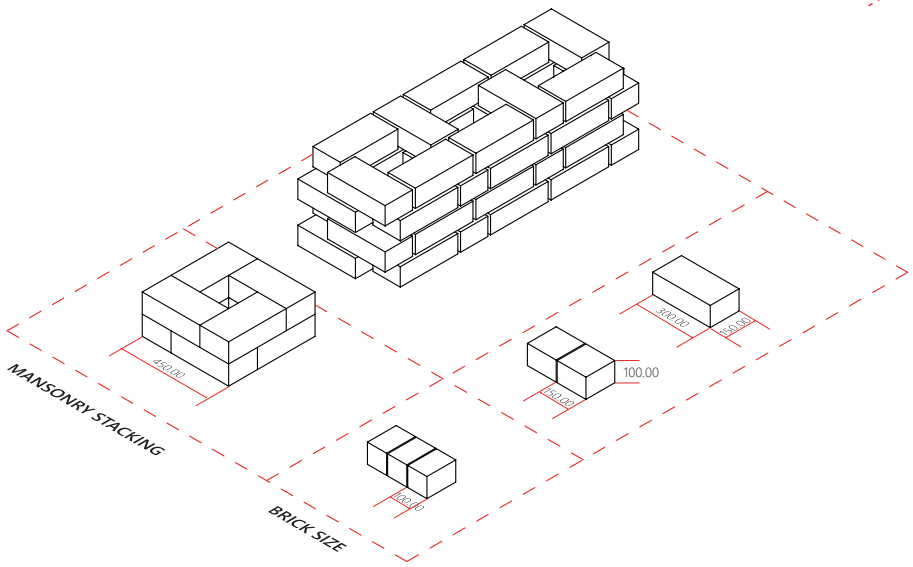


Figure 70: Brick Stacking system

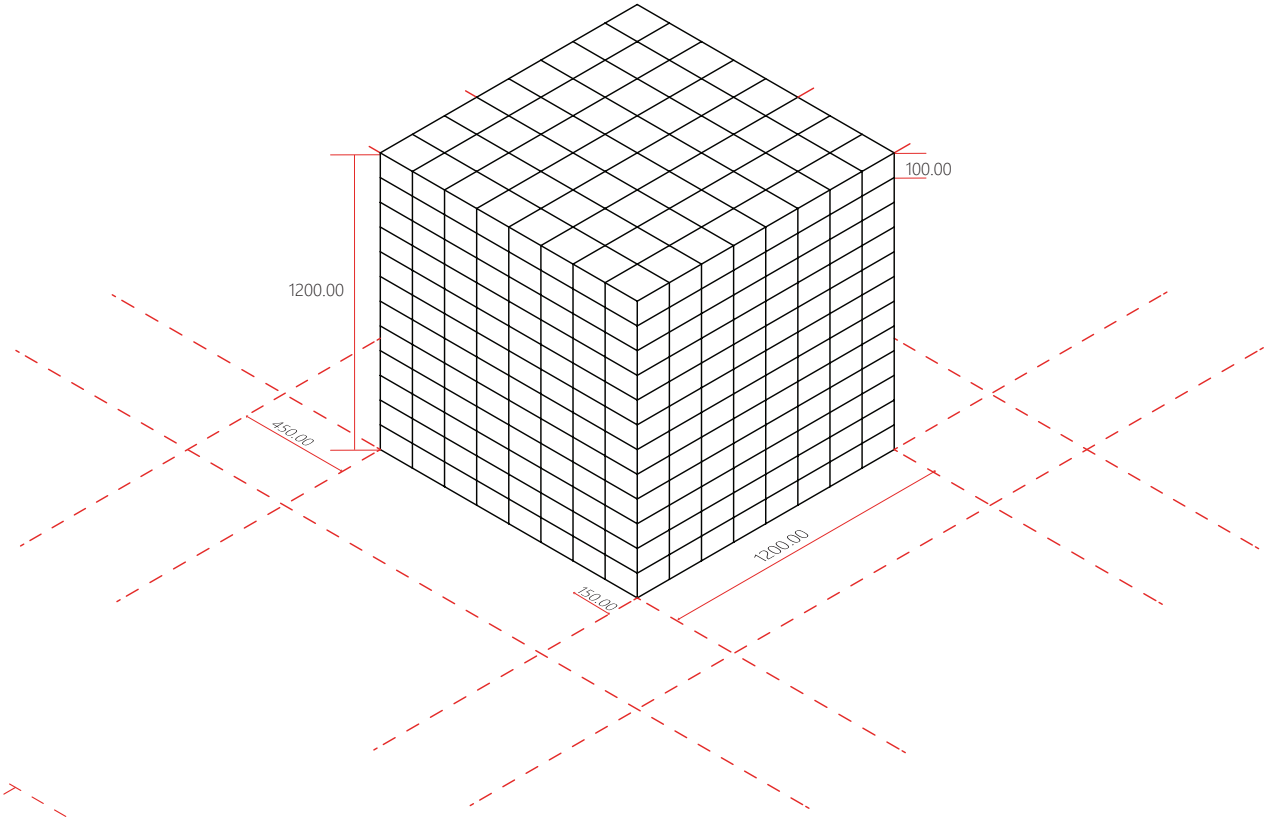


Figure 71: Modular logic - Tartan Grid

4.2.3 I Block Sytem And The Grid

For this case, the Tartan Grid follows the constraints of this specific stacking system logic, which as a result, produces a 450 mm thickness wall. This decision is due to the stacking orientation the blocks create using one vertical block (150 mm width) and one horizontal block (300 mm length). The result of this rationalization is a wall with cavities in its interior structure. This design decreases the material requirement of the constructions and increases the wall thickness necessary for structural performance.

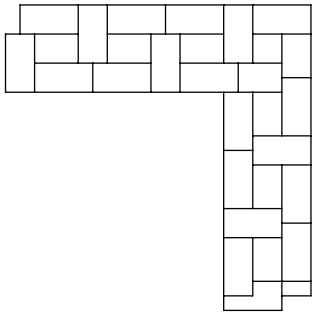


Figure 74: Brick stacking

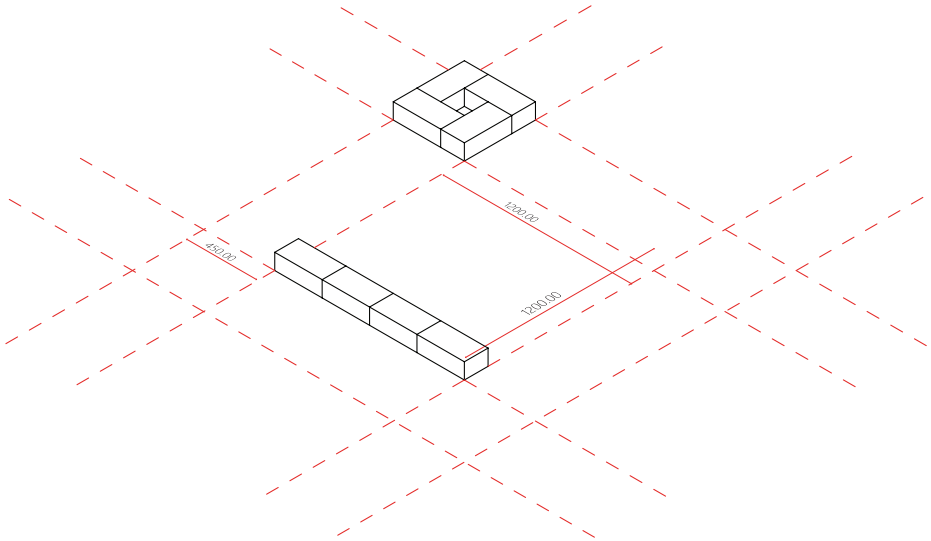


Figure 72: Brick modularity

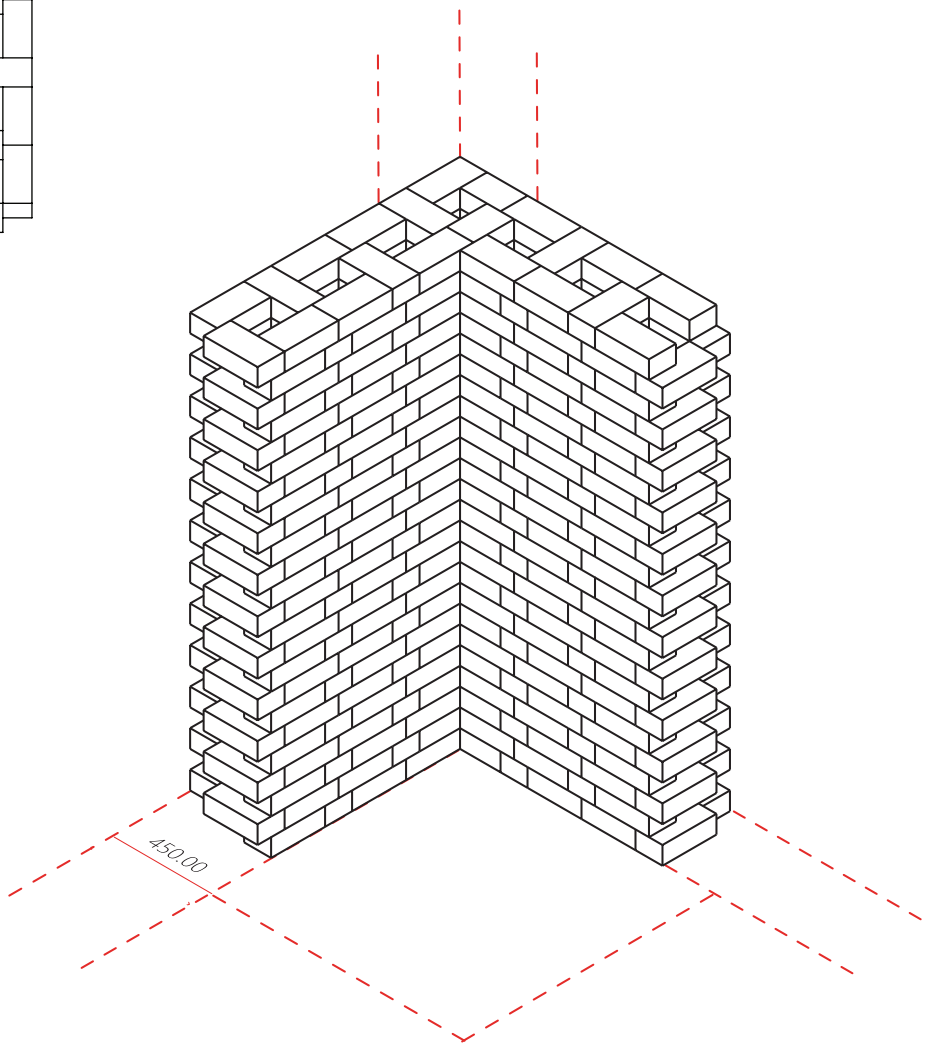


Figure 73: Masonry Wall placement

4.3.1 I Wall Generation

The wall component has been generated unparalleled to the roof system library. This strategy breaks down the wall's structure into repetitive parts, allowing the implementation of blocks in several cases along with the project. Thus, following the Tartan Grid and the block dimensions (150mm x 300mm x 100mm). A simple classification of the wall types narrowed down the wall blocks into four simple categories that defined several scenarios:

- Column (C_01): Are used to fill the tartan grid-gap between the corridors and wall unions.
- Close Wall (CW_01): Are placed in spaces contiguous to other rooms, internal or external facades.
- Window (W_01): Place in rooms next to halls and exterior facades.
- Door (D_01): Access points to small rooms.

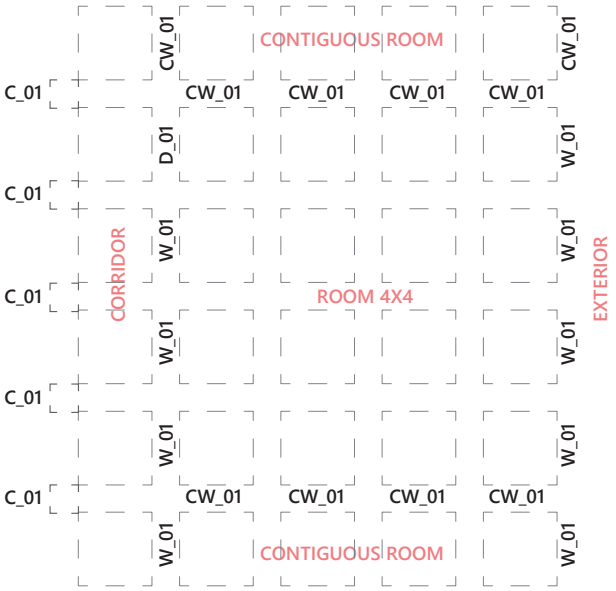


Figure 75: Wall placement diagram

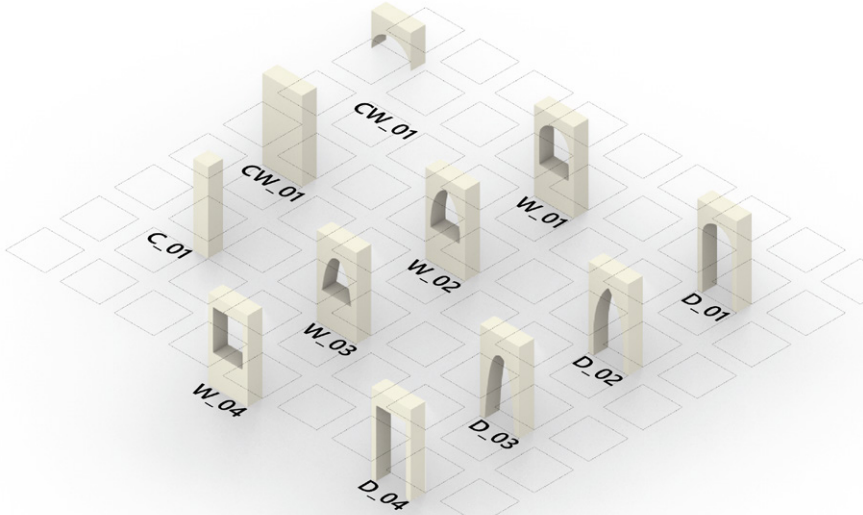


Figure 77: Wall typologies

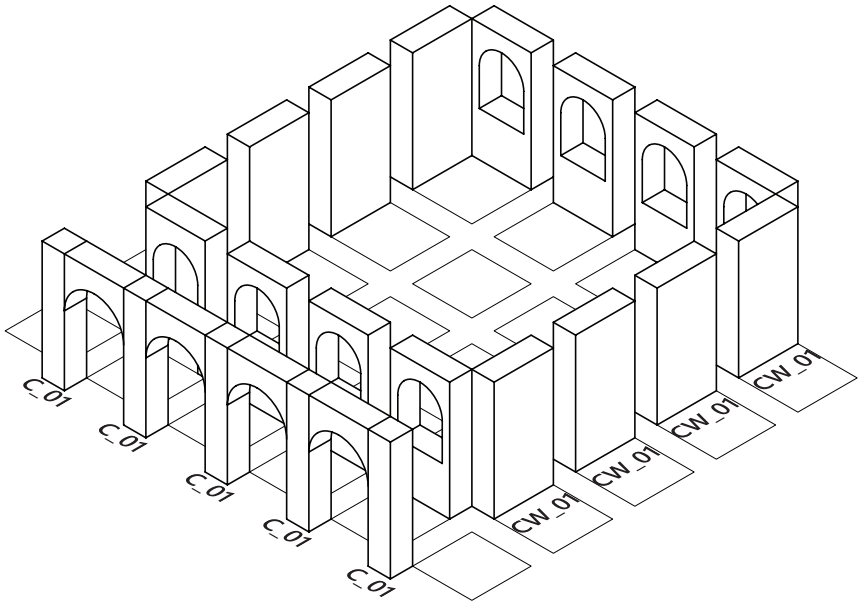


Figure 76: exemplification wall placement

4.3.2 I Parametrization of Wall Modules

Following the tartan grid, the parametrization of windows, walls, and doors based its initial dimensions on the brick size (150 mm width x 300 mm length x 100 mm height). Thus the initial size of the components has three primary parameters: The wall heights increase every 1200 mm, leaving a module of 2400 mm wall height. The thickness of the wall is increment by 150 mm. This dimension represents the number of blocks aligned in the structure (simple wall, double wall, or wall with cavities). The wall width adjusted tree units to the thickness parameter to achieve 450 mm wall depth, following the analysis of the construction process and the Tartan grid. The wall sets its length every 1200 mm, according to the existing grid.

Following the same principle, the windows and doors heights increment by 100 mm, meaning that the disposition of window gaps respect the block line and stacking system.

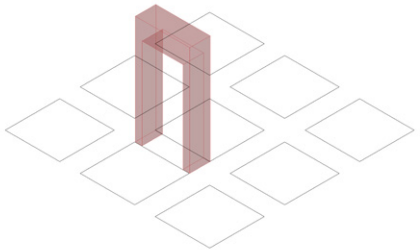


Figure 78: Alternative door - hortogoanal dintel

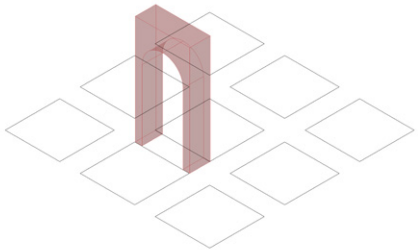


Figure 81: Roman Arch door

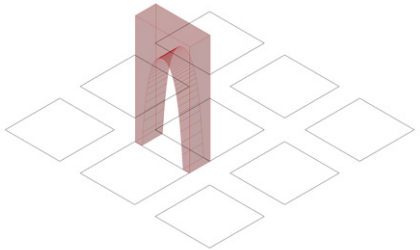


Figure 79: Cathenary door

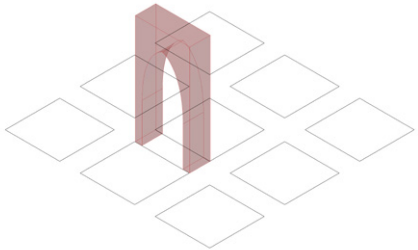


Figure 80: Gothic door

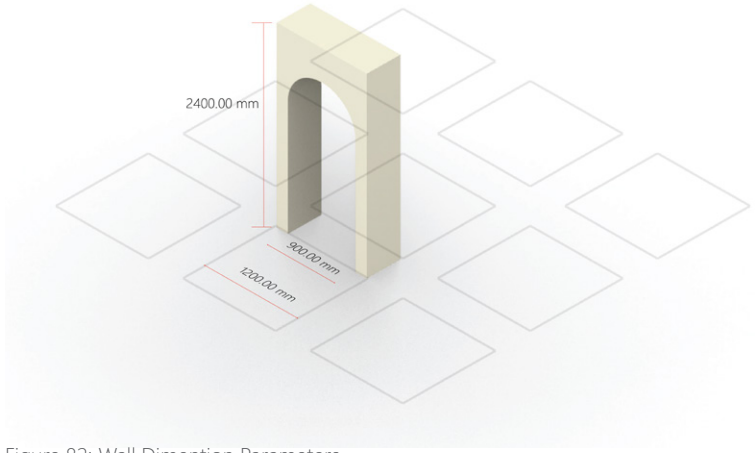


Figure 82: Wall Dimention Parameters

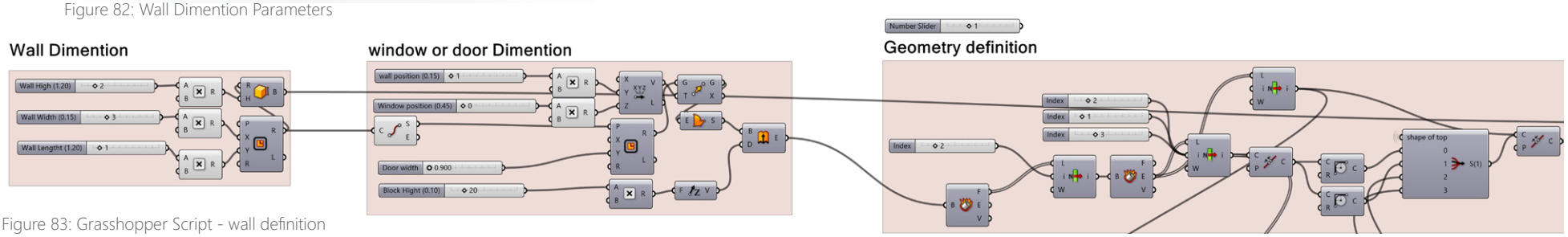


Figure 83: Grasshopper Script - wall definition

4.4.1 I Main Room Vaults

The common denominator for all the building components to be implemented into the code, is the tartan grid. Similarly to the wall placement, the roofs are also assigned as objects onto the matrix, by the corner values. However, in this case they can be placed only after the walls have formed the rooms. For this step, the half wall thickness of the tartan grid is used in both x, y directions, therefore forming a new tatami grid of 1.65m x 1.65m, as it is shown in the Figure 83.

This workflow was chosen by linking the shaping with the construction process. The roofs were deemed necessary to be treated as rooms and not to be cut into smaller pieces that would be assigned in parts of the tatami grid. This approach was also investigated, but the results from the relaxation process were not accurate to be used, therefore the conclusion that each room should be relaxed as one mesh was strengthened. Again, because of the room sizes standardization decided during the configuring, this was feasible by creating the library illustrated in the Figure 85.

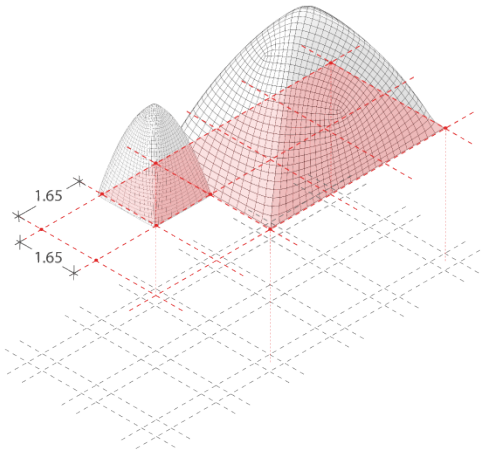


Figure 84: New tatami grid for the placement of roofs.

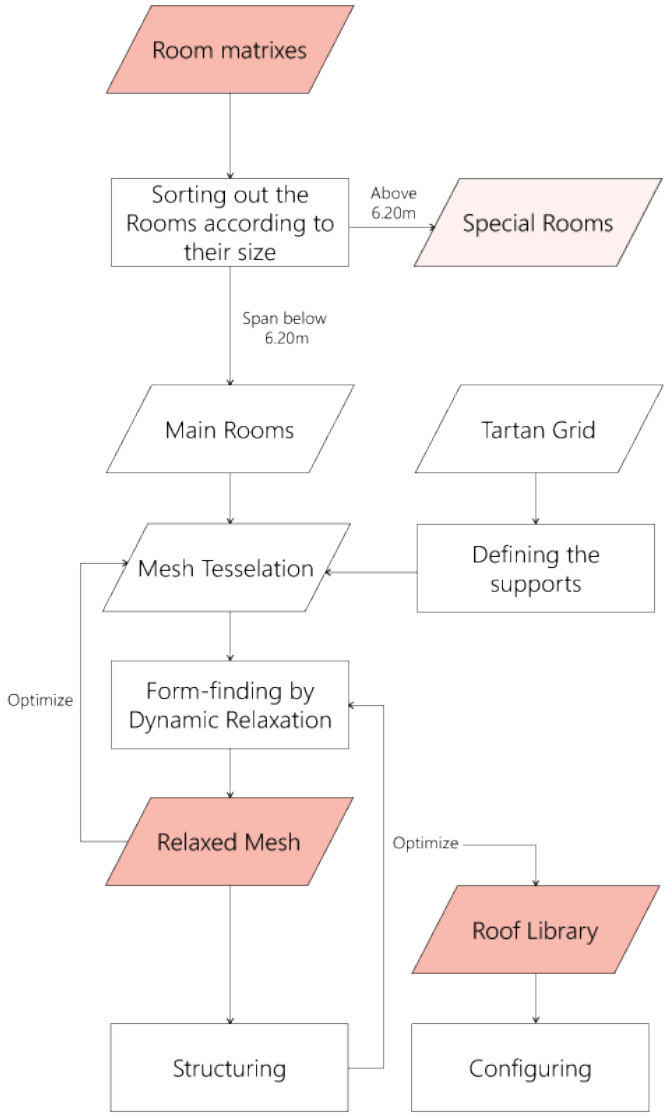


Figure 85: Flowchart for producing the Main room relaxed ceiling mesh.

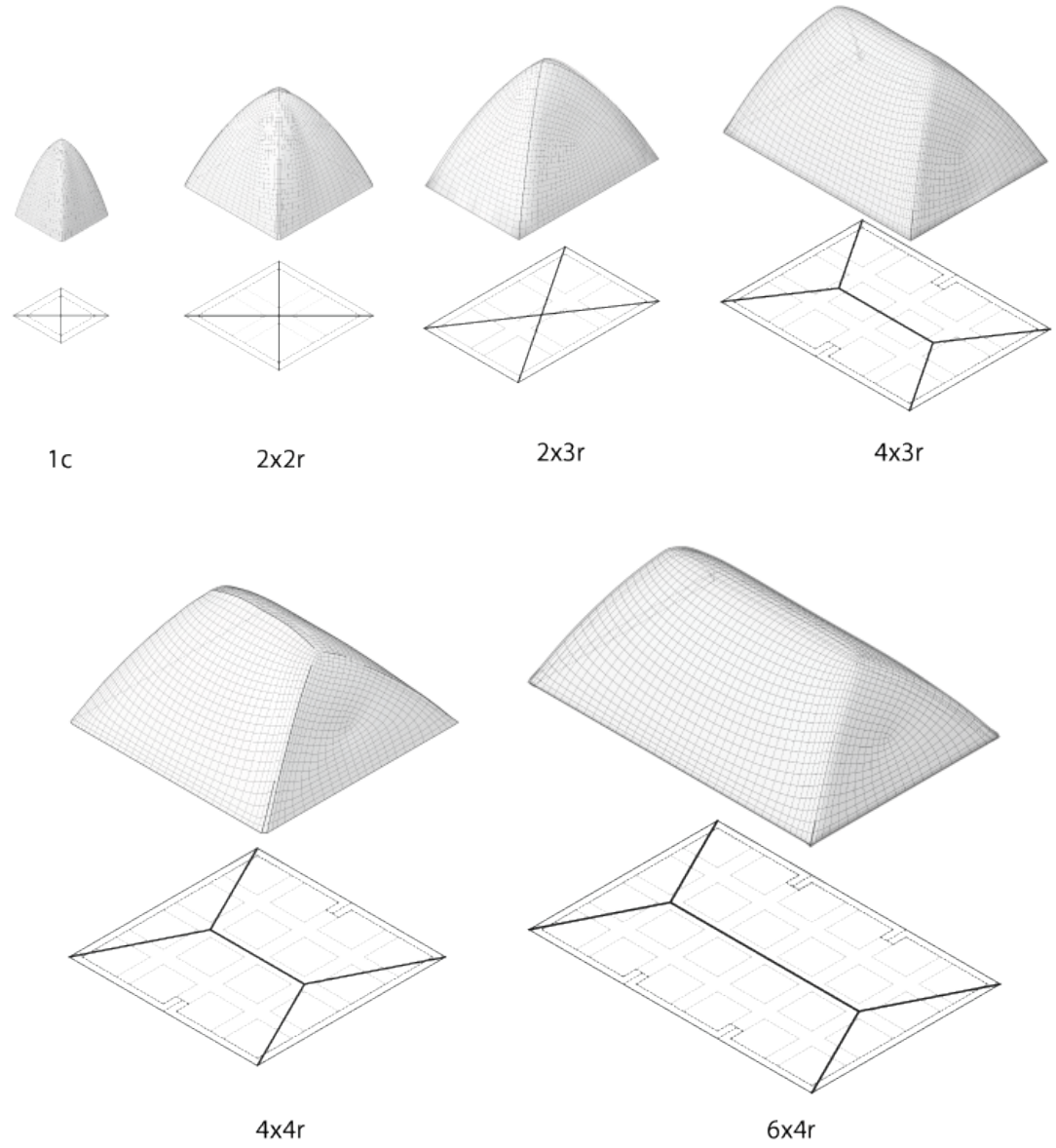


Figure 86: Roof Library for the Main Rooms

4.4.2 I Vault Tessellation logic

As the goal was to create modular but easy to construct roofs with the same tools, the tessellations used in the end were the most simplified ones. This outcome was also, derived by the wall design that had the doors and windows integrated, leaving the base for the roofs a continuous and planar outline with no openings Figure 86. This decision, might had resulted in a repetitive, non-interesting roof pattern, if there weren't the special rooms that required a different approach to attract the attention. Consequently, the tessellation logic was linked with the construction, that as explained in the Figure 66, a rib system was chosen out of interlocking blocks that have only one way of being assembled, in order to simplify even more the work on site.

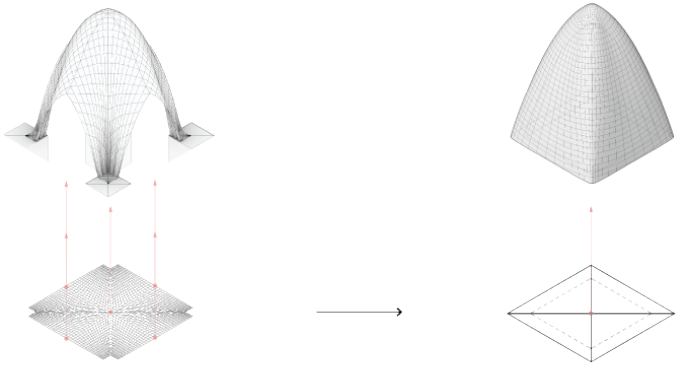
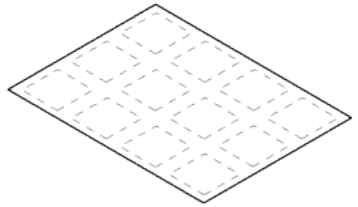
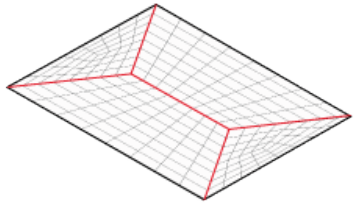


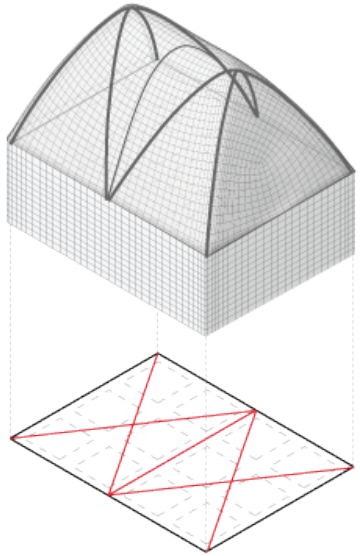
Figure 87: Mesh edges as supports result to a simplified mesh topology.



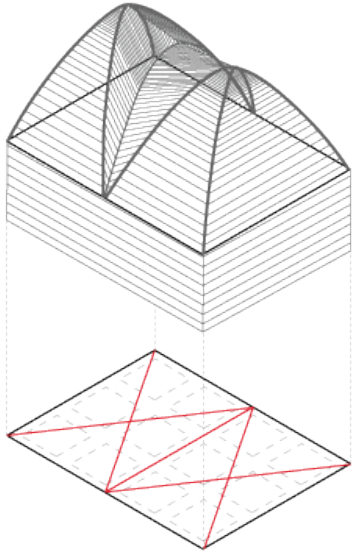
1_Room 12T
(size of a matrix 3x4 modules)
| Configuring



2_Mesh Tessellation using quads split subdivision | Shaping



3_Rib curves defined by the relaxed room box | Structuring



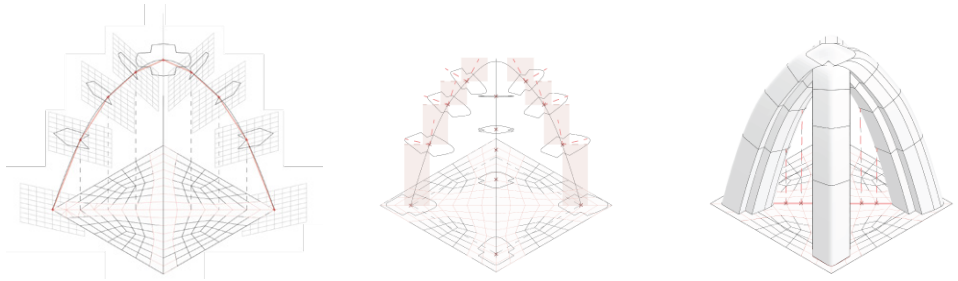
4_Brick pattern as infill between the ribs | Construction

Figure 88: From the tartan grid to the constructed room

4.5.1 I From curves to rib geometries

The process for the rib design started of the 2d tessellation, only after the optimized relaxed ceiling mesh was concluded by the structuring part Figure 89. Since, we are working with ribs, the diagonal and transversal curves needed were retraced by making sections on the relaxed mesh, as it is explained in the ***Fig XX*** (structuring). By multiplying these rib formations, we are able to construct all the specified roof sizes for the main rooms, as shown in the library of the Figure 94.

The first step to define the rib pieces by projecting the 2d tessellation is depicted in Figure 88. In this point, the demanded cross-section from structuring was checked, which was for this proposal equal to an area of 60 x 45 cm, but with further calculations it was concluded with the structuring part that it could be further optimized to a smaller one. Then, the outcome is evaluated, by applying constraints related to construction. For instance, one block should be of a size and weight that one man could lift it. For this to happen, the rib blocks had to be a lot smaller than the divisions derived by the projection of the 2d tessellation. Consequently, the final ones were derived in proportion, from larger on the base to shorter towards the top.



1_ Tessellation projection on the normals to the curve points

2_ Custom cross-section

3_ Block extrusion & length optimization

Figure 89: Design method for the rib blocks in steps

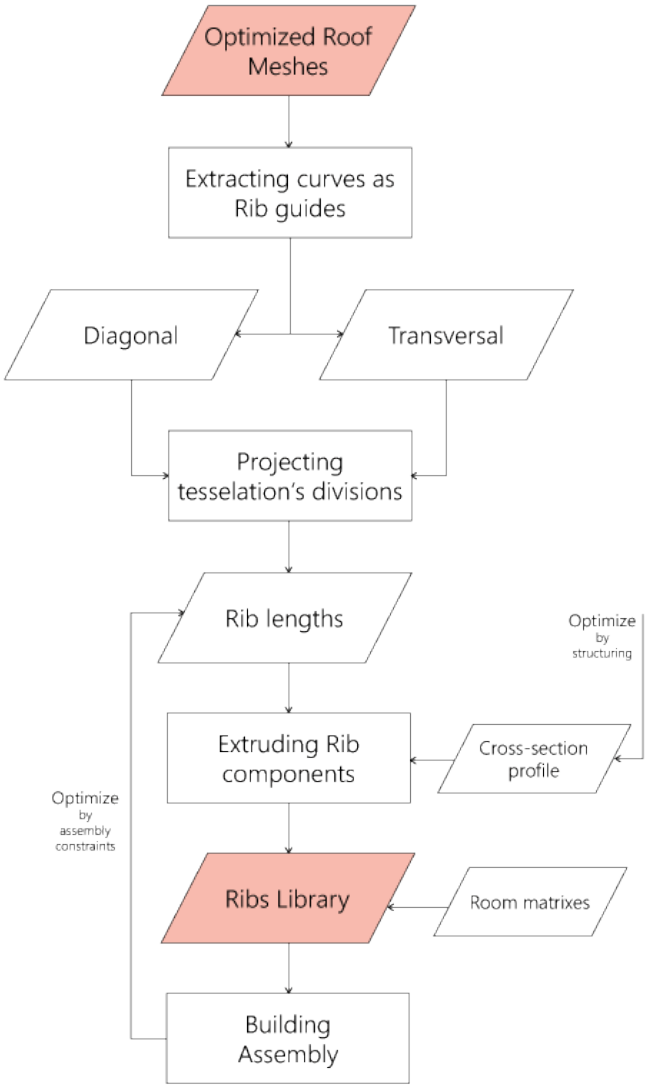


Figure 90: Flowchart for the ribs' design

4.5.2 I Rib blocks

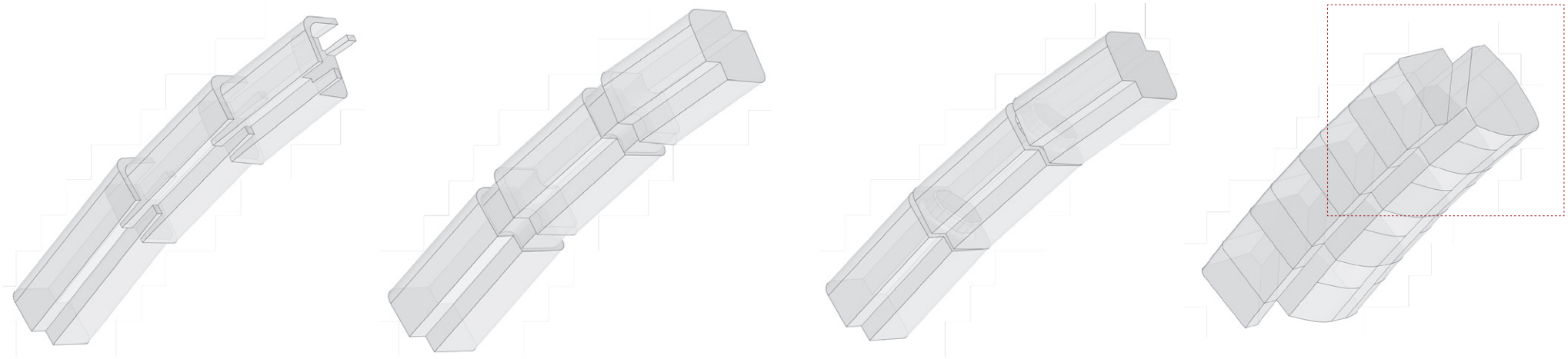


Figure 91: Connection study between the rib blocks .

As far as the cross-section is concerned, a unique design was produced – Figure 91 – by implementing the structural thinking of carrying the vertical forces efficiently to the walls and therefore to the ground, but also providing a smart way to lay the roof bricks for the vault formation. Depending on the position of the rib, whether it is diagonal or transversal, a different cross-section is needed.

Moreover, the interlocking mechanism was investigated with different ways as it is shown in the Fig 90, while in the last illustration is shown the chosen solution that carries the minimum amount of shear forces and deemed as the most stable one. The criteria for this design was that the block should be able to be casted into the mould by a mixture of gypsum and has the easiest way for stacking on top of each other, while forming an arc. In the mutliple experiments, an effort to standardize more the unique pieces was carried out, however due to time constraints we did not develop it further.

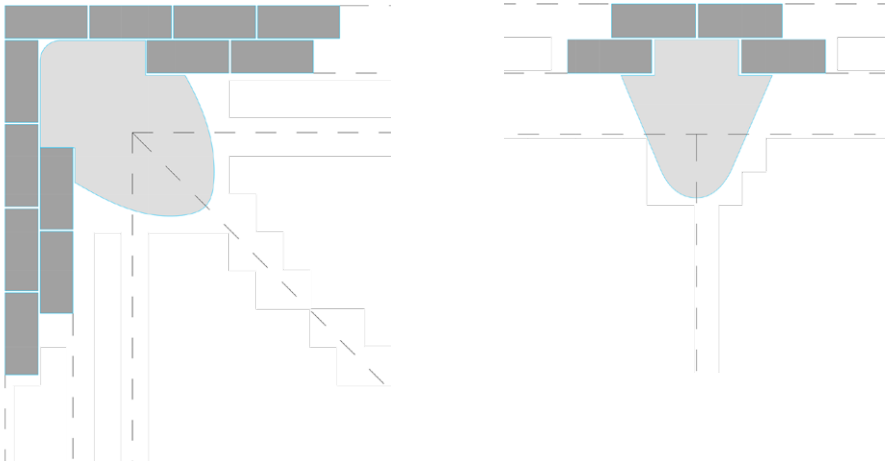


Figure 92: Cross-section design integrated with structuring and construction part.

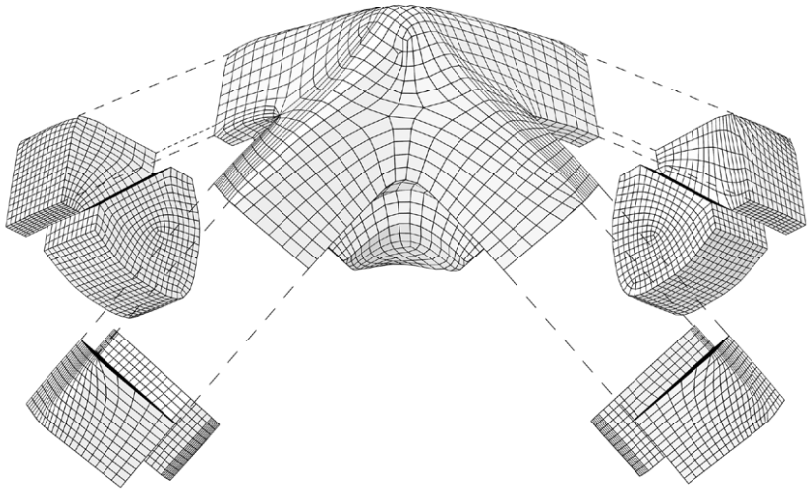


Figure 93: Cross-section design integrated with structuring and construction part.

The keystones both in the apex and in the base of the ribs require special attention, as they hold together the structure and could potentially connect a different number of ribs. Thus, the concept regarding the geometry of this unique blocks is explained in Figure 93. For this project, only four ribs are connected every time in the apex, so the final model of the keystone is shown in Figure 92 After the final step depicted in the diagram where the result is a polyhedron, smoothing processes are applied for construction purposes. While for the keystone on the base there are variations in the ceiling plan from connecting 1 rib until 3 in one room. However, the same logic can be applied with even more ribs being connected for the needs of another project.

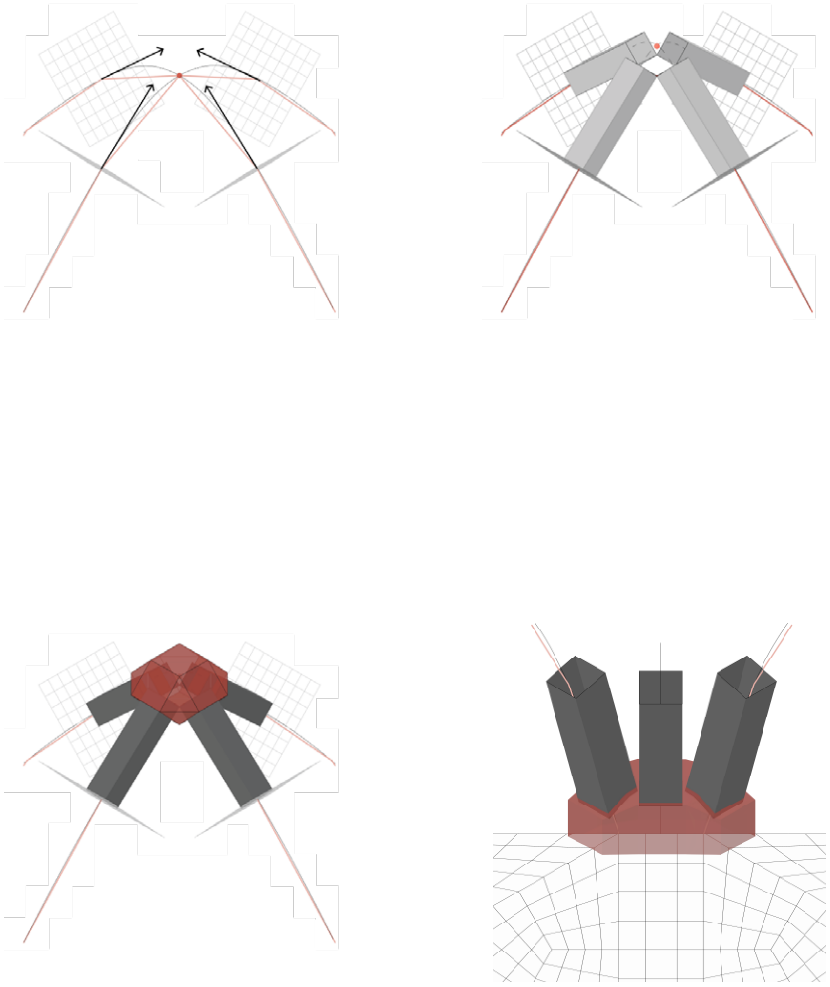


Figure 94: Step analysis for the design of the keystone on the apex and on the base of the rib.

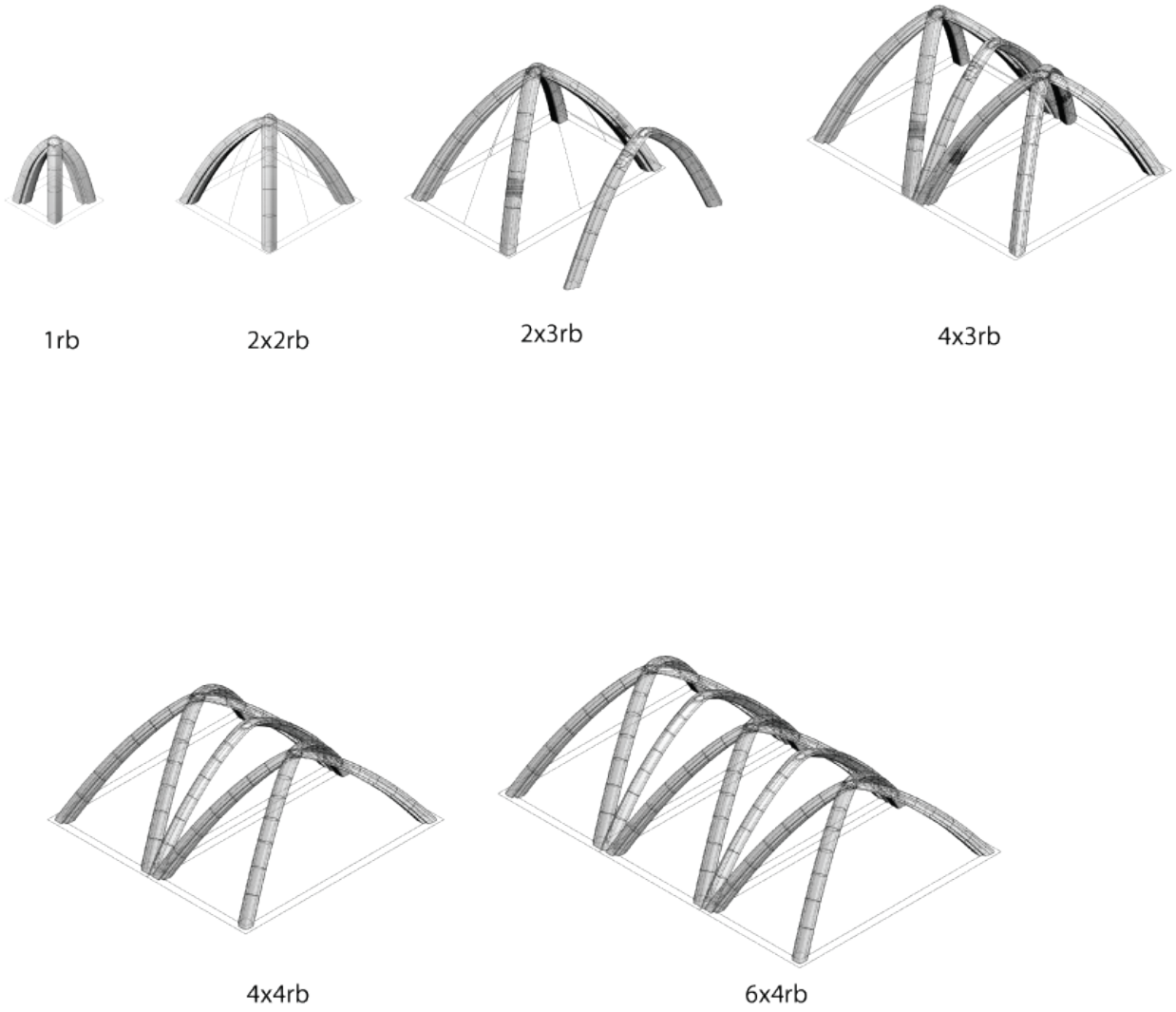
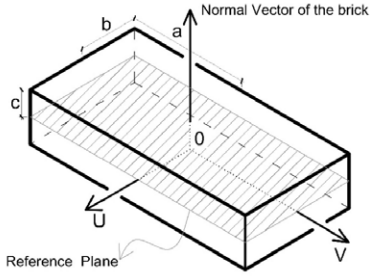


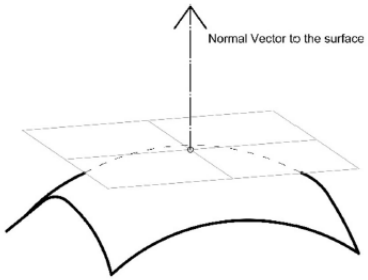
Figure 95: Rib formation Library for the Main Rooms

4.6 I Automate brick laying

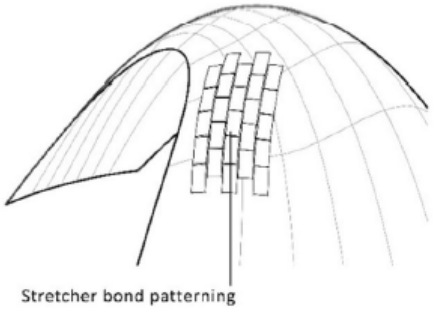
In our first structural system chosen, a filling is needed. We wanted to produce a brick laying tool that could apply different patterns of a common brick on a vault. it would help us digitalise the construction process and be more precise in the laying guide and more efficient in the material usage.
A python script has been started on ghpython, inspired by the work of Rajadeh Sassone as the plugins he states in the paper are no longer supported.

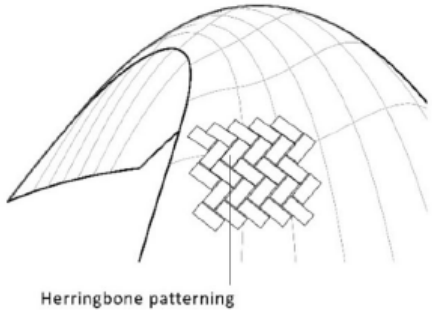


1: Parametrization of the brick



2: Normal vector at anypoint of a surface





3: Brick laying with starting point and pattern

Figure 96: Parametrization of brick laying
Source:Rajabzadeh-Sassone2017_Article_BrickPatterningOnFree-FormSurf-2

Directions	Flip brick
1	0
2	1
3	0
4	1
5	0
6	1
7	0
8	1
9	0
10	1
11	0
12	1
13	0
14	1
15	0
16	1

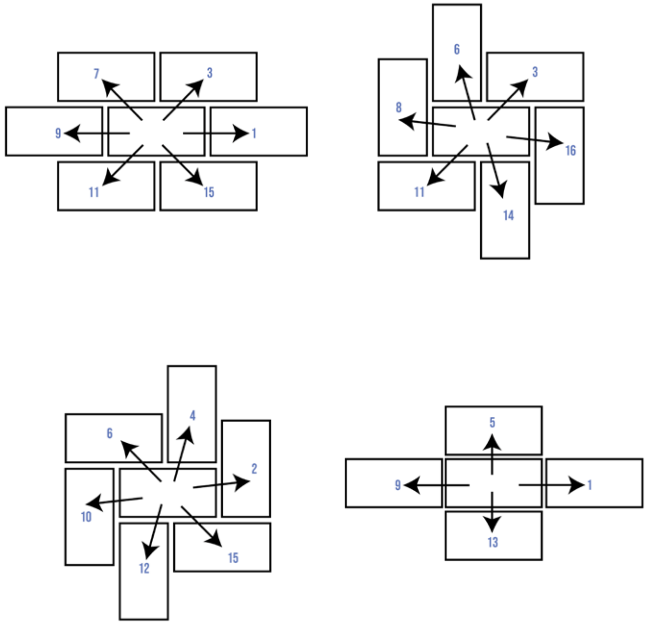


Figure 97: coordinates of brick centers regarding pattern

Table 1 The coordinates of bricks regarding the brick dimensions and mortar gap

Point	X	Y
1	2a + d	0
2	a + b + d	a − b
3	a + d/2	2b + d
4	a − b	a + b + d
5	0	2b + d
6	−a + b	a + b + d
7	−a − d/2	2b + d
8	−a − b − d	a − b
9	−2a − d	0
10	−a − b − d	−a + b
11	−a − d/2	−2b − d
12	−a + b	−a − b − d
13	0	−2b − d
14	a − b	−a − b − d
15	a + d/2	−2b − d
16	a + b + d	−a + b

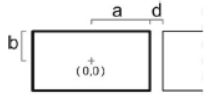


Figure 98: coordinates of brick centers regarding pattern
Source:Rajabzadeh-Sassone2017_Article_BrickPatterningOnFree-FormSurf-2

The development of the script was stopped halfway due to time constraints. It can at the moment compute the vectors for the nexts steps for the Stretcher bond and Herrigbone pattern. the strategy was to report those vector on the starting point of the surface using cricle intersection, and compute the next step as those diagram shows. Some tests would need to be done in order to validate this methods. In a more pragmatic note, this script was stopped because it is too precific compared to scope of the course.

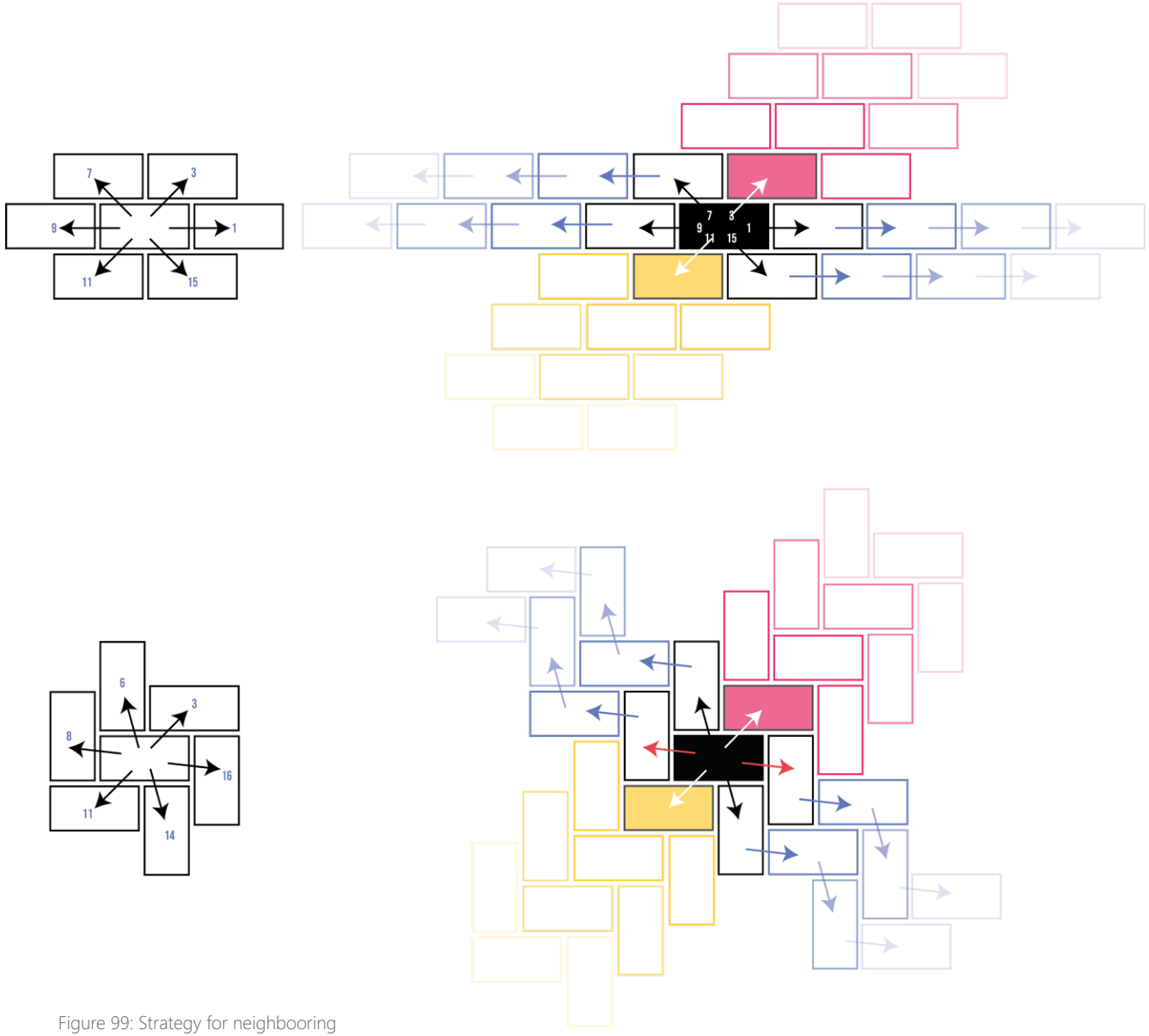


Figure 99: Strategy for neighbooring

4.7.1 I Concept and Tesselation

The muqarnas is a traditional Islamic architecture element. The decorative composition consists of blocks with concave interior surfaces. A layer of stones is projected beyond the previous layer, creating rich and articulated faces that progressively escalate, and as a result, create three-dimensional shapes.

The geometrical composition of these Muqarnas bases its form on an octagonal base. The pattern repetition rotates every 180 degrees per section. And, its number of sides beginnings at 8, and it doubles the amount every two levels starting from the second position. Thus, layer one has eight vertices, the following two 16; the next ones 32; progressively, this pattern repeats, doubling the size and reducing the radius to the point that the geometry becomes a circumference. Finally, the union of these polygons results in the creation of blocks used to construct the Muqarnas.

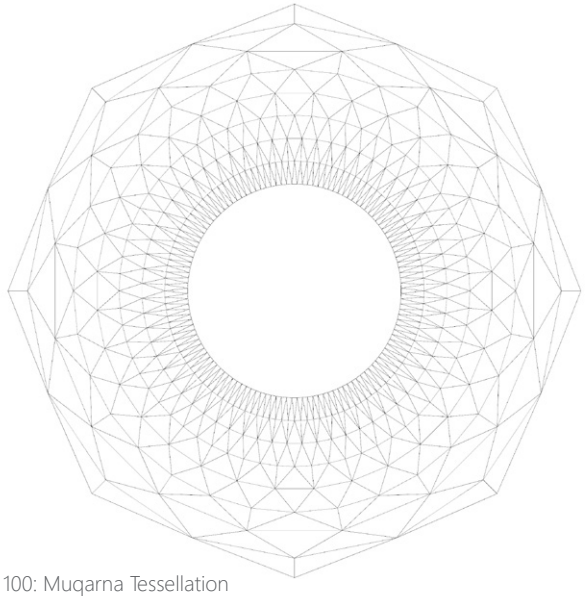


Figure 100: Muqarna Tessellation

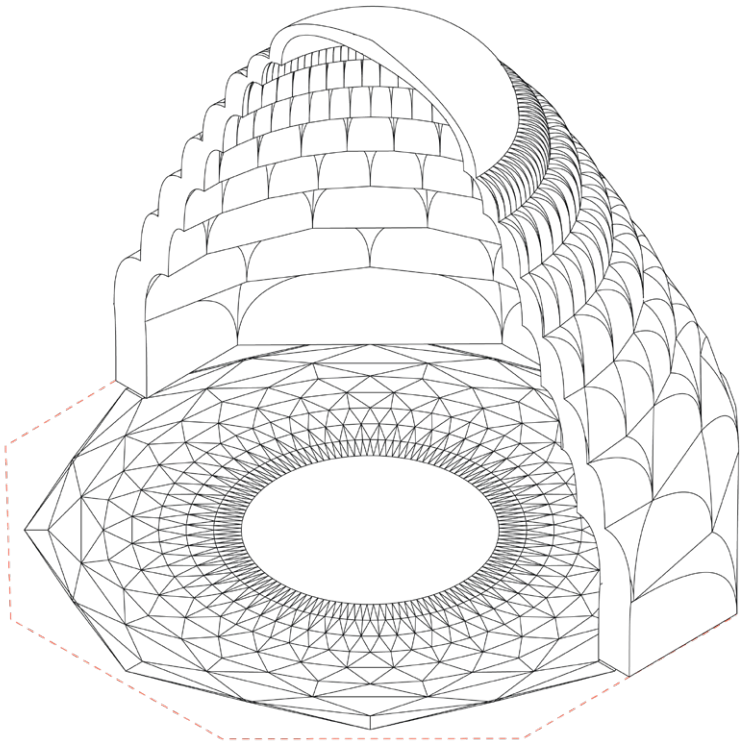


Figure 101: Muqarna - bloack projection

Figure 102: Reference Muqarna
Source: https://www.alamy.com/muqarnas-of-west-iwan-friday-mosque-isfahan-iran-image62556343.html

4.7.2 I Muqarna Blocks

Due to its geometry and structural capabilities, this construction concept was assigned to rooms with big spam varying in area. Therefore, the block's dimensions are not fixated; nevertheless, the tessellation logic allows the design and projection of several options following the sense of octagonal bases and doubling the number of vertices every two levels. Many constraints guide the partition of the blocks. For instance, the vertices of the octagon are the initial guide. However, this is insufficient to guarantee a block that complies with a lightweight standard for the building process. Therefore, further separations are necessary to achieve a block that maintains the dimensions not superior to 500 mm x 500 mm x 500. Despite the geometrical achievement of this element, further research is necessary to optimize the system. For instance, the material properties of the blocks limited to compressive strength do not allow sharp corners. Thus, it is essential to increase the width of the Muqrna wall to guarantee the mechanical properties of each piece.

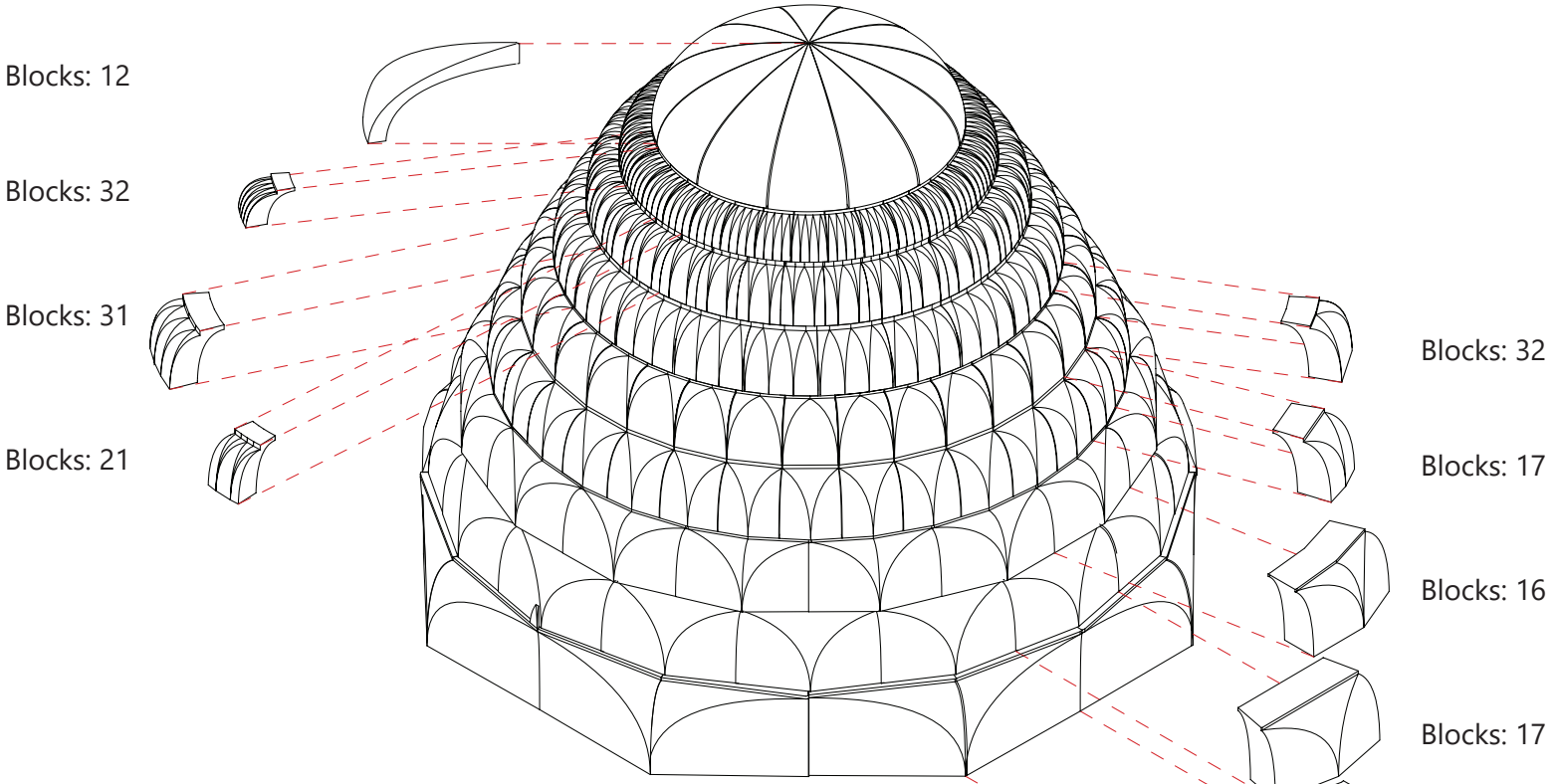


Figure 103: Block partition

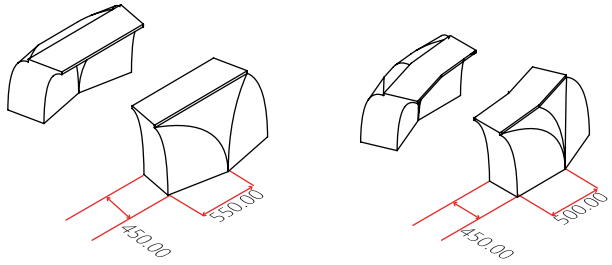


Figure 104: Block dimantion

4.7.3 | Assembly Process

Using as a reference the Domo of Brunelleschi in Florence, the Muqarnas follows the same principles. It uses an octagonal base geometry to create a large spam dome. The construction system produces rings that reduce its shape to create an escalation system that works on compression. As a principle, the bottom components required a bigger block size that progressively decreased its formation to facilitate the assembly sequence. Each ring uses one specific kind of block due to the reasons above. However, this arrangement allows a simple construction logic, as the worker can quickly identify the scale progression. Despite the number of pieces in the construction, it is relevant to remark that one type of stone has been used per ring, except for the initial base that required mirror pieces to produce the circumference. This block game simplifies the production process of the block, as it is just necessary to create one mold per each Muqarna layer. (this may variate depending on the room-scale)

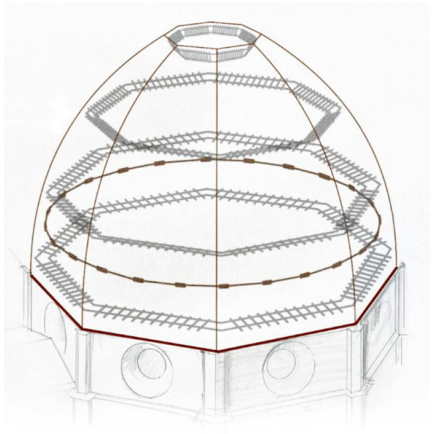


Figure 105: Domo Di Santa Maria by Brunelleschi
Source: https://www.exploringart.co/brunelleschi-santa-maria-fiore-dome/

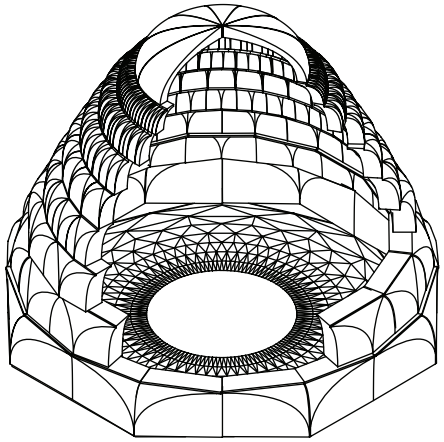


Figure 107: Assembly sequence of Muqarnas blocks

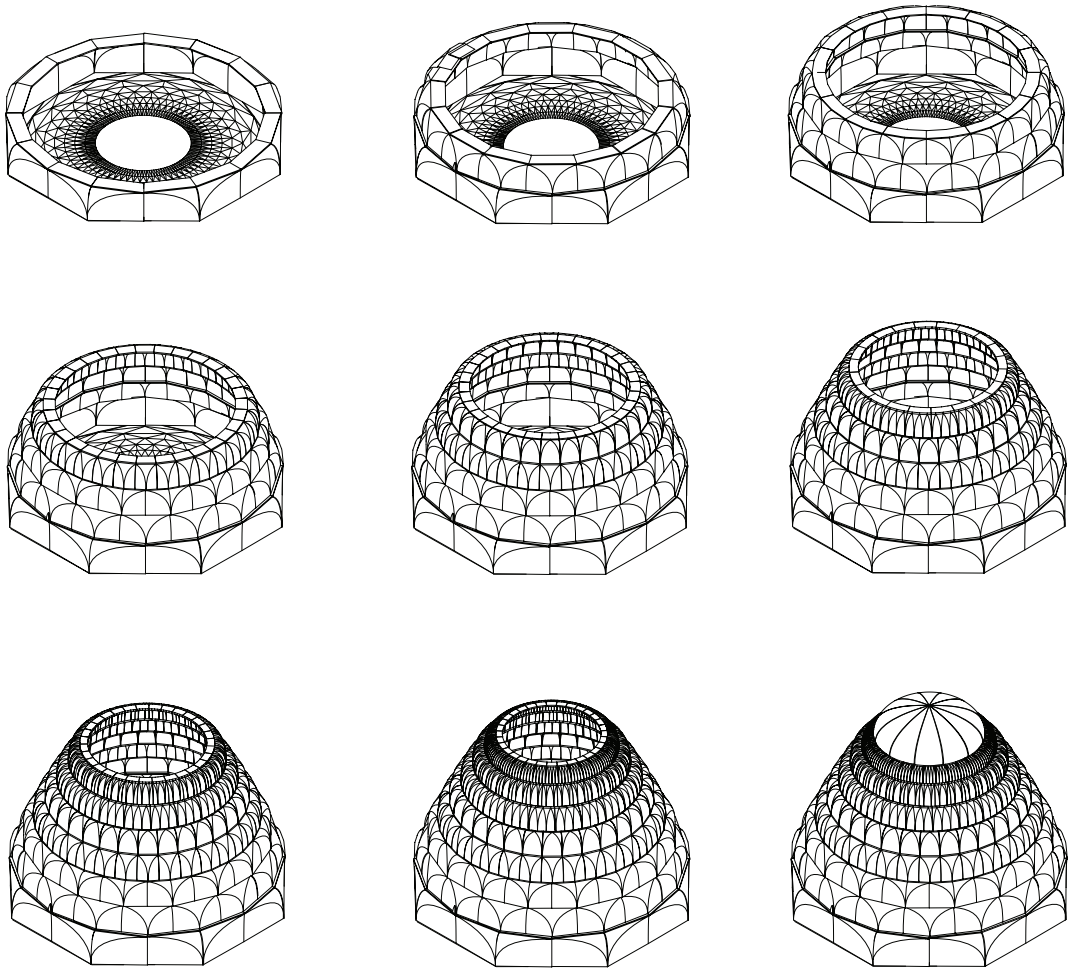


Figure 106: Muqarnas construction sequence

4.8 | Reflection

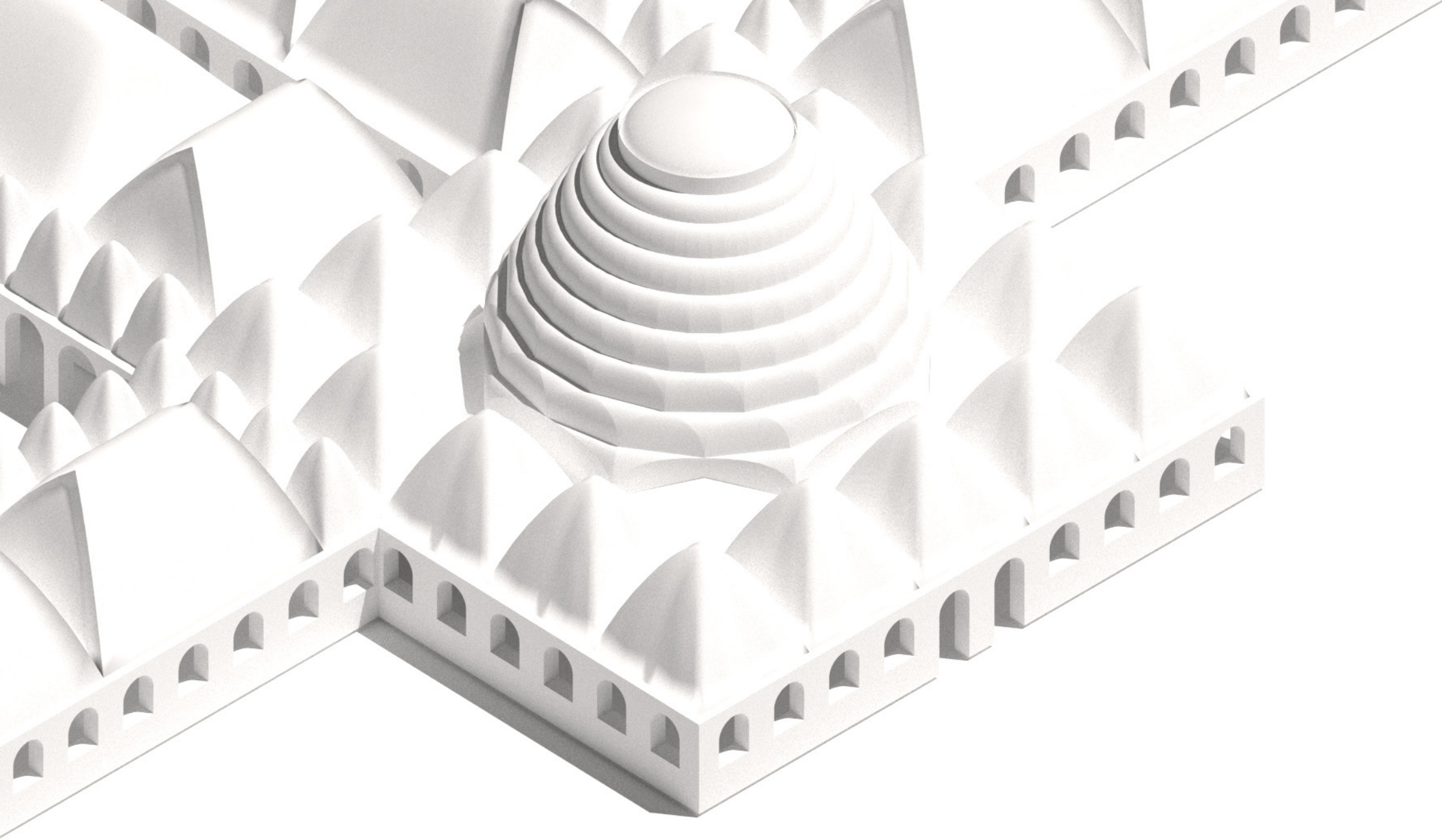
Shaping was a process that connects configuring and structuring. Thus, it merely stopped because we had to submit a final example.

One of the main things that required a lot of investigation was to standardize the rib blocks to the minimum amount of different parts, by inserting wedge-shaped blocks. By studying the angles produced between the pieces and specifically for this interlocking design this approach could not be further developed in the time of Earthy 4.0. A new system for connections should be designed to accommodate this idea.

The Tartan Grid as a primary guideline of the project helped the group to coordinate several aspects. However, the standard block size was the keystone of the project. The brick dimension clearly defines the stacking system of the walls. Consequently, this decision led to the specified width of the grid and the general modularity system of the libraries.

The module of 300 x 150 x 100 mm perfectly fits into the constraints of the construction system. Likewise, this decision reflected the coordination of the block productions process, as the hydraulic compress machines require similar measurements. Thus, knowing the brick construction requirements helped narrow down wall and roof designs.

In general terms, the brick system has coordinated the gamification process satisfactorily. Despite the design's success, time constraints restricted the further development of techniques such as the Muqarnas.
For instance, the block modularity accomplishes the form-finding of the object. Nevertheless, the pieces in more detail require a more defined process to guarantee the proper material strength in compression. The blocks present narrow and sharp edges not characteristic of earth construction systems. Finally, considering the time, the initial prototype is a good approximation, but additional research could improve such matters mentioned in this text.



5.1 | General Logic

FLOWCHART & ORGANIZATION

The structuring step of the project starts with the sizing and classification of the rooms into two categories - **standard** and **special** - based on their individual matrices and function, leading to two different ceiling designs and structural solutions - **ribbed vaults** and **muqarnas**.
The analysis is then executed in a series of design iterations and adjustments until a final (and stable) structure is obtained and validated for construction.

Disclaimer: Due to time constrains and the complexity of the project, the detailed structural analysis was developed only for the most occurring conditions in the project - standard rooms. As a rule of thumb, the structural analysis for the muqarna ceilings should follow the same methodology and similar steps. The main difference would be the approximation strategy for construction and its analysis. Meanwhile the ribbed vaults were simplified to a beam + shell structure, the blocks which compose the muqarnas would be simplified to a spatial truss and analysed accordingly.

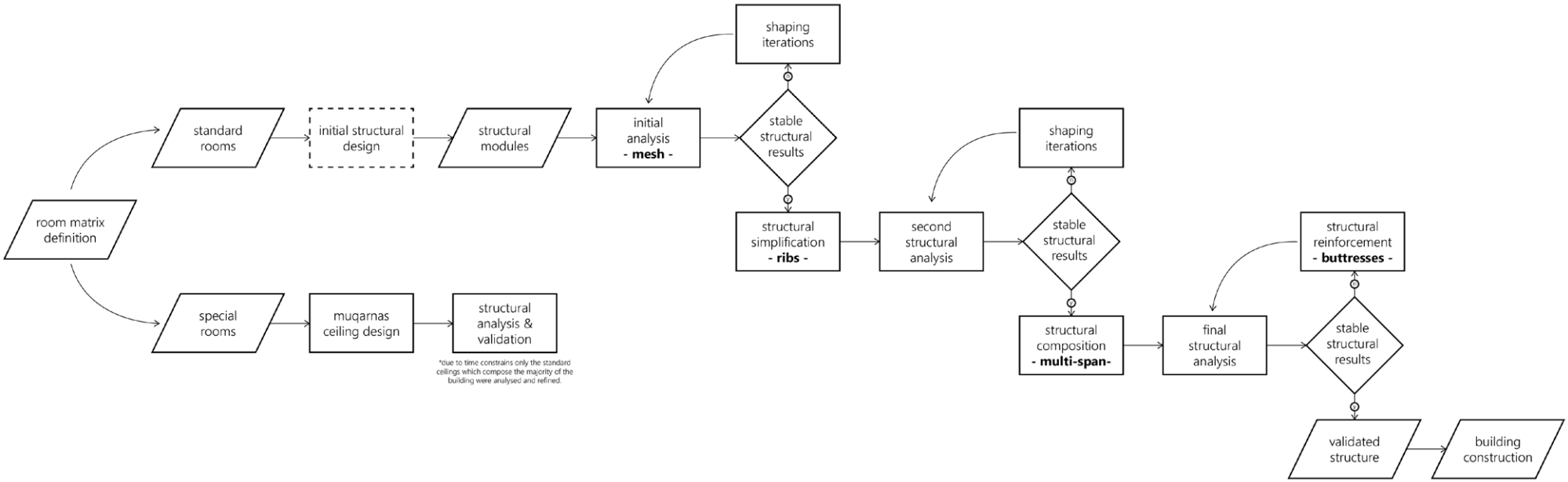


Figure 108: Structuring Process Flowchart

PROCESS FLOW & METHODOLOGY

The structuring process is deeply entwined with the shaping process, demanding several design iterations in search of the ideal form and sizes for the ceilings, at first, during the initial design, and for the structural elements at last, during the final analysis.
For the first structural analysis, different parameters for the dynamic relaxation of the ceiling meshes were studied to identify the ideal apex height and form.
Moving forward, different dimensions for the cross sections of walls and ribs were tested in the second step of the structural analysis to simplify and optimize the initial design for construction.
At last, structural modules were combined, windows and doors were added to the model and a final analysis executed. Where necessary, reinforcements were proposed with the addition of buttresses and the whole structural design understood, experimented and validated.

INITIAL INPUT & INITIAL STRUCTURAL DESIGN

Before performing any structural analysis, it is necessary to define a logic for the structural design and develop an initial scheme to be evaluated and refined.
It starts with input from the configuring process - the room matrix generated with standard sizes for all spaces in the project and a division between regular and special shapes.
Next step was limiting the spans to a maximum of four grid units (6.60m including the tartan grid) due to the low tensile strength of earth blocks.
Finally, a minimum number of structural modules was created. These could then be combined and replicated to form all the regular room shapes. The less variety of modules the better in terms of constructability and ease of assembly.

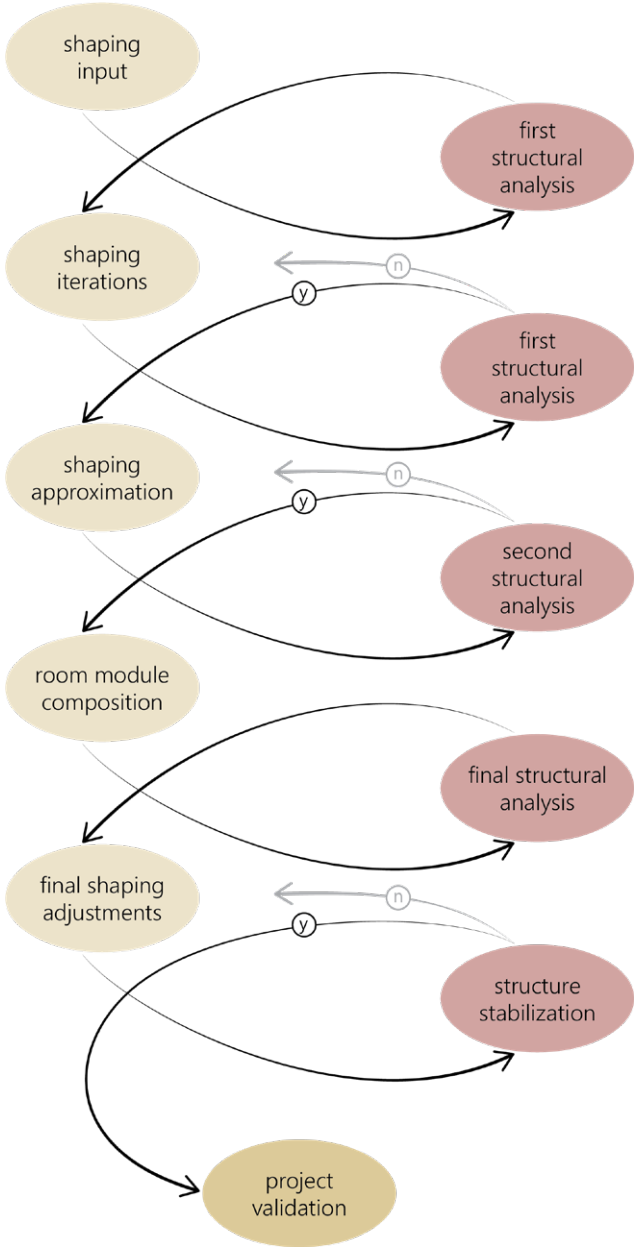


Figure 109: Detailed Structuring Process Flowchart - Structuring & Shaping Iterations

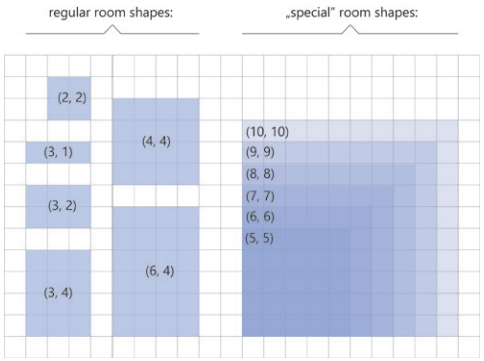


Figure 110: Room Matrices

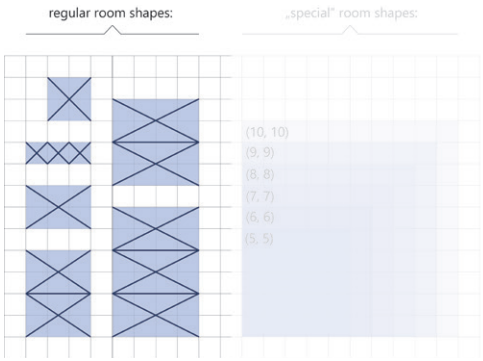
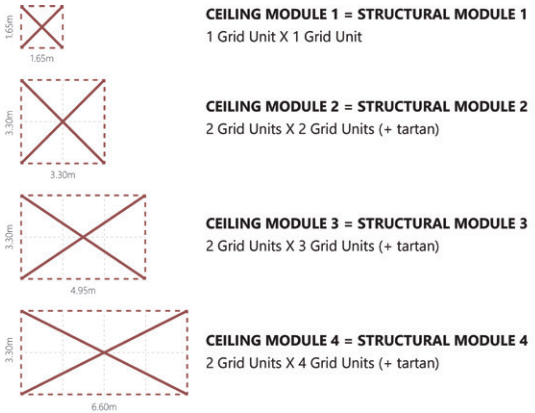


Figure 111: Initial Structural Layout

STRUCTURAL MODULES

There are six regular room shapes which compose the whole building, plus the combination of the small 1x1 modules which form the corridors. Deconstructing them into smaller units for ease of design and construction, it is possible to compose all with four different modules - 1x1, 2x2, 2x3 and 2x4 grid units.



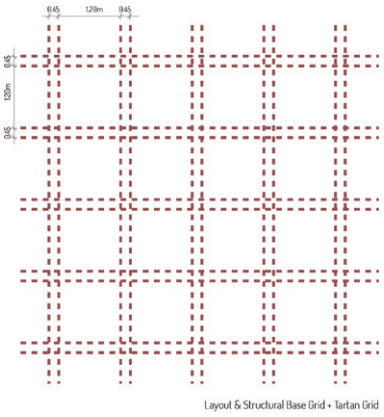


Figure 112: Structural Modules & Grid

5.2 | Material Properties & Input

EARTH MIX & PROPERTIES

The material mixture used to produced the blocks is composed of clay, sand, cement and aggregate and according to the literature consulted it has the following properties:

density	1,200 kg/m^3
young's modulus	80 MPa
compressive strength	1 MPa
tensile strength	0.1 MPa

Figure 113: Material Properties Table - *source: Earthy 2.0 Adobe CC Report, 2019*

In addition to the values above, the structural analysis demanded additional information to calculate the loads related to the ceiling infills, built with the same blocks as the ribs and walls, and the roof finishing, comprised of only sand.

density - sand	1,400 kg/m^3
density - infill	1,200 kg/m^3
density - air	1.2 kg/m^3
gravity	9.81 m/s^2
wind speed Zaatari	10 m/s

Figure 114: Additional Material Properties Table - *source: https://weatherspark.com & Granta Edupack*

ALLOWABLE VALUES

By having the compressive strength confirmed by literature, a rule of thumb was applied to define the tensile strength and maximum displacement, as described below:

compressive strength	1 MPa
tensile strength	1/10 of compressive strength
displacement	1/250

Figure 115: Allowable Values Table - *source: Earthy 2.0 Adobe CC Report, 2019*

5.3 | First Structural Analysis

SHAPING INPUT

As one of the outputs from the shaping process, a ceiling mesh is created for each one of the structural modules described above. Through dynamic relaxation of these meshes, the ideal ceiling form (conditional to the loads and spring strength considered in the simulation) was obtained.
The initial objective established was to maintain a proportional relationship between ceiling and wall heights, avoiding exaggerated high ceilings in small 1x1 spaces such as corridors and toilets, and investigating how it would affect the structural system in order to optmize the final design.

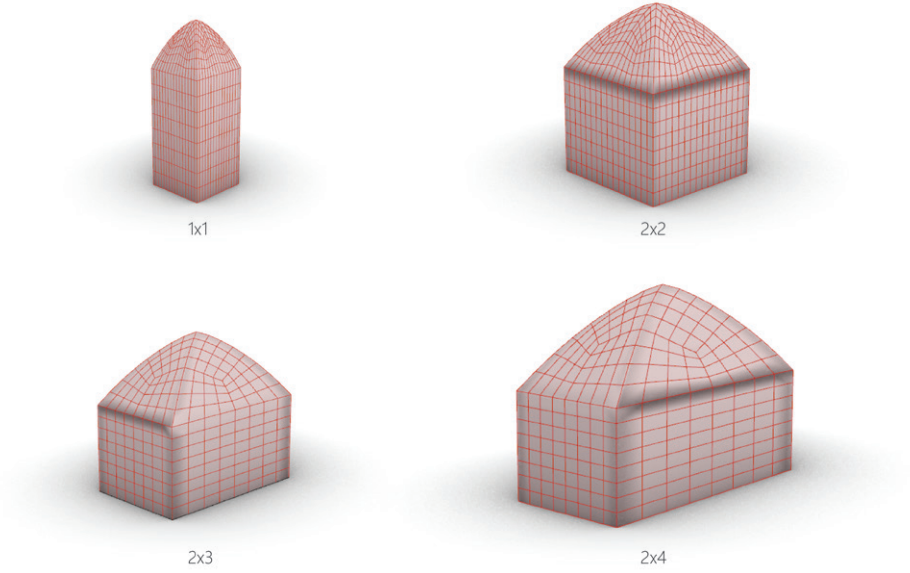


Figure 116: Dynamic Relaxation Shape Output - Low Ceilings

STRUCTURAL ANALYSIS OUTPUT

The first structural analysis was executed by simplifying the building to a single unified **mesh** for walls and ceilings, resulting from the dynamic relaxation step. The simulation calculated the walls as shells with a thickness of 45cm (equivalent to three layers of 15cm wide blocks) and had the diagrams below as output, indicating accumulated tensile stress around the „wall to ceiling" transition and accumulated compressive stress at the base of the walls.
The **tensile** stress can be explained by the horizontal component of the roof loads, more significant due to the low height of the ceiling.
The **compressive** stress can be explained by the additional wind load applied on the opposite face.
As a conclusion, the ceiling height should be increase as an attempt to reduce the tensile stress at the top of the walls.

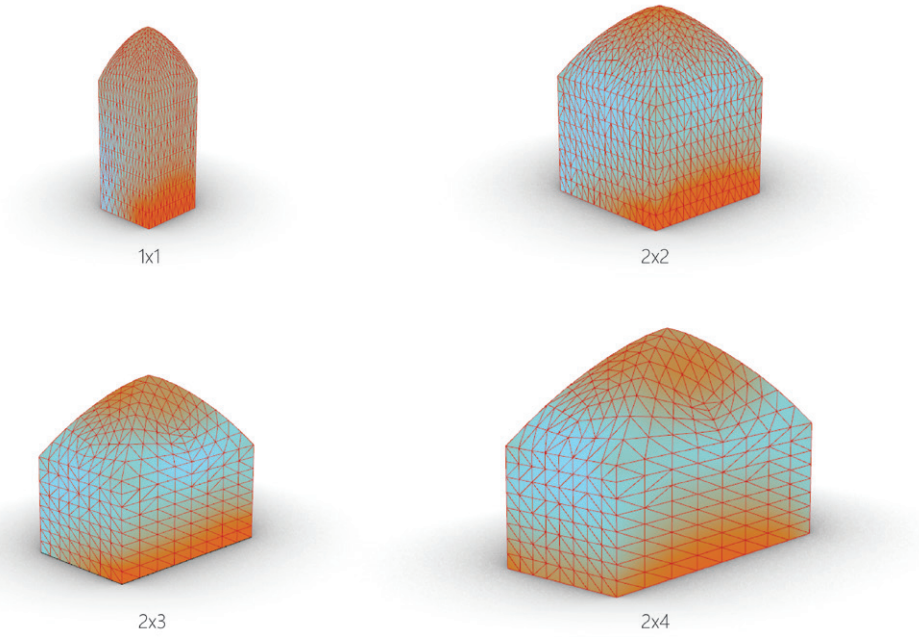


Figure 117: First Structural Analysis Output - Stress Diagrams

SHAPING ITERATIONS

In an attempt to reduce the tensile stress at the „wall to ceiling" transition by reducing the horizontal component of the ceiling loads, the dynamic relaxation parameters were adjusted (higher vertices loads and lower spring strength) to increase the ceiling heights.
As a result, the „pointy" ceilings are expected to have a line of thrust more vertical and closer to the vertical plane of the walls, thus forming a more stable structure and less susceptible to accumulated tensile stresses.

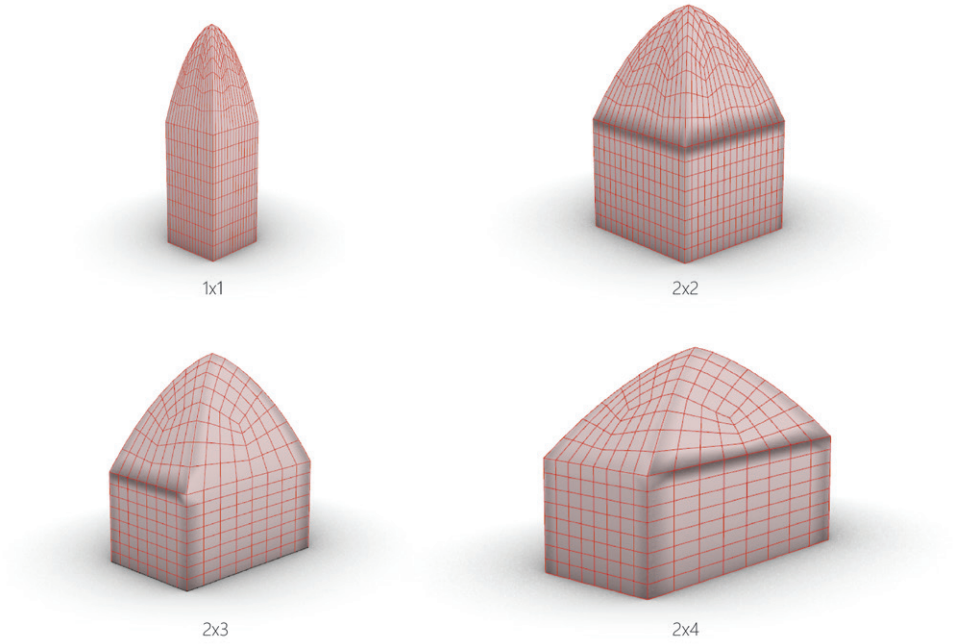


Figure 118: Adjusted Dynamic Relaxation Shape Output - High Ceilings

STRUCTURAL ANALYSIS REVISED OUTPUT

The structural analysis was performed according to the same methodology as the previous one and, as it can be visualized on the diagrams below, had similar results. Zones of tensile stress can still be found at the „wall to ceiling" transitions, although the stress values are lower than the ones previously obtained.
By comparing all the results of this first structural analysis, it is certain that higher ceilings collaborate in reducing the tensile stresses, despite not eliminating them. Only by extending the curvature to the ground and integrating wall and ceiling profile line into one catenary arch it would be possible to reach the ideal compression-only shell structure.
Since the values obtained are still within the limit values defined above, this structure can be considered viable and ready for further analysis.

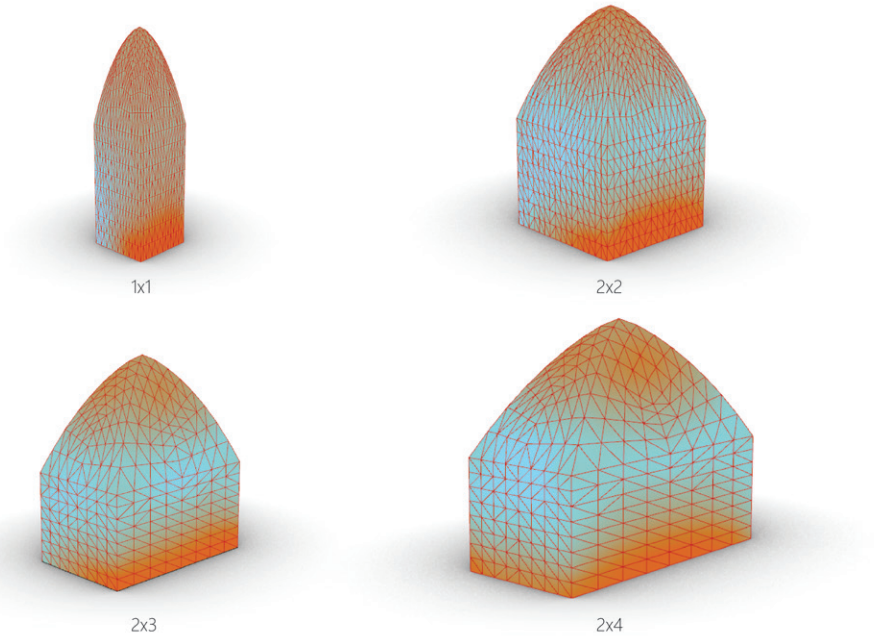


Figure 119: First Structural Analysis Output Revised - Stress Diagrams

5.4 | Structural Approximation & Simplification

FROM SHELL TO RIBS

Higher ceilings might not completely eliminate the tensile stresses on the shell structures analysed above, but do reduce their values. Therefore, they are the chosen option to proceed with the structural design for the project.
Since a shell structure was merely a simplification and is not the ideal solution for the material, design, required modularity and ease of construction, it was approximated and simplified to a modular ribbed vault system.
By taking the resulting geometry from the dynamic relaxation process, three vertical clipping planes are used to extract the ceiling catenary arches - two diagonals and one transversal - which will be transformed into ribs. The transversal arch will also serve as the ceiling extrusion profile for the multi-span rooms.
Walls will still be considered as meshes, as a simplification of a block & mortar stacking system, on which the ribs will be supported.

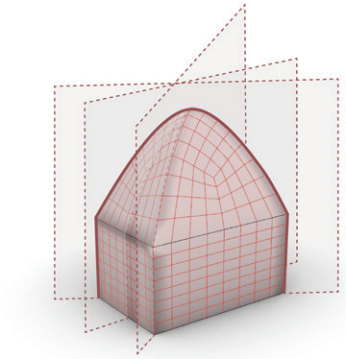


Figure 120: Structural Approximation

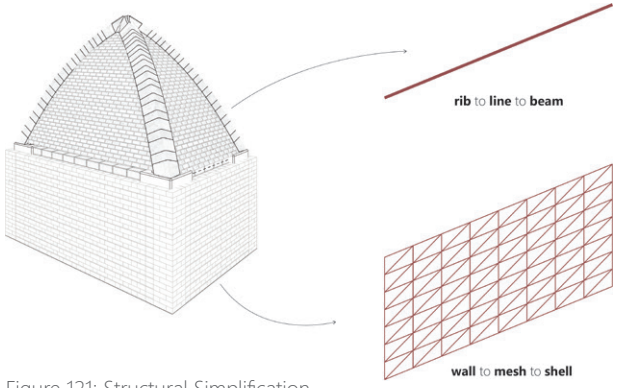


Figure 121: Structural Simplification

SIMPLIFICATION

The ribs were simplified to single polylines, which were then divided into 50 segments each, to approximate them to the catenary lines, and simulated as beams. The walls were simplified to meshes, refined with Weaverbird, and calculated as shells. To ensure the correctness of the load transfers and analysis, the vertical lines were aligned to the intersecting nodes with the rib lines.

PARAMETERS - CROSS SECTIONS

All ribs have rectangular cross sections with dimensions multiple of 15cm to match the adobe block dimensions. Standard dimensions applied are 45cm x 60cm, which have room for optimization to 30cm x 45cm for the shorter spans (1x1, 2x2 and 2x3). There is also room for optimization of the transversal ribs, which have a separated cross section modelled, and that could be reduced to a square cross section with a 30cm side for the shorter spans.

ribs	45cm x 60cm (standard) and 30cm x 45cm (optimized)
walls	45cm (or 22.5cm half of the tartan grid)

PARAMETERS - LOADS

Four types of loads were considered for all following analysis and applied either as **unform** or **point loads**.
Self-Weight is based on the gravity and material properties and it is applied as a gravity load on the Z axis to the whole model.
Infill and **Sand** are applied as point loads on each segment of the deconstructed rib on the Z axis and calculated based on the material properties and on the equivalent ceiling area to each point defined on the beam.
Wind is also applied as a point load, but only on one of the longest sides of the structure, on a horizontal direction perpendicular to the facade (X or Y axis).

self-weight	variable - based on the material and cross section properties
infill	variable - 10cm thick layer of adobe blocks
sand	variable - 10cm thick layer of sand (same area as the infill)
wind	0.06kN - strongest yearly wind in Zaatari region = 10m/s

PARAMETERS - SUPPORTS

The „naked vertices" at the bottom of the wall mesh, also the ones which are the closest to Z = 0, are all considered to be the structural supports, resembling a continuous foundation for the wall.

5.5 | Second Structural Analysis

APPROXIMATION TO REALITY

With all the parameters defined, the four single-span structural modules were simulated and evaluated based on three analysis - stresses, utilization and deformation - with positive results in all scenarios.
As expected, the results show tensile stresses at the top of the ribs and on their underside, mostly caused by the higher loads at the lower parts of the ribs leading to compression on the lower outter side. Tensile stress is also found at the top of the walls, since the load from the ribs being transferred at the corners creates compression at those locations and tension at the top of the walls which tend to deform following the line of thrust from the ribs (as seen at the bottom right image).
In terms of utilization of the cross sections, as it can be observed in the image on the right, the values are well under 10% and opportunities for optmization, specially at the higher sections of the ribs. However no optimization based on higher utilization is being proposed in this design due to the low tensile strength threshold and limitations to work with multiple values of the block sizes. This would be more effective with cast-in-place pieces such as concrete shells.
Finally at the bottom image it is possible to visualize the exaggerated deformation effect of the loads applied on the model, to illustrate its structural behaviour.
The output of this analysis is a pass to all scenarios with a possibility of structural optimization by adjusting the cross sections based on the block sizes.

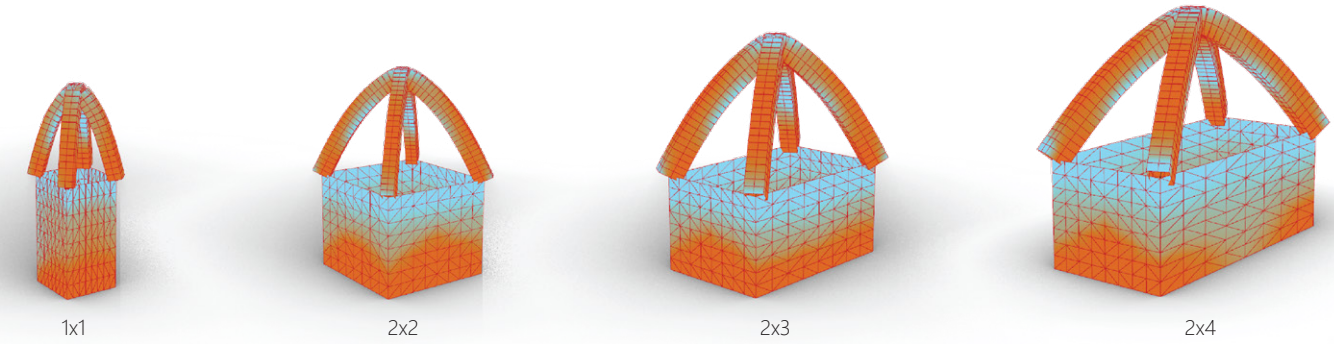


Figure 122: Second Structural Analysis Output - Stress Analysis

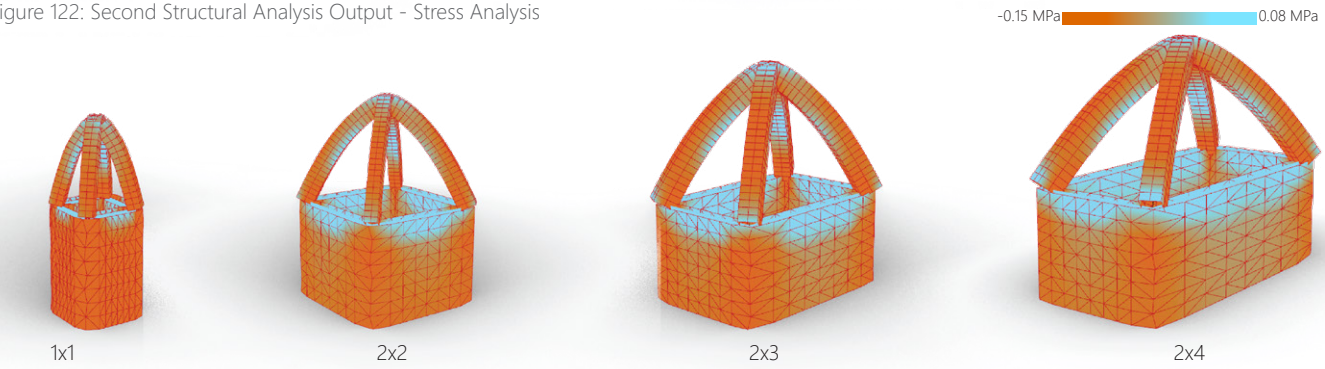


Figure 123: Second Structural Analysis Output - Utilization Analysis

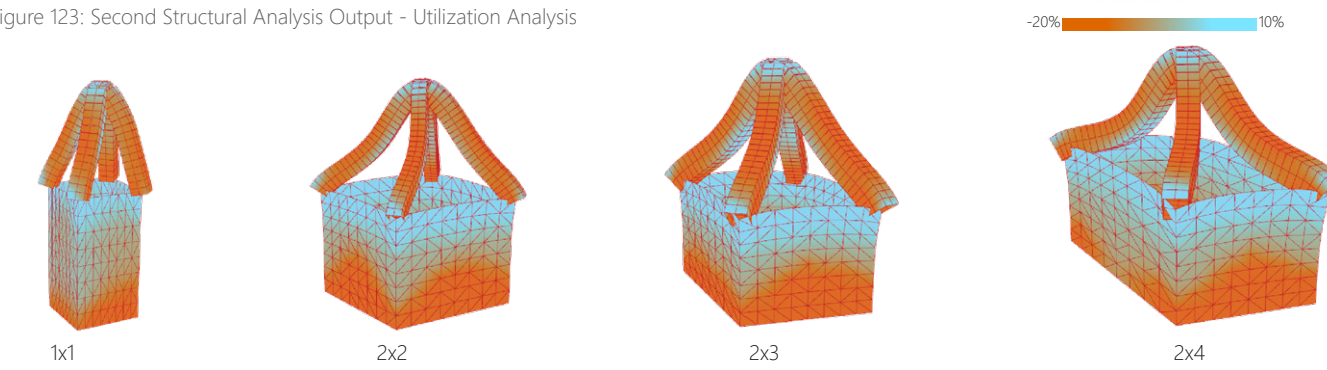


Figure 124: Second Structural Analysis Output - Deformation Analysis

5.6 | Shaping Adjustments & Composition

OPTIMIZATION

As mentioned above, the positive results from the second structural analysis, with values well under the thresholds of compressive and tensile stresses, allow for further shaping iterations and adjustments to the cross sections. Trials were performed with utilization as one of the parameters, although the results were not satisfying, since higher percentages would also mean tensile stresses above the maximum acceptable.
Still a few alternatives were investigated, based solely at manually adjusting the cross sections of ribs and walls by multiples of 15cm, respecting the adobe block dimensions. The outcome of such studies would allow for reductions on the rib cross sections to 30cm x 45cm, except for the larger 2x4 module, however to standardize construction and allow for expected higher stresses once modules are combined into multi-span spaces to form larger rooms, it was decided to maintain all cross sections at the same value - 45cm x 60cm.

ROOM COMPOSITION

With the structural modules validated it is possible to combine two or three units to form the remaining „regular shape" rooms - 4x3, 4x4 and 6x4 grid units - as seen in the bottom right picture, and perform the final structural anaysis on these more complex scenarios.
In a pre-analysis assessment, it is expected that the structures composed of only two modules will not have any structural issues. On the other hand, the largest one with three modules might face stability issues at the middle section of the longest wall, where the horizontal component of the loads transferred from the ribs might lead to excessive deformation of the walls, demanding reinforcements either to the cross section or with buttresses.

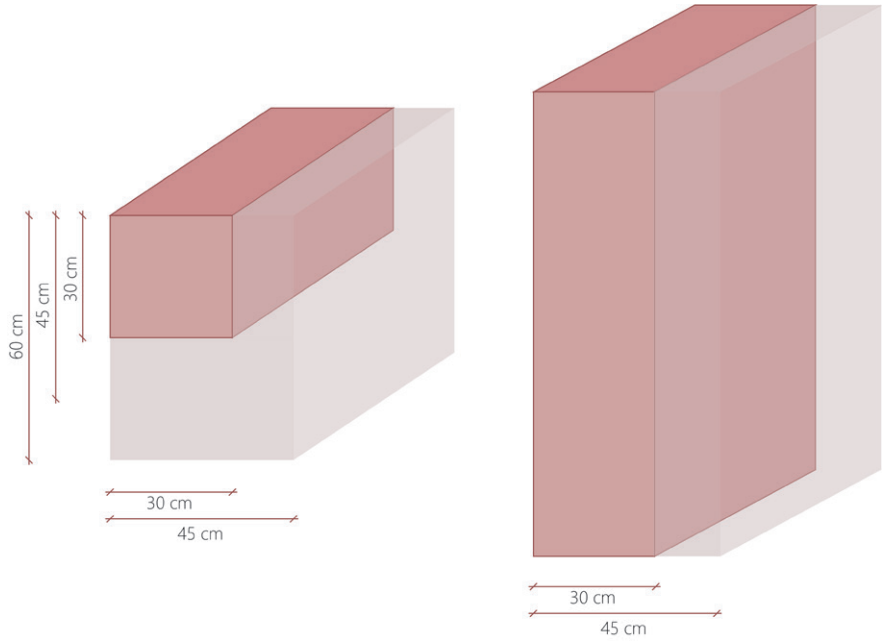


Figure 125: Beam & Wall Cross Section Size Iterations

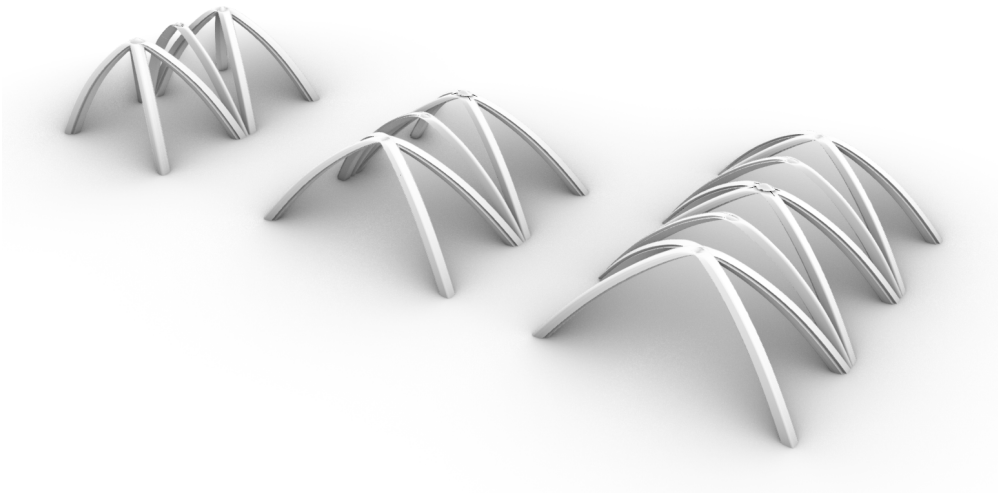


Figure 126: Large „Regular Room" Shape Structural Composition

5.7 | Final Structural Analysis

FINDINGS & CONCLUSIONS

The simulations were performed with the same parameters as the previous analysis and once again evaluated based on stresses, utilization and deformation. The results although were not entirely positive and require further improvements. The 4x3 model is the only one which can be considered stable and validated for construction, with results within the acceptable thresholds.

The remaining two models failed on the stresses analysis, showing excessive tensile stresses leading to excessive deformation and possible structural collapse.

As it can be observed in the image at the top, there is a concentration of tensile stresses at „rib x rib" and „rib x wall" connections and at the top of walls, specially at the middle section of the longest ones, crossing the tensile strength threshold.

The utilization diagrams also confirm the locations described above as the peak of utilization due to tensile stresses.

These results can be explained by the large span, (above 6 meters), by the load concentration at the weakest portions of the walls and the excessive horizontal resulting forces (with no perpendicular reinforcements) and the lines of thrust from the ribs, which all fall out of the walls.

When compared to the acceptable limits, the tensile stresses are twice or three times above the threshold, demanding a structural redesign.

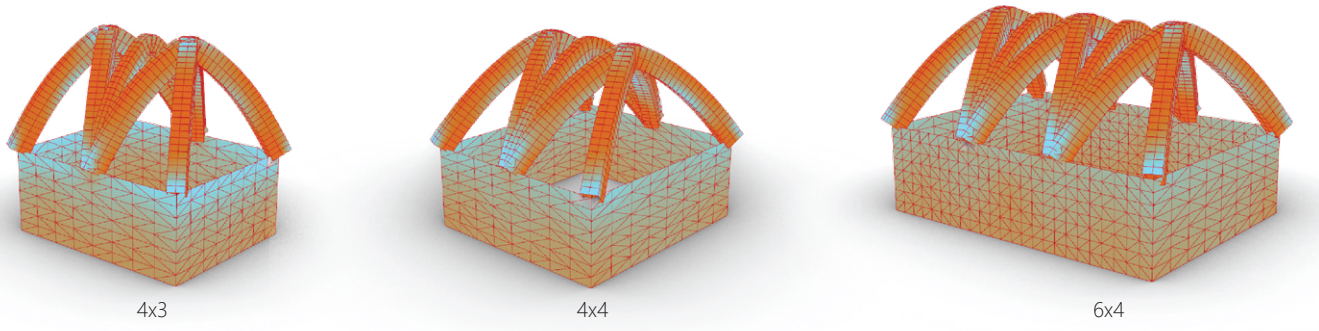


Figure 127: Final Structural Analysis Output - Stress Analysis

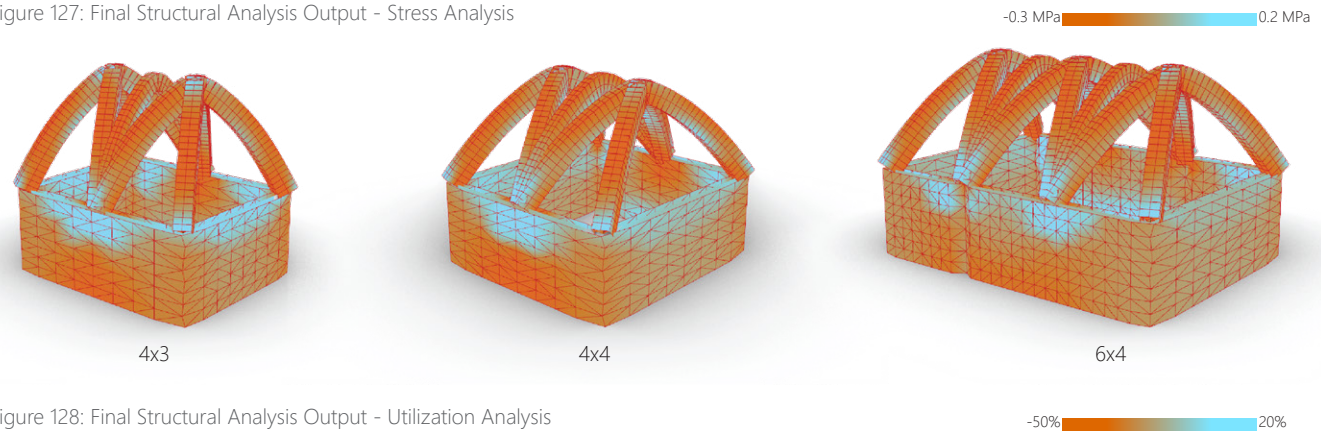


Figure 128: Final Structural Analysis Output - Utilization Analysis

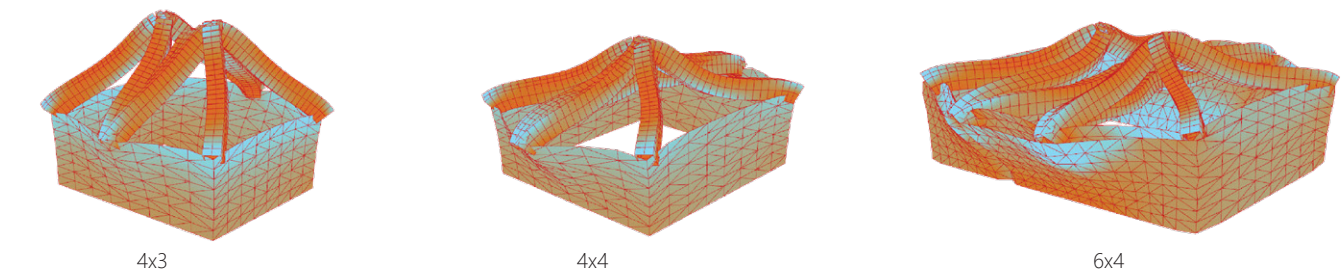


Figure 129: Final Structural Analysis Output - Deformation Analysis

5.8 | Structure Stabilization

ALTERNATIVES & CONCLUSIONS

The output from the previous analysis indicated that the two larger „regular room" shapes were not stable and required adjustments. The first iterations focused on increasing the cross sections of walls and ribs, which did not produce tangible improvements and were disregarded.
The next step considered was to increase the height of the ribs, also discarded after an initial assessment. Improvements were not remarkable due to the additional material and weight, plus the single-span module would be modified in order to keep the modularity and ease of construction, leading to an excessive increase on the material to be used.
At last, buttresses were introduced at the „rib x wall" connection points to contain the lines of thrust from the ribs and to compensate for the excessive horizontal components of the resulting forces at that connection.
As it can be observed in the top image, by adding the buttresses the stresses are better distributed througout the walls and, despite a tensile stress concentration still occurs at the start/end of the ribs, the values are within the acceptable limits.
From the diagrams at the bottom image, it can also be observed that no exaggerated deformation is obtained.
As a result of this analysis, 80cm long buttresses are the most adequate solution to stabilize the structure of the largest rooms and validate the building structure.

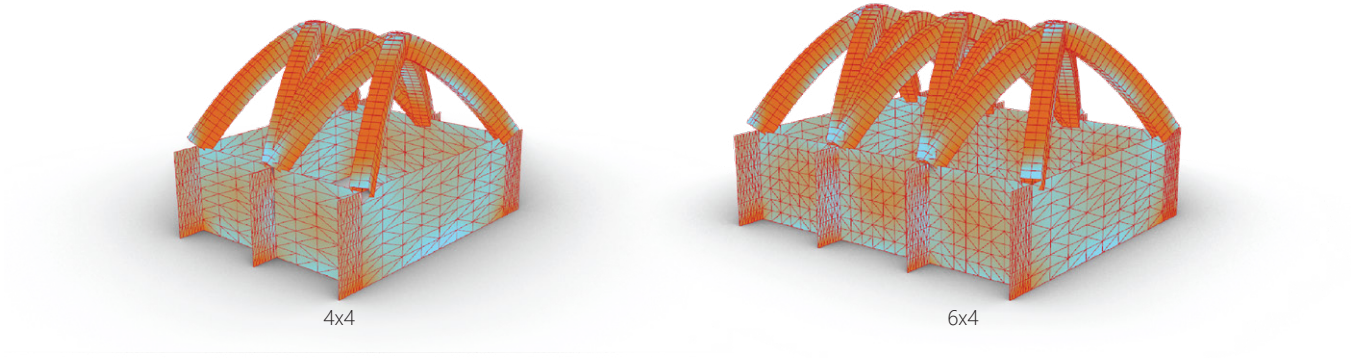


Figure 130: Stabilized Structure Output - Stress Analysis

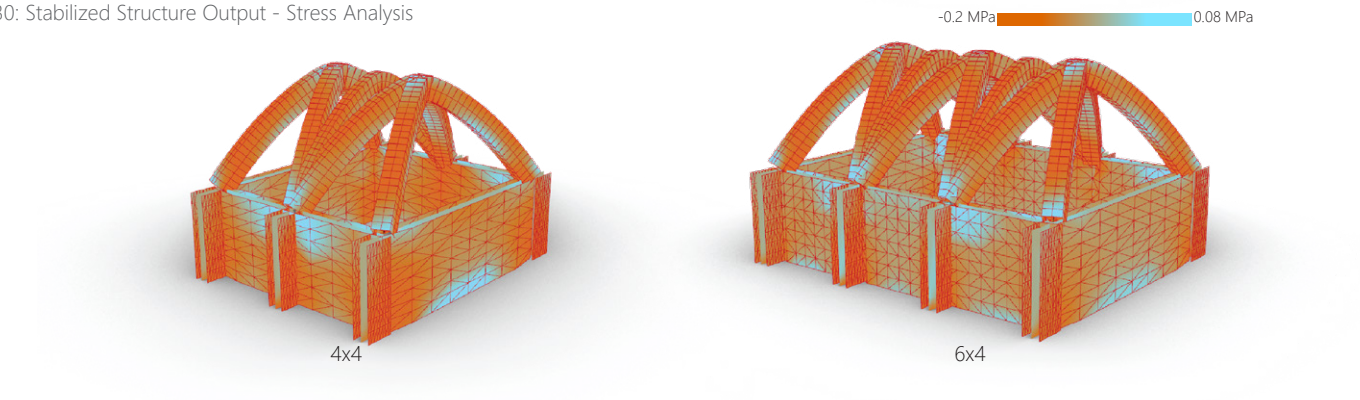


Figure 131: Stabilized Structure Output - Utilization Analysis

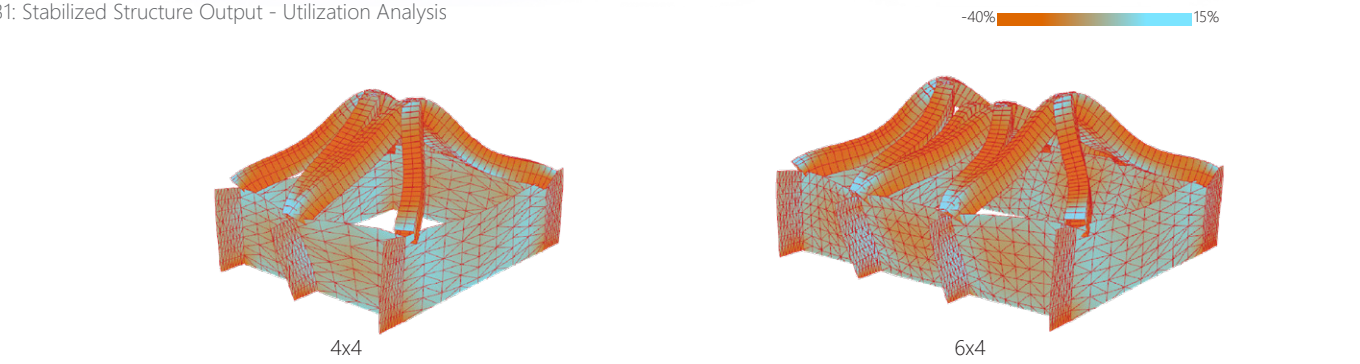
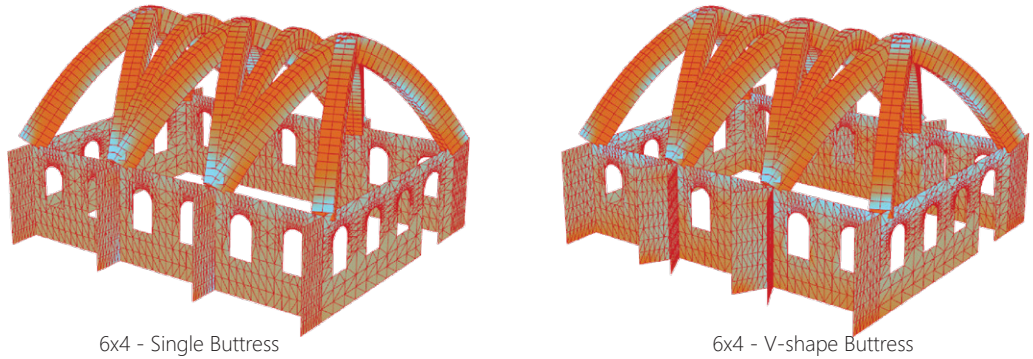


Figure 132: Stabilized Structure Output - Deformation Analysis

5.9 | Structural Case Study - Worst Condition

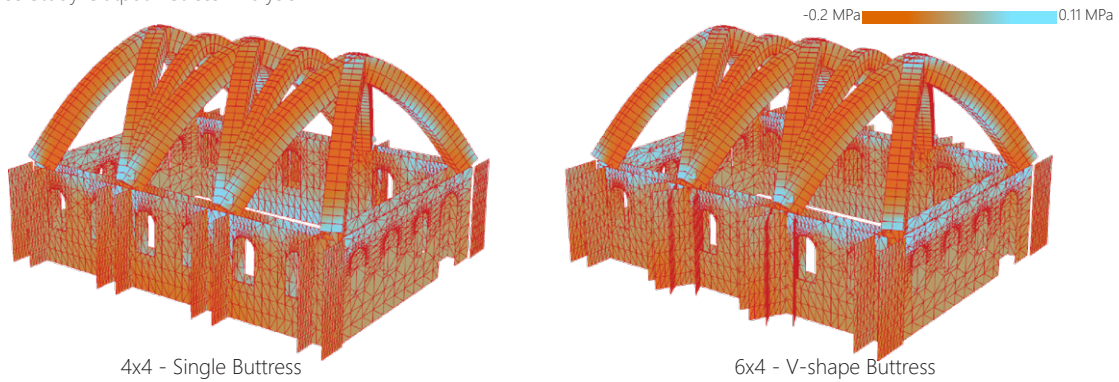
SHAPING ITERATIONS

The last step of the structuring step of this project is the analysis of the worst case scenario for the building structure to determine if all the previous findings are valid or if any additional reinforcements are necessary. The chosen condition was the largest „regular room" with all windows and doors.
As it is possible to observe in the images on the right, by adding openings to the walls the mesh becomes weaker and a single buttress at each „rib x wall" connection point is not sufficient to stabilize the structure. Despite the numbers not being excessively high, capping at 15% higher than the maximum allowable tensile stress, the ideal structure would remain entirely below this threshold.
The weakest points identified on the structure were the „rib x wall" connections, due to the horizontal component of the resulting forces, and the bottom of the buttresses, due to the wall deformation caused by the horizontal forces mentioned above.
The first iterations involved increasing the lenght and width of the buttresses, which did not show any tangible results and were quickly discarded.
The best results were achieved by modifying the buttress design to a V-shape which closely follows the lines of thrust from the ribs, creating a more suitable structural reinforcement and avoiding excessive horizontal forces at the „rib x wall" connections.
As it can be observed in the image on the bottom, by adding a V-shape buttress the structure can be stabilized and the deformation contained.



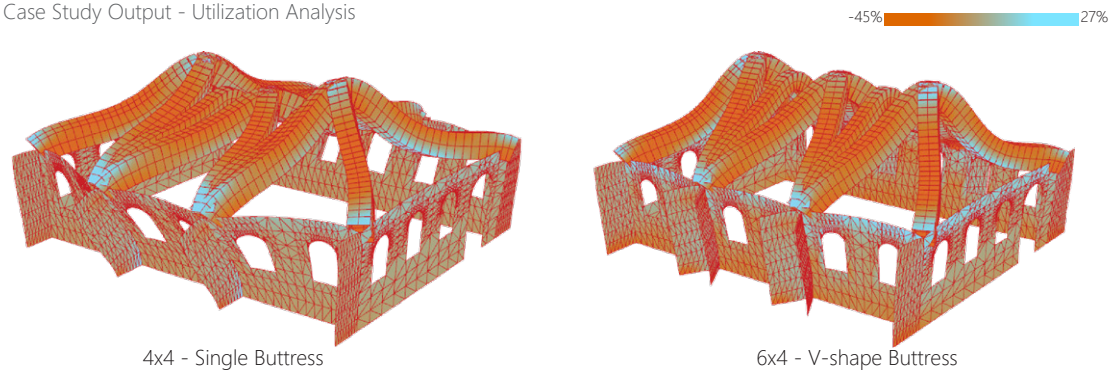
6x4 - Single Buttress | 6x4 - V-shape Buttress

Figure 133: Structural Case Study Output - Stress Analysis



4x4 - Single Buttress | 6x4 - V-shape Buttress

Figure 134: Structural Case Study Output - Utilization Analysis



4x4 - Single Buttress | 6x4 - V-shape Buttress

Figure 135: Structural Case Study Output - Deformation Analysis

5.10 | Reflection

STRUCTURAL DESIGN

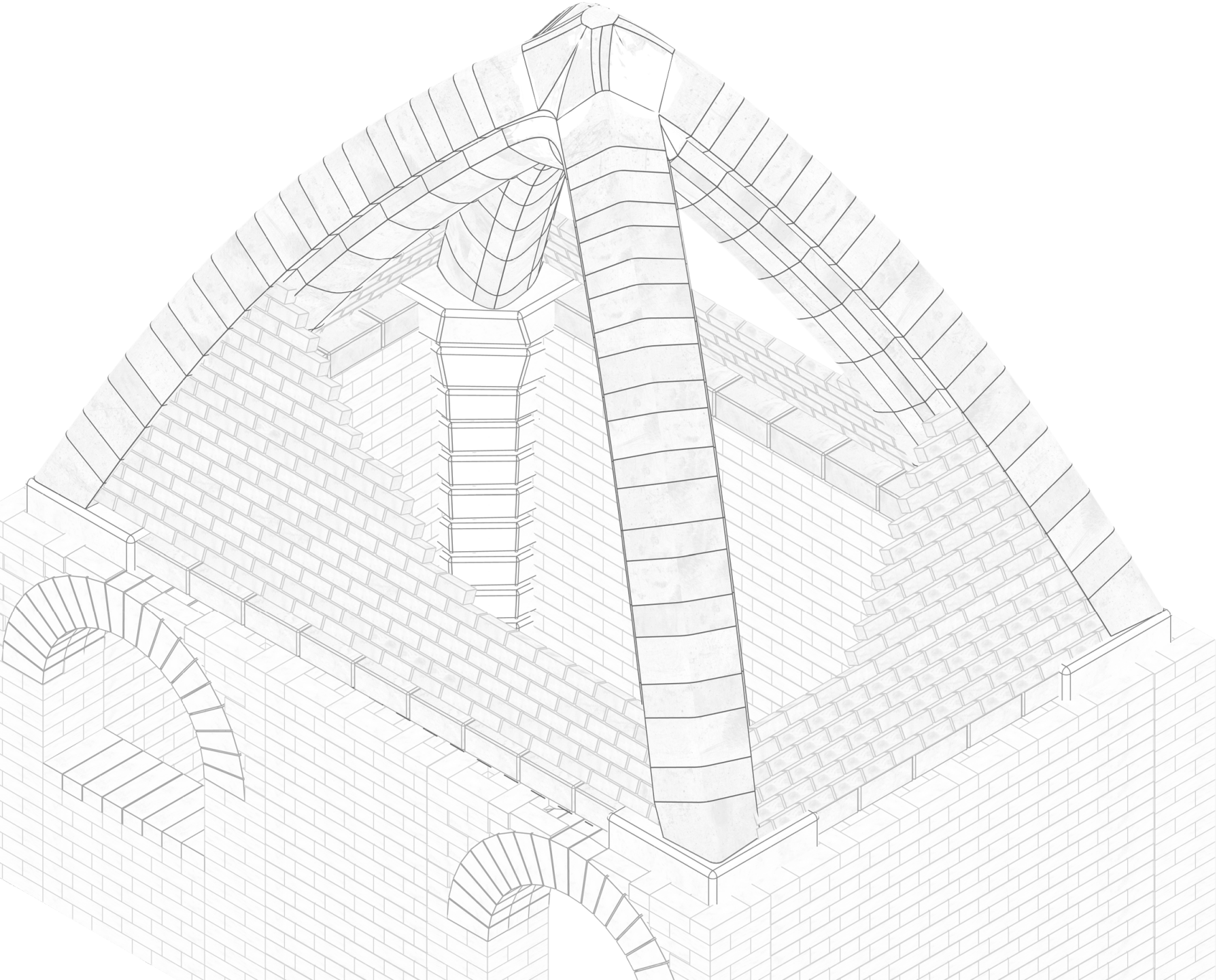
Looking back at the structural design process, the integration with shaping could be improved. Earlier iterations with the structural analysis would have facilitated the decision making in terms of ceilings and room sizes.

STRUCTURAL SIMPLIFICATION

The simplification of the ribbed vaults to lines and beams with uniform and point loads applied was a good decision to perform an analysis closer to the real scenario. In order to improve it, an analysis of the block wall simplified to a spatial truss could be executed and compared to the analysis already performed considering the wall as a mesh. By having both results it would be possible to define which analysis is more efficient in terms of precision and approximation.

MISSING STEPS

The structural analysis performed for the „regular room" shapes is complete and well detailed, providing a full insight on the structural design of our project. However, due to time constrains and the complexity of the project, the same study could not be replicated for the „special room" shapes, which have a unique ceiling design and a more challenging structure to be analysed, comprised of custom made blocks with different sizes and dimensions.
If time would allow, a detailed structural analysis of the muqarnas ceiling, by simplifying it to a spatial truss, would be an interesting challenge and a great addition to the project documentation and validation.
Another good addition to the project would be a complete structural analysis of the whole building, to optimize the use of buttresses and even cross sections of ribs.



CONSTRUCTION

6.0 | Process and Tools

PREPARATION OF SOIL

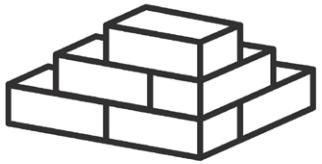
CLAY 35 %
SAND 35 %
CEMENT 7 %
WATER 10 %
AGGREGATE 13 %

MATERIAL BINDER MIX

MANUAL STEP

COMPRESS EARHT BLOCK MACHINE PROCESS

360 BLOCKS
PRE HOUR



CURE / DRYING BLOCK STAGE

32 HOURS DRYING
PROCESS

6.1 | Material Classification

OPTIMAL SOIL COMPOSITION

CLAY (0.002 mm)
35%

SAND(0.06 - 2 mm)
35%

CEMENT
7%

Aggregate (2.5mm)
13%

3.84

Av. load Kn/

BINDER PROCESS

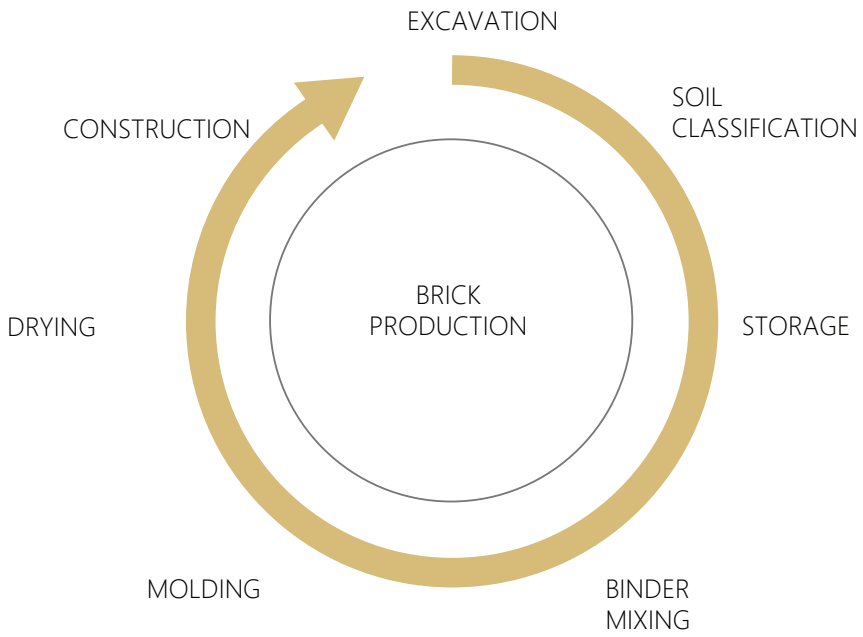


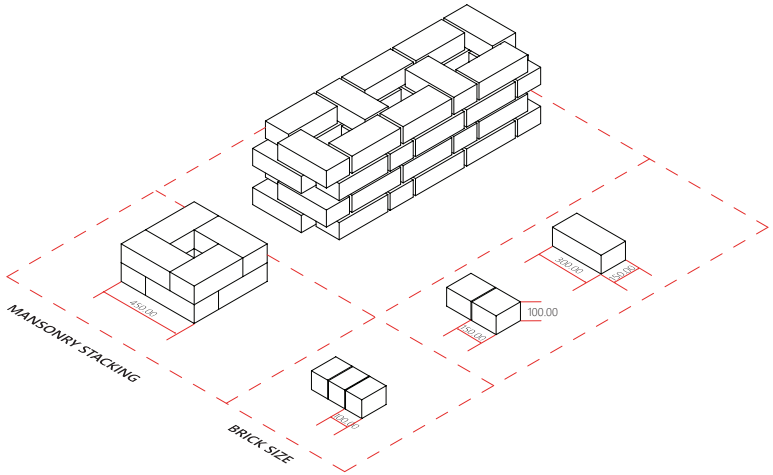
Figure 136: Brick Production process Diagram

6.2 | Earthen Block Production

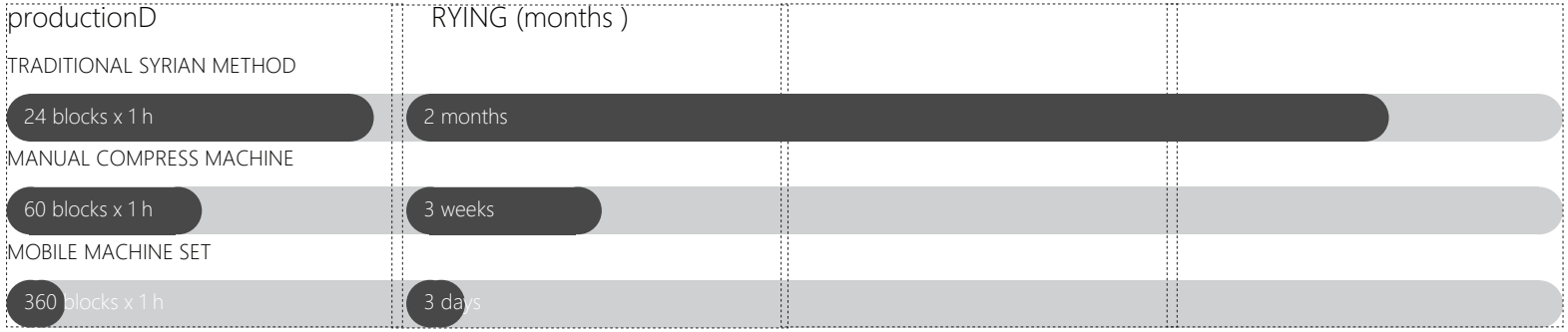
Traditional earthen blocks use a wood formwork box to mold the units. Generally, this process may take a considerable amount of days to produce a certain quantity. Moreover, the handcraftsmanship is physically demanding, and the worker 's skills condition the product quality, which means that a highly specialized workforce is necessary to achieve such a project. Due to the above reasons, the group has chosen a more industrialized method that increases the material performance, mechanical properties of the material, and production execution in terms of time. Thus, construction analysis aims to tackle these issues by implementing the mobile machine set from OSKMA. This hydraulic compress unit can produce 360 units per hour and integrate a supply chain of material binders that guarantee product quality due to its filtration system.

The decision to use a specific block dimension comes from the machinery requirements. This brick size bases its metrics on a 295 x 140 x 90 mm rectangular shape. However, due to the absence of mortar, these dimensions are readjusted proportionally to the current project brick system (300 x 150 x 10 mm). Consequently, the Tartan grid, the standard block unit, and the production process integrate a block system coordinated from the design to the building stage.

OSKAM V/F



TIMEFRAME BRICK PRODUCTION



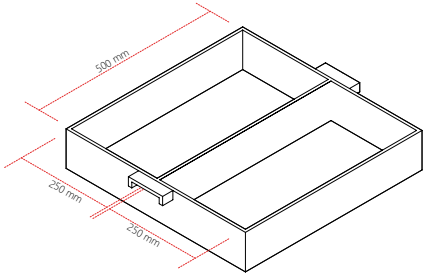


Figure 137: Traditional Brick Mold

6.3 | Brick Study

The brick stacking system selected for this process aims to increase the thickness of the wall. Therefore, Using a specific sequence, bricks are aligned in a horizontal and vertical direction. The result of this block game produces gaps in the internal sections of the wall. Wich increases the structural performance due to its width. The principle of inverted T repeats itself following the masonry structure. This step-by-step position simplifies the construction process. However, due to its interlocking strategy of double-wall, it decreases the necessity of block production, among other benefits.

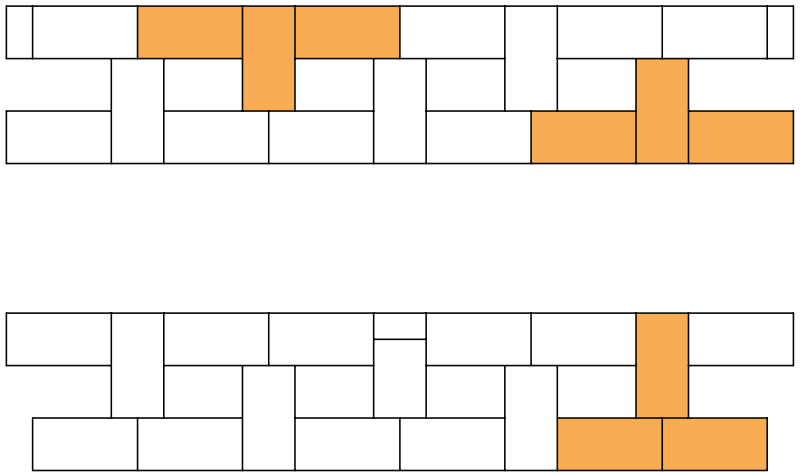


Figure 138: Floor plan brick pattern

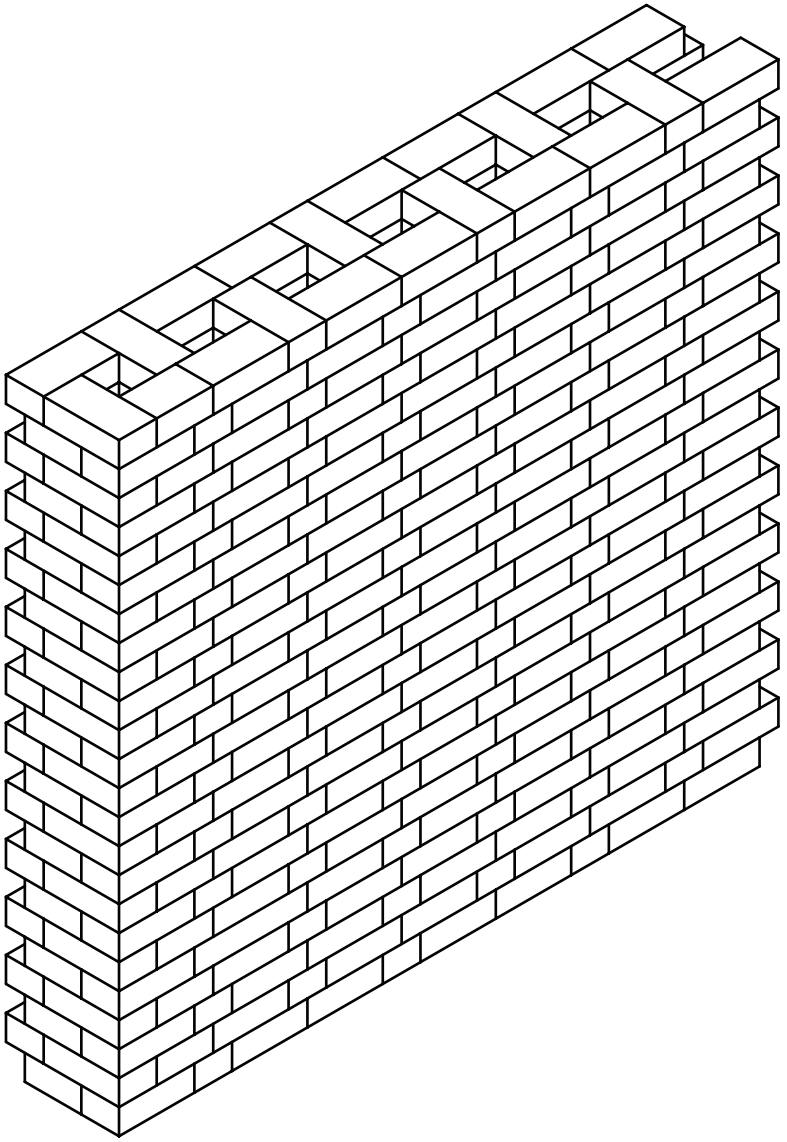


Figure 139: Isometric solution wall

6.3.1 | Corner Connection

The corner wall system guarantees stability throughout the interlocking section of two wall faces. In the detailed brick pattern, the general grid uses standardized blocks sizes. However, the corner connection requires a slite change on the interlocking section of the walls. Thus, using a 3/4 block of the same initial dimension, the additional piece can place the wall in the other direction thanks to the extra block.

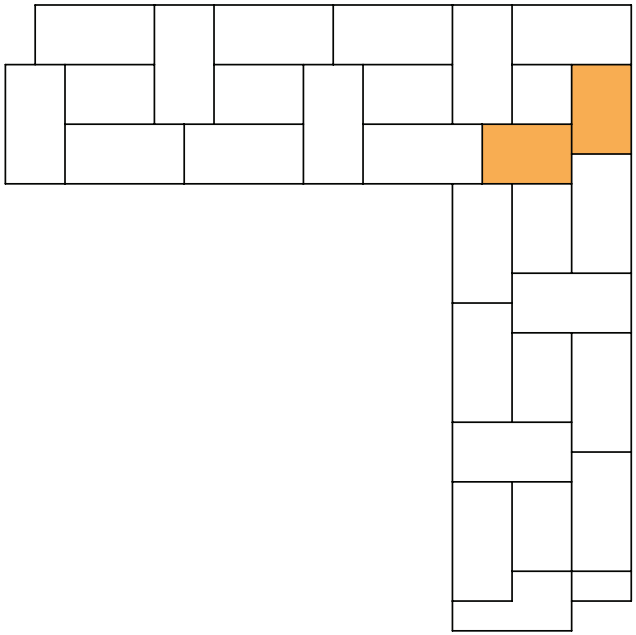


Figure 140: Floor plan corner

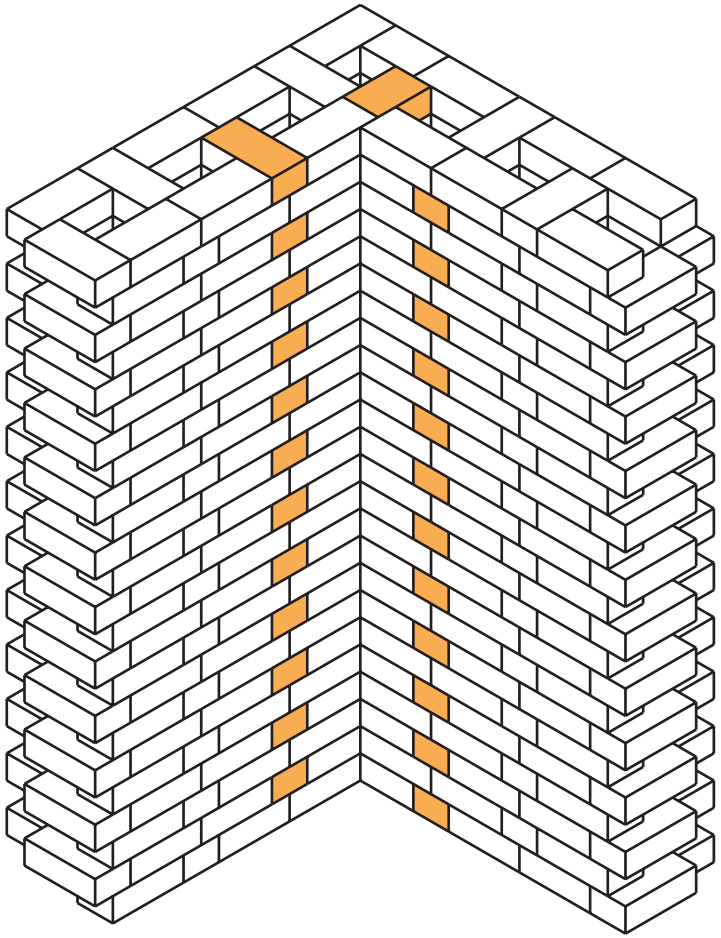


Figure 141: Isometric solution corner

6.3.2 | Windows and T Connections

The T intersection follows the same stone strategy as the L or corner stacking. However, in this case, the unique 3/4 stone is used in three points, the outdoor wall and the two intersections that connect to this point. As specified in the table below, the window solution is necessary to implement two more pieces, excluding the corner solution. The first unique stone is placed in the arch, while the second one encloses the lower frame of the window.

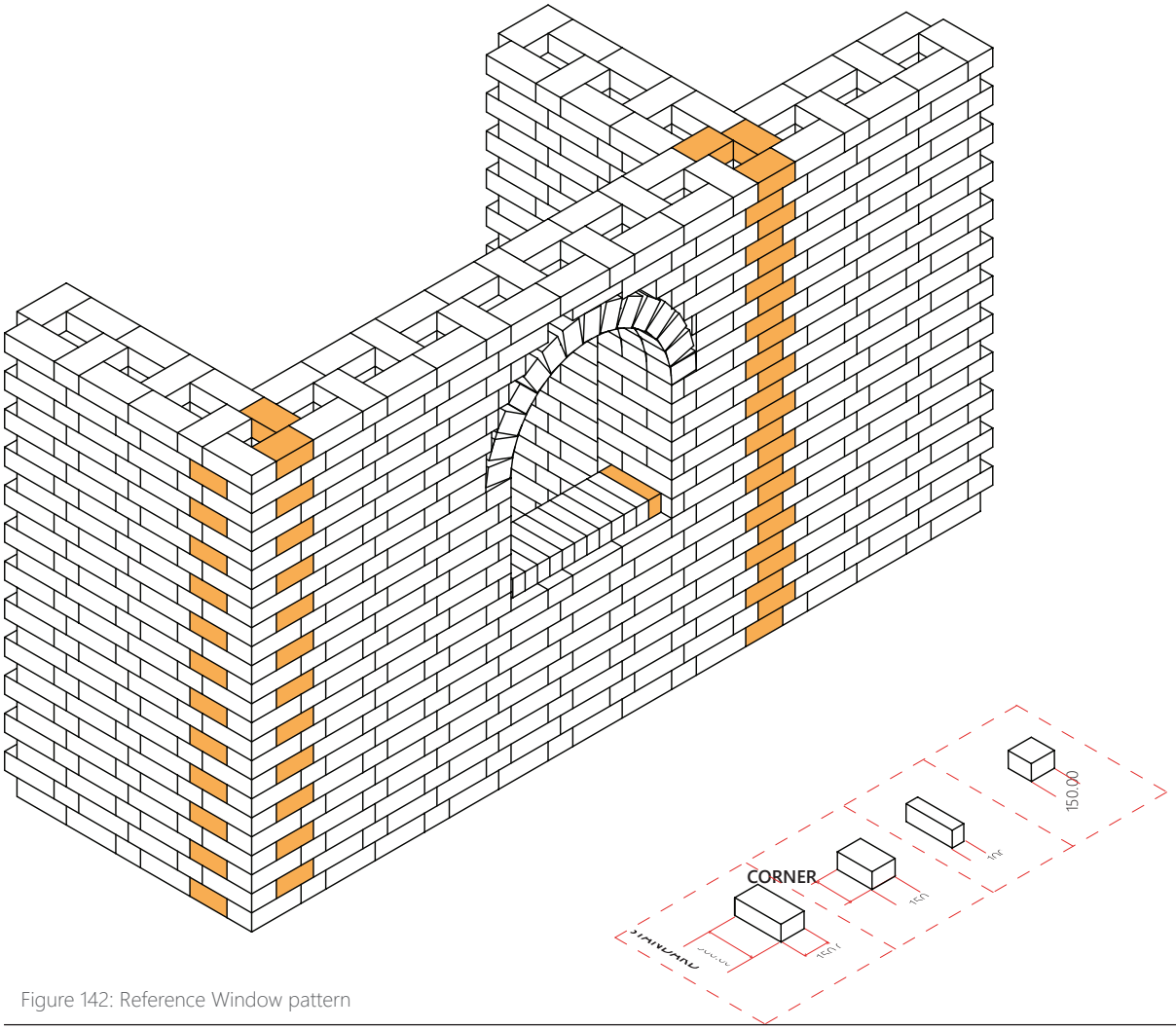


Figure 142: Reference Window pattern

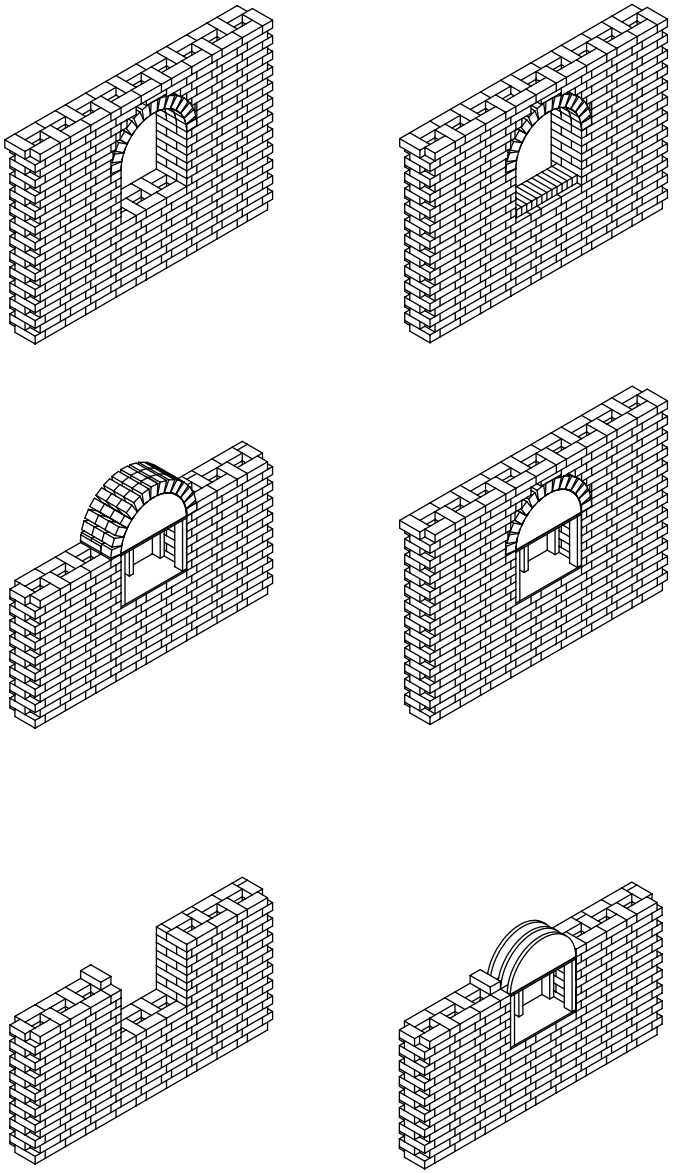


Figure 143: Window building process

6.3.3 | Doors

Following the same window construction process, the door uses plywood formwork for the arch generation and wood beans to support from the floor.

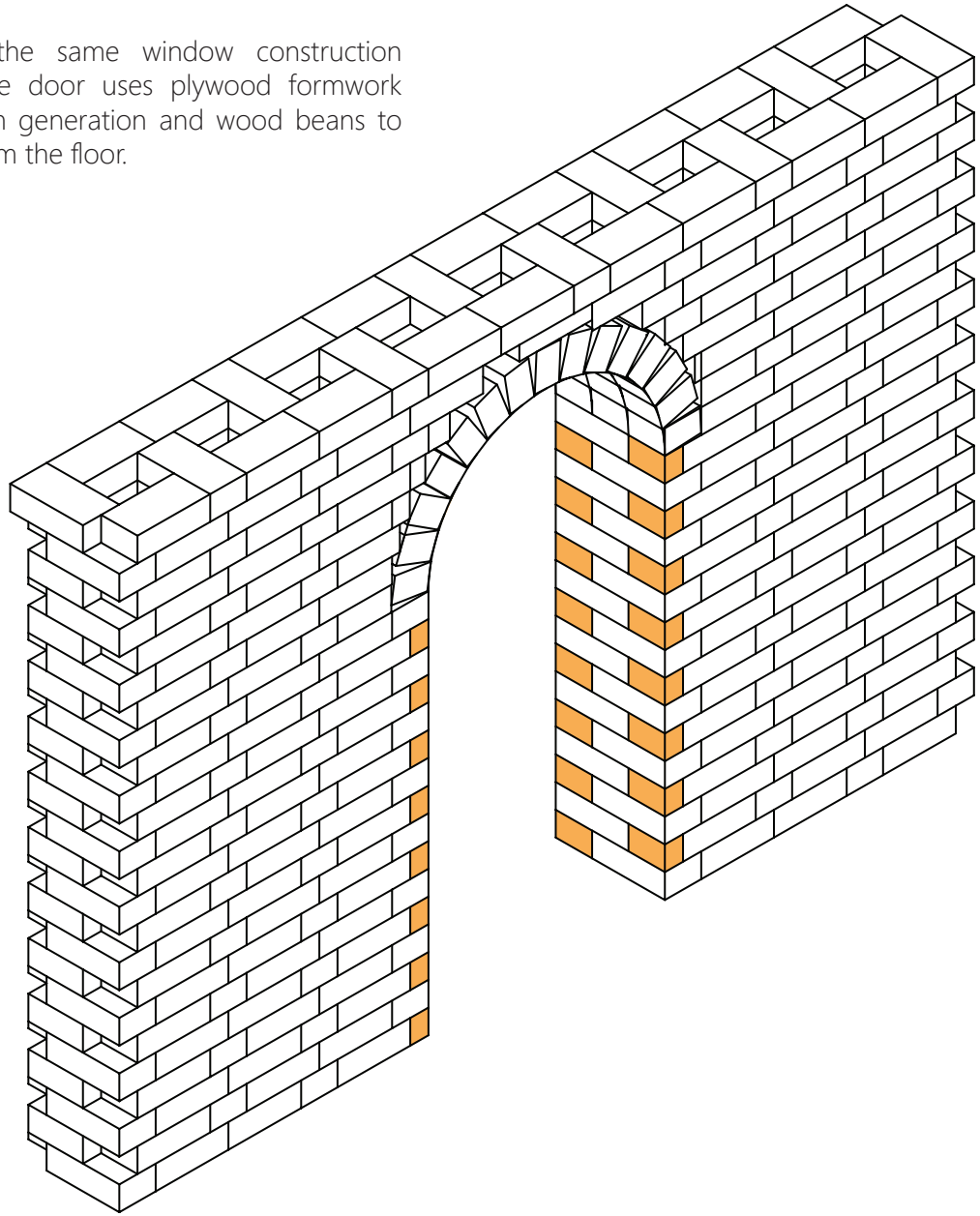


Figure 145: Door wall solution

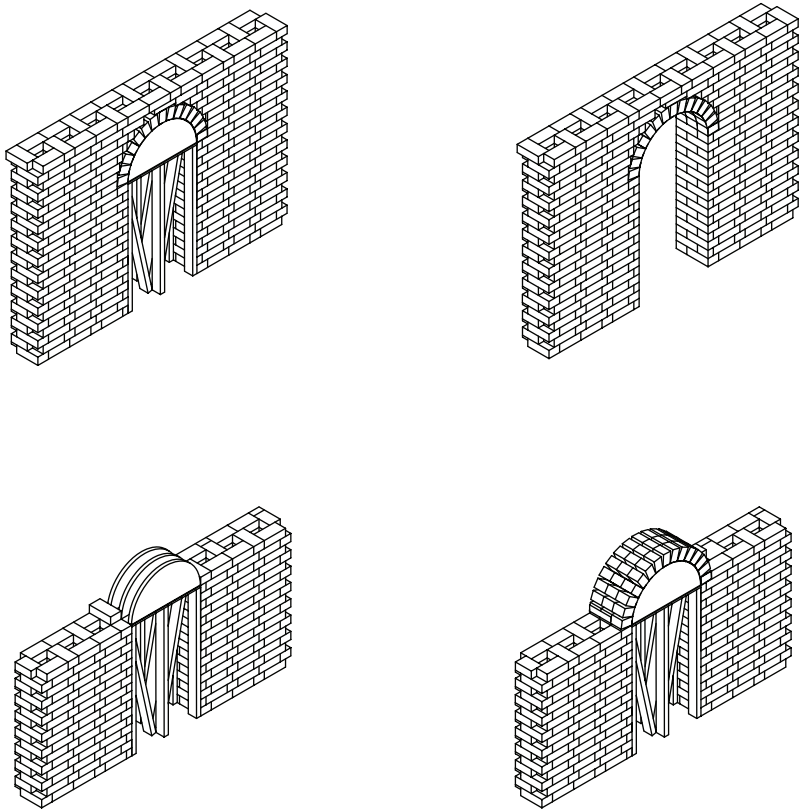


Figure 144: Door building process

6.4 | Assembly from the Walls to the Vaults

For the construction of the main rooms, once the foundation and the walls are constructed, there are certain blocks illustrated in the Figure 117, that ensure the right connection and therefore durability of the construction. These, have a different size acoording to where they are placed and are designed in such a way to ensure the right connection between the distinguishing building components.

Additionally, they might differ in shape, because their funtion is really unique and the demanded solution intricate. For instance, the ribs which were extensively analyzed in the Shaping section are such an example that require unique molds to be produced. The decision to follow this approach, made to achieve first of all, feasibility and ease in the assembly of the pieces without demanding any tool that is not found inside the camp. Even though, the proposed system of set of rib blocks is over simplified, by assigning from the beginning specific rooms into the code, one cannot ignore still the large number of molds that are needed.

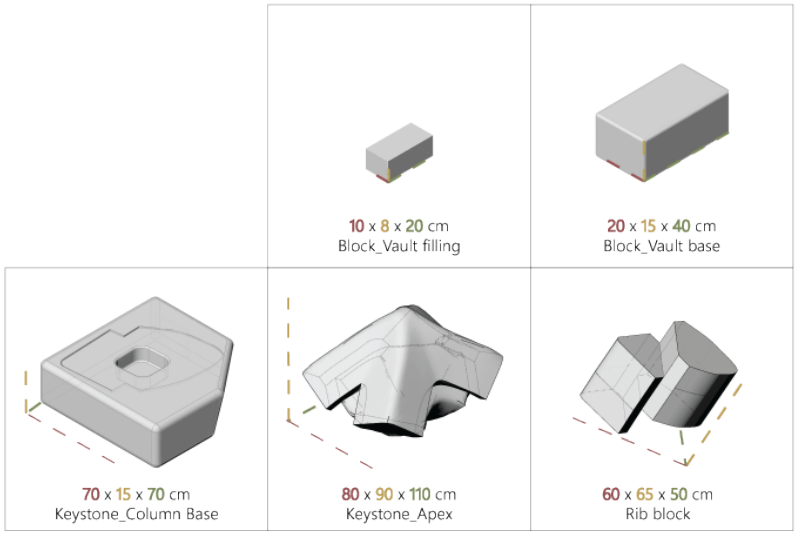


Figure 146: Special blocks for the vault assembly

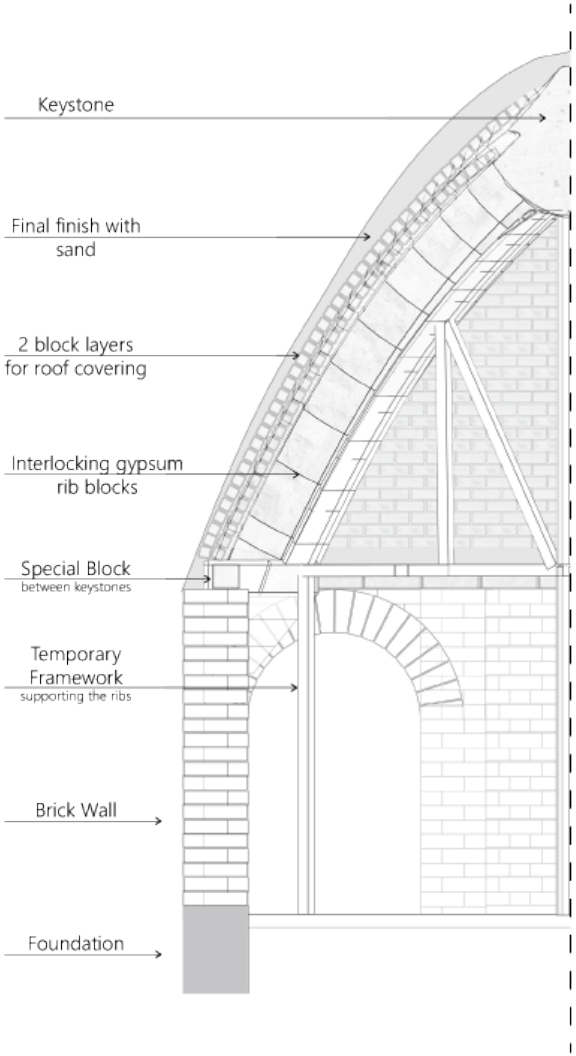


Figure 147: Section of the room under construction

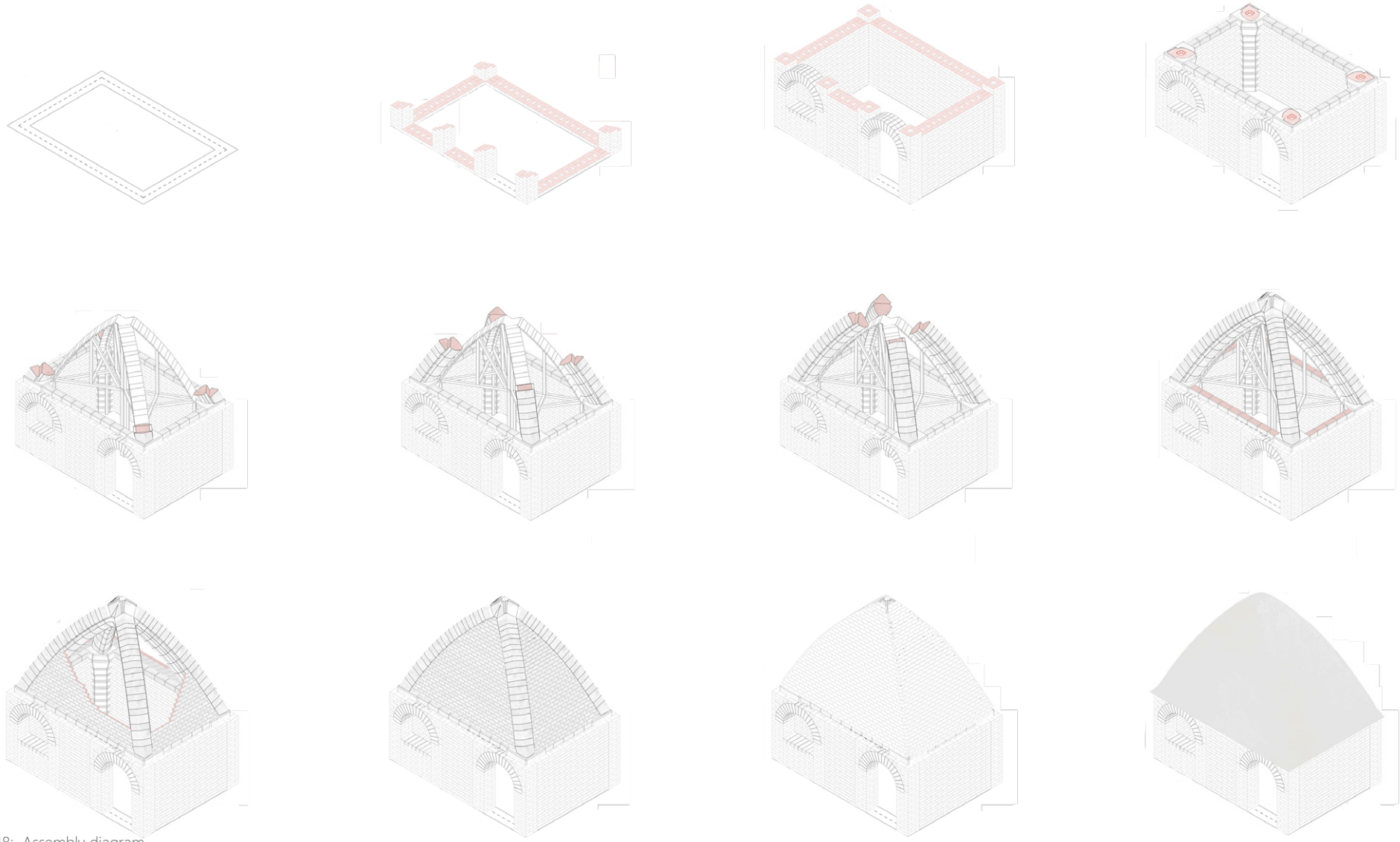
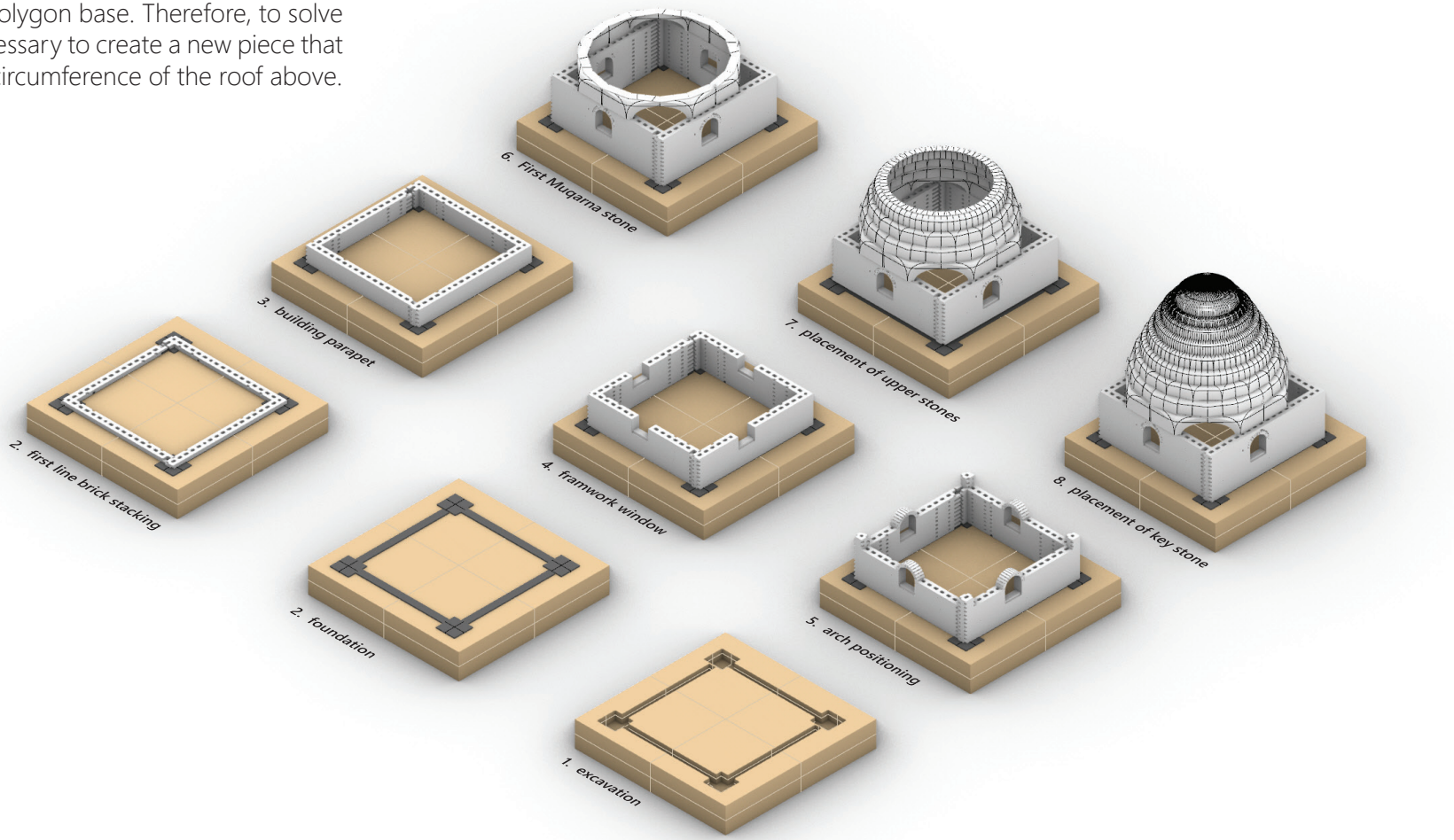


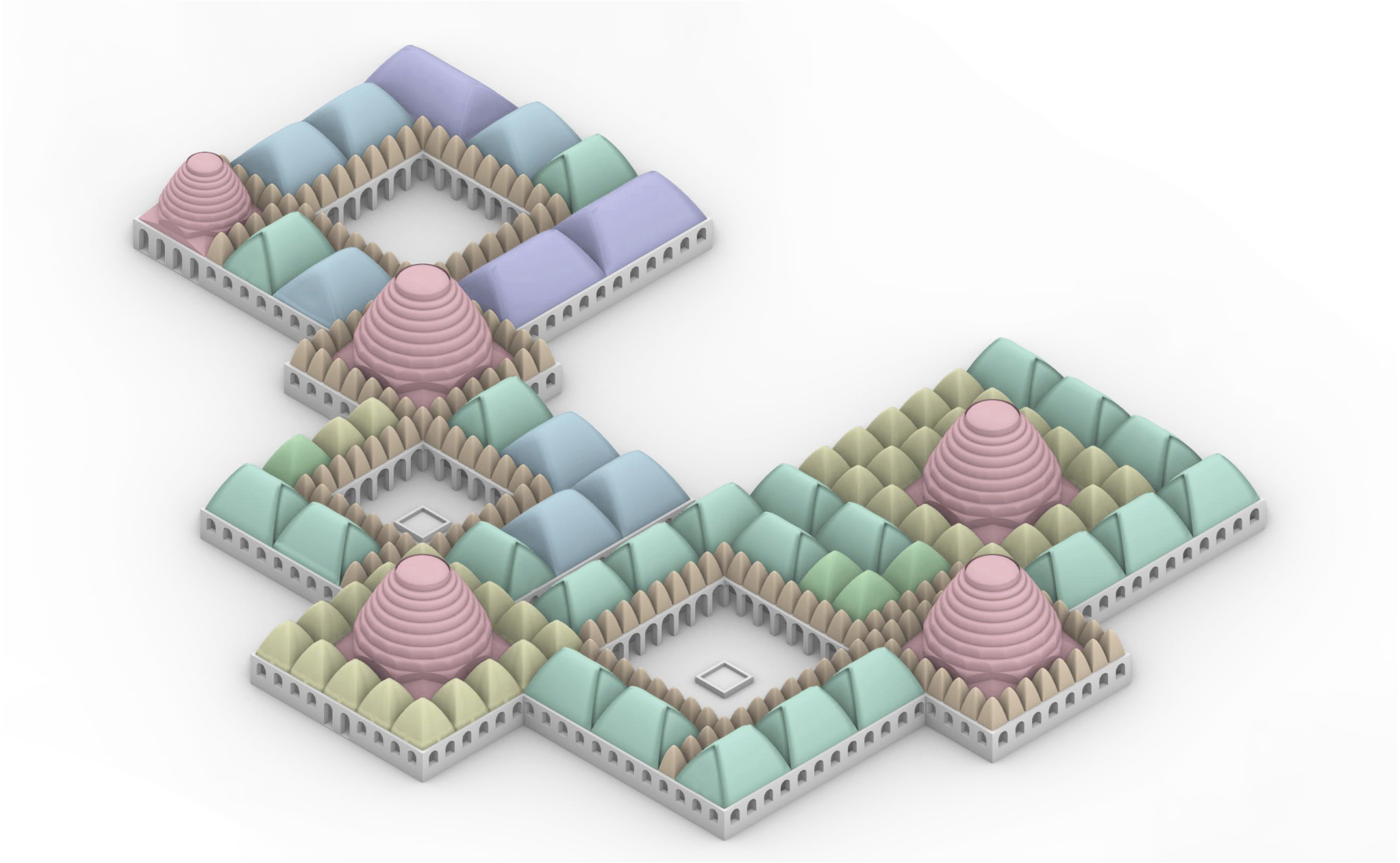
Figure 148: Assembly diagram

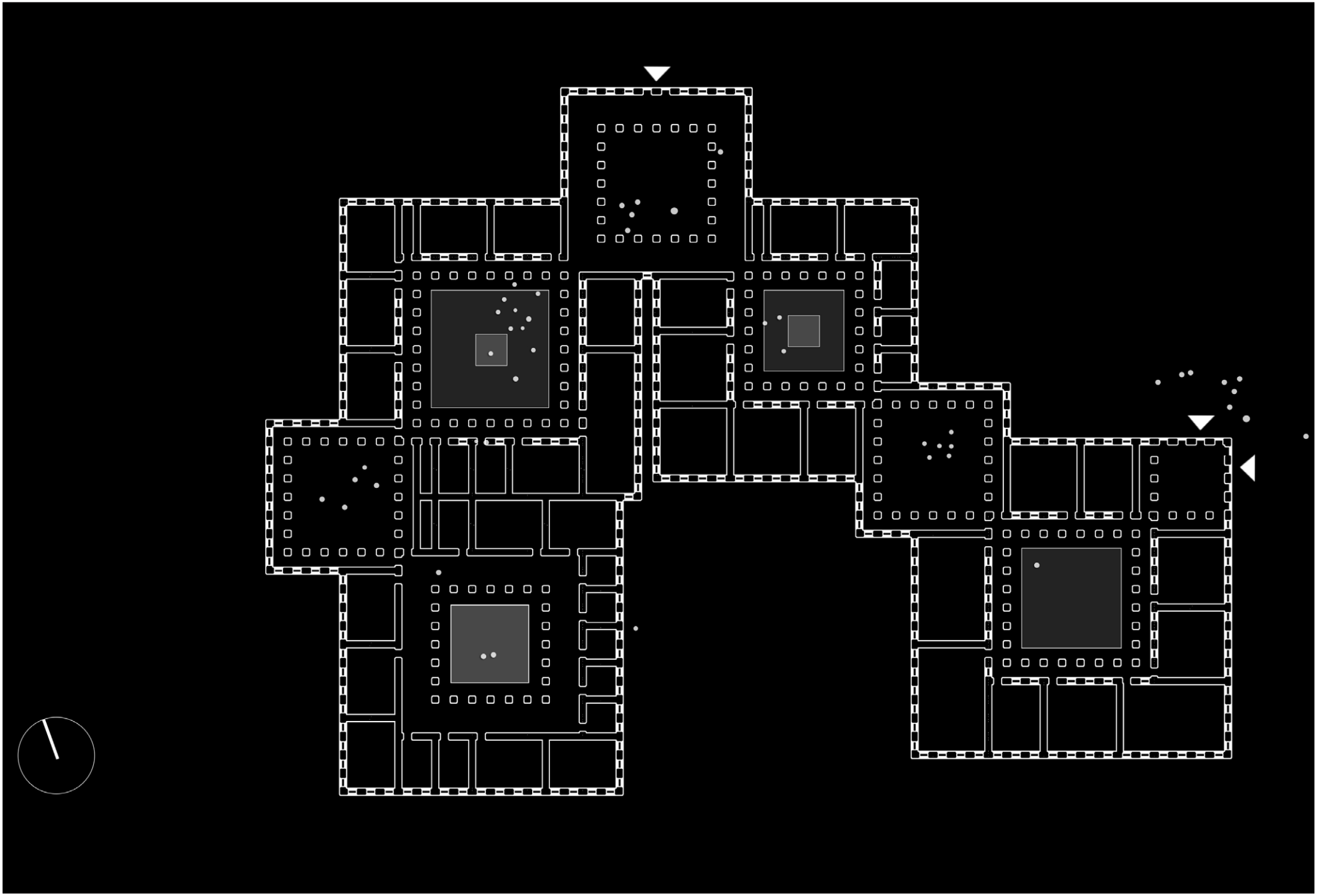
6.5 | Construction Phase Muqarnas

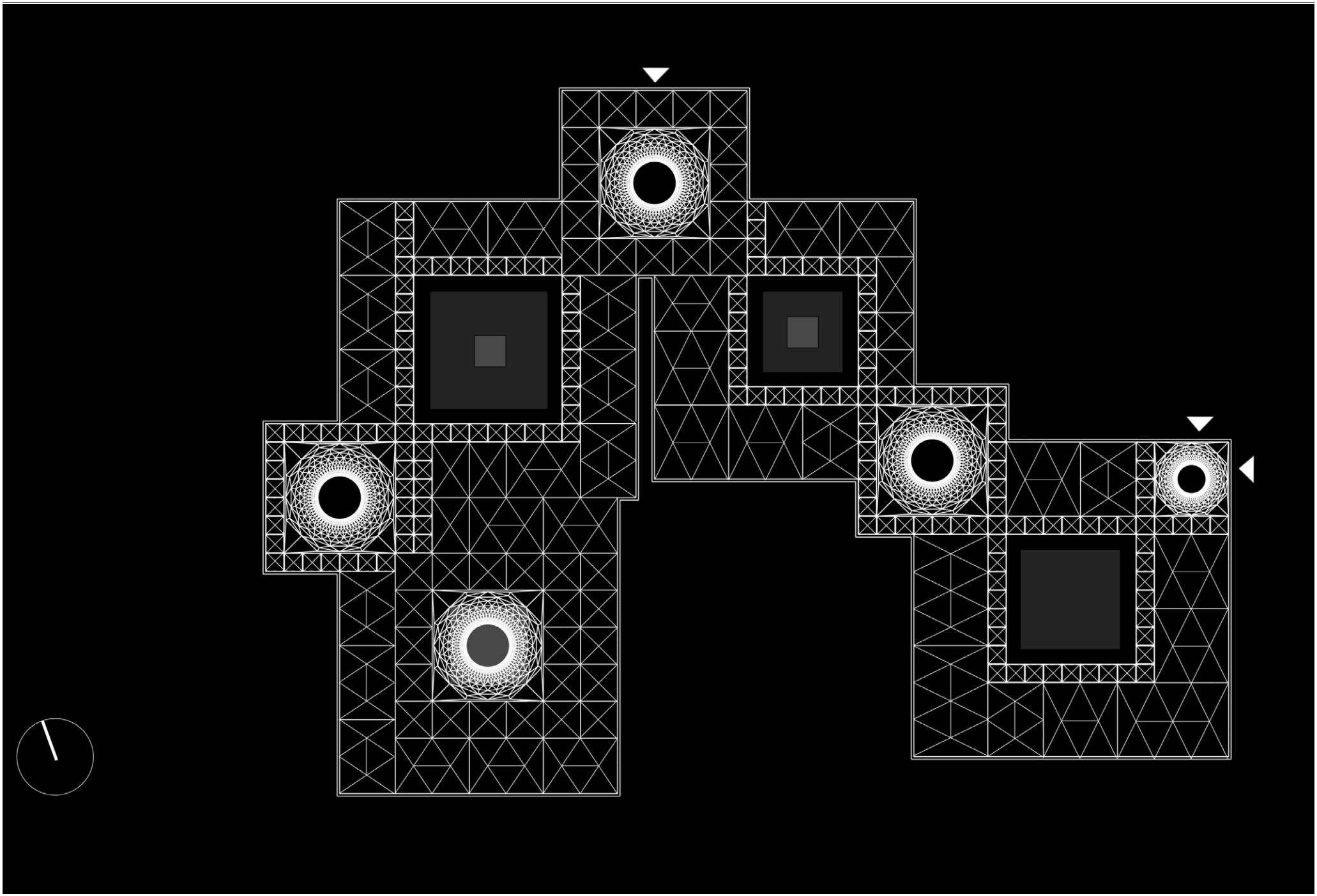
The location of the Muqarnas indicates in the unique room size project distribution. The parameters of the self-standing structure reflect the description of the block section. In this part, the diagrams describe the connections of the excavation, foundation, wall construction, and roof placement. Contrary to the initial description of the blocks. The traditional composition differs from the initial plan, as the wall base is orthogonal, and the Muqarnas follows an octagonal polygon base. Therefore, to solve this construction detail, it was necessary to create a new piece that could join the wall corner to the circumference of the roof above.

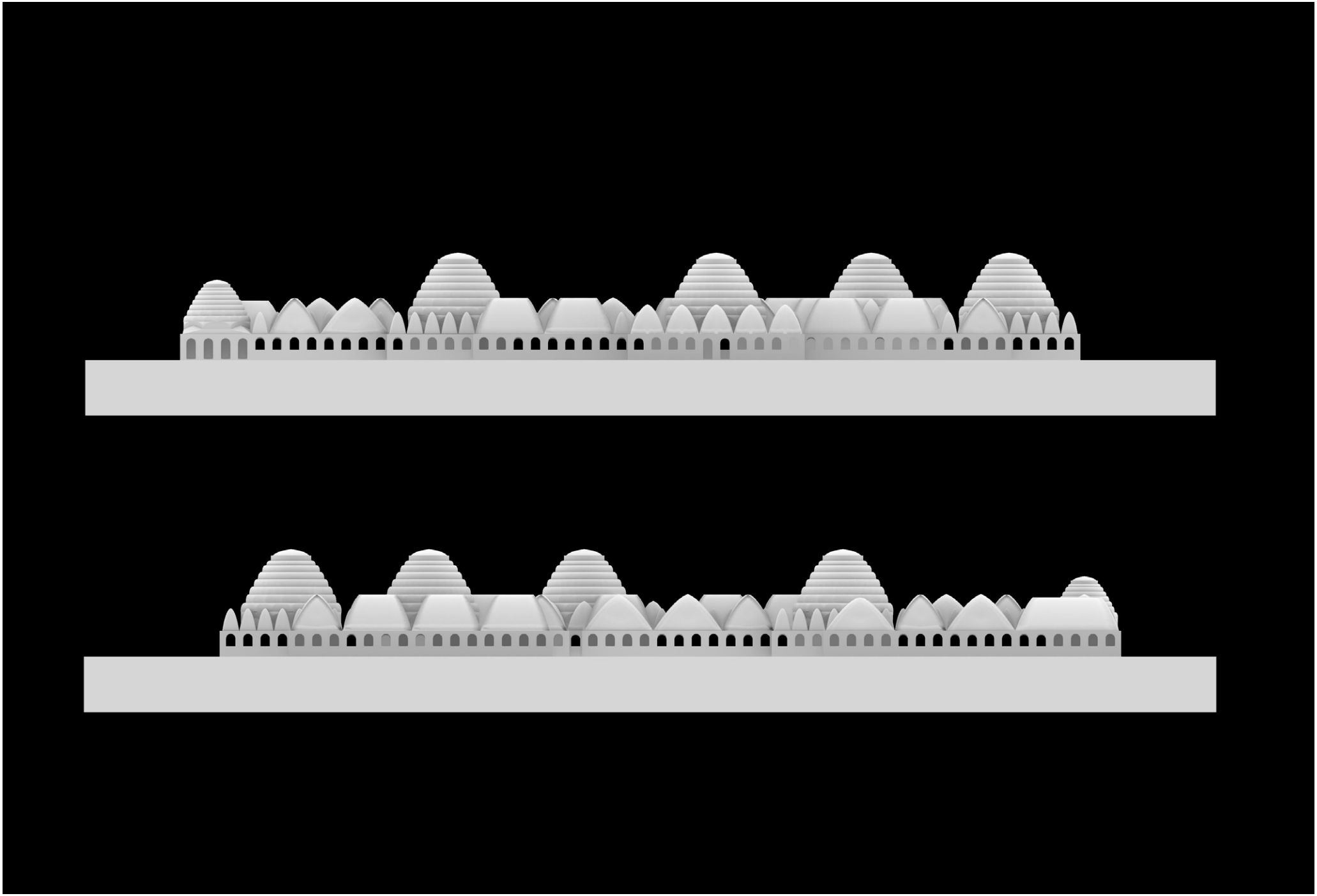


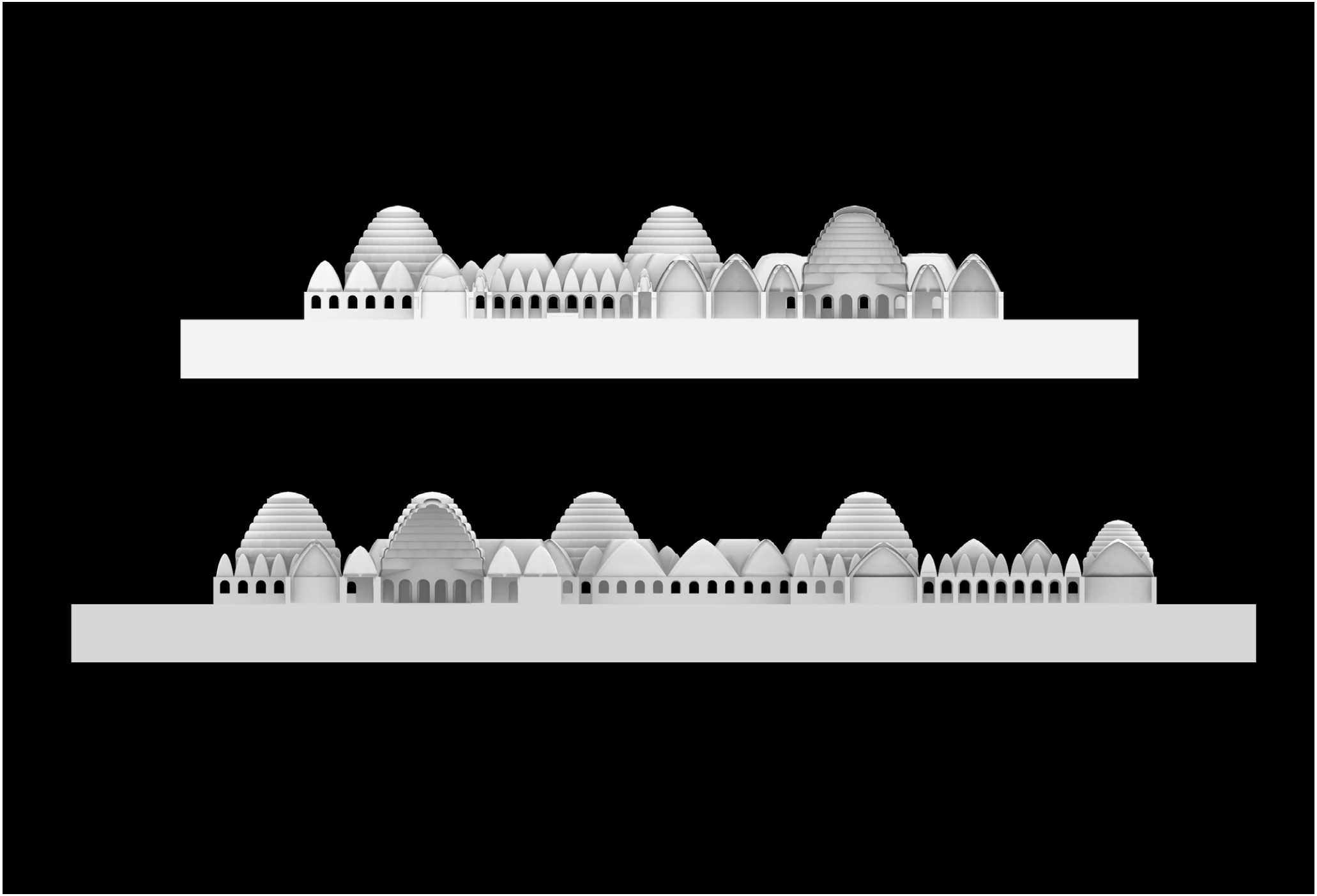
7.1 | Final Design Impressions

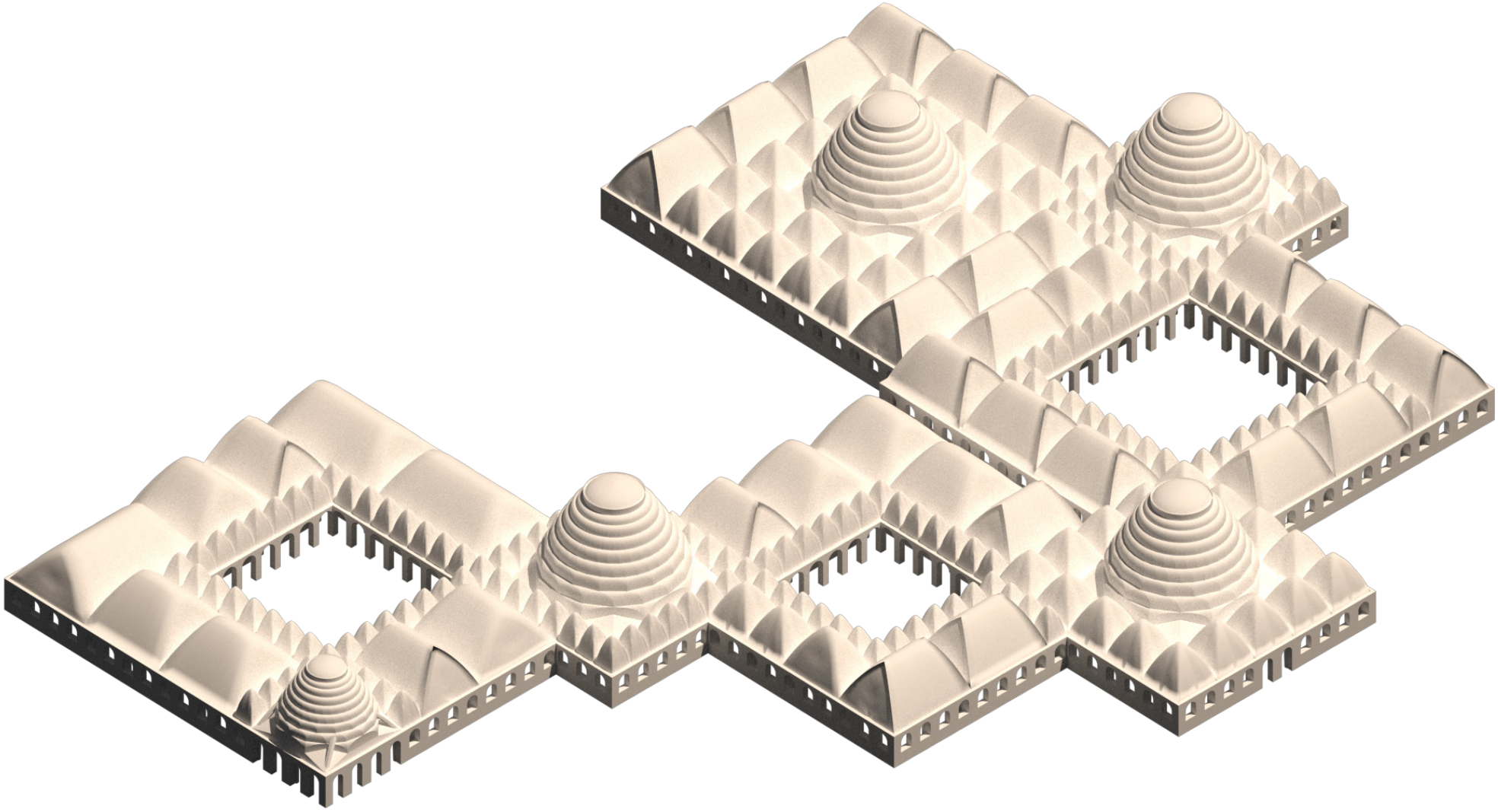


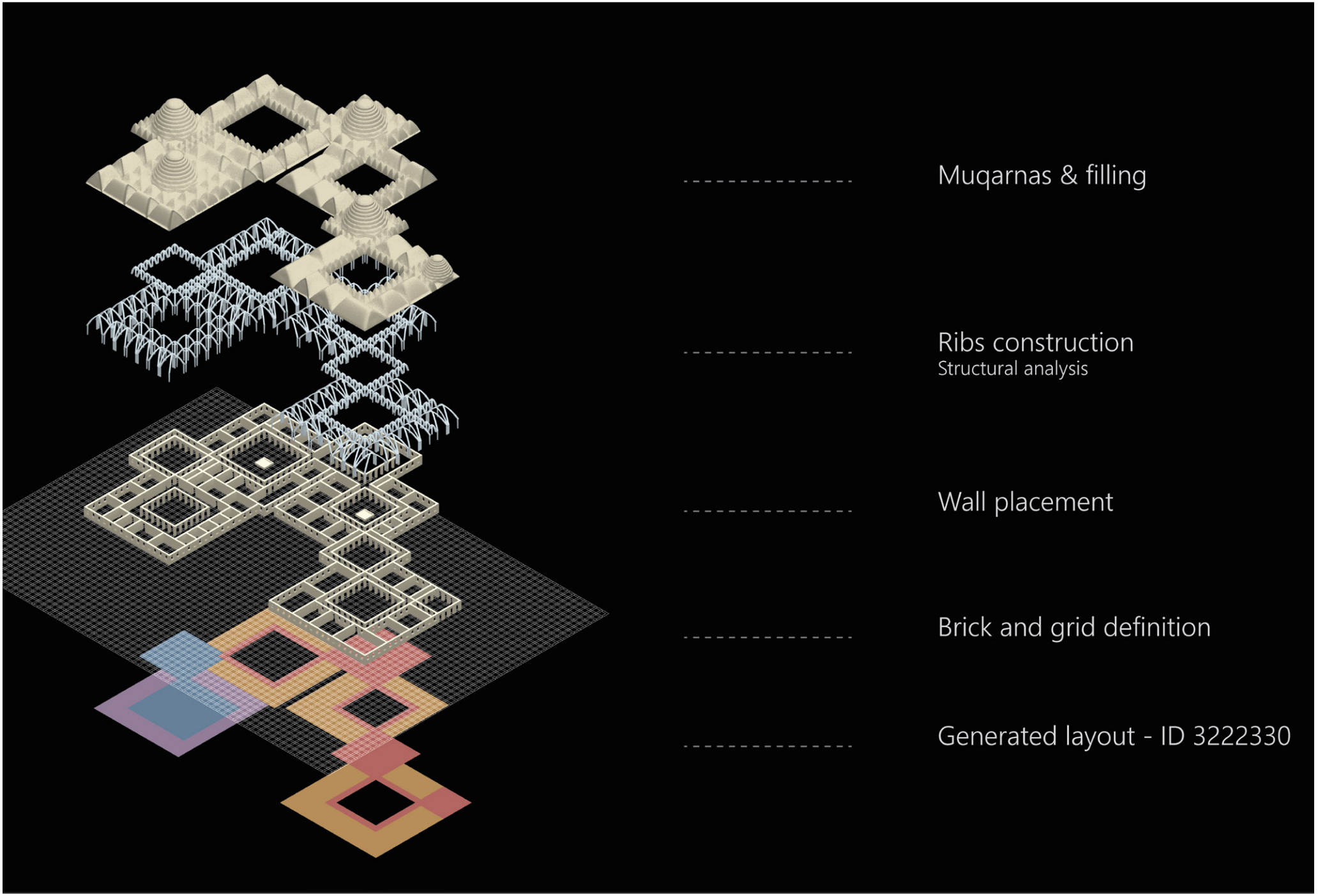











Muqarnas & filling
Ribs construction
Structural analysis
Wall placement
Brick and grid definition
Generated layout - ID 3222330

8 | Conclusions

LIMITATIONS

On a broad sense, the main limitation of Earthy is the short time allocated for the course. It could easily be extended over a whole semester and there would still be content to be taught and work to be done even without changing the scope of work from the studio. Going into the specifics, the lack of prior knowledge in coding, meanwhile one of the main attractions of the course, also becomes an obstacle once the time is short and scripts must be quickly developed to keep the project moving. Despite the strong intention in generalizing the use of Python for all steps of the process, we fell the constrains in time and experience to apply it in parts of the urban analysis, shaping and structuring phases. At several moments we caught ourselves in the learning curve spending great amounts of time to solve simple problems and create existing scripts from the scratch. All extremely valid but nonetheless a limitation in diving deeper in generative design.

EXPECTATIONS

The possibility of developing a whole project based on math and generative design through Python and Grasshopper is the selling point of Earthy and, therefore, the source of the biggest expectations from us all. Learning new skills related to computational design, developing a new design methodology based on mathematics, understanding generative design and learning how to apply it were some of the topics we were enthusiastic to investigate and get a chance to work with.

REFLECTION

The course Earthy was a very intense and all-around introduction to generative design. Thinking about Architecture in the language of mathematics was an eye-opening experience. The beginning of Earthy was tough for our group since we only had a basic understanding of python and computational methodologies. However, after a few weeks, we got caught up in the idea of creating a fully generative design. In the end, we achieved to automate wide parts of the design process, but the more parameters and options we considered, the more complicated was the linking between individual parts. Thinking of a computational workflow as a set of methods with a clearly defined input and output was a huge help to improve the interaction.