Terra Tetris

DATE	:	03-11-2020	
REPORT	:	EARTHY 2020	
AUTHORS	:	GROUP 3	
		ADITYA SOMAN	4999487
		CHRISTINA KOUKELLI	5115736
		DION VAN VLERKEN	4976541
		NEHA GUPTA	5081874
		VICENTE BLANES	5102219
TUTORS	:	PROF. DR. IR. SEVIL SARIYILDIZ	
		DR. IR. PIROUZ NOURIAN	
		DR. IR. FRED VEER	
		IR. SHERVIN AZADI	
		IR. HANS HOOGENBOOM	
		IR. DIRK RINZE VISSER	
		IR. FRANK SCHNATER	

FACULTY OF ARCHITECTURE AND THE BUILT ENVIRONMENT
TU DELFT 2020-2021

CONTENTS

1 I INTRODUCTION ...4

- SITE
- DESIGN PROBLEMS
- DESIGN PROPOSAL
- FLOWCHART
- ABOUT SCALES

2 I CONFIGURING ...10

- SCALE 03
- SCALE 02

3 I SHAPING ...30

- SHAPING GOALS
- SHAPING PROCESS
- FIRST TESSELLATION APPROACH
- FINAL TESSELLATION
- ROOF FORMING
- SIMPLIFICATION

4 I STRUCTURING ...39

- METHODOLOGY
- MATERIAL PROPERTIES
- STRUCTURAL ANALYSIS PROCESS
- EVALUATION AND DISCUSSION
- CONCLUSION AND LIMITATIONS
- KIT OF PARTS
- CONSTRUCTION
- ADOBE 2.0

5 I CONCLUSION ...70

- LIMITATIONS
- EXPANSION PACK
- REFLECTIONS

References ...73

Appendix ...74

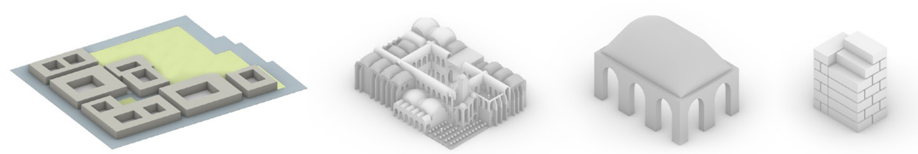
01 INTRODUCTION

In this report, the design process for the TerraTetris project will be explained. TerraTetris was developed as a participatory design game to enable the dwelling organisation and configuration in the Zaatari camp in Jordan for the Syrian refugees. It was realised by looking into the existing housing situation, identifying the problems and focusing on the opportunities that would give way to propose a more systematic and modular approach as part of a universal system. This would involve at a great level the users, their housing preferences and living conditions. The proposal is carried out in various interrelated scales in a feedback-loop logic, connecting the design phases from urban configuration to the construction details under one manifold system logic.

As for the general framework, the course AR3B011 EARTHY at TU Delft, Building Technology Master programme has been developing for the last couple of years a set of projects under the same theme. The aim is to continually improve the living conditions in the refugee camp, by experimenting with building earth structures under compression using computational methods and tools. In this way, a knowledge database is created, where diverse topics are being addressed each time, adding innovative solutions and computational approaches to the previous ones that add value to the configuration logic, form-finding, structural and construction processes, by using only earth and local-based materials.

The report will follow the same logic, starting by elaborating on the general idea behind the TerraTetris game logic and its objective and diving into each aspect of design and the scales in respect. The configuration steps will be explained, followed by the shaping process and lastly the structural and construction part with the detailed elements will be addressed, highlighting their relation and connection every time.

In all, TerraTetris became a participatory design game, which defines a set of rules and guidelines to the refugees to build their own house, by explaining the steps and options available and providing them with a kit of parts and a construction manual. This offers a combined computational and manual approach, providing an open-ended and flexible system, which can lead to various possible housing solutions and configurations. In the end, it has a universal logic which can be applicable in different scenarios and situations, that exceed the ones covered in this theme, extending its possibilities.



SITE

ZAATARI REFUGEE CAMP

Al Zaatari Refugee Camp opened on July 28, 2012 in the Mafraq Governorate of Jordan as part of the UN-sponsored relief effort to house those displaced by the Syrian civil war. It is close to Jordan's northern border with Syria, has become emblematic of the displacement of Syrians. Since then, from a small collection of tents, the camp has evolved into an urban settlement of some 76,000 residents, reflects both the needs and aspirations of the camp's residents and a transition to a more predictable, cost-effective, and participatory platform for the delivery of assistance (UNHCR, 2020).

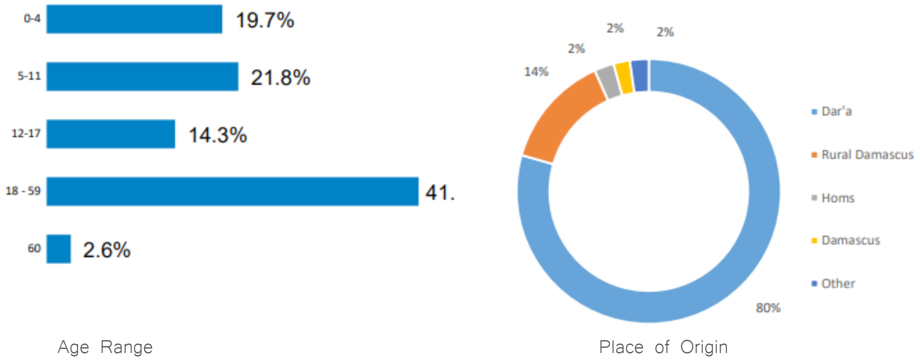


Figure1: Demographics for Zaatari Camp
https://reliefweb.int/report/jordan/zaatari-refugee-camp-factsheet-january-2020

SHELTERS

UNHCR is responsible for coordinating shelter assistance and camp infrastructure improvements on behalf of all humanitarian partners. Working to ensure equitably and gender-appropriate access to adequate shelter and basic facilities, together with the provision of sustainable energy supply. There are over 26,000 pre-fabricated shelters, and each includes a latrine and kitchen to ensure the privacy of the refugees. A household addressing system is in place and is updated regularly. To accommodate Persons with Disabilities need some shelters have been adapted to their needs and conditions. In 2019, 1,000 vulnerable families in Zaatari camp received shelter maintenance assistance through shelter maintenance project (UNHCR, 2020).

GROWTH OF CAMP

Jordanian authorities supplied land for Zaatari's development and provide security within the camp. The camp includes 530 hectares (220 square meters) of land encircled by an 8.3-kilometer ring road and measuring 3.5 kilometers from east to west. The west or "old side" of town was settled first, in July 2012; the old side of town includes the downtown and slums of Zaatari. Tensions erupted during the planning of the city because the Jordanian government had (and continues to have) a vested interest in the Syrian refugees leaving at the end of the conflict. Stakeholders must continuously balance the needs of the refugees and Jordanian concerns that Zaatari will become a permanent city (Ledwith & Smith, 2014).

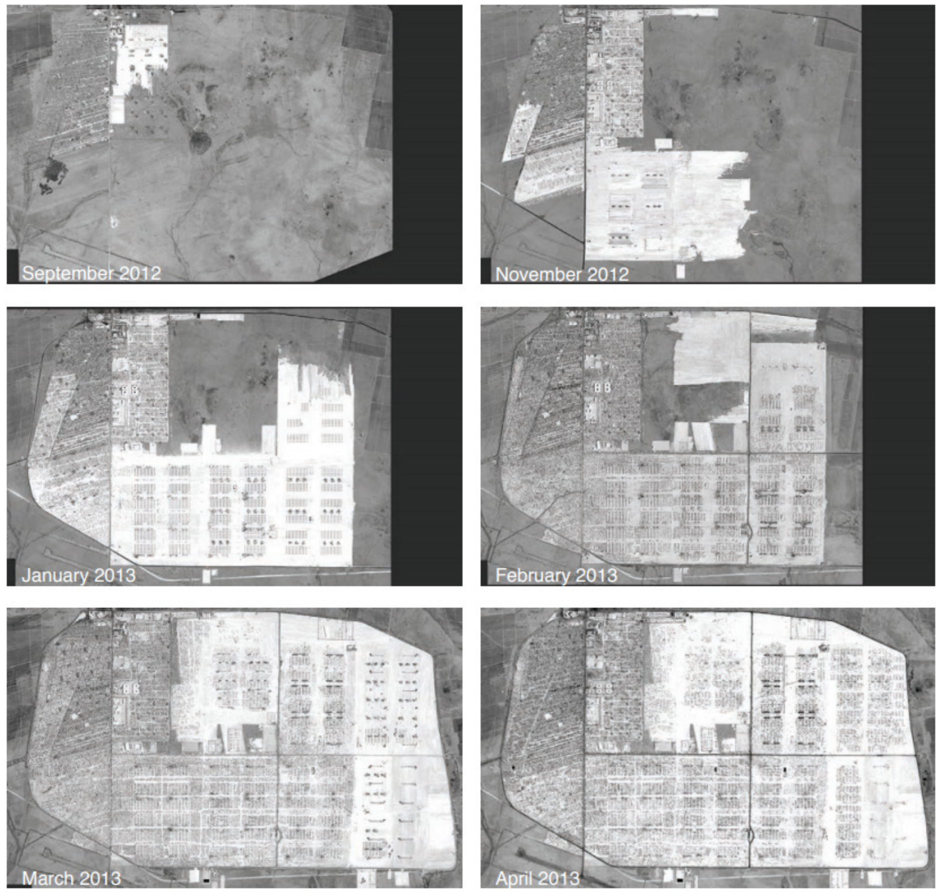


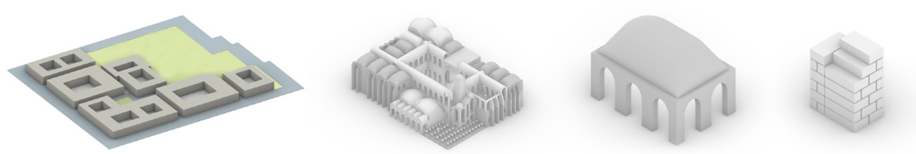
Figure2: Satellite image showing growth of Zaatari camp
Zaatari - The instant city, 2014

Figure3: Shelter types in the camp
Zaatari - The instant city, 2014

LAYOUT

The formal layout of the camp is a grid system with caravans placed in rows; the spacing of the caravans is designed to accommodate vehicles, guard against fire, and promote hygiene. Since the caravans were not donated at the start of the camp, the original planning fabric from the tent infrastructure is visible in the old camp. As caravans replace tents in the old town, the close spacing requires surveyors to place more caravans than advisable in a given area.

The informal layout of the camp arose after the residents received the caravans. The residents, rather than maintaining the row shapes, re-position their caravans in "little compounds" — typically with a U-shape or a courtyard shape — so that they may live together with their extended families. Other rearrangements of the camp allow refugees to move closer to people from their villages; these unsanctioned modifications result in a redrawn, "maze-like" map (Ledwith & Smith, 2014).



DESIGN PROBLEMS

From the site analysis and observations of the current situation, certain issues were identified as problematic in the area, which became the focus of the current design proposal. Those were mainly addressing the housing configuration and area densification in an urban level. More specifically, it was generally observed that the camp is highly dependent on external resources and it lacks self-sufficiency, which limits its potentials for independent growth. Apart from that, the relocation of the communitites to their new housing is most of the times interrupted and is driven by an unorganized grouping of the caravan structures. This, in combination to a poorly designed street network, lead to non-usable interstitial spaces and a densification of the area due to many ground floor structures. Lastly, all these factors are associated with an insufficient reaction of the housing clusters to the adjacent facilities, providing only generic solutions, which are not apt to grow or adjust to change. All the above consist the main design problems, that are relevant in this case and relate to the design objective, which will be further elaborated.

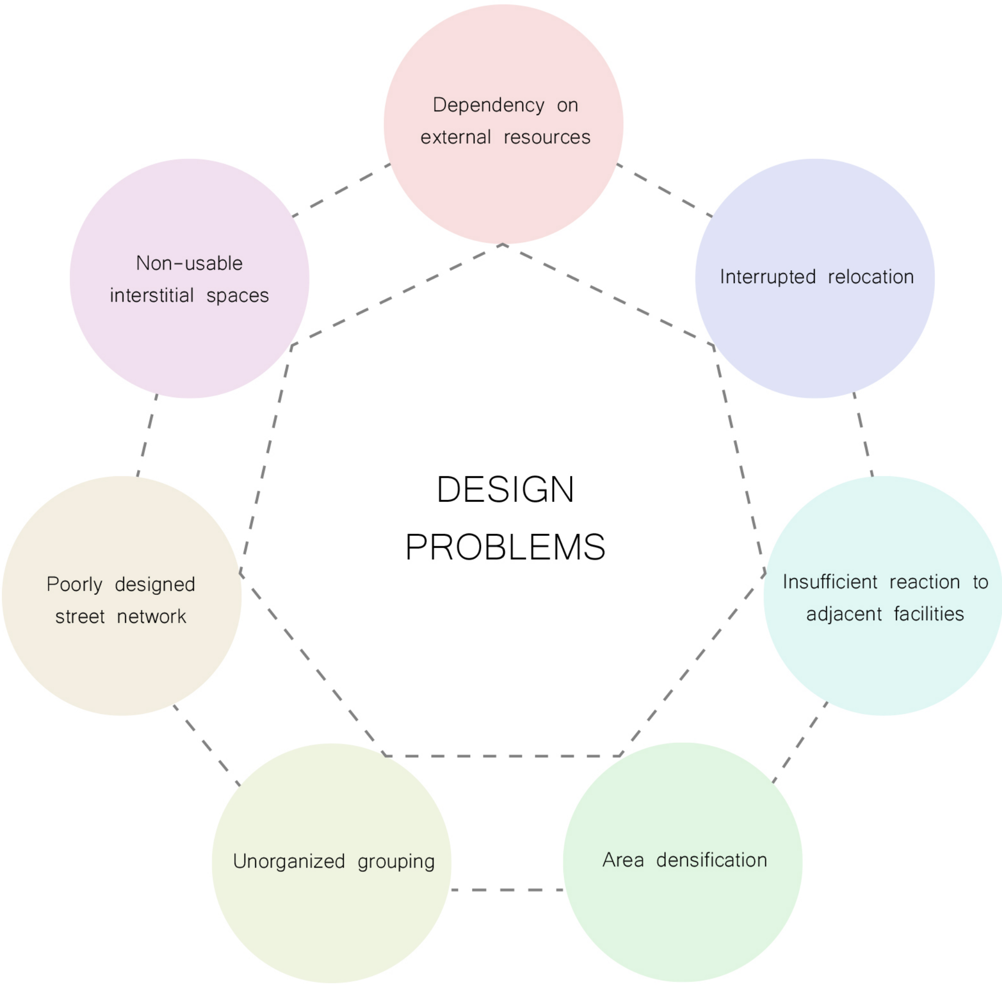
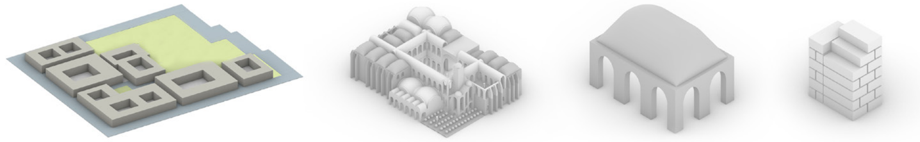


Figure4: Design problems overview diagram

Figure5: Site photos of the current situation
Sources: https://theinterrobang.com/photos-show-the-enormity-of-syrian-refugee-crisis/
https://deeply.thenewhumanitarian.org/refugees/community/2018/10/11/we-need-to-help-jordans-other-refugees



DESIGN PROPOSAL

INTRODUCTION TO TERRATETRIS

In order to tackle the design problems which we found in the Zatari camp we came up with the solution in terms of a participatory game called as terratetris which would create a platform for collaboration between the camp dwellers and the designers

ABOUT THE GAME

Terratetris is a participatory game that we designed for the camp dwellers of Zatari for engaging in the development of the camp. The intention of playing the game is to involve the camp dwellers to participate in the decision-making process involved in planning of their Earthy houses which is a more permanent alternative to their current container/tent dwelling.

GAME INSTRUCTIONS

The participants playing the game are the chosen representatives of the individual family who will try to put forward the interests of the family involved in the game. Depending on the type of dwelling the number of participants may vary. For example, in a family dwelling where there are 3 families of 5 members each the number of participants playing will be three. similarly, in a communal dwelling there may be many more participants since it comprised of a lot of smaller families and individuals.

The participant must fill in a list of his /her preference in terms of importance of the various spaces in the house and the priority for the direction of the spaces if any. If the participant has no preference regarding space planning, then they can rely on the Architects decisions which are the default value sin the game.

The Architects decisions are based on the REL-Charts and the Bubble diagrams made for the various dwellings based on the level of privacy and the ideal connectivity between the spaces.

Terra
Tetris

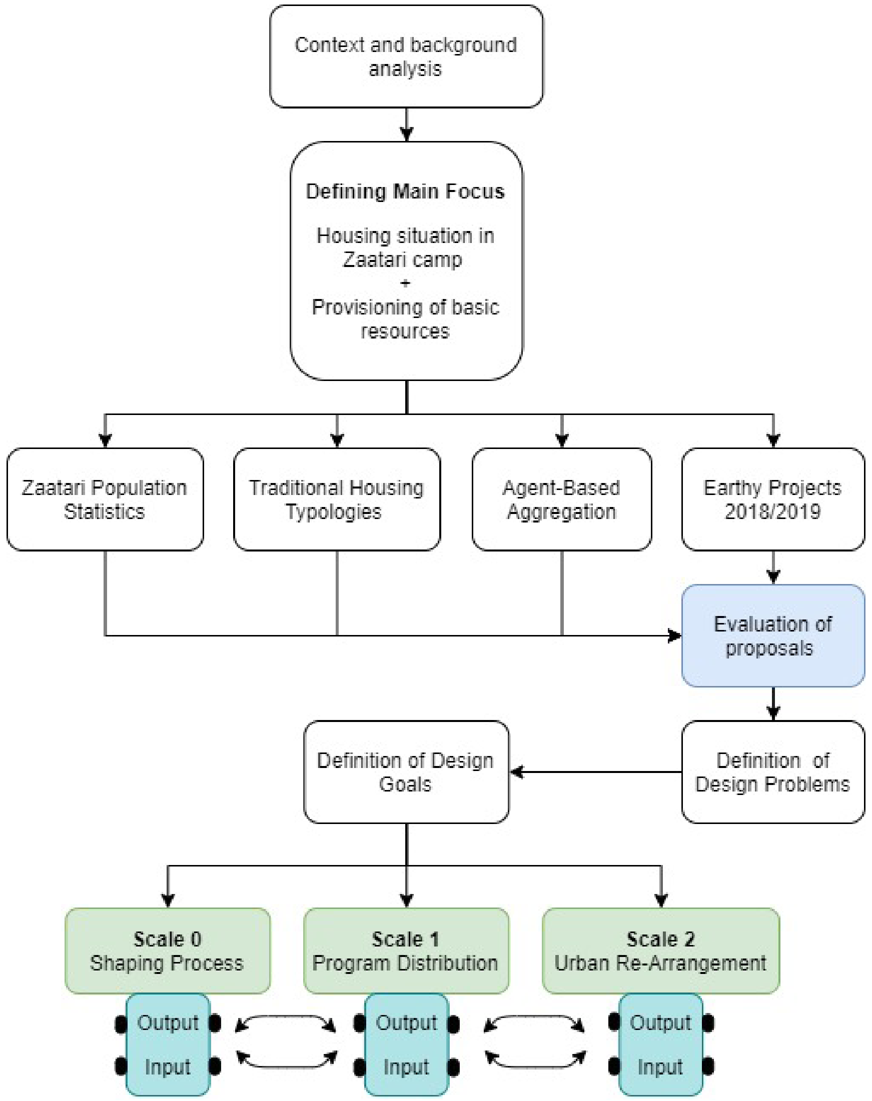


Figure6: Flowchart showing the workflow of the project

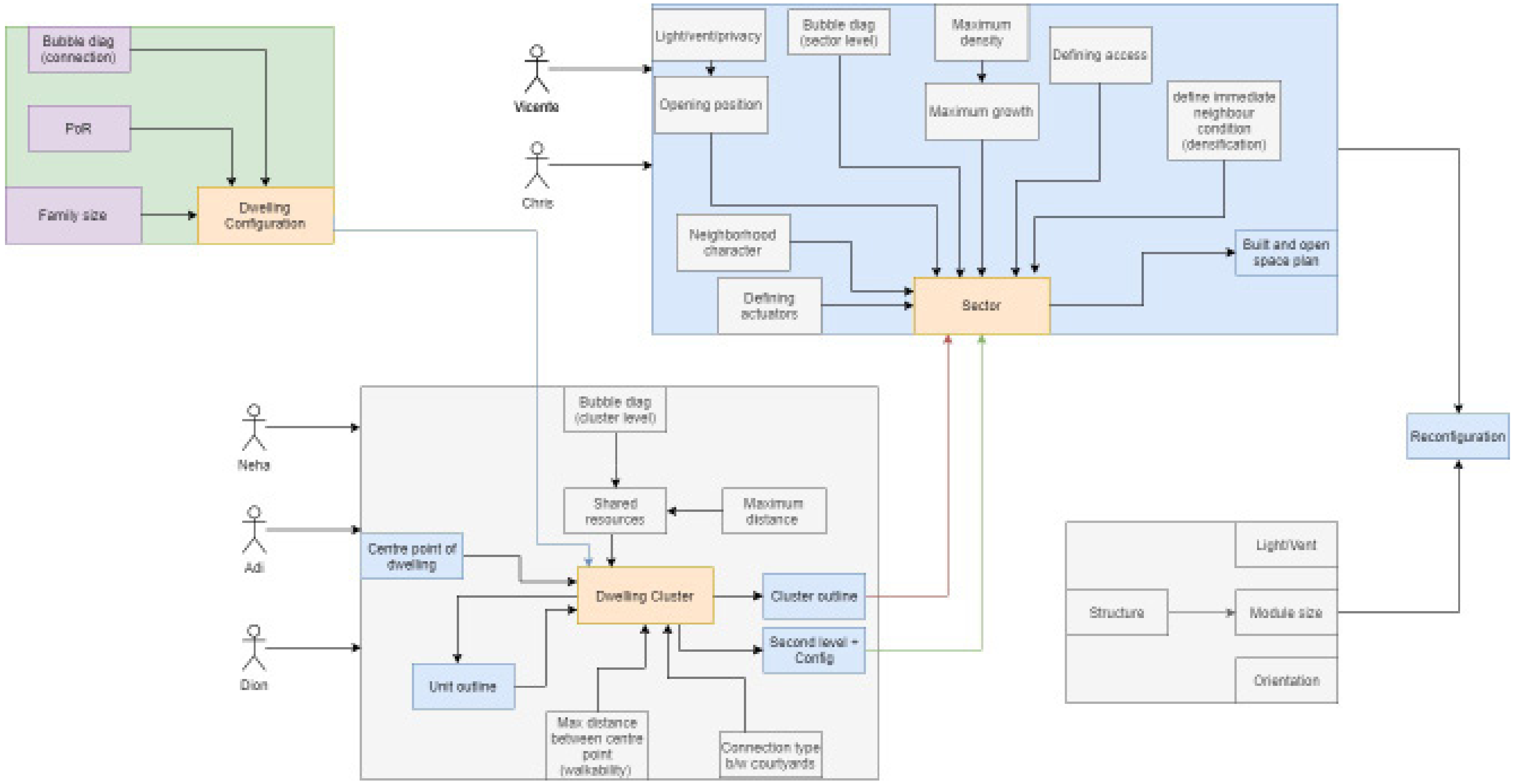


Figure7: Flowchart showing the internal team working plan

ABOUT SCALES

In this process, a systematic approach is being followed, in which four scales are being identified, based on the focus each time and moving from large to small scale. These are interrelated to each other and work in a feedback-loop logic.

Starting off with scale 03 which is the urban aggregation, this provides information to scale 02, the dwelling configuration. This information contains the dwelling outline and location. From its part, scale 02 feeds scale 03 with the required bounding box and courtyard size. At the next stage, scale 02 provides scale 01, which is the unit development, with the information about the connection of the spaces and the option for growing on a second level. In return, it receives as an input the amount of sizes required for connection. Lastly, at the smallest scale 00 is the construction, which takes as input the output of scale 01 in terms of interconnection between the spaces, while at the same time informs scale 01 about the grid size that is used and certain structural contraints. This whole process is continuous and gives a clear overview of the information interchange between all stages and levels of the design iteration process. Significant characteristic of the process is the user's input and the participatory design, as it will be explained later on.

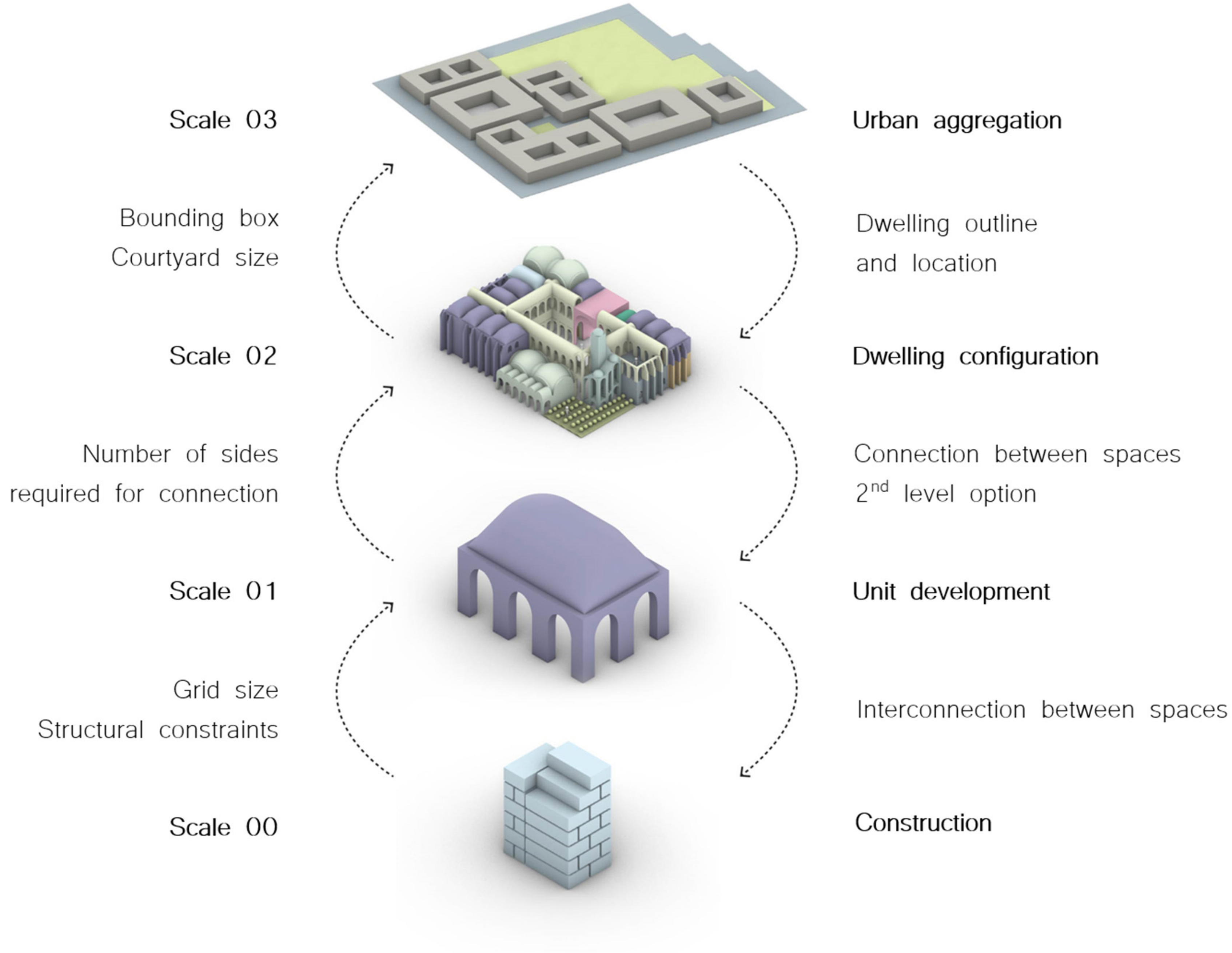
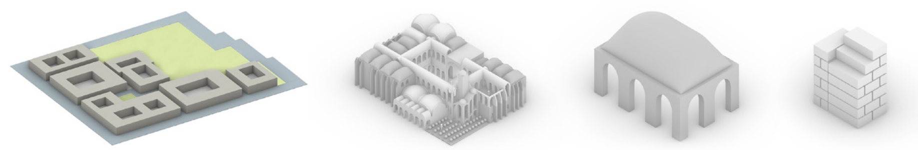
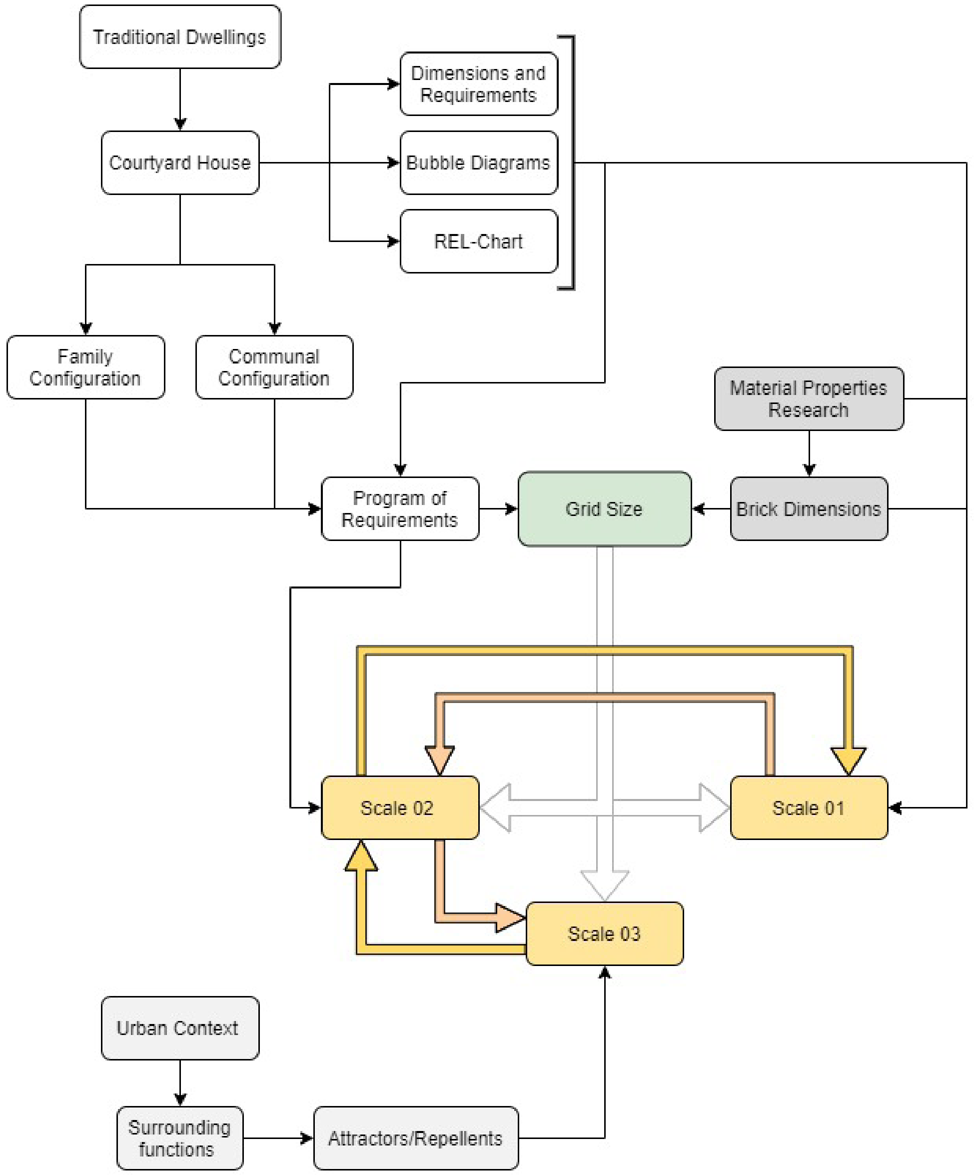


Figure8: Scale overview and feedback-loop logic



02 CONFIGURATION



CONFIGURATION GOALS

1. CONFIGURATION APPROACH:

The process of Architecture designs many times is compared with Art. It can be argued that logic and reasoning is not a necessity for generation of art. But when it comes to Architectural design of a project which is meant for human habitation it should rely on logic and reasoning since the result has consequences on the comfort and wellbeing of the occupants. Adapting to such an approach will also lead to a process for which each step can be traced back, analysed and modified for a better result .This was the main idea that led us to create the approach for configuration of Terratetris.

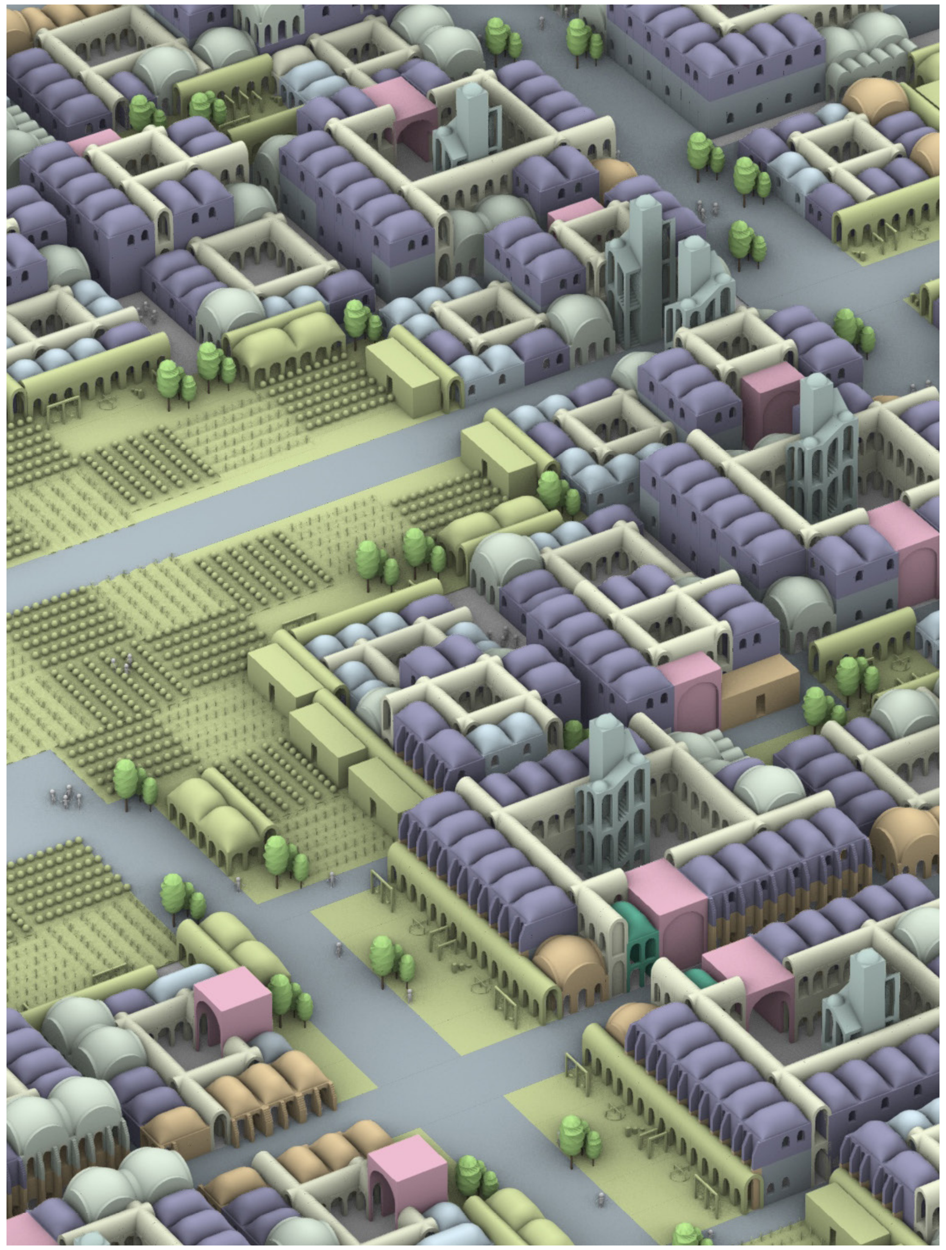
2. MAIN GOALS:

To breakdown the human process of Architectural design as accurately as possible into procedural functions.

To consider the opinions of the camp dwellers as much as possible in the configuration of the camp as a whole and also at an individual house dweller.

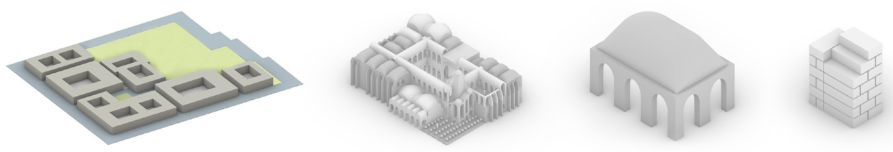
To think of the housing problem at a system level rather than at a acse level for a wholistic and integrated output

To generate a system that is flexible for accepting conflicting decisions and can potentially generate a lot of options for the participants to evaluate



Human Logic

Procedural Rules



SCALE 03

Four main objectives were proposed at an urban scale level as a combined strategy to address some of the main design problems identified during the research process.

1.1) Maintain the current grouping of families:

As mentioned before, the idea of distributing the different families in “Family Dwellings” and “Communal dwellings“ is proposed as a way to reinforce the sense of community between the people on the neighborhoods and as a strategy to reduce the amount of infrastructure and space required per family by sharing basic functions without losing privacy. The functions that will be shared on a communal dwelling are mainly service areas like kitchen, toilets, showers and storage.

For this purpose, the initial step would be to gather all the information about the current grouping of families through a survey per block. Based on the program of requirements, it was determined that the minimum size of a family for a “Family dwelling” arrangement should be of 4 people. This means that families of 3 or less would be gathered into “Communal Dwelling” arrangements. Moreover, the reusable caravans will also be considered in the survey for the future intervention.

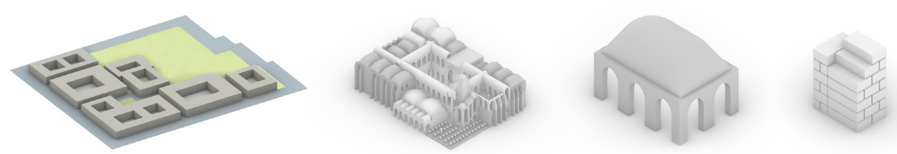
1.2) Interaction with surrounding context

In the current situation it’s possible to see how the houses are reactive the commercial function, concentrating along the street as an opportunity for the people to establish a source of income on the camp. Taking the same concept, it is proposed to establish the same reaction from the housing arrangements to all functions on the camp. Mapping these functions on each block and establishing their relationship with housing units, to determine if they will act as attractors or repellents.

Commercial functions for example, will attract the dwelling configurations and will create hybrid functions, allowing the families to accommodate commercial functions on ground floor level and shifting house functions to the first floor. Additional combination of functions could also create different types of hybrid configurations that could bring more variety to the urban blocks. As an opposite example, infrastructure for water treatment, waste treatment or similar functions would act as repellents, using farming areas as a buffer zone. These implementations will increase the way housing units react to their surroundings, creating more variation between blocks and giving more identity to each neighborhood.

Figure9: Grouping of familyes - Famyly / Communal Configurations. Source: Group autorship.

Figure10: Interaction with sorrounding context - Attractors, Repellents and Hybrid Funtions. Source: Group autorship.



1.3) Introduction of communal functions:

Urban farming areas, workspace areas, and storage areas are defined as the communal functions, proposed as an additional approach to increment the sense of community between the users, and to reduce the dependency on external resources. The idea is to give a starting point for the creation of self-sustainable communities, allowing them to produce for their own consume and giving them space for further infrastructure additions like solar energy capitation, water harvesting/treatment systems and waste management that could be implemented on later stages to extend the potential of this approach.

The distribution of this spaces will be mixed with green areas that will contain additional covered areas for sitting spaces and study areas with tables for kids, as well as playgrounds areas, gardening and other additions that could be of benefit for the people. The connectivity of these functions should respond to the previous objectives.

1.4) Rearrangement Strategy:

The concept of Urban Tetris is proposed for this project, as a way to include all previous mentioned objectives in one urban rearrangement strategy. To avoid the constant relocation of the users, the dwelling arrangements (Family and Communal) will be implemented one by one, where (if possible) only the families that will be accommodated on each step will be moved on the process.

This rearrangement (as the Tetris game) will be directional, following the attractors (the existing or proposed functions) attraction/repulsion effect which ideally will be different on each urban block to create more variety. On this manner, the attractors will determine the start and the order of the rearrangement process. The dwellings that are in direct contact with the attractors, depending on the function, will create the hybrid configurations mentioned before (as the preciously explained example of commercial/housing).

The size of people on one block, based on the program of requirements, will determine the amount of square meters of green areas and communal areas that need to be implemented, converted in grid units. As a result, the total number of modules of each function (housing, communal areas and green areas) will be known and the order of implementation will be established. The process of implementation of the modules will be divided in a certain amount of repetitions to achieve an interspersed arrangement of functions.

For example: 60 Housing, 30 Green Areas and 15 Communal modules could be divided in 3 repetitions. Therefore, 20 hosing, 10 green areas and 5 communal modules will be distributed on each repetition.

Figure12: Implementation of communal functions and green areas.

Figure11: Rearrangement Strategy - Urban Tetris.

2 TETRIS REARRANGEMENT STRATEGY:

2.1) Defining the Modules

The first step of the configuration strategy is determining the size of the modules that will be introduced on the grid. In case of green spaces and communal modules, the size is predefined. For house modules, it will respond to the configuration logic of the arrangement.

This is an example of the interconnection between scales. The input data for this process comes from scale 02, where the bounding box of one house is determined by the number of people and the participatory process that will be explained later on. The input from Scale 02 is:

- Size of the courtyard in both axis (X and Y).
- Size of space in 4 directions (South, East, North and West).

Every hosing module will always start from a central cell, which will always be the center of the courtyard. If a configuration is composed by 3 families, 3 set of values will be given. Based on the conditions of the urban grid, in which this configuration will be placed, the interconnection between these modules will be defined.

As seen in figure 4, configurations for houses are always centric, meaning that the modules will always connect to a central one. The biggest module of an arrangement will always be the central one, to which the secondary ones will connect. For green areas, this configuration is linear, meaning that the following module will be connected to the previous one and so on. For communal spaces, they will be assigned individually.

2.2) Conditions of the Grid

The conditions of the grid are determined by:

- The size of the modules (Size of the grid).
- The attractors acting on the urban block.

As explained before, the attractors (the existing or proposed functions) will define the order and direction of the reconfiguration process by polarizing the grid. Each cell will be assigned with a value correspondent to its distance towards the attractor/s. The closer the cell, the higher the value and therefore, the higher the priority for the implementation of modules on that position. This will create a gradient of values reacting to one or multiple attractors at the same time, described as the grid polarization.

As mentioned before, the attractors can also act as repellents, which will be represented with negative values. The different values caused by different attractors will be added on each cell, creating a combined value that will diversify this gradient effect on each block, depending on the present functions and their position.

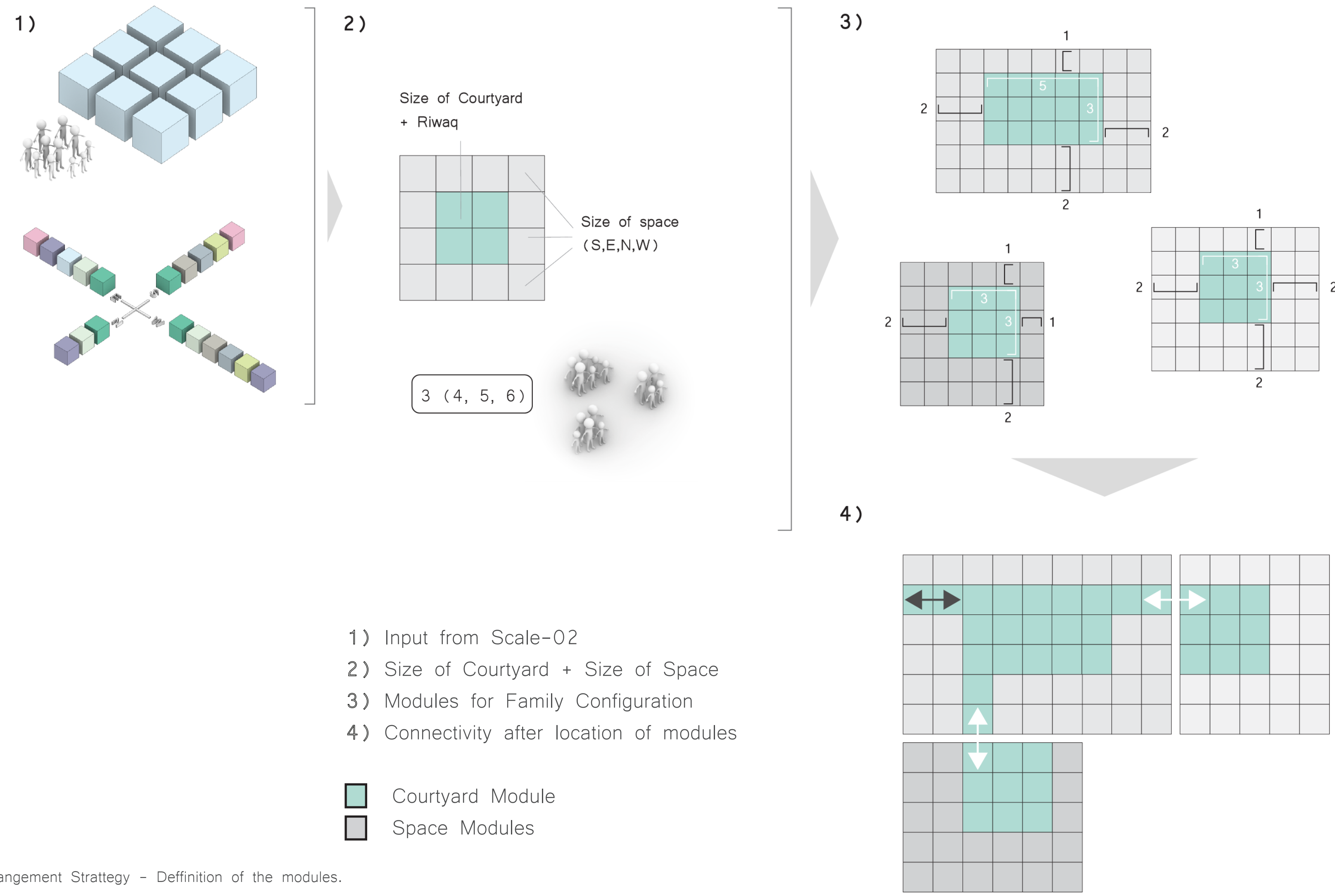


Figure13: Tetris Rearrangement Strattegy - Deffinition of the modules.

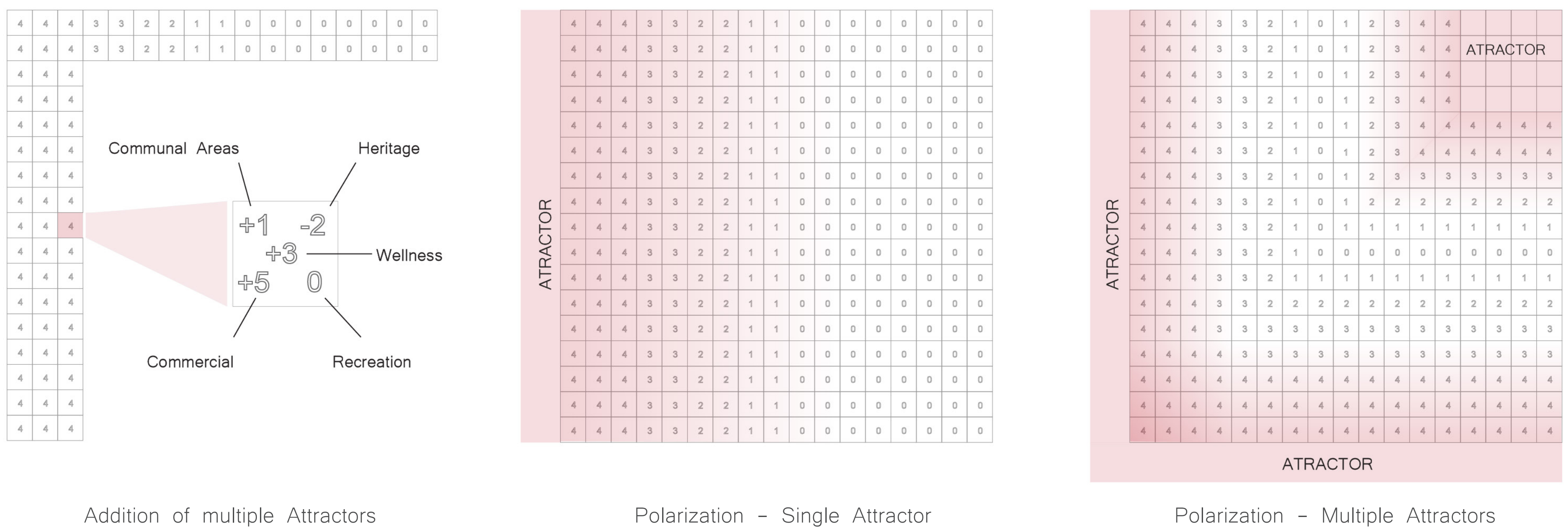
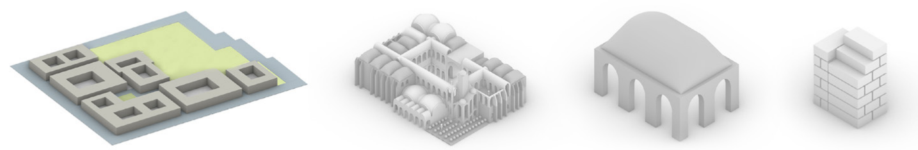


Figure14: Tetris Rearrangement Strattegy - Polarization of the grid.



2.3) Implementation of Modules

a) Selection of positions: A module will always start from one central cell, which is the center of the courtyard. The process of selecting the position of this module starts by obtaining the dimensions of the block. This information comes from scale 03, including the dimension of the courtyard in each axis and the dimension of the spaces on each direction (S, E, N, W). This will result in the bounding box of a module, which will dictate the amount of cells required for the comparison process.

Available centers: As a first step, the algorithm will detect the cells that fulfill the required amount of cells around themselves for this particular block size. These will be the available centers.

b) Selection of center: Once the list of available centers is defined, the values of the cells around those centers will be added to determine a combined value for each center. For example, if the bounding box of a house is of 7x5 modules, 35 values will be added together to form a combined value. These combined values will be compared to select the highest one, which will be the selected center.

c) Secondary modules: For family arrangements of 2 or more modules, the next step is to determine the possible positions for the connected module. As we showed before, the family dwellings composed of more than one module have a centric structure, where all modules are connected to a central one. Therefore, the same process will be repeated, but this time the available centers will have two requirements:

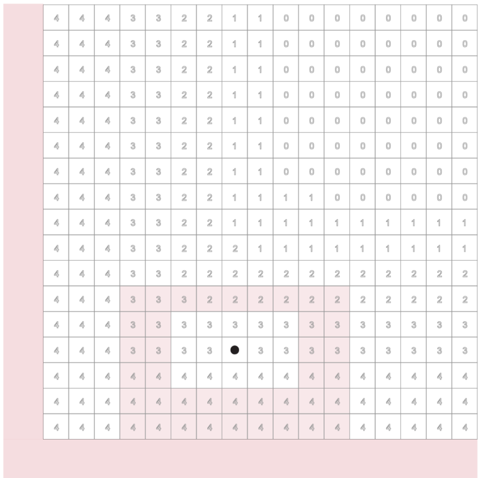
- They need to be at a specific distance to connect their courtyard to one of the faces of the main module (S, E, N, W).
- They need the required amount of cells around themselves according to the size of the module.

Again, the same process of comparison of values will happen between the available centers, selecting the highest one and repeat the process for the next one.

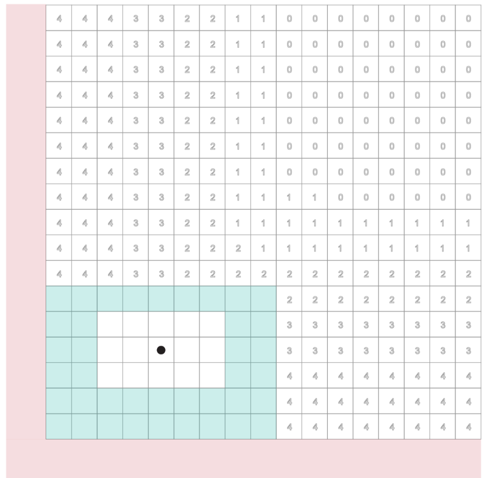
d) Updating List of cells: To avoid modules choosing repeated or overlapping positions, the modules will update the list of available sells after every selection process. This will happen in two specific situations:

- In the case of a family configuration, the list of available points for the secondary modules will be updated after selecting the positions, so avoid repetition or overlap between secondary modules.
- When the family configuration or communal configuration is finished, it will disable the selected cells and the cells around the configuration. This will provide a boundary area that will be defined as circulation, green area or additional space for the configuration in later steps.

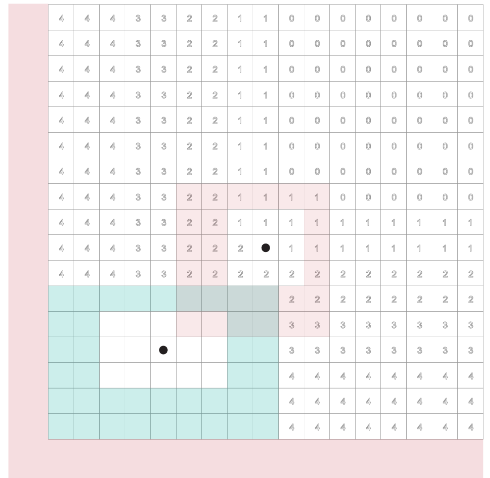
This process will be repeated for each configuration.



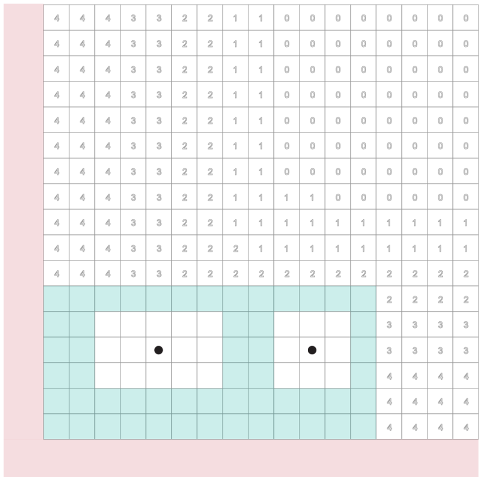
Comparison of values between available centers



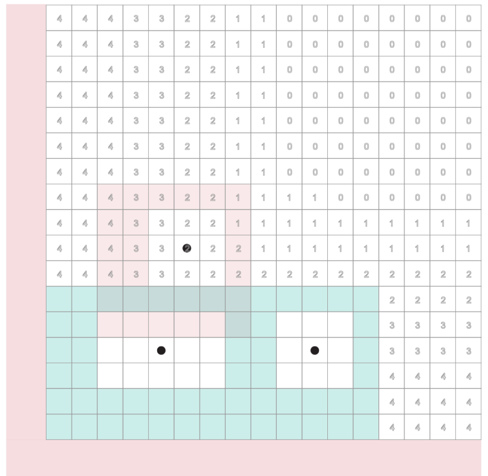
Selection of highest value



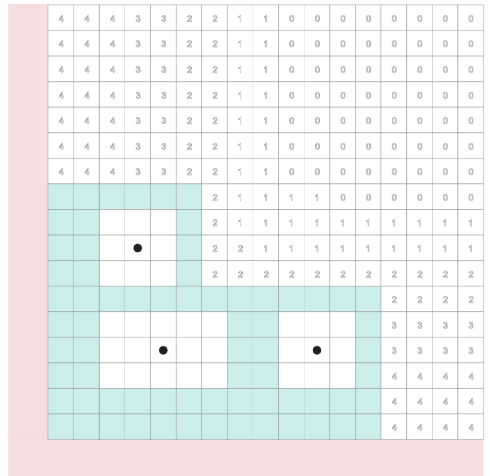
Comparison of values between connectors



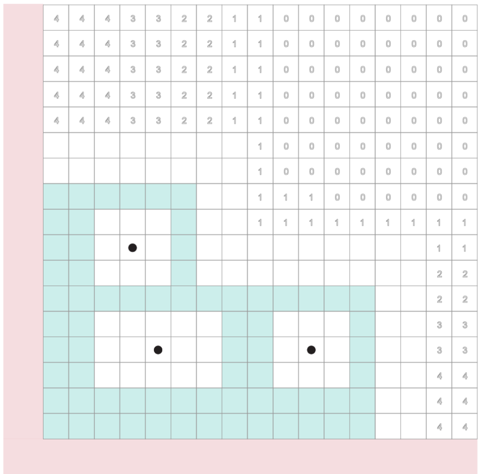
Selection of highest value adn update the list of connectors



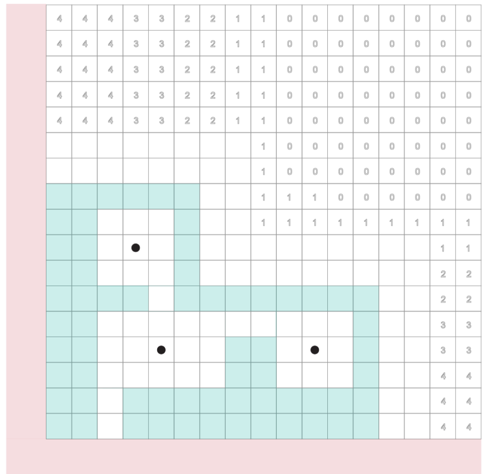
Comparison of values between connectors between connectors of multiple Attractors



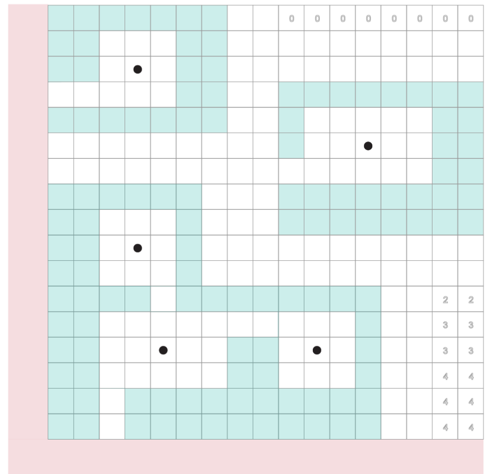
Selection of highest value and update the list of connectors



Cull cells in boundary and update the list of cells and values

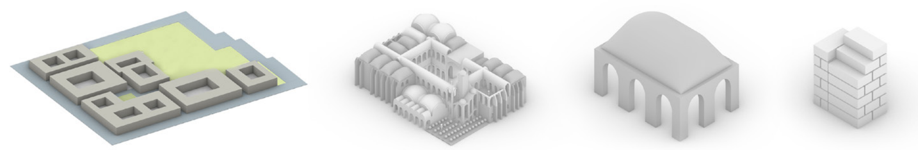


Connectivity between modules and circulation



Repeat same process for next configurations

Figure 15: Tetris Rearrangement Strattegy - Implementation of modules.
Source: Group Autorship



2.3) Urban Script

The digital implementation of the Urban rearrangement strategy was scripted in Grasshopper following the same process described on title 2.3) Implementation of Modules.

Parameters in rhino: The following parameters are the geometrical inputs for the script.

- Boundary of the Urban Block: The area in which the cells will be implemented for the distribution process.
- Grid Size: The size for the grid sell and the module sizes.
- Boundary of the attractor/s: The position of the attractors, from which each sell will measure its closest distance.

Configuration and Module Sizes:

First, an initial number is given, which dictates the number of modules per configuration. In case the configuration is of 3 modules, number is 3 is given. This will inform that 3 sets of Module Sizes are required for that specific configuration. In case of a communal configuration, number 1 is given and therefore, only one set of Module Sizes.

The size for each module will be composed of 6 numbers: A,b,c,d,e,f. A and b will dictate the number of modules of the courtyard in each direction (starting from the central cell). C, d, e and f will dictate the number of modules on each direction of the courtyard. These will come in the following order: South, East, North and West. The information for the module sizes can be inputted in two different ways to the script:

- Randomly generated: An extension was scripted to create random values for the housing modules and arrangement configurations.
- Manual input: Values provided by Scale 02-Script can be introduced manually to the script.

The distribution process requires to update the list of available cells after each process is finished. Therefore, the plugin of Anemone was used to create introduce loops to the script and enable this process.

Single module arrangement: If the configuration is of one module (for a communal dwellings) the process is finished after the first loop, culling of the cells inside the selected module plus the ones around itself to leave the space for circulation/ future growth. In this case, the second loop is not required.

Multiple module arrangement: If the configuration contains multiple modules, the biggest one will be selected as the central one. After completing the process in the first loop for the main module, the following set of sizes will be inputted on the second loop. In this loop, only the cells that allow the correct connection between main and secondary module will be shown as available positions.

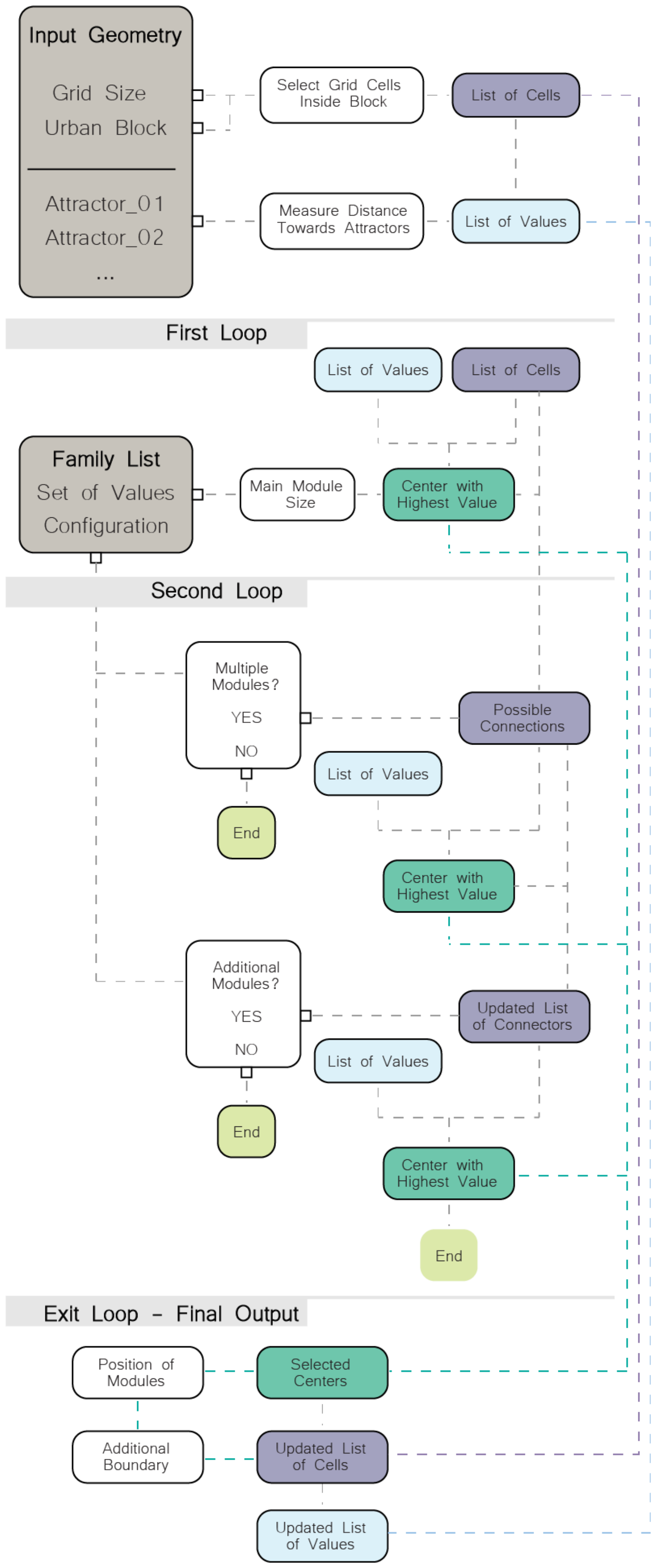


Figure 16: Urban configuration - Script Flowchart.

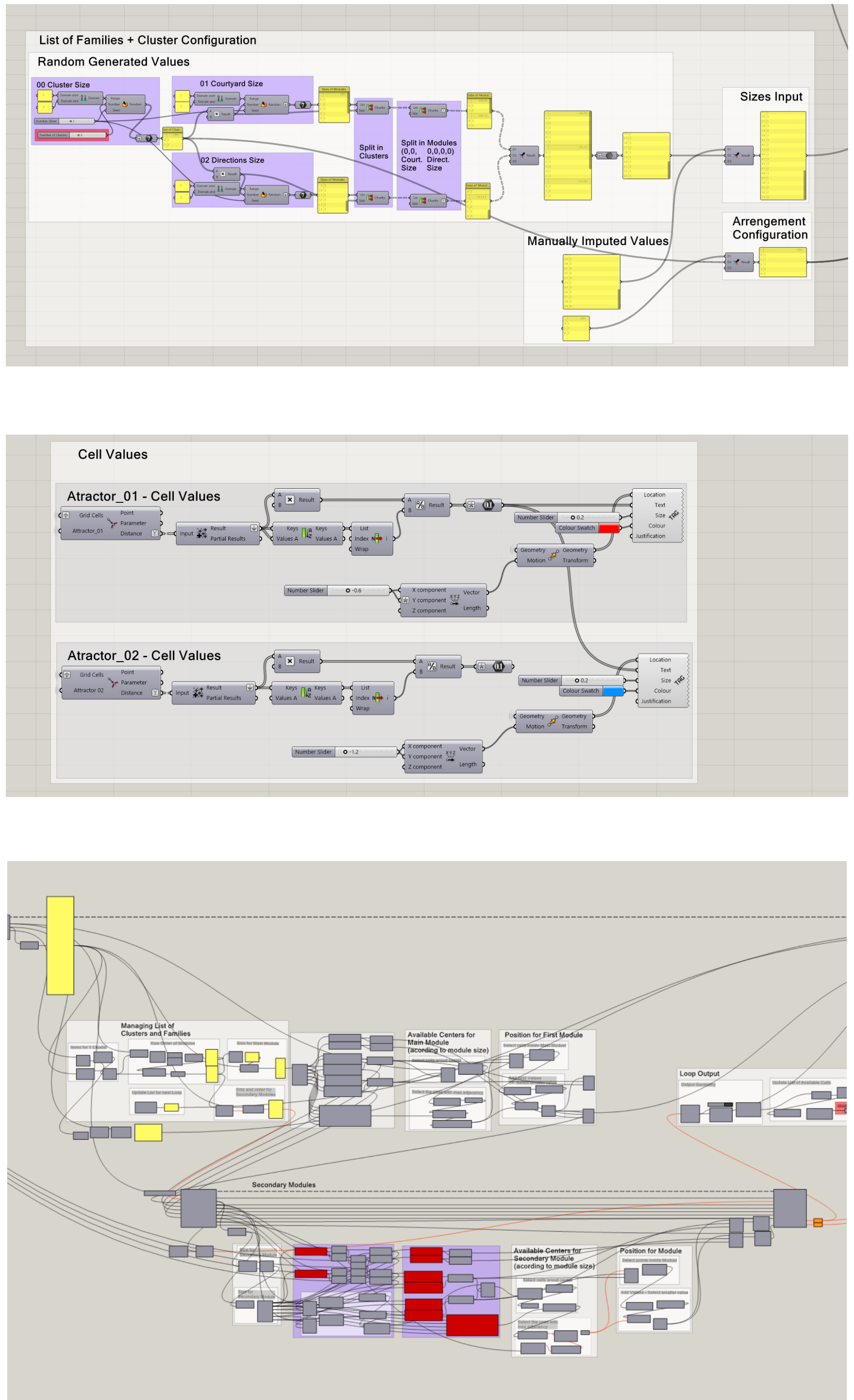
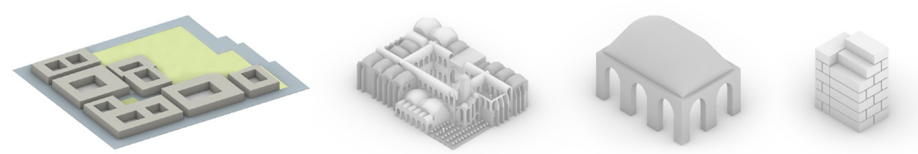


Figure 17: Urban configuration - Script Flowchart.



After selecting the position for the secondary module, the available connections will be updated to avoid the overlap between secondary modules (in case the configuration is of more than 2 modules).

Outputs for Scale-02:

- Position of the modules: The position of each module and the connected modules for each configuration.
- Unavailable directions for secondary modules: When a secondary module connects the main module, it will cull the direction that is being used for the connection. This will tell the process in Scale-02 that the functions assigned to that direction will go to a second floor.
- List of remaining cells: The available cells for green spaces, communal spaces and circulation/future growth.

2.4) Implementation of Tetris Rearrangement Strategy

As we can see in the following sequence of images, the process explained before is applied step by step into the real dimensions of an urban block. We can se how the reaction of the grid could variate depending on the attractors present on the blocks, which could also represent functions proposed by other groups for the improvement of the camp situation.

2.5) Further improvements of the urban strategy for TerraTetris 2.0

Further research and definition should go into the different interactions between house units and the rest of functions on the camp. For the example shown on the project, only one option is used, where multiple objects interact with the cells creating the polarization effect on the grid, but all of them with the same strength and direction of attraction.

Initially, the surrounding space on each arrangement was proposed for the dwellings to grow in case it is not required for circulation/interconnection purposes. All access to the arrangements will always be from the main module of the configuration, to the shortest path towards the street. The interconnection between these accesses and the green areas should define the required circulation space on the grid. The remaining cells should be assigned as space for the spaces to grow. In case there is interaction between two houses, the connection should be established in such a way that the spaces of the house don't lose efficiency and quality.

Initially, the surrounding space on each arrangement was proposed for the dwellings to grow in case it is not required for circulation/interconnection purposes. All access to the arrangements will always be from the main module of the configuration, to the shortest path towards the street. The interconnection between these accesses and the green areas should define the required circulation space on the grid. The remaining cells should be assigned as space for the spaces to grow. In case there is interaction between two houses, the connection should be established in such a way that the spaces of the house don't lose efficiency and quality.

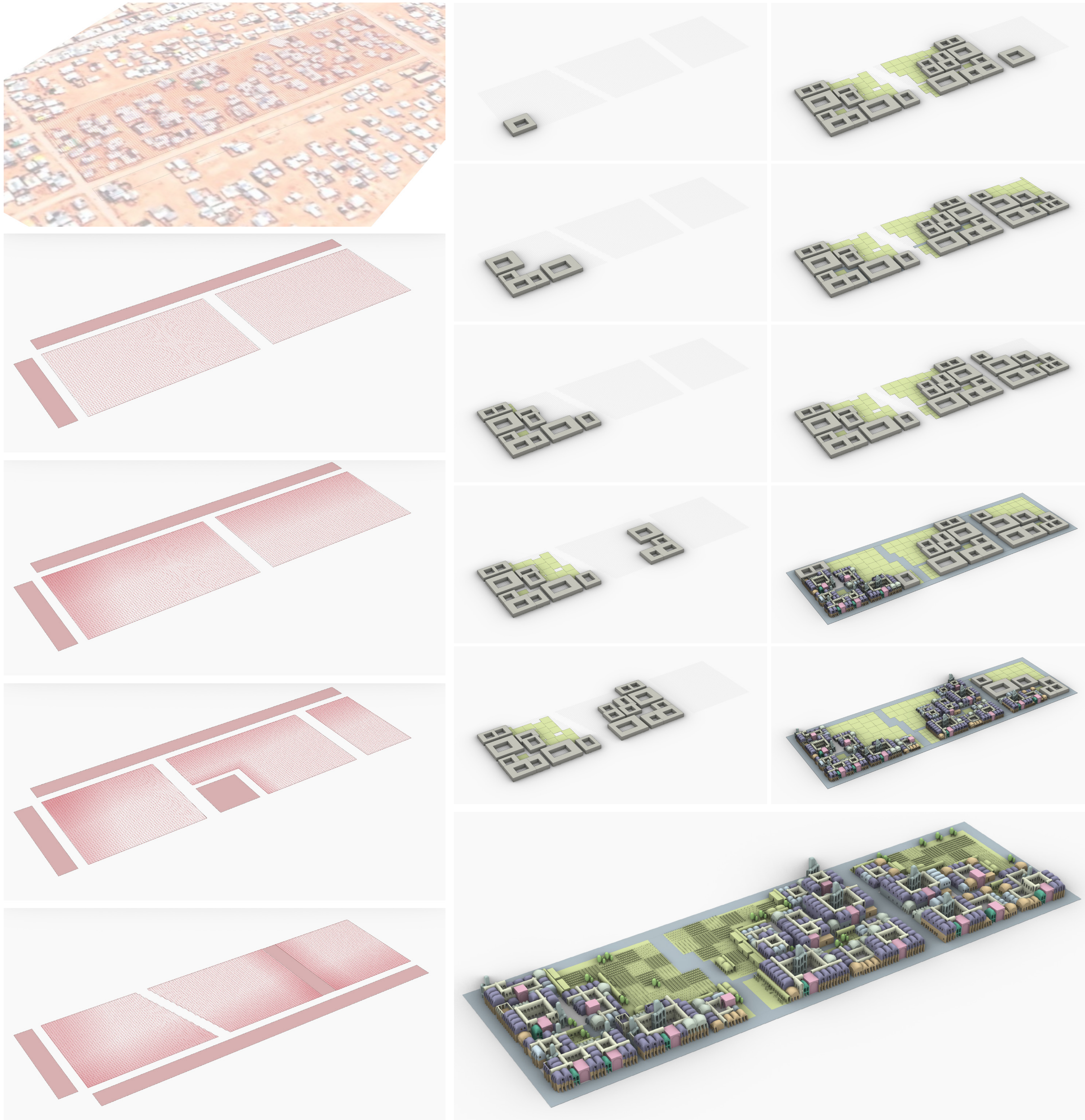


Figure 18: Tetris Rearrangement Strattegy - Process Overview.

MANUAL APPROACH

CARCASSONNE GAME LOGIC

The game of Carcassonne was studied to derive at logics to connect the spaces by users. The manual approach was taken into consideration to understand how a human mind works with intuition to develop and understand a parallel logic for computation.

The first step is to convert the configuration grid into smallest tile that can make up any space. This tile is then divided into four parts which define its connection system. All of these four parts are given a connection type. There are 5 types of connections.

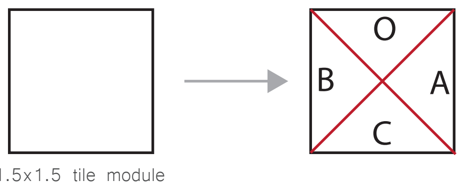


Figure19: Division of tile into 4 parts

CONNECTION TYPES

A - CONNECTED TO ADJACENT FUNCTION (as per bubble diag)
B - CONNECTED TO ITS OWN KIND
C - CONNECTED TO THE CORRIDOR (RIWAQ)
D - CONNECTED TO THE COURTYARD
O - CONNECTED TO AN OUTER FACE (light/vent)

TILE TYPES

This results in seven types of tiles for the game.

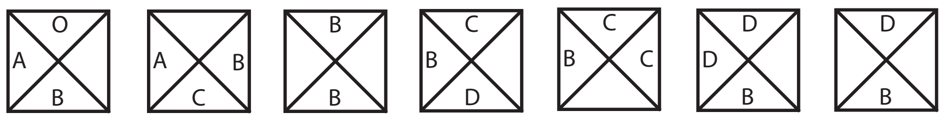


Figure20: Tile types

CONNECTION RULES

1. The connection can only connect to its own type.
2. Few connections are left blank on the tile, which can connect to any type. This allows enough flexibility for users to orient the spaces as per their needs.

MODULES

The family size of the house predefines the maximum number of tiles needed for a module. A minimum number of tiles are always needed for a module, however, with the increasing size spacer tiles can be added. These tiles can be rotated to generate spaces in various orientations. The minimum room tiles consists of connection to outside and connection to the corridor.

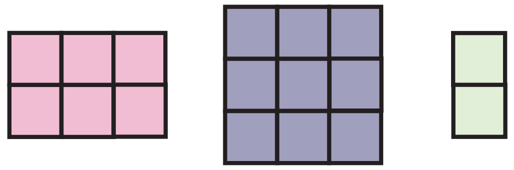


Figure21: Assigning Module to spaces

MODULE TYPES

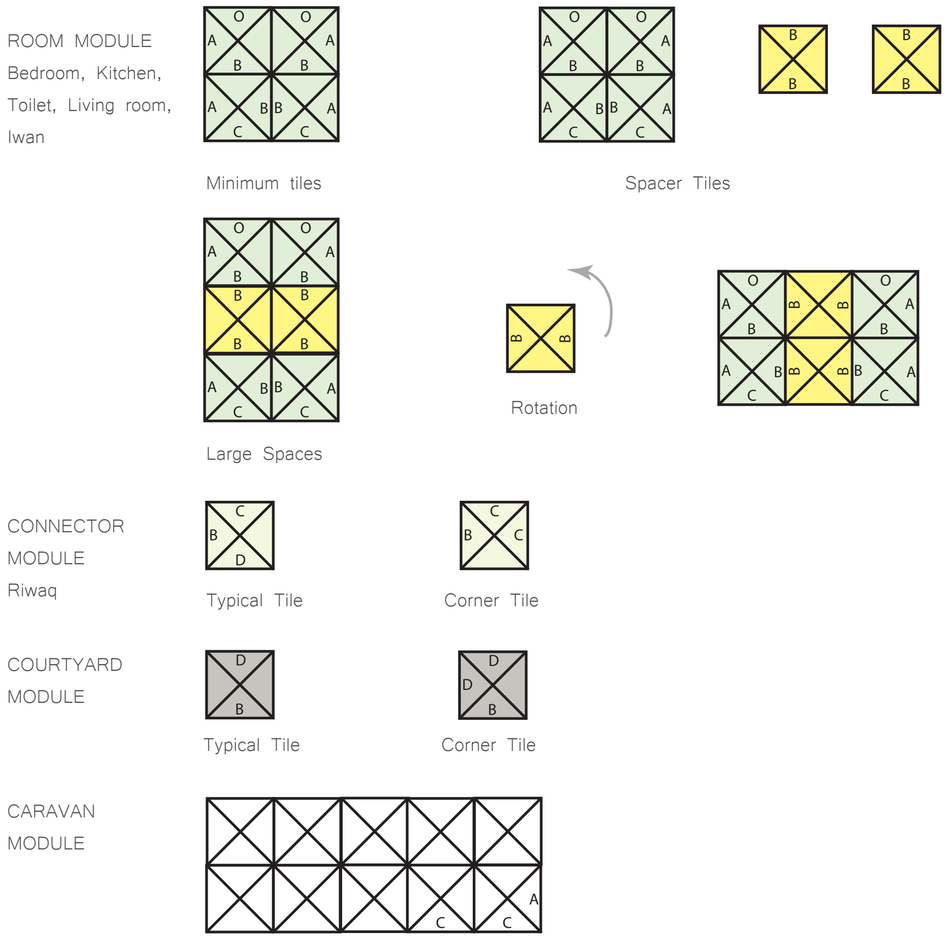


Figure22: Module Types

TYPICAL SPACES DIVIDED INTO TILES

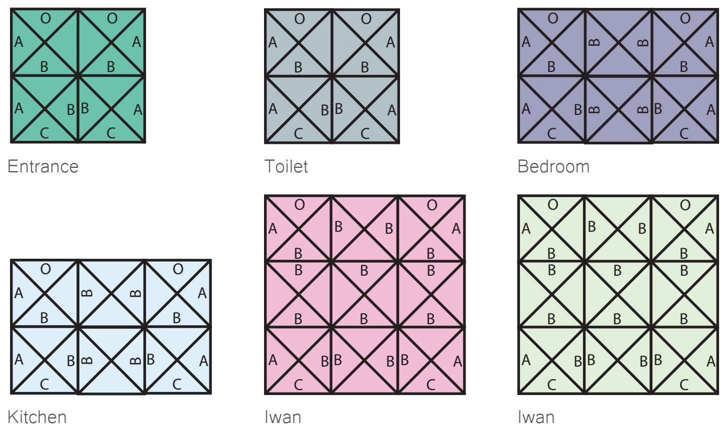
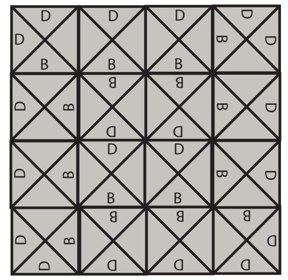


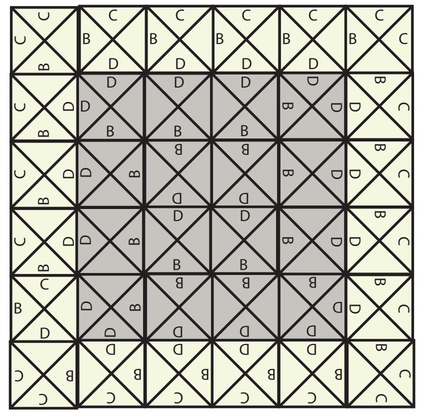
Figure23: Minimum room tiles

GAME CONFIGURATION

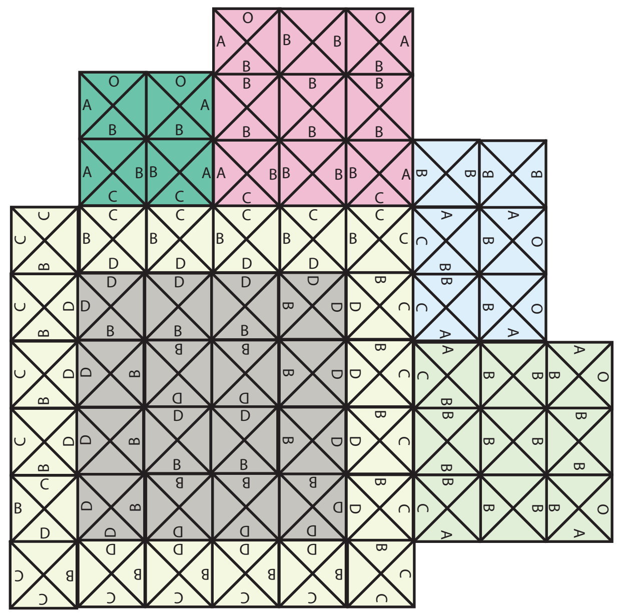
The game can then played starting with the number of tiles needs for every space and connecting them in possible ways.



STEP 1 - Starting with the courtyard module and defining the proportion of the space.

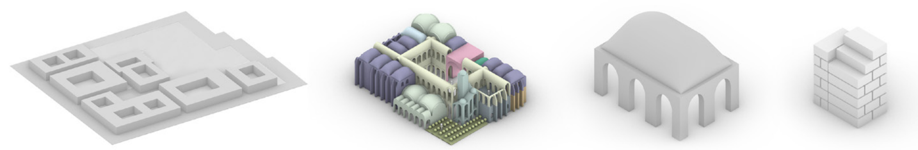


STEP 2 - Connecting to corridor(Riwaq) to ensure connectivity to every space.



STEP 3 - Configuring the unit, module by module as per the preference and priority of the user (Bubble diagram and Rel Chart)

Figure24: Game Configuration by users

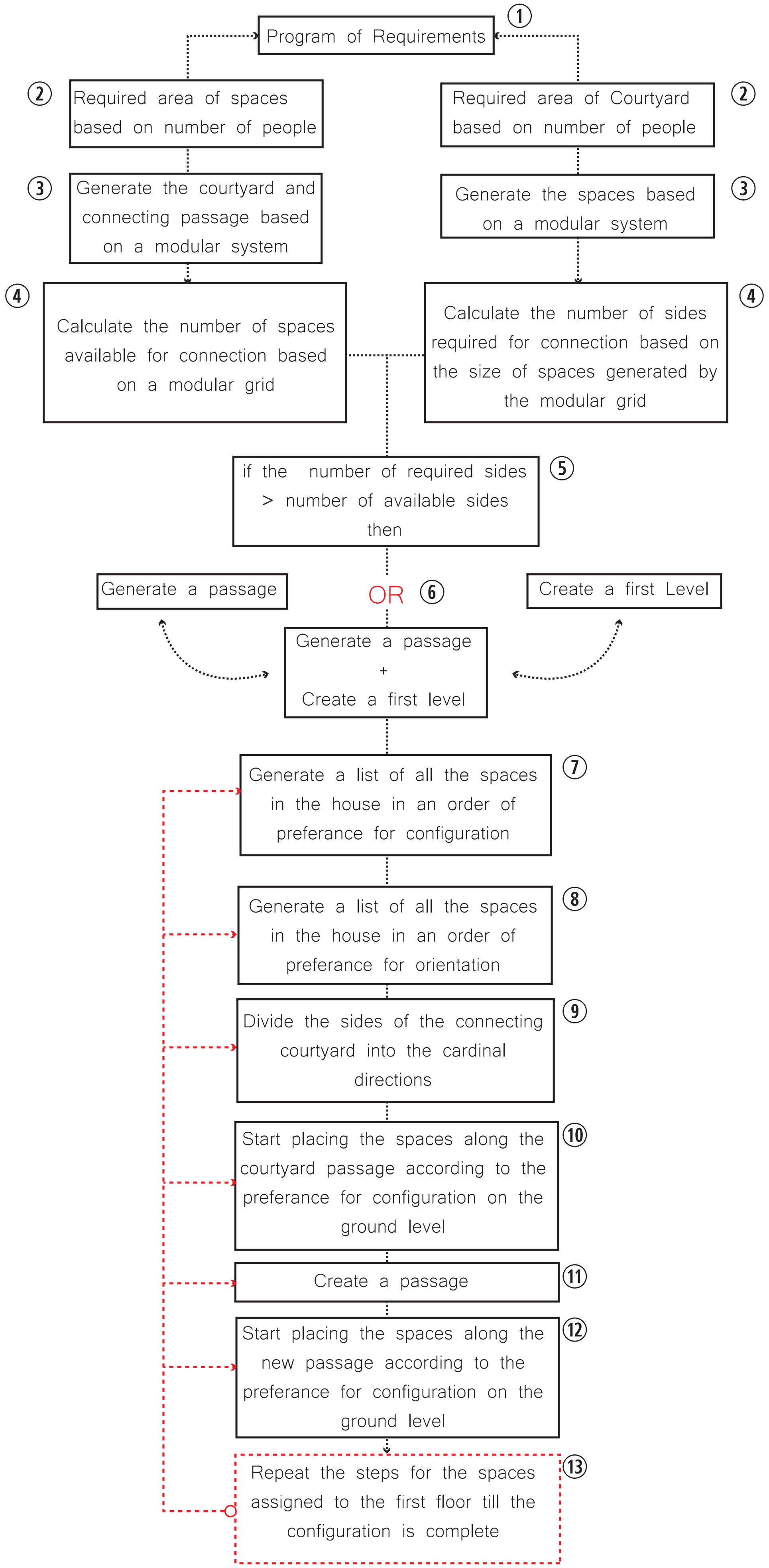


SCALE 02

2 MAIN OBJECTIVES

1.To replicate procedurally the human process of space planning:
The level 2 logic of Terratettris can generate multiple design options for the same program of requirements for a house. This is done by replicating the human design process by means of breaking it down into procedural functions. The whole concept rests on the idea that Architectural planning is a step by step process of arranging spaces based on a decision tree. The decision tree is a culmination of priorities and functions which the designer uses in order to make logical decisions for the floor planning of the various spaces in the project according to the program of requirements.

2.To create a generative system for design which is flexible and can respond to user choices:
Since the tool relies on a decision tree to generate the spaces each step of the design can be traced back and the design options generated can be analysed for various parameters. The modifications to the elements of a decision tree in evidently leads to a different design and becomes the primary tool of the designer to manipulate the system according to his/her priorities



3. PROCESS

We as a group decided to go for a courtyard type of house since it had contextual benefits in terms of climatic comfort. A typical courtyard house is configured in a way that all the spaces in the house connect to the courtyard and the courtyard becomes the primary source of access to all the spaces in the house.

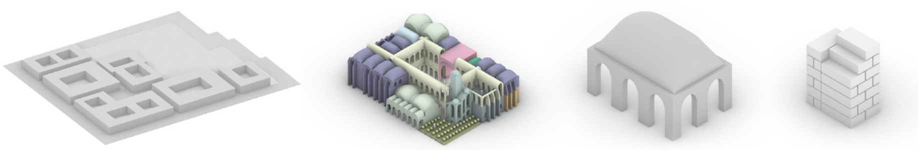
This provided us with a starting point for our breakdown of the process. As the first step we generate the courtyard and the connecting passage around it. Then based on the size of the connecting passage we calculate the available sides for connection with the other spaces in the house.

The size of the courtyard is dependant on the number of people in the house and a per person area consideration. It can be argued that the courtyard can also be calculated based on the number of sides needed by the various spaces in the space program so that the size will always be enough for the spaces to fit in . The main problem with this approach is that the courtyard becomes so large that it no longer functions as a courtyard but behaves more like an open space both spatially and climatically.

The second step in the process is to start arranging the spaces along the courtyard till all the available connecting spaces along the passage are filled up. The remaining spaces then can either go on the first floor or can connect to an extra passage originating from the main courtyard passage. A combination of both is also possible .

The order of arranging spaces along the courtyard is based on the decision tree as generated by the player in the game as described further in the report.

The flowchart on the left describes the steps involved in the process of configuration for terratettris.



4. THE CASE

In order to explain how the configuration logic works a case was considered where a communal dwelling is to be made for 13 people. The dwelling is adjacent to the market street , hence the communal house turns into a hybrid typology which will accomodate the market function on the ground floor of the building on the side facing the market street.

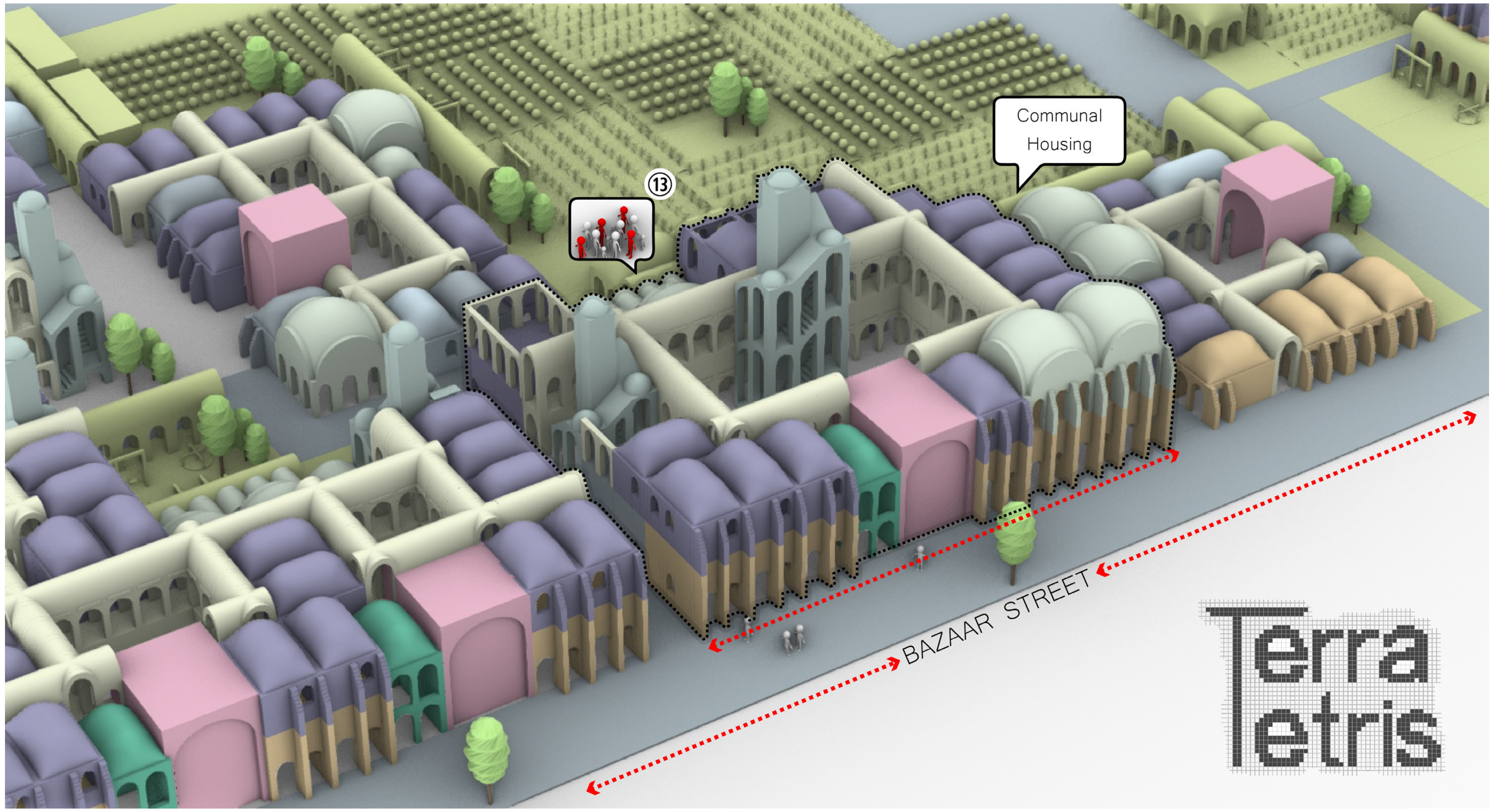
The complete logic (Geometry + Code) behind the configuration of this house will be explained step by step simultaneously in the report.

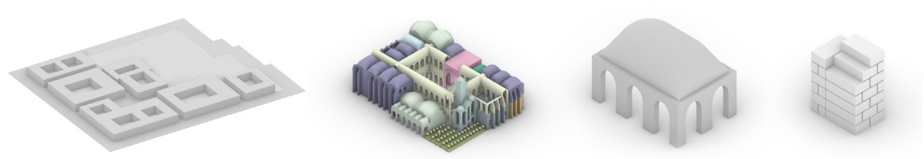
As a first step among the 13 people the decision makers for the families involved in the composition are chosen to play the game (players marked in red).
The selected players have to represent the interests of their indivudual families in the participatory game.

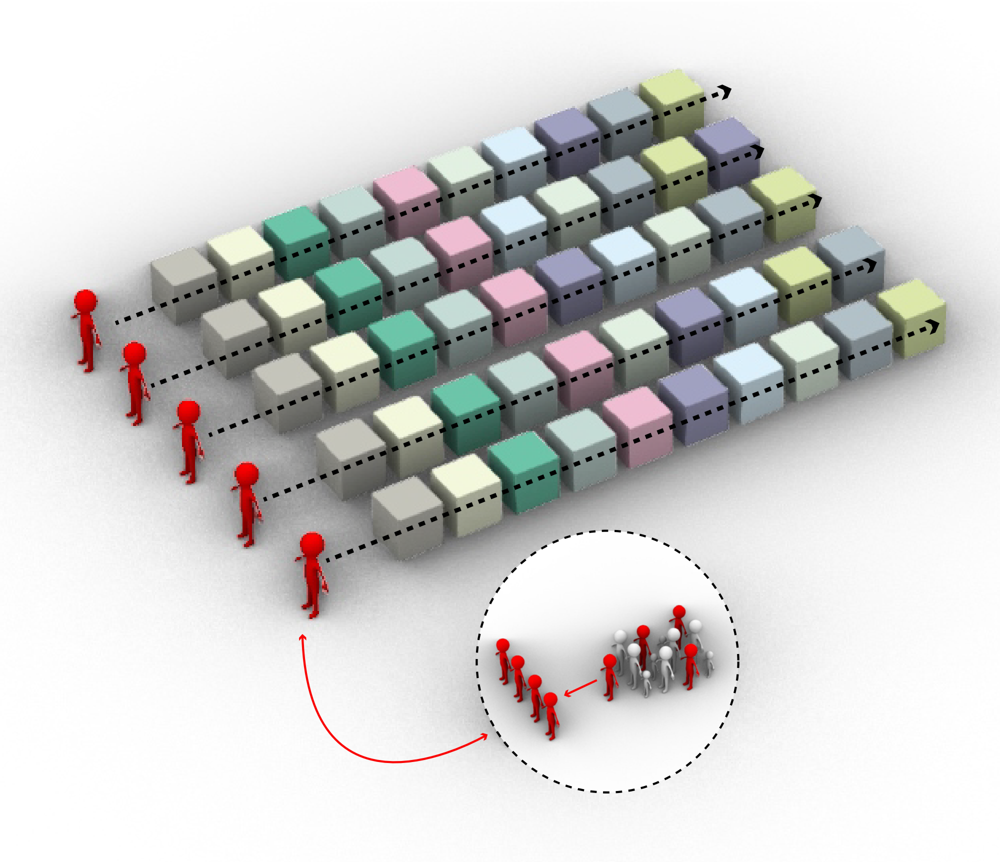
The participants have to choose two things. The preferance of the spaces in the house according to them and the orientation of the rooms (East,West,North,South) as per their needs.

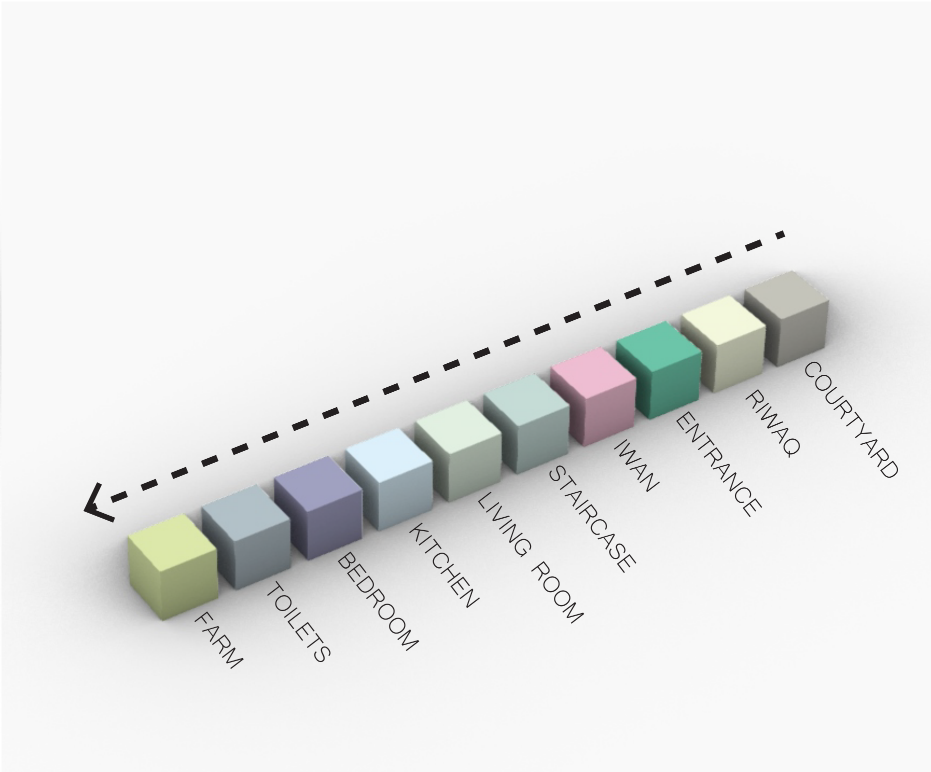
If the participant doesnt choose the preferance or partially fills the choices then the rest of the data required for the script is autocompleted based on the Architects choice of options.

The Architects choice of decisions is based on the Relcharts which shows the relationship between the spaces and the Bubble diagrams which indicate connectivity between spaces and the level of privacy .
The orientation of spaces is base das per climatological needs of the spaces as analysed by the architect.







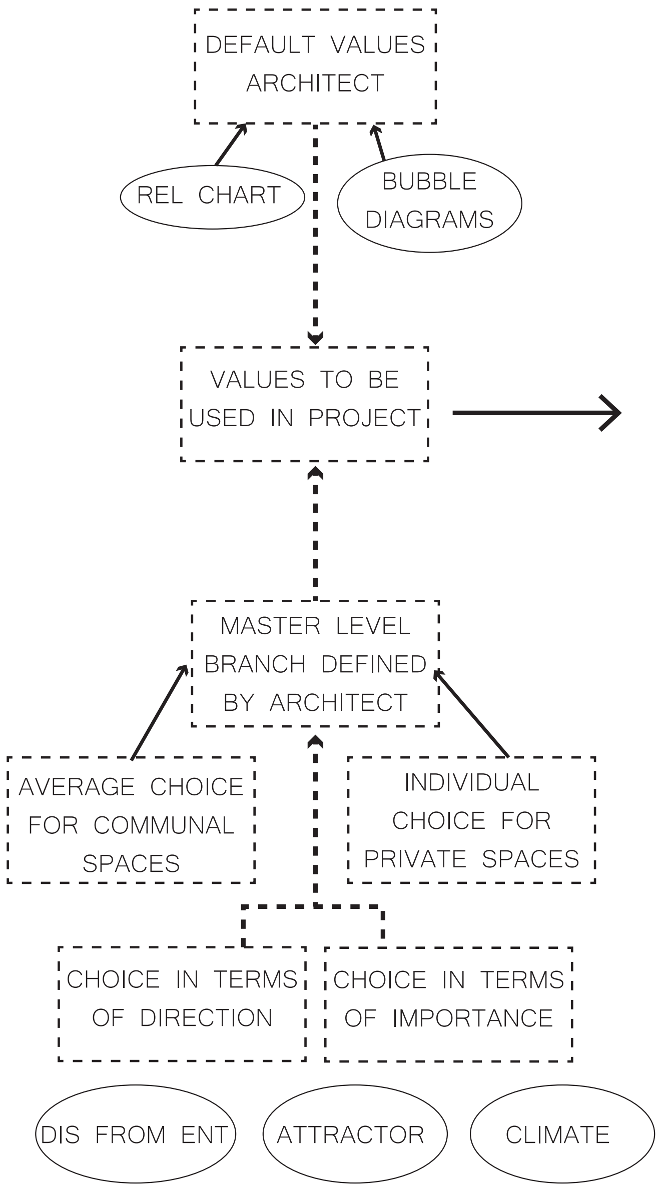


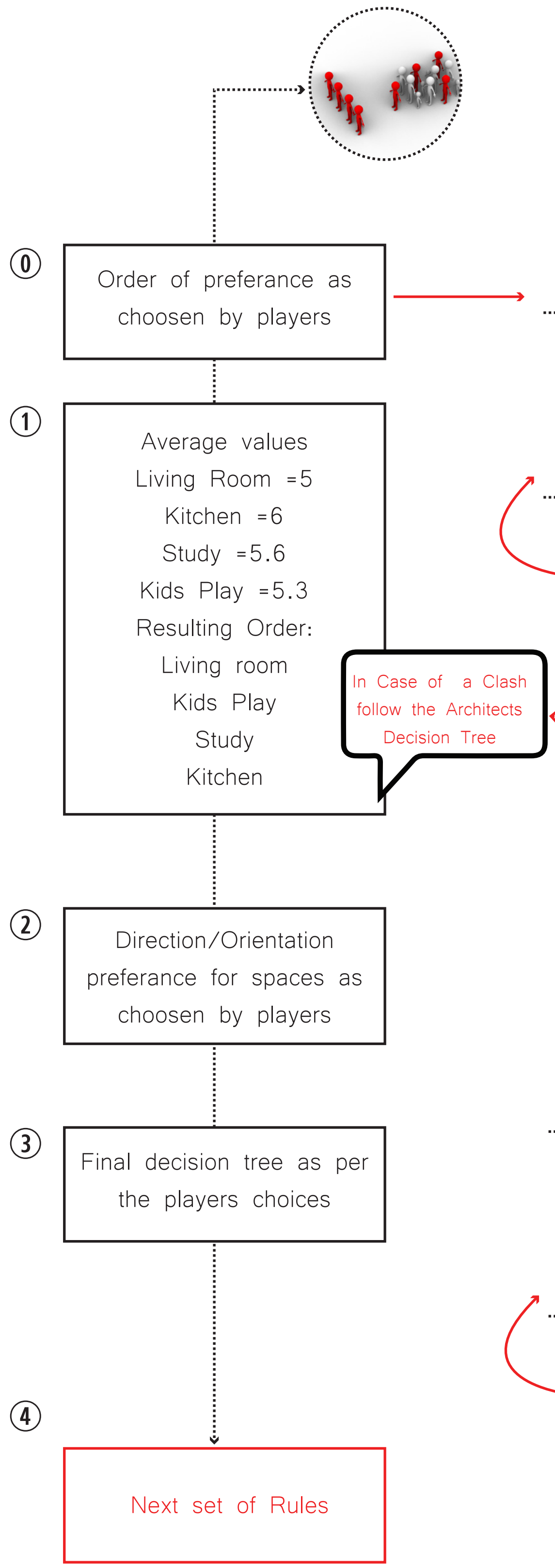
Rule 1: Preference of arrangement

5. THE DECISION TREE:

Decision tree is defined as the list of rules and their powers as defined in the configuration algorithm.
The example on the right shows a typical construction of a decision tree. Each player chooses the priority or preferance of arrangement of spaces as per his wishes. The higher the priority the earlier will be the placement of the particular space in the layout. The average of the values are counted and the spaces are ordered accoringly in the list. In case of similarity of the calculated value the Architects decision tree is reffered to solve the decision.
Similarly the orientation of the spaces are also determined and a list is made.

The master level structure is also present in the decision tree where the common spaces in the house will always be placed first and the living private spaces can be arranged later or can go on top floors . This has been made due to the structural considerations and privacy priorities for the living spaces.





PRIORITY LIST AS PER ORDER SELECTED BY PLAYER 1

COMMON SPACES
- 0 COURTYARD
- 1 RIWAQ
- 2 ENTRANCE
- 3 IWAN
- 4 LIVING ROOM
- 5 KITCHEN
- 6 STUDY
- 7 KIDS PLAY
- FARM

PRIVATE SPACES
- 0 BEDROOM 1
- 1 TOILETS

PRIORITY LIST AS PER ORDER SELECTED BY PLAYER 2

COMMON SPACES
- 0 COURTYARD
- 1 RIWAQ
- 2 ENTRANCE
- 3 IWAN
- 4 KIDS PLAY
- 5 LIVING ROOM
- 6 KITCHEN
- 7 STUDY
- FARM

PRIVATE SPACES
- 0 BEDROOM 1
- 1 TOILETS
- 2 BEDROOM 2

PRIORITY LIST AS PER ORDER SELECTED BY PLAYER 3

COMMON SPACES
- 0 COURTYARD
- 1 RIWAQ
- 2 ENTRANCE
- 3 IWAN
- 4 STUDY
- 5 KIDS PLAY
- 6 LIVING ROOM
- 7 KITCHEN
- FARM

PRIVATE SPACES
- 0 BEDROOM 1
- 1 TOILETS

FINAL PRIORITY LIST AS PER ORDER

COMMON SPACES
- 0 COURTYARD
- 1 RIWAQ
- 2 ENTRANCE
- 3 IWAN
- 4 LIVING ROOM
- 5 KIDS PLAY
- 6 STUDY
- 7 KITCHEN
- FARM

PRIVATE SPACES
- 0 BEDROOM 1
- 1 TOILETS

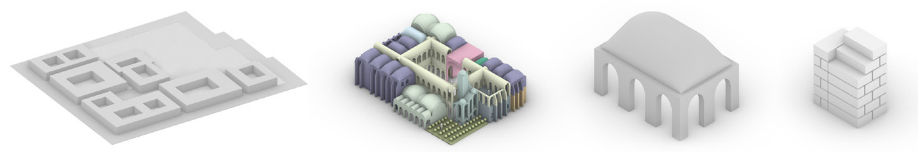
- 0 BEDROOM 1
- 1 TOILETS
- 2 BEDROOM 2

- 0 BEDROOM 1
- 1 TOILETS

PRIORITY LIST AS PER ORDER ARCHITECTS PRIORITY

COMMON SPACES
- 0 COURTYARD
- 1 RIWAQ
- 2 ENTRANCE
- 3 IWAN
- 4 LIVING ROOM
- 5 KITCHEN
- 6 STUDY
- 7 KIDS PLAY
- FARM

PRIVATE SPACES
- 0 BEDROOM 1
- 1 TOILETS



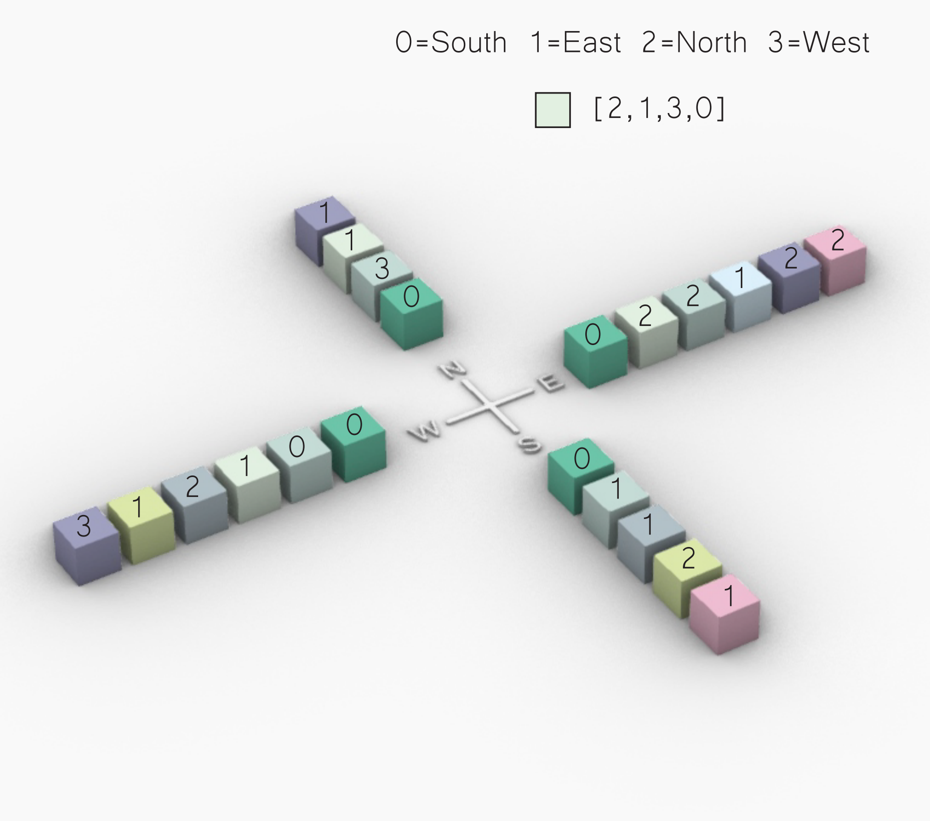
6. ARCHITECTS DECISION TREE

In the previous section the architects decision tree and its role in the configuration logic was explained. In this part we discuss more about the fundamentals which govern that decision tree. The relchart and the bubble diagrams as shown on the right together govern the decision tree that is considered the default value .

Selection of Orientation of the spaces:
The default selection of the orientation of spaces is dependant on the climatological factors and the choices are made between the four cardinal directions as shown in the figure below. Based on the preferance a list is made , for example consider living room as a space the priority for it is North, East, South followed by west considering the protection from the harsh sunlight hence the list created will be Living room = [2,1,0,3] where 0=South, 1=East, 2=North, 3=West.

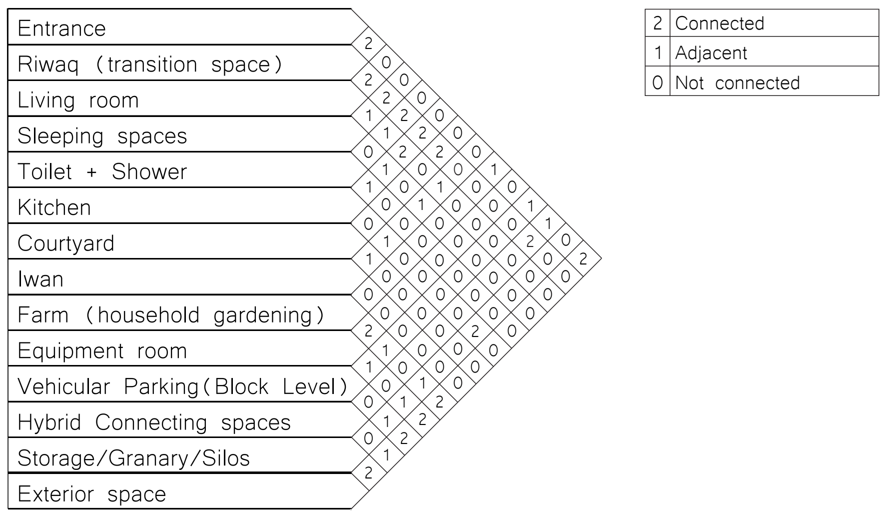
Why is this important?
The logic for configuration works on searching for connection spaces betwwen the spaces in the house and the corridors so If the space is not available in the direction of the first priority then the code can search for the next priority in the list. This ensures that the choice and decision of the user is considered while planning of spaces in the house.

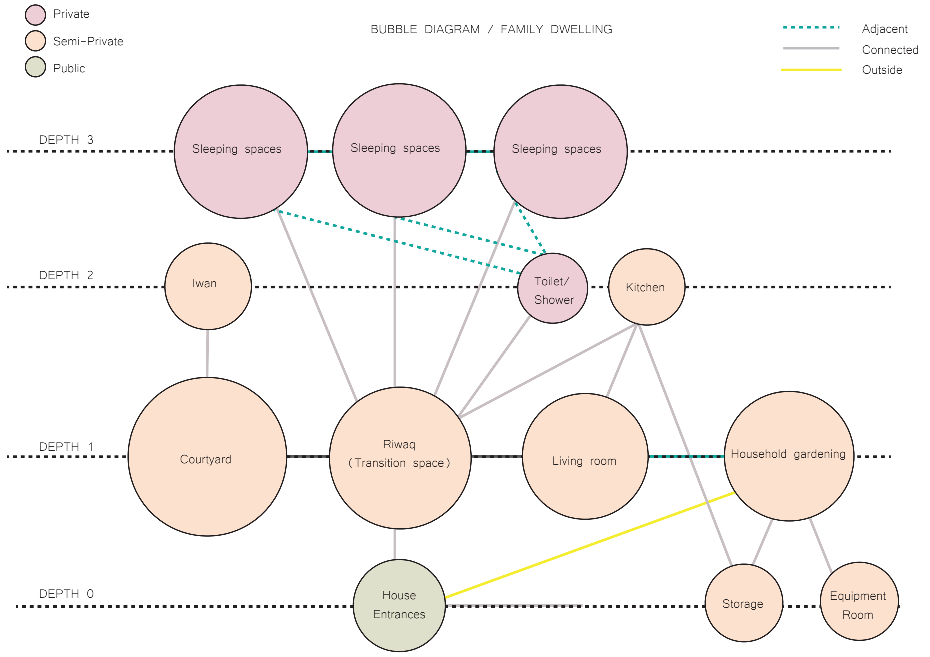
Finally the decision tree is also different for different type of Housing since thenature of spaces in the house tend to be different.

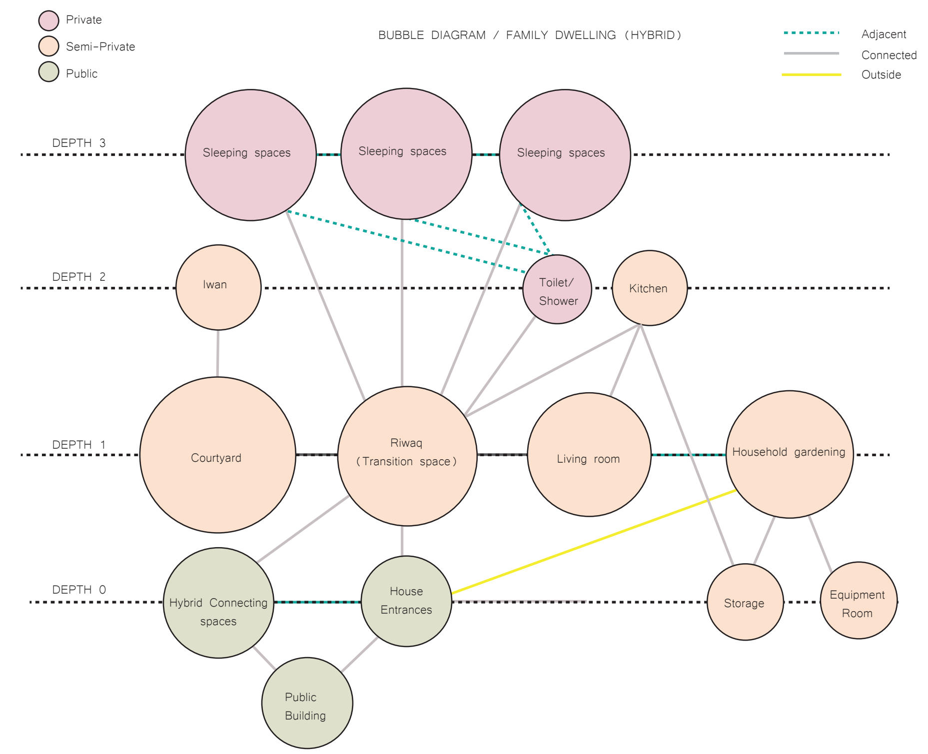


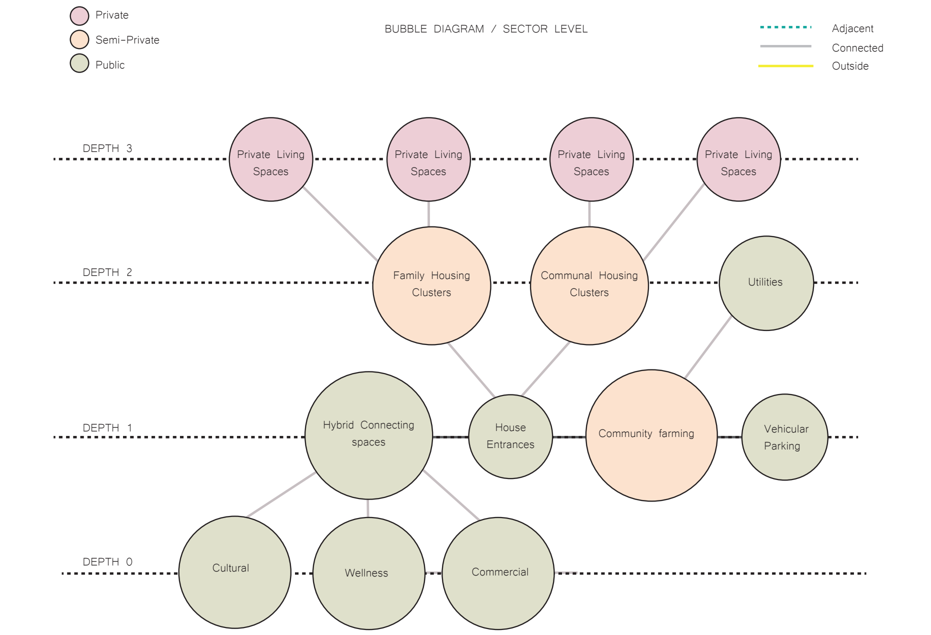
Rule 2: Determining priority numbers

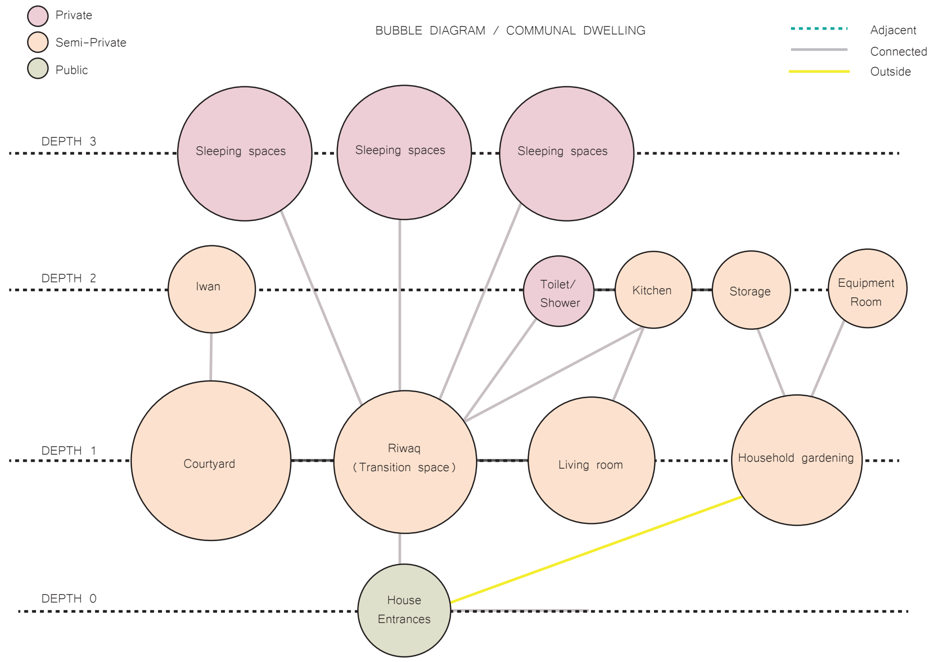
Relationship chart

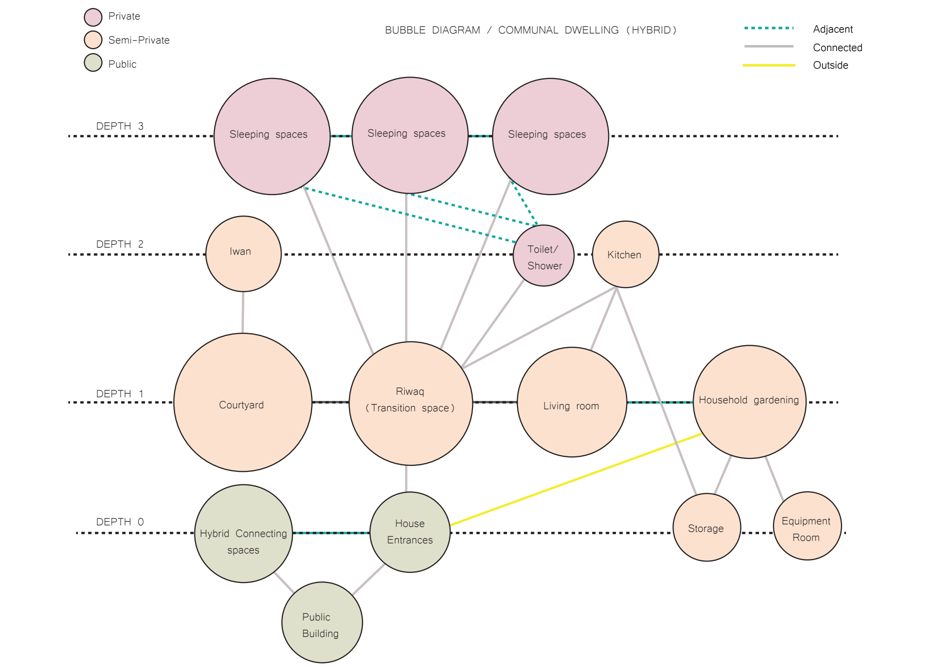












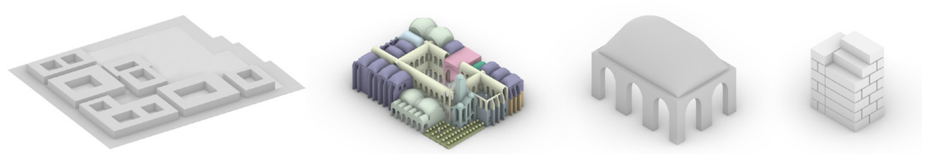
7. CODE AND ITS GEOMETRIC IMPLICATIONS

The idea behind the configuration logic was conversted into code in python. The example shown on the right is meant for communal dwelling where only the number of people in a tenament and the grid size is required to run the code.

The code was made in an object oriented manner so that it could be modular , reused and modified easily to add more and more functions.

In the code the outputs exists as tuples and lists which when taken into grasshopper have geometrical implications.The digram on the left indicates what each functions in the class does, the actual code can be accessed via our git.

In our process, we aso found out that some of the functions in the code would function better if a user input is taken before running it or at every instance of running it because of the abstract nature of desired output needed from it. This is furhter elaborated when the geomtrical implications are discussed.



```
# The aim of this script is to generate the necessary areas for the communal or family dwelling based on the number of people convert it into the measurements of the modular system
# The no. of modular blocks for each space are then converted into clusters of necessary proportions.
# The logic for connecting them with the courtyard is established and the blocks are split into lists based on directions and floors.
import math
module_size = 2.25 # considering a grid of 1.5m x 1.5m
final_module_matrices = []
class communal_dwelling :
```

(0) def __init__(self,nos):
Initialization function

(1) def area_distribution(self):
We take out the areas required for each space based on research and referances . The number of people and the area required per person is used to generate the areas for each space

(2) def blocks_house (self):
Taking out modules from the areas

(3) def matrix_generator_for_courtyard(self,variable_a):
Generating the (Xaxis, Yaxis) number of modules for the courtyard

(4) def matrix_generator_for_incremental_spaces(self,variable_b):
Area to number of even modules which can form a rectangle

(5) def matrix_generator_for_toilets(self):
Generating the number of Toilets required and generating the list of tuples (Xaxis, Yaxis) number of modules for each toilet.

(6) def Bedroom_matrix (self):
Generating the number of Bedrooms required and generating the list of tuples (Xaxis, Yaxis) number of modules for each toilet.

(7) def Incremental_space_matrix (self):
Generating the (Xaxis, Yaxis) number of modules for incremenral spaces

(8) def priority_list_definition (self):
Generating the priority list for all the spaces in terms of orientation

(9) def Final_list_of_modules_of_spaces (self):
Generate a final list of tuples indicating the number of modules in (Xaxis, Yaxis) for all the space arranged as per priority list

(10) def sides_available_for_connection (self):
Generate a list of available sides (length of each side of riwaq/ module size) in each direction.

(11) def smallest_side_connection_per_function (self):
Generate the minimum sides needed for connection for each sace in the space program

(12) def Sorting_into_directions_as_per_priority(self):
Algorithm to sort the spaces into the direction based on the sides required and the sides available

(13) def Sorting_into_directions_as_per_priority_for_bounding_box(self):
The biggest spaces in all the directions are deconstructed

(14) def Bounding_box(self):
Generation of a Rectangle based on the max distances of spaces in all the directions (Bounding box for the floor plan)

(15) def Sorting_into_directions_as_per_priority_combined_dwellings (self,S,E,N,W):
The algorithm is implemented for houses which are connecting with the main house in case of family dwellign where the connecting side needs to be eliminated in the spaces available for connection

(16) def First_floor_list_of_modules_of_spaces (self):
Generate a final list of tuples indicating the number of modules in (Xaxis, Yaxis) for all the space arranged as per priority list for the first floor raw input of the remaining spaces used to generate this list.

(17) def Extra_passage_generation (self,,S,E,N,W):
Extend one/multiple sides of the riwaq to create an extra passage for connections

(18) def Sorting_into_directions_as_per_priority_first_floor(self):
Algorithm to sort the spaces into the direction based on the sides required and the sides available for the spaces assigned on the first floor

(19) def Shift_cells (self):
Shift tall he cells to the outermost edge of the bounding box on the ground floor

Classified under intuitive decisions taken by the user

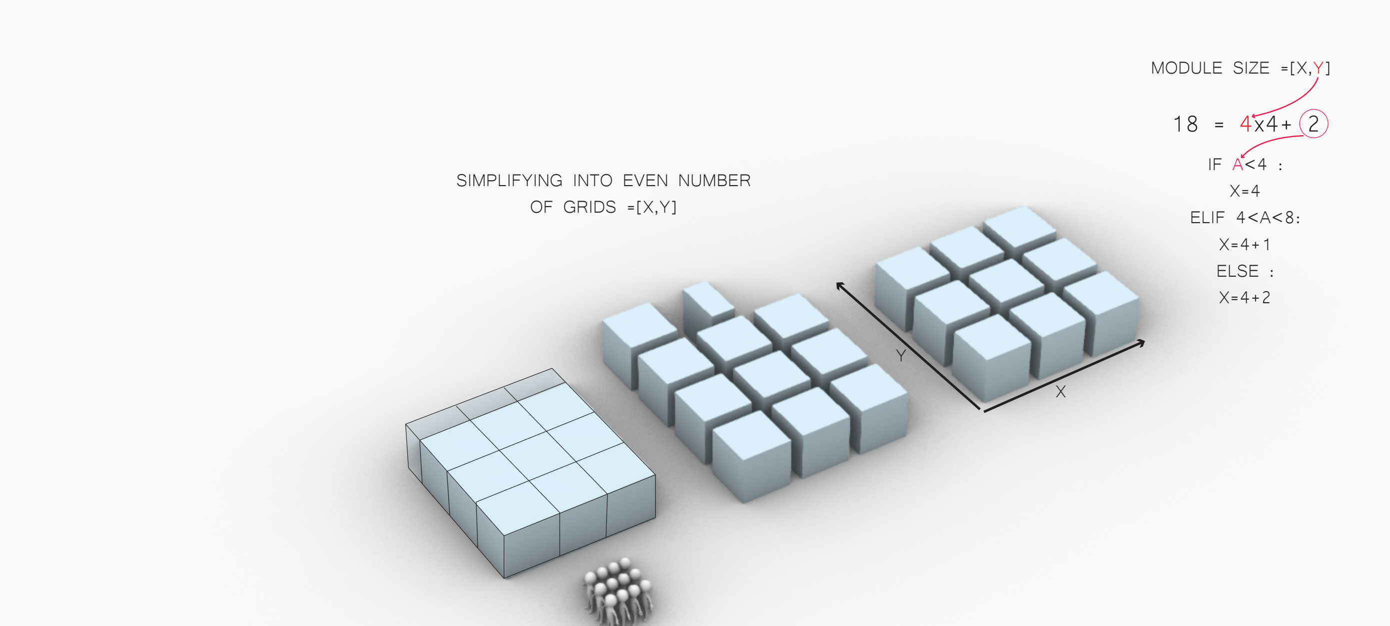
7. CODE AND ITS GEOMETRIC IMPLICATIONS

In Rule0 the per person areas for each space in the space program is calculated. This furthermore is converted into the number of modules based on the size of the grid in this case 1.5 x 1.5 . The number of modules is always converted into an even number so that the spaces will always have rectilinear proportions.

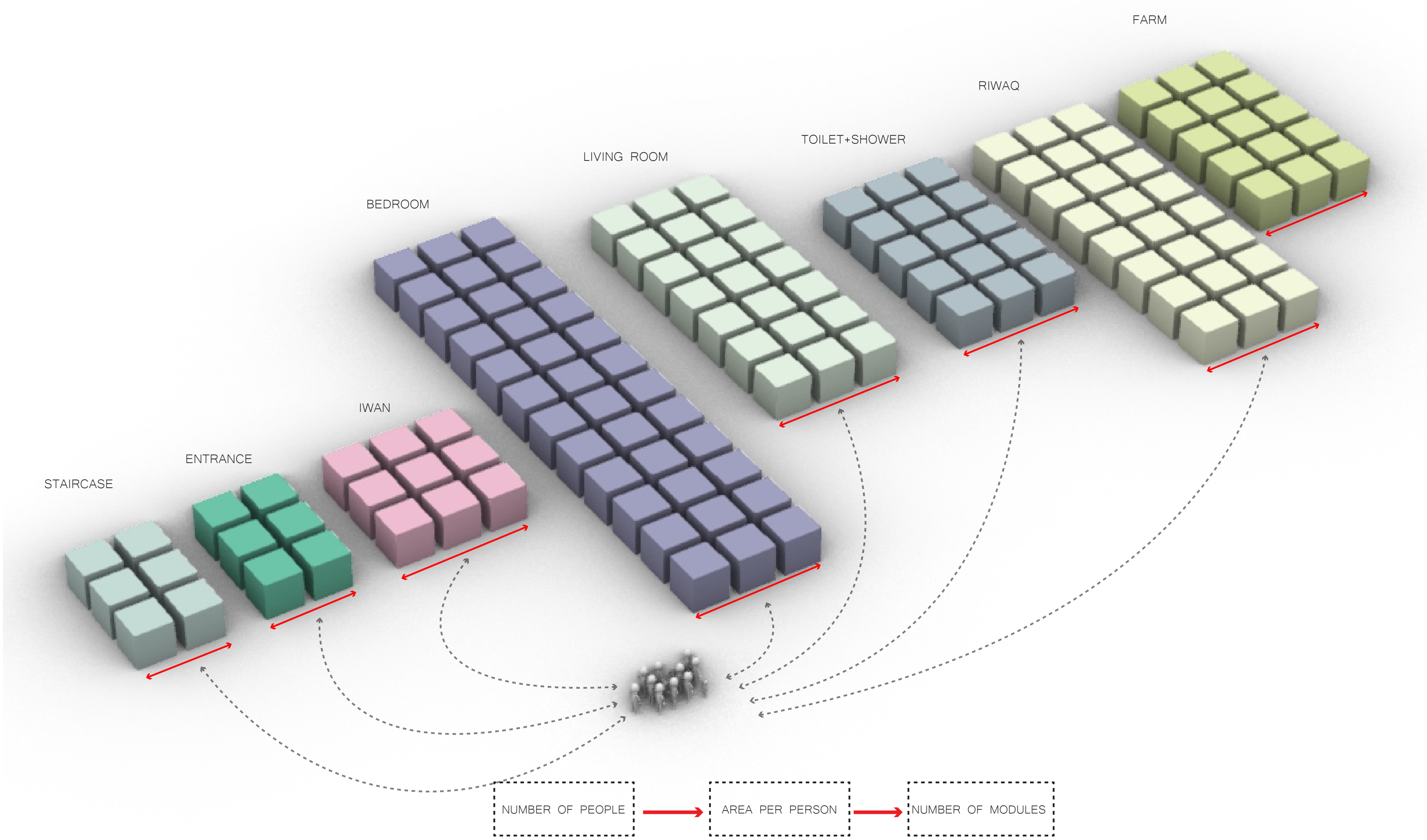
```
⓪ def __init__(self,nos):
   Initialization function

① def area_distribution(self):
   We take out the areas required for each space based on research
   and referances . The number of people and the area required per
   person is used to generate the areas for each space
```

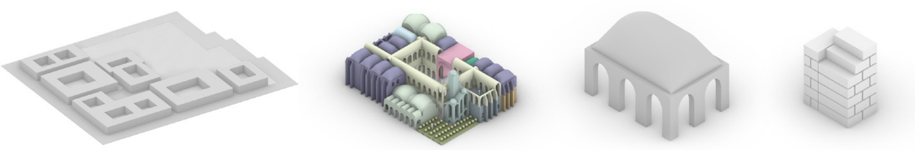
```
② def blocks_house (self):
   Taking out modules from the areas

③ def matrix_generator_for_courtyard(self,variable_a):
   Generating the (Xaxis, Yaxis) number of modules for the
   courtyard
④ def matrix_generator_for_incremental_spaces(self,variable_b):
   Area to number of even modules which can form a rectangle
```

```
def area_distribution(self):
    num = self.nos
    Entrance = num * 0.45
    Living_room = num * 5
    Kitchen = num * 1.2
    Courtyard = num*5
    Iwan = num * 0.9
    Farm = num * 3.0
    Equipment_room = num * 0.2
    areas = {"Courtyard": Courtyard, "Entrance": Entrance, "Iwan": Iwan,
            "Living_room": Living_room, "Kitchen": Kitchen, "Farm": Farm}
    return areas


def blocks_house (self) :
    area_com_dwg = self.area_distribution()
    courtyard_blocks = list(area_com_dwg.values())
    courtyard_blocks_values =[]
    for i in courtyard_blocks:
        a = int(int(i) / module_size)
        if a < 1 :
            b = 2
        else:
            b=a
        courtyard_blocks_values.append(b)
    return courtyard_blocks_values
```



Rule 0: Area per person to modules sizes





7. CODE AND ITS GEOMETRIC IMPLICATIONS

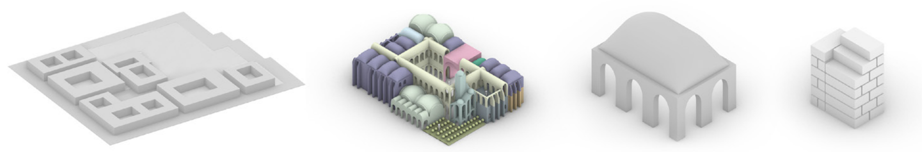
In Rule1,2 there is not a direct geometric implication but is more the basis for the space aggregation .In rule 3 the first starting point of the house configuration happens by means of generating the courtyard and the connecting corridor along it. The courtyard+corridor is also generated based on the number of people the house is accommodating. The area needed is converted into blocks in rule 0 which is further processed in this rule. The number of blocks is divided into two parts the first one contains the perfect square which lies under the total number of blocks and the second one is the reminder blocks. The remainder blocks are further analyzed, and the courtyard then can attach 1 or 2 layers of blocks to the perfect square to create a rectangular space or it jumps to the next perfect square. This approach ensures that a well-proportioned space is generated for a courtyard and the chances of it being too long or narrow are avoided. If the number of people are too low to generate a substantially big enough courtyard then the minimum size is considered.

```
② def blocks_house (self):
   Taking out modules from the areas
③ def matrix_generator_for_courtyard(self,variable_a):
   Generating the (Xaxis, Yaxis) number of modules for the
   courtyard
④ def matrix_generator_for_incremental_spaces(self,variable_b):
   Area to number of even modules which can form a rectangle
```

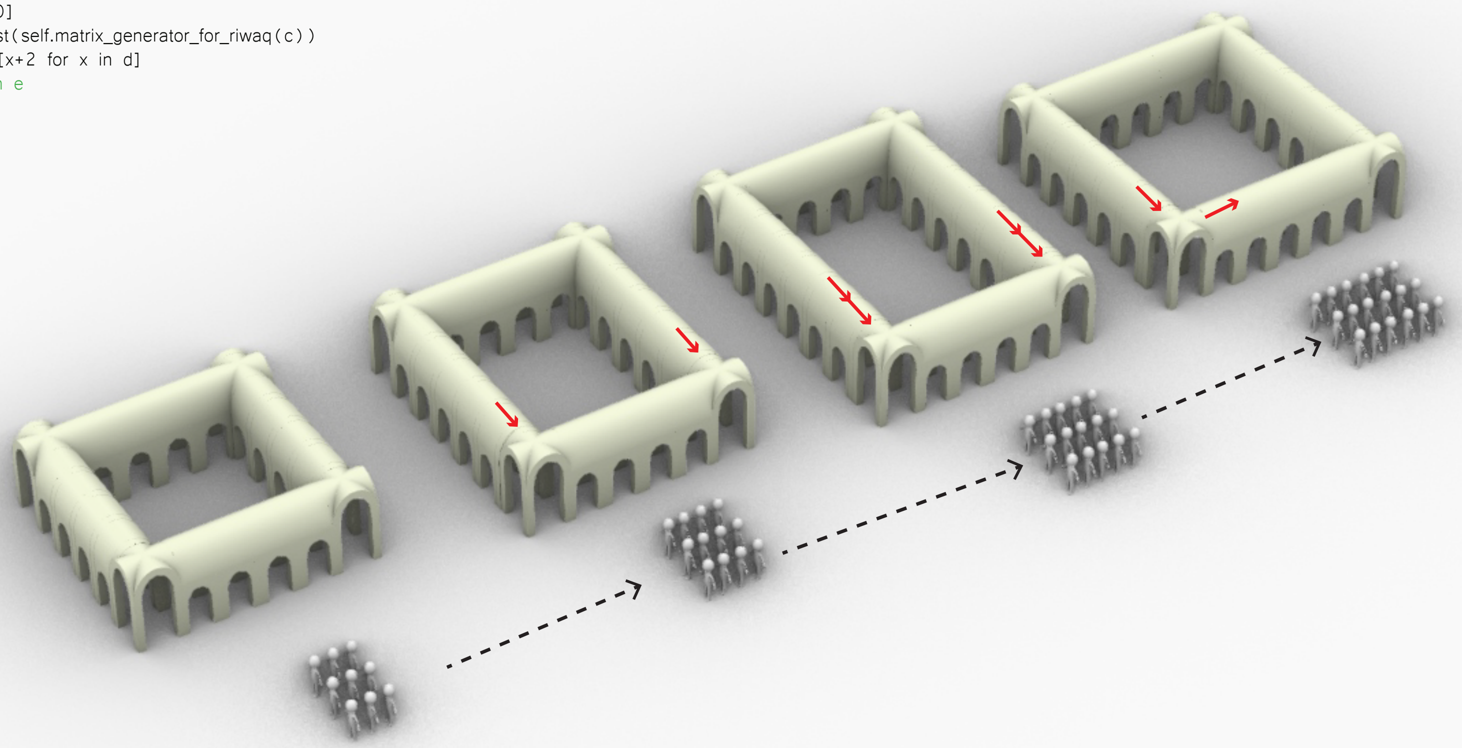
Rule 4 is meant to generate the tuple for incremental spaces. If all the spaces in the house are to be classified, then there are two types of spaces the first one being incremental spaces which increase in size if you increase the number of people. Spaces like the courtyard, Living room , Toilet and shower areas come under the umbrella of incremental spaces. The second type of spaces are the fixed spaces which increase in number if the number of people which need to be accommodated increase. Spaces that are included in this category are the Bedrooms, Iwan, Entrance.

Based on the logic defined in scale 01 unit development for the ease of construction and utility the kitchen lies in between the two categories. The X axis of the kitchen is fixed to have two modules, but the Y axis is incremental up to 4 modules after which the kitchen increases in number. This is done to ensure a better internal layout for the kitchen.

In the diagram on the left you can see that depending on the increase in number of people the space increases either in both axes or one of them. In the example case when 6 people were added in the configuration the living room expanded by 2X2 blocks and when three people were added it increased by 1X2 modules. These logics can also be modified according to the designers preference.

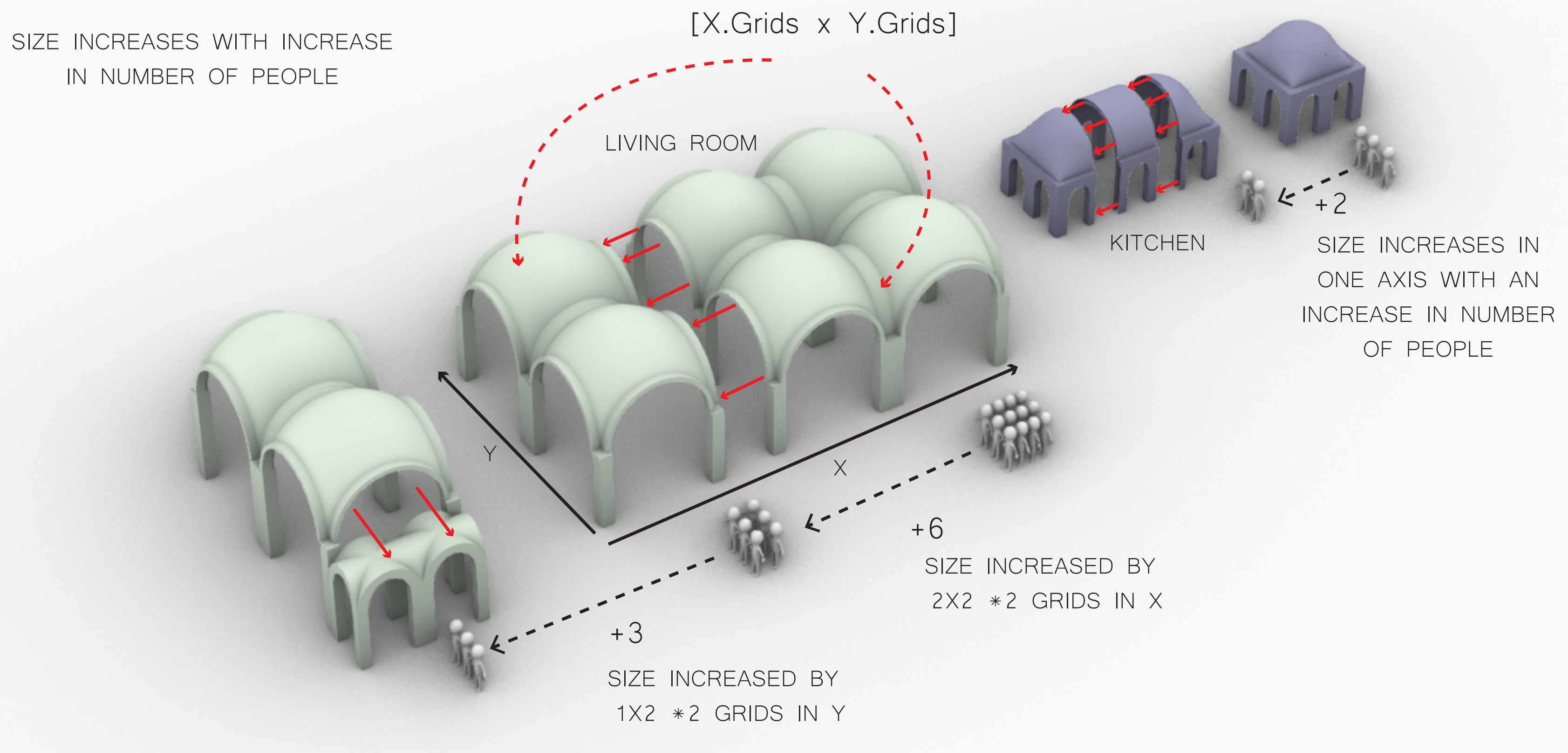


```
def riwaq_matrix (self):
    b = self.blocks_house( )
    c=b[0]
    d= list(self.matrix_generator_for_riwaq(c))
    e = [x+2 for x in d]
    return e
```



```
def matrix_generator_for_riwaq(self,variable_a):
        a= variable_a
        perfect_square = []
        y_grid = []

        for i in range(1, a+1):
            b = i*i
            if b <= a:
                perfect_square.append(b)

        c = math.sqrt(perfect_square[-1])
        d = a-perfect_square[-1]
        if d < c:
            y_grid.append(c)
        elif c < d < c*2:
            y_grid.append(c+1)
        else:
            y_grid.append(c+2)
        matrix = (int(c), int(y_grid[0]))

        if matrix[0]<=4:
             matrix_mod = (4,4)
        else:
             matrix_mod = matrix

        return matrix_mod
```

Rule 3: Generation of Riwaq



Rule 4: Generation of Incremental space

7. CODE AND ITS GEOMETRIC IMPLICATIONS

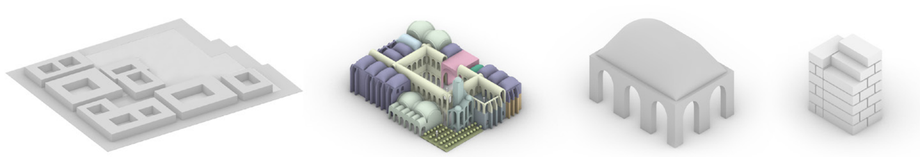
Rule 5 generates the tuples for the spaces designated as fixed spaces in the space program. The fixed spaces increase I number with an increase in the number of people as seen in the image on the right. The function in this case considers the number of people and divides the value by a designated number which indicates the count of the fixed spaces per the number of people . For example for every two people one bedroom is considered .If the number of people are odd then the number is changed to the next even number so that enough bedrooms are available for all .

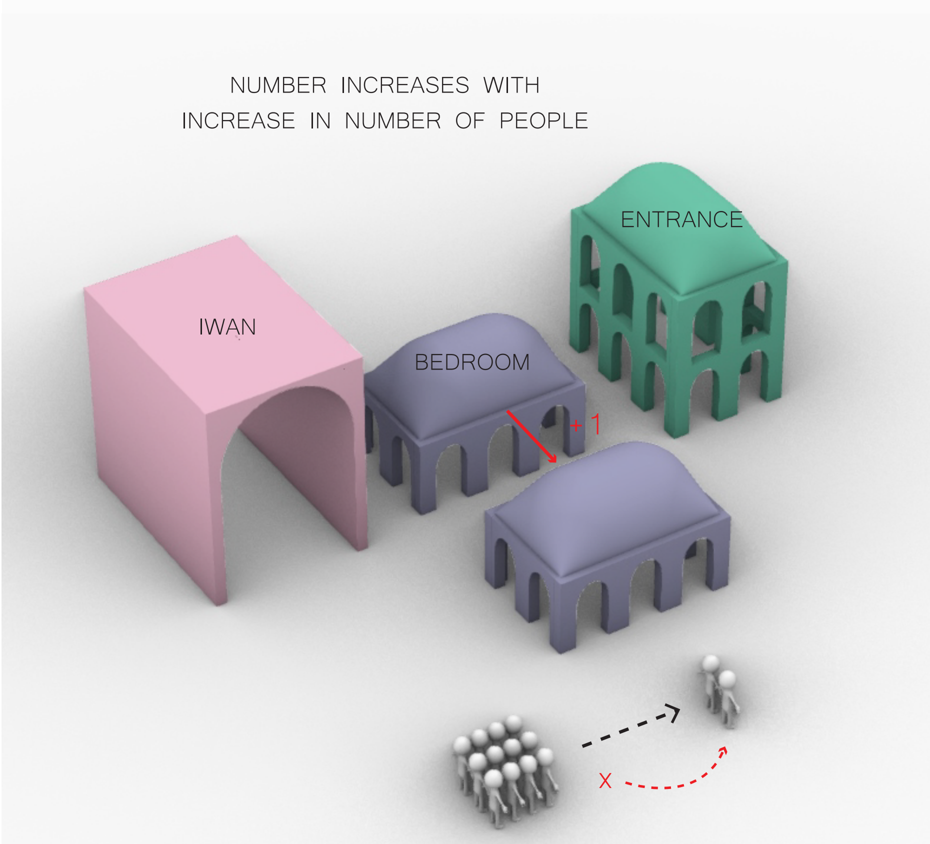
Rule 6 starts the space aggregation process. It checks for the number of sides available for connection on each direction of the connecting passage of the courtyard also called as riwaq . The length of riwaq is divided by the grid size to indicate the number of sides available for connection.
The final output for rule 6 is a list which indicates the sides available in all the 4 directions. Another feature is added to the code which is specially useful while working with the combined family dwelling type of configuration.This feature allows the user to eliminate sides of the connecting passage for connection. This is done when the configuration needs to be worked out for houses which are connected on one or more sides with others. (See the diagram on page 14)

Rule 7 allows the user to generate one or more passages along the courtyard till the end of bounding box for increasing the number of possible connection sides. This can be specified in a similar way as rule 6 where either 1 or 0 indicates the existence of a passage in that direction.

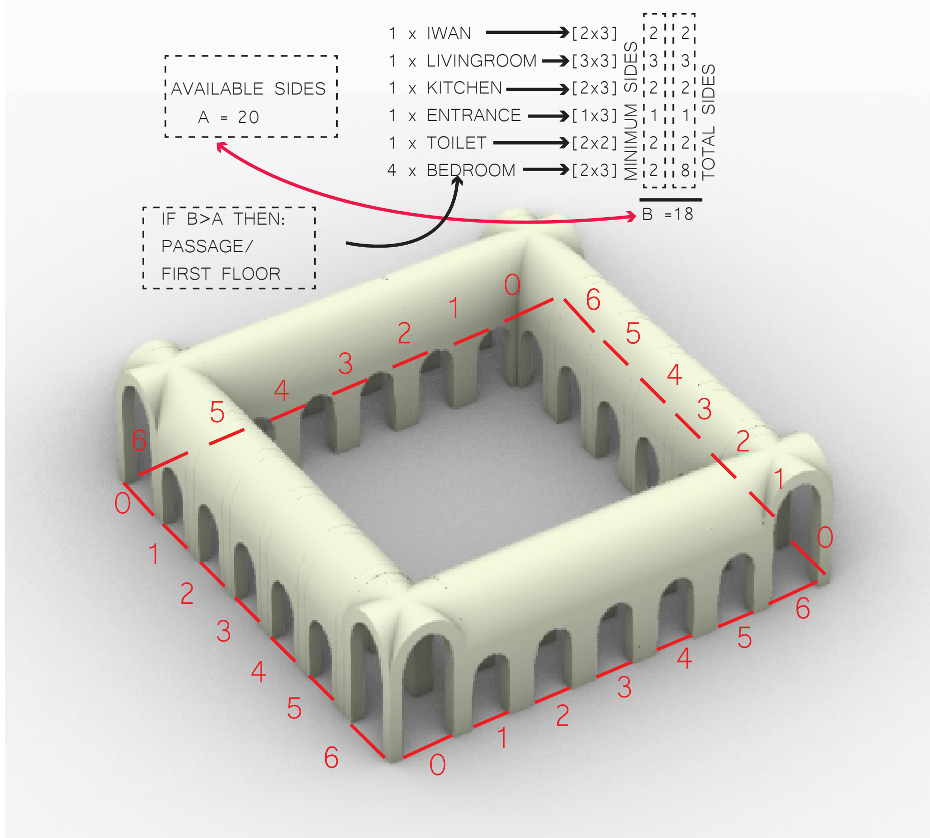
```
⑨ def Final_list_of_modules_of_spaces (self):
   Generate a final list of tuples indicating the number of modules in
   (Xaxis, Yaxis) for all the space arranged as per priority list

⑩ def sides_available_for_connection (self):
   Generate a list of available sides (length of each side of riwaq/
   module size) in each direction.
```

```
def Final_list_of_modules_of_spaces (self):
 # Priority List : Entrance,Staircase,Iwan, Living Room,Kitchen,Farm,Toilets,Shower,Bedroom
 # 0 = South , 1 = East , 2= North , 3= West
 toilet_modules= self.matrix_generator_for_toilets( )
 bedroom_modules = self.Bedroom_matrix( )
 Final_list_of_modules_of_spaces=[(1,3),(2,3),(3,3)]
 Living_Kitchen_Toilet = self.Incremental_space_matrix( )
 for i in Living_Kitchen_Toilet :
      Final_list_of_modules_of_spaces.append(i)
 for i in toilet_modules:
      Final_list_of_modules_of_spaces.insert(-1,i)
 for i in bedroom_modules :
      Final_list_of_modules_of_spaces.append(i)

 return Final_list_of_modules_of_spaces
```

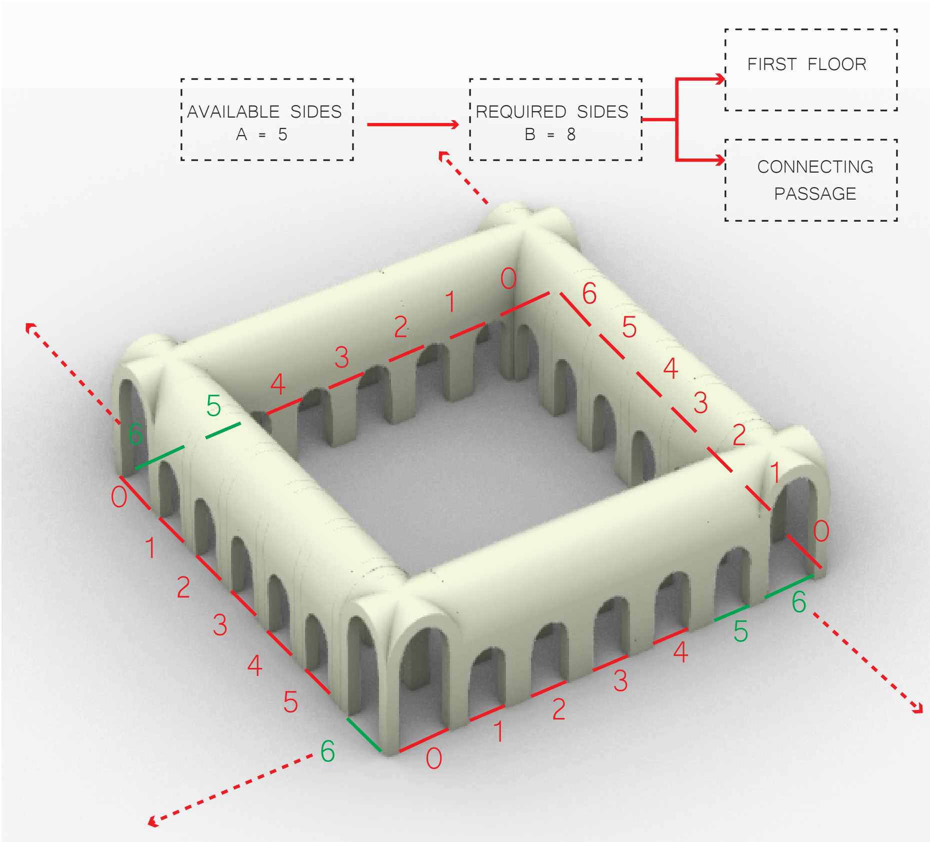




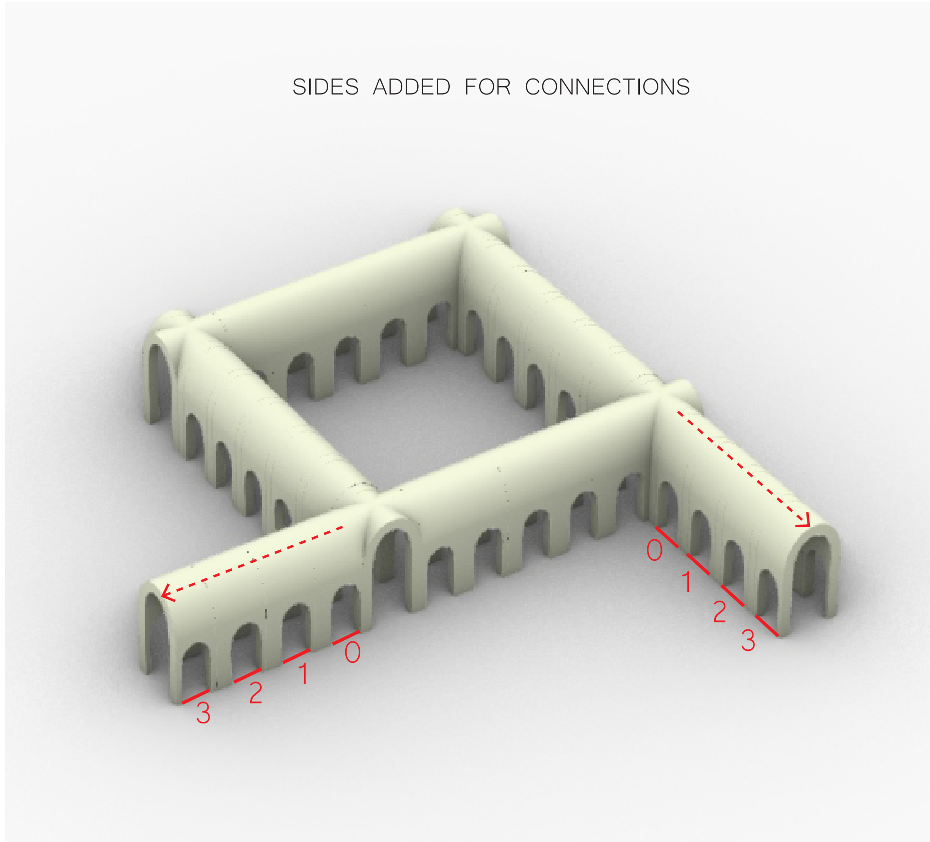
Rule 5: Generation of fixed module



Rule 6: Analysing available spaces



Rule 6: Analysing available spaces



Rule 7: Passage generation

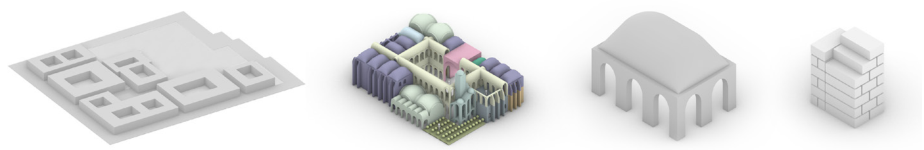
7. CODE AND ITS GEOMETRIC IMPLICATIONS

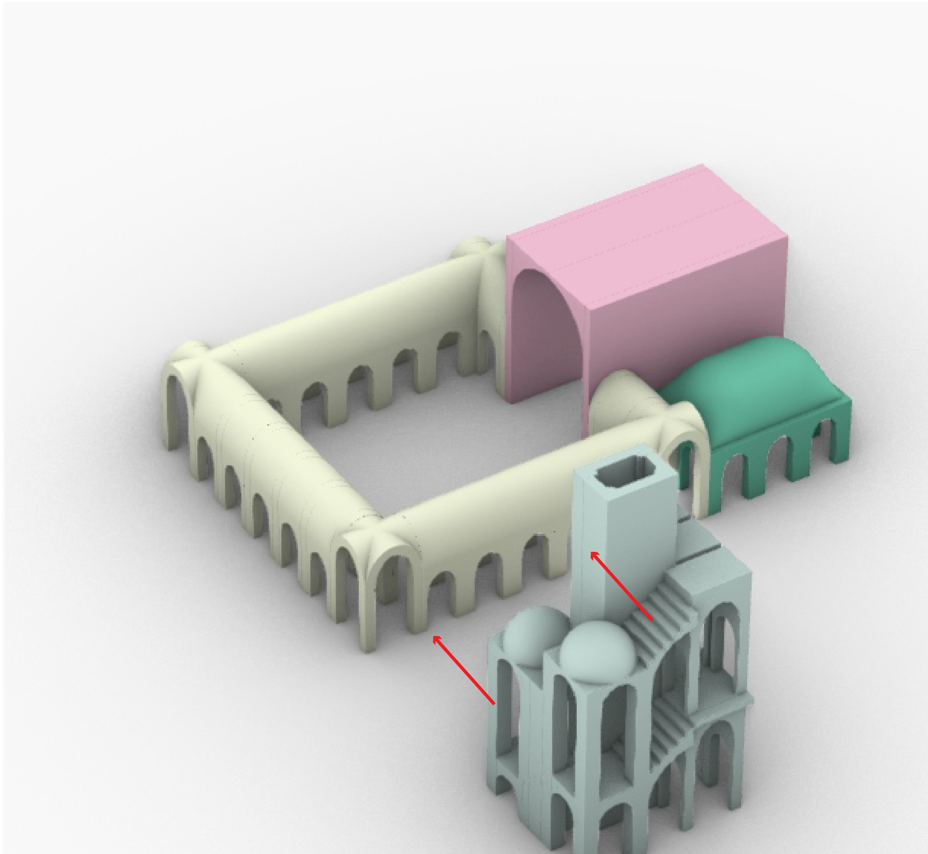
Rule 8 and Rule 9 start with the placement of the generated spaces along the connecting passage according to the priority number. The code in this function is basically a loop inside a loop which iterates over the available spaces with the minimum number of spaces required for connections. It also iterates over all the sides of the Riwaq based on the priority number till it manages to find a place for the space to fit in . If it doesn't find a place for the space to fit in the ground floor it creates a list of spaces (Tuples that indicate the number of grids in X and Y axis) that are placed on the first floor.

Rule 10 talks about the intuitive decision making which a designer makes while thinking about floor plans. For example if the previous steps are giving a 2X2 module on a side where there are mostly 2X3 modules then it is a choice of a designer to provide more space to the 2X2 module to have a common edge on the boundary . These logics are based on choice and cannot be hard coded hence there should be a user input required for all these instances of intuitive logic. This rule is not code in the python script yet .

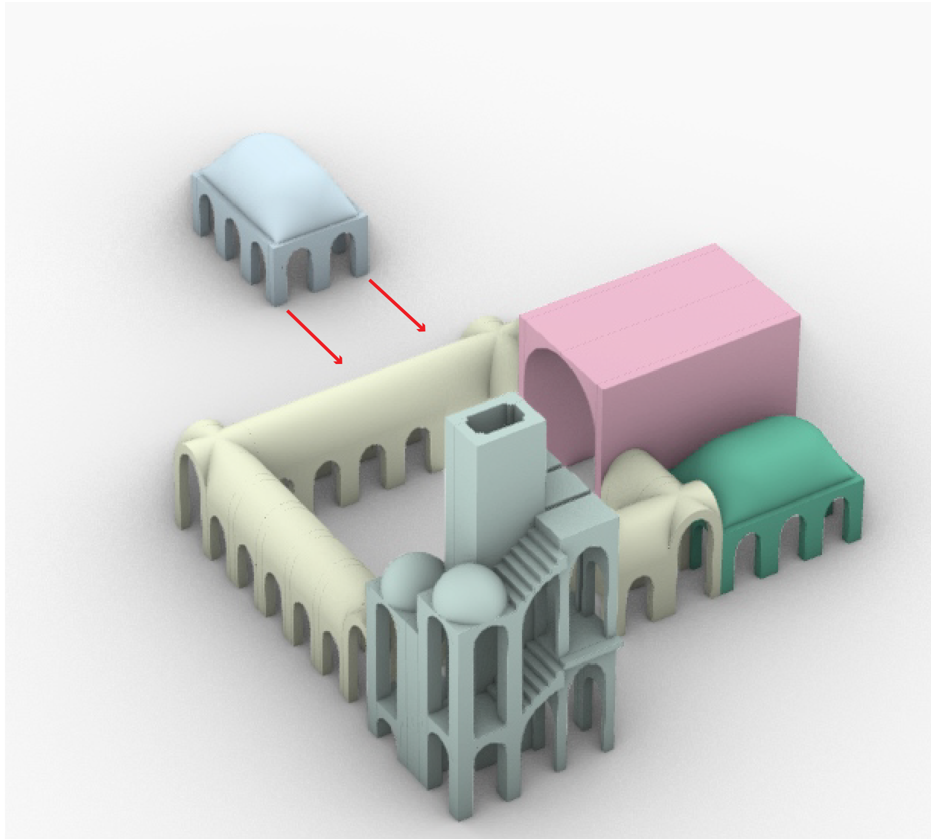
```
⑨ def Final_list_of_modules_of_spaces (self):
   Generate a final list of tuples indicating the number of modules in
   (Xaxis, Yaxis) for all the space arranged as per priority list

⑩ def sides_available_for_connection (self):
   Generate a list of available sides (length of each side of riwaq/
   module size) in each direction.
```

```
def Sorting_into_directions_as_per_priority(self):
     # Priority List : Entrance,Staircase,Iwan, Living Room,Kitchen,Farm,Toilets,Shower,Bedroom
     # 0 = South , 1 = East , 2= North , 3= West
     priority_list_all_spaces = self.priority_list_definition( )
     Direction_Lengths = self.sides_available_for_connection( )
     lengths_spaces= self.smallest_side_connection_per_function( )
     final_list_of_modules = self.Final_list_of_modules_of_spaces( )
     Result_of_lists = [[], [], [], []]
     Remainder_spaces = []
     for i in range(len(lengths_spaces)):
          inputValueCurrentlyBeingRead = lengths_spaces[i]
          pList = priority_list_all_spaces[i]
          for j in range(len(pList)):
               selectedPriorityValue = pList[j]
               spaceAvailableInSelectedLocation = Direction_Lengths[selectedPriorityValue]
               if spaceAvailableInSelectedLocation >= inputValueCurrentlyBeingRead:
                    Direction_Lengths[selectedPriorityValue] = spaceAvailableInSelectedLocation - \
                         inputValueCurrentlyBeingRead
                    Result_of_lists[selectedPriorityValue].append(
                         final_list_of_modules[i])
                    break
               elif j == (len(pList))-1:
                   # print("for loop complete")
                    Remainder_spaces.append(final_list_of_modules[i])
     East_Direction = Result_of_lists[1]
     North_Direction = Result_of_lists[2]
     West_Direction = Result_of_lists[3]
     South_Direction = Result_of_lists[0]
          return ["SENW",East_Direction,North_Direction,West_Direction,South_Direction],["Remainder_
spaces".Remainder spaces]
```

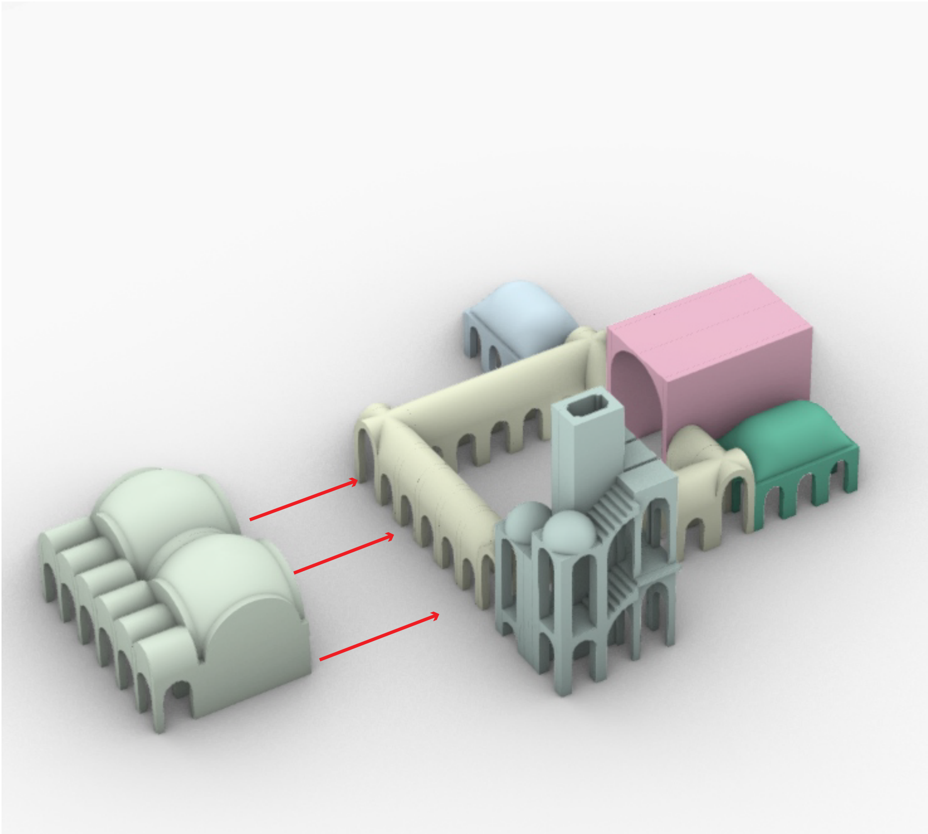




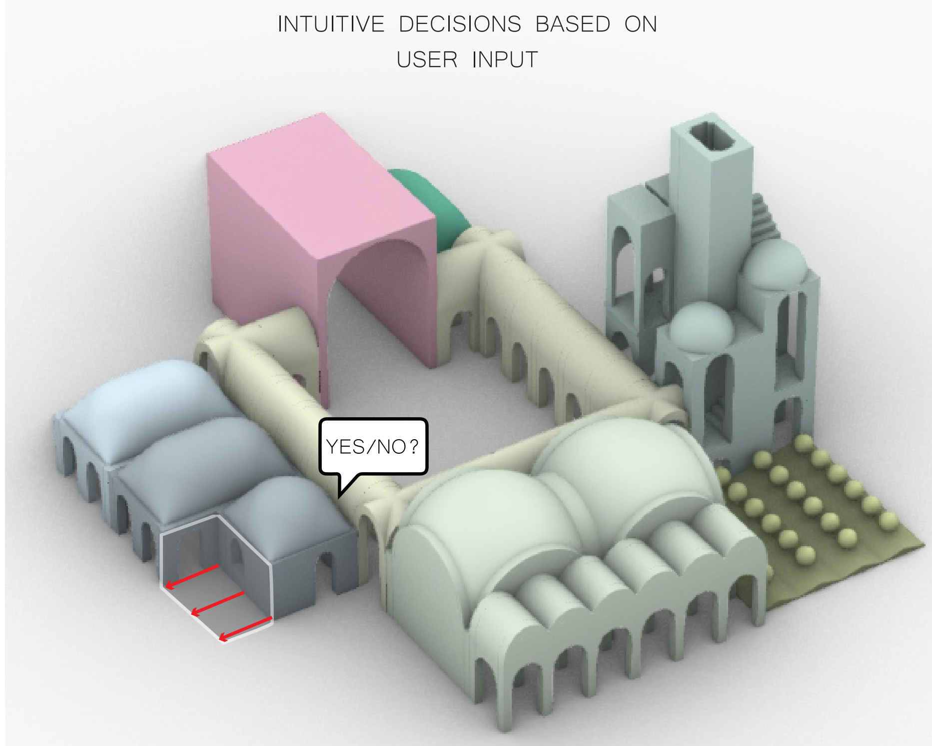
Rule 8: Staircase placement



Rule 9: Arranging rest of spaces as per Rule1



Rule 9: Arranging rest of spaces as per Rule1



Rule 10: Intuitive decision making

RECAP OF ALL THE RULES FOR GENERATING THE COMMUNAL HOUSE FOR THE CASE

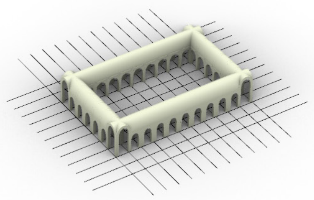


Figure27: Step 00 Generating the courtyard

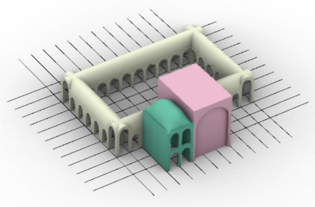


Figure28: Step 01 Adding the Entrance and the Iwan

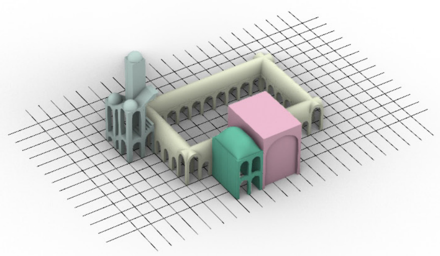


Figure29: Step 02 Adding the staircase

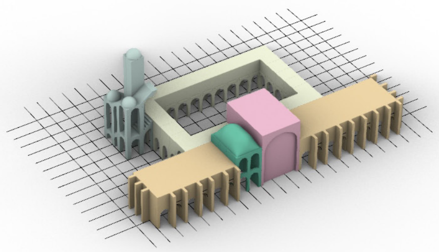


Figure30: Step 03 Generating the hybrid spaces

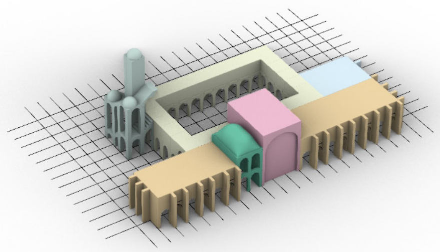


Figure31: Step 04 Adding the kitchen

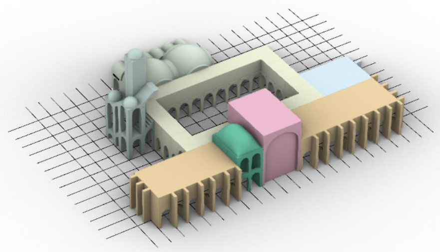


Figure32: Step 05 Adding the living room

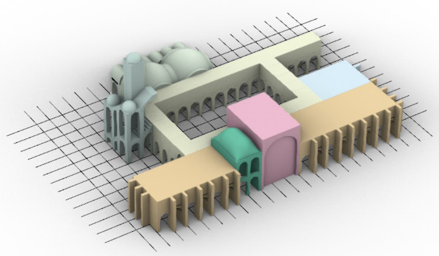


Figure33: Step 06 generating a passage

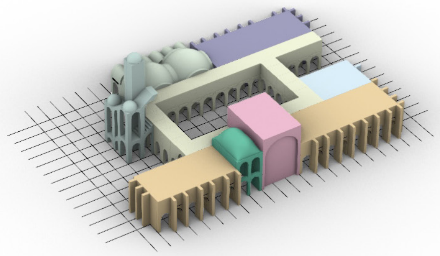


Figure34: Step 07 Adding the bedrooms

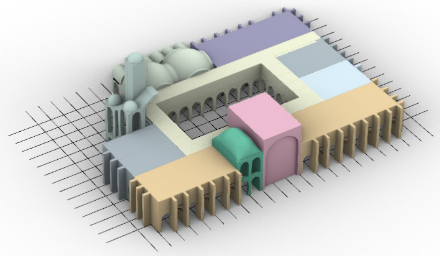


Figure35: Step 08 Adding toilets + showers

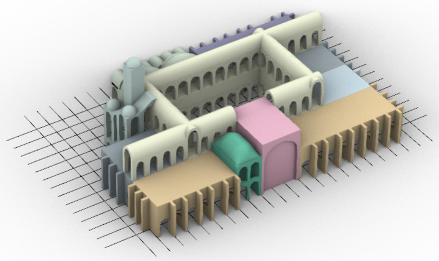


Figure36: Step 09 Generating first floor Riwaq

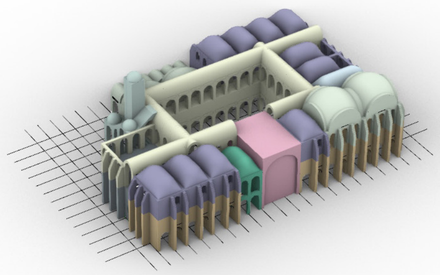


Figure37: Step 10 Adding the spaces on first floor using same logic

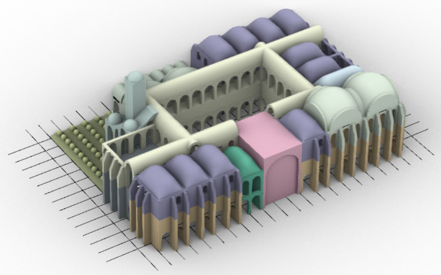
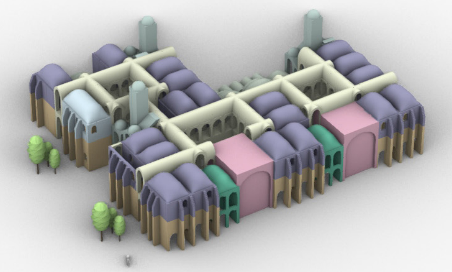


Figure38: Step 11 Generate Farms

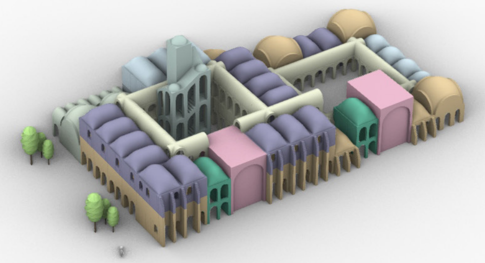
GENERATIVE DESIGN VARIANTS OF THE SAME HOUSE WHEN THE GAME IS PLAYED BY DIFFERENT PLAYERS

GENERATIVE DESIGN VARIANTS OF THE VARIOUS HOUSES ON THE SITE SHOWING THE POTENTIAL OF THE SYSTEM

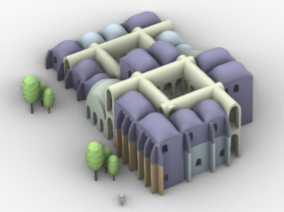
00



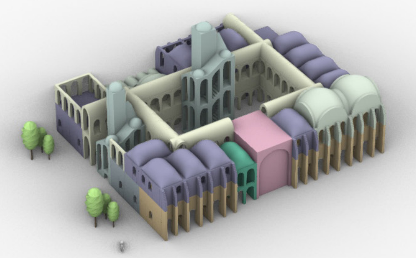
01



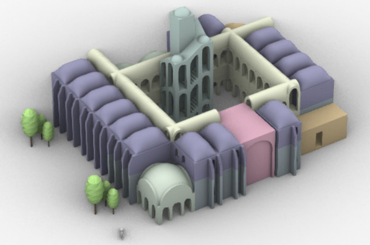
02



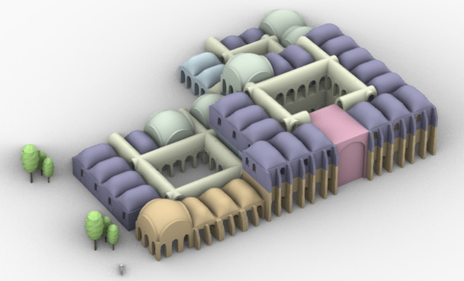
03



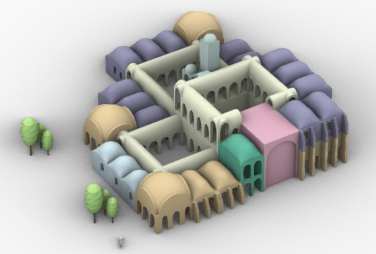
04



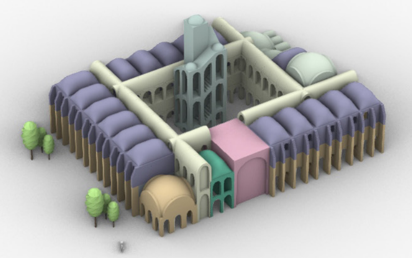
05



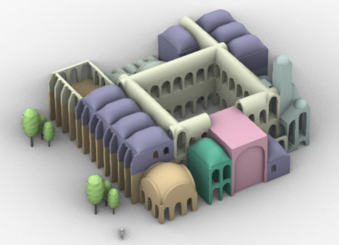
06



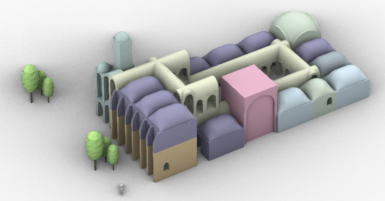
07



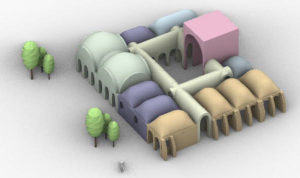
08



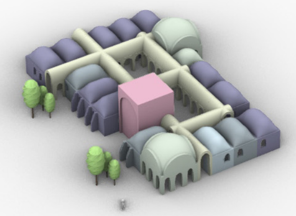
09



10



11



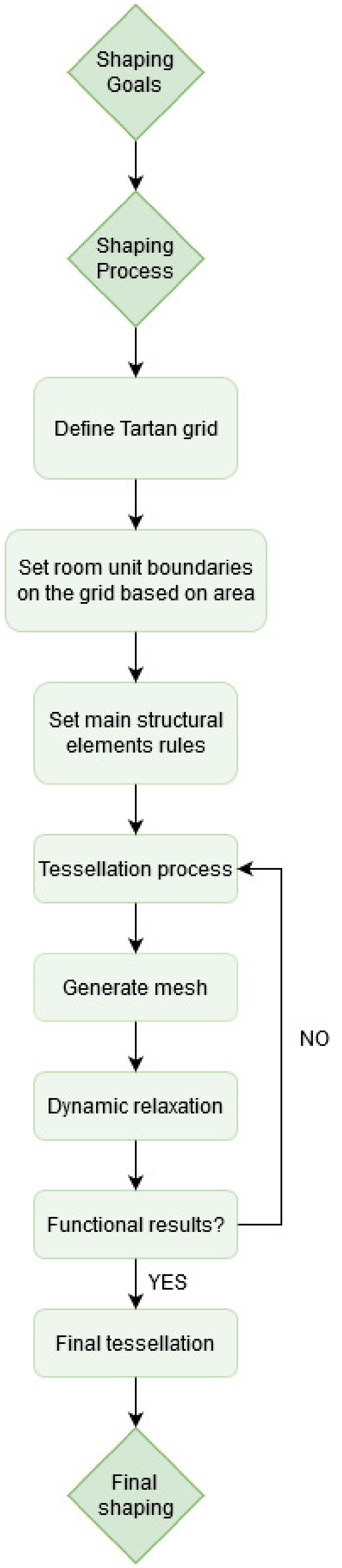


Figure39: Flowchart of the shaping methodology and process

03 SHAPING

SHAPING GOALS

The same systematic approach is followed during the shaping process and the unit development. The main component that connects all the focus points and scales of the process is the tartan grid and the measurement units used, which dictate also the shaping and dimensions. The tartan grid is used in the end as a reference guideline for the urban aggregation, module development and construction. This grid is 1500*1500mm. All the rest of the units used are subdivisions and multiples of each other and correspond at the same time to the ergonomics, the construction and workability requirements, like the brick size and opening sizes.

The goal for the form-finding is to create modular and easily repeatable units that can be related to each other to create various adjacencies and connections as part of a flexible and open-ended system. Important aspect of the entire process is also to enable a certain level of participatory design with the user's preferences to be able to influence the final design. In this case, as it will be explained later, certain aspects of the shaping process are left to the user to decide out of a set of options, which again, thanks to modularity, don't interfere with the general form-finding and room connections.

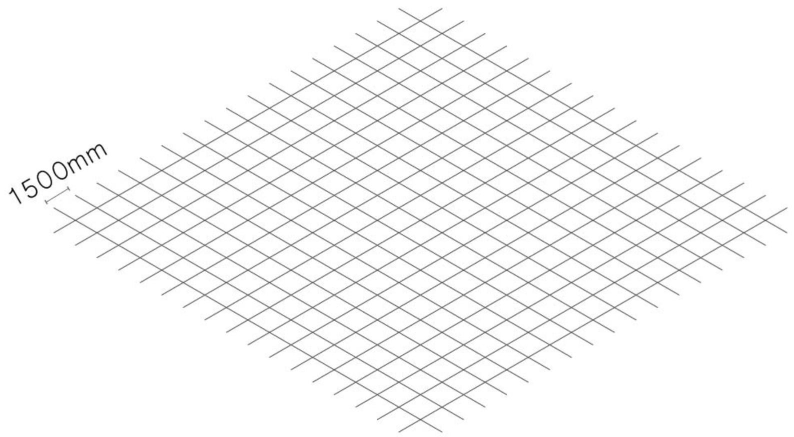


Figure40: Tartan grid

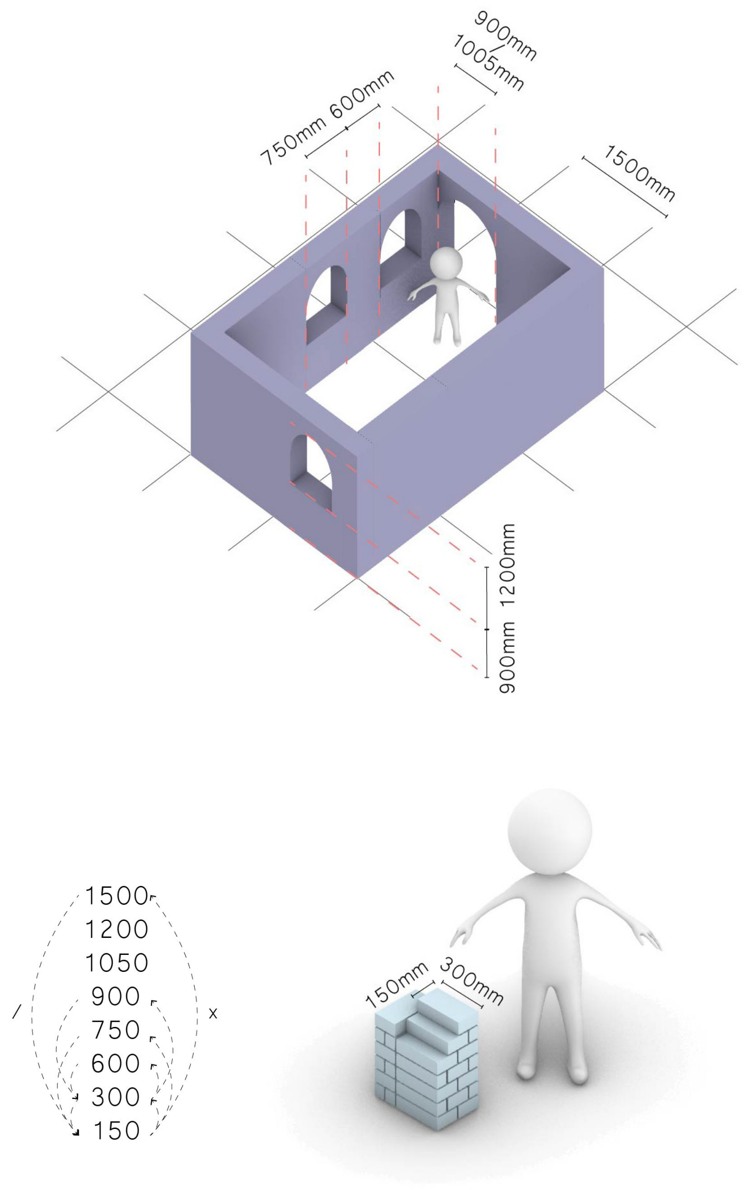
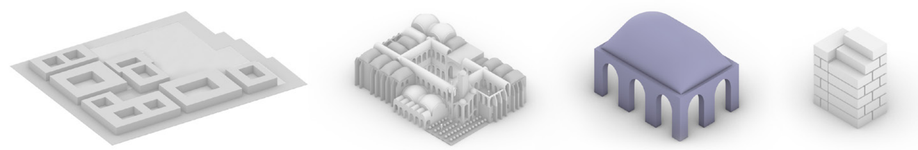


Figure41: Relations of dimension units and connection to ergonomics, workability and construction



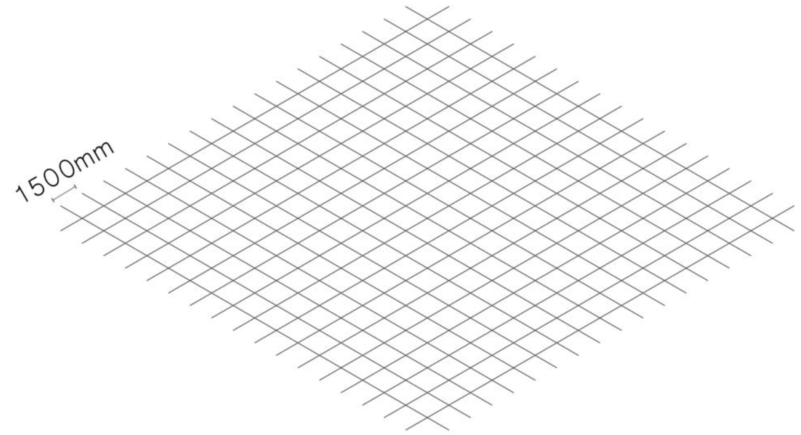
SHAPING PROCESS

The room types identified by the configuration process relate as well to the tartan grid as in number of cells per type. As a result the room types are distinguished as follows:

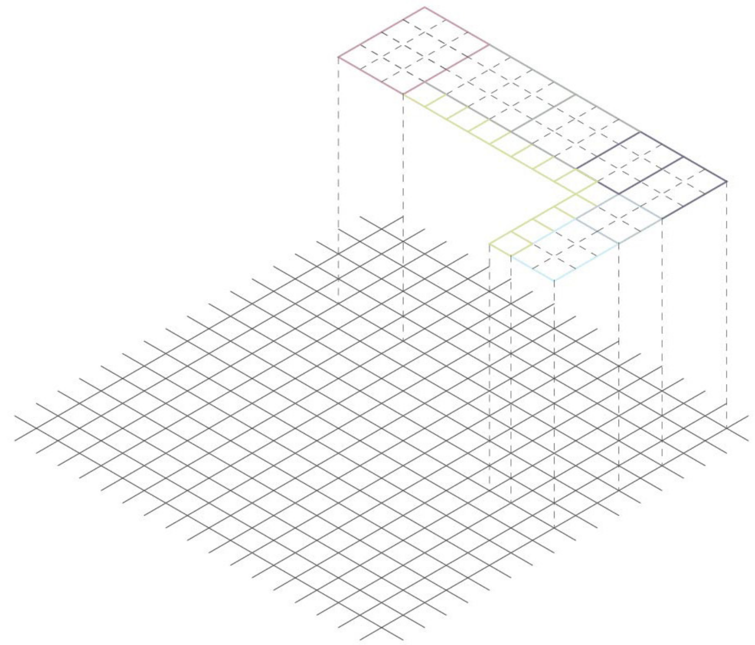
-Riwaq: 1x1
-Bedroom / Kitchen: 2x3
-Toilet/Shower: 2x2 or 2x3, depending on family size
-Living room: 3x3 or 4x3, depending on family size
-Iwan: 3x4

From those, the riwaq, bedroom, kitchen and iwan are fixed spaces, while the toilet/shower and living room are incremented spaces, meaning that they have different sizes, while the other ones increase in number, if needed. This modularity is reflected on their form-finding process as well, where both the wall and roof types should follow the same logic and flexibility to adapt and connect to each other.

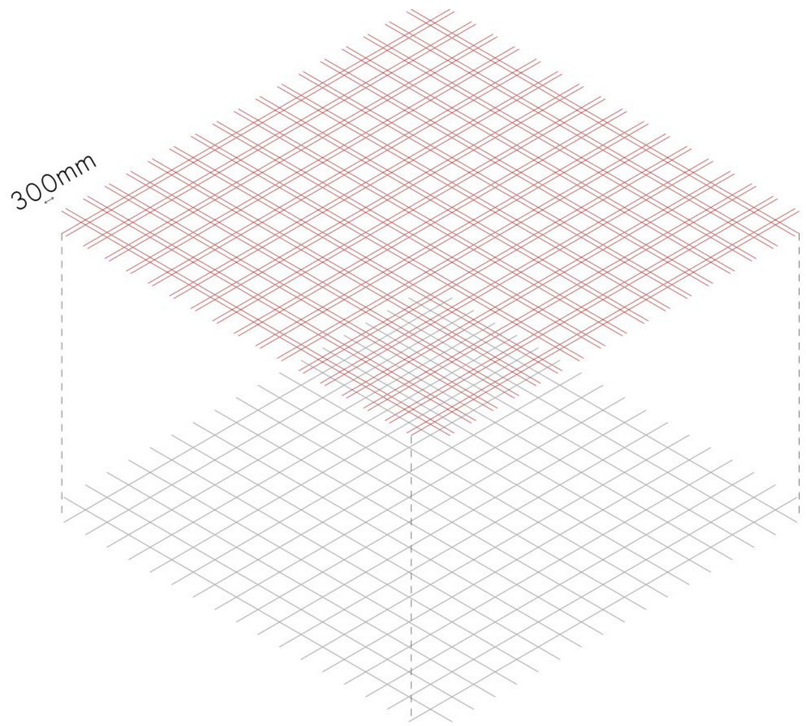
These room types and cell sizes are defined by the program requirements for the area needed for each space. For each space, this area is then projected on the tartan grid and the adjacency and connection between the spaces are defined. For each space, first the walls and columns are determined, in order to proceed to the roof shaping. For this, a secondary grid is used as a guideline for the development of the units. This originates from the fact that one of the intentions is for the rooms to work individually as a structure, because it enables a certain level of flexibility in combining the rooms together with more freedom. This secondary grid is an offset of 150mm on each side of the tartan grid, which relates to the column size, which is close to 300mm.



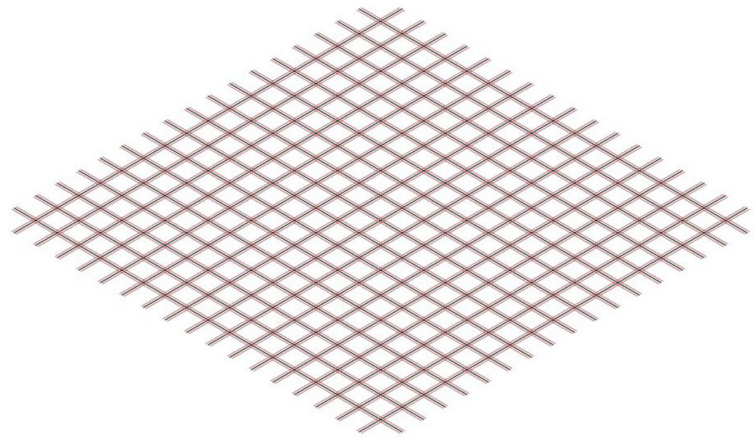
Tartan grid 1500*1500mm



Alignment of modular room types as inner floor areas on the grid

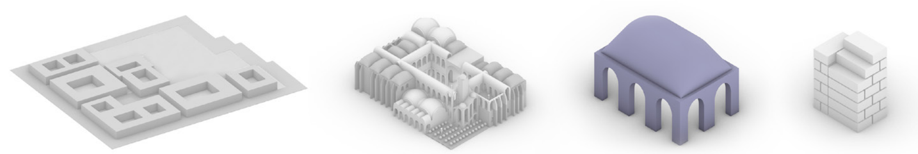


Secondary reference grid offset 150mm on each side of the tartan grid



Alignment of modular room types as inner floor areas on the grid

Figure42: Double reference grid



With this base, a systematic approach is also used in defining the main structural elements. The edge elements are centered to the secondary grid, which means that they are eccentric to the tartan grid, in order to avoid overlapping. To avoid further conflicts and to try to achieve some efficiency in the placement of the structural elements, each unit is designed with cornered walls in a diagonal symmetry logic. Each room type contains in the end the information of both the required area as in grid cells and the type and main structural configuration needed for its stability. In this sense, the rooms can be placed next to each other by aligning similar elements next to each other, wall to wall and column to column. And since the dimensions function in a modular way, it is easier to avoid misalignment. Only exception is the iwan, which due to architectural reasons, it has the back wall entirely closed and a full opening to the courtyard in front, while the sides leave only one opening connection to the adjacent riwaq, but even in this case, the dimensions follow the same modularity logic and grid base.

The sequence which is then followed to place the rooms is by starting on one corner, for example, and then placing the adjacent rooms next to each other by other mirroring it or leaving it as it is based on wall to wall and column to column logic. By doing so, a basic minimum main structure is predefined for each room type to ensure its independent stability. At this point, the automatic generation of the room types as in wall structures is finished and the rest of the openings is left to the user's preferences. The user has, therefore, the freedom to choose from a kit of parts, whether they would prefer to have an opening as in door or window or to close it off entirely with a wall strucutre. In this sense, different variations of the same room type can occur and possible connections between types, for example creating a door or window opening between two adjacent kitchens in the large communal dwellings.

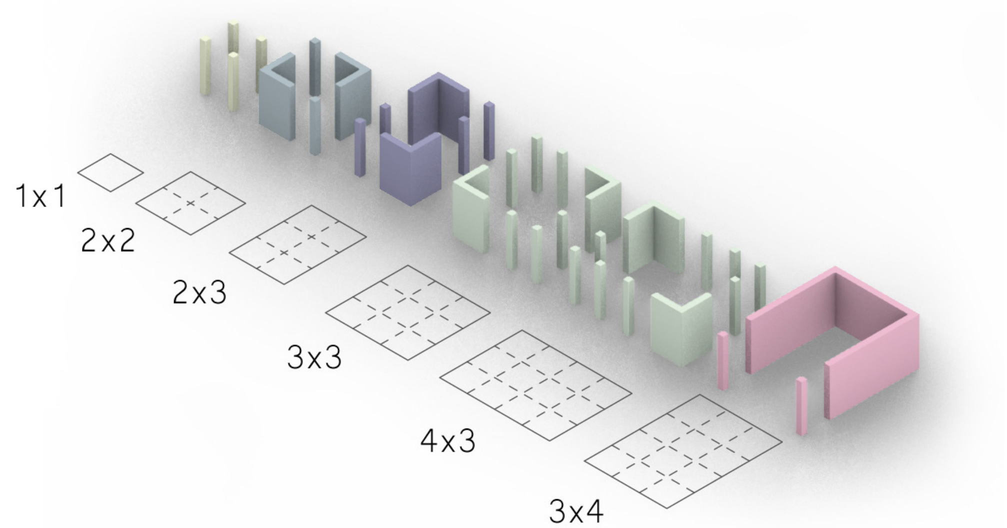


Figure43: Main structural element configuration per modular type size and structural independency logic

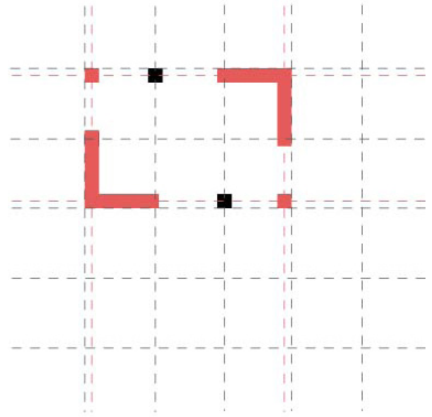
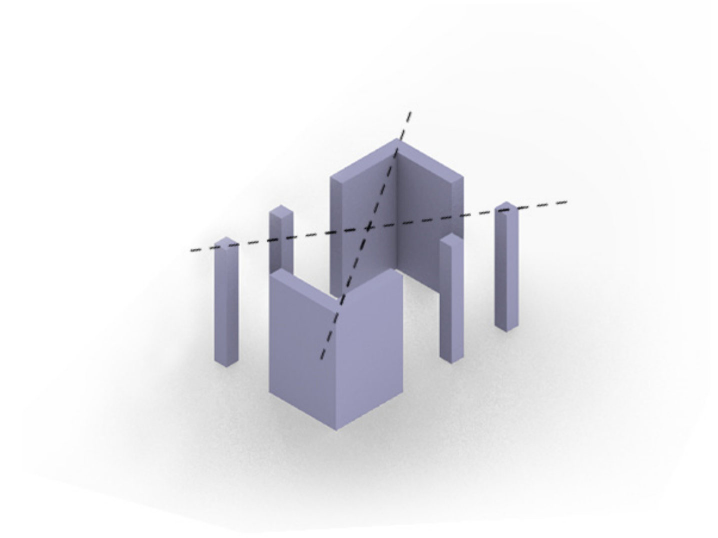


Figure44: Alignment of edge structural elements to secondary grid axes (center offset to the interior)



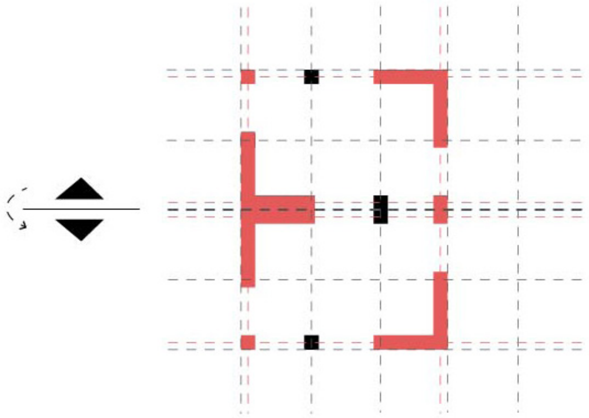


Figure45: Mirroring and alignment per structural element

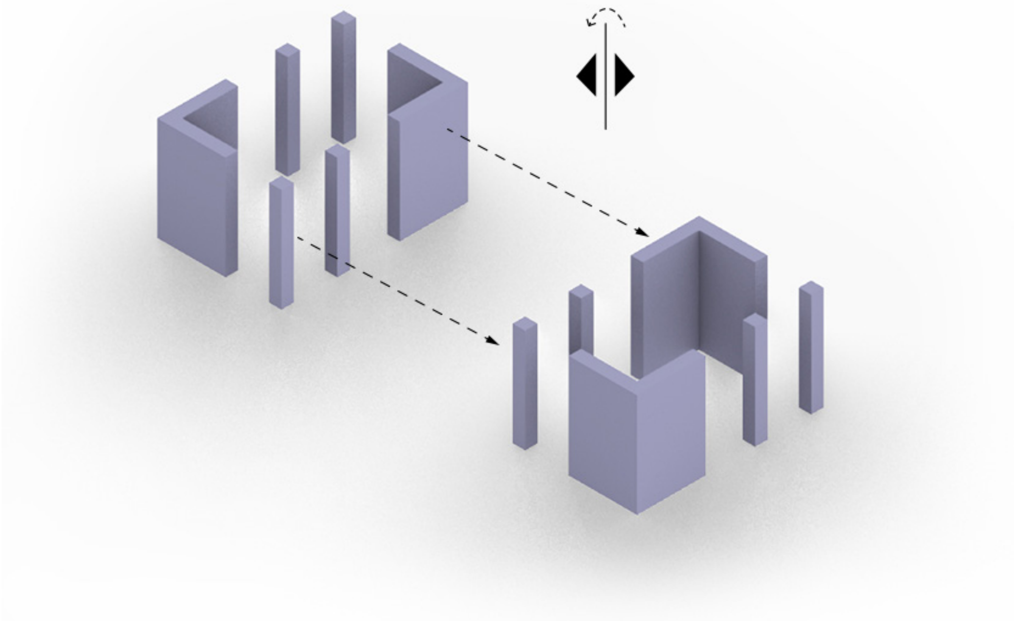
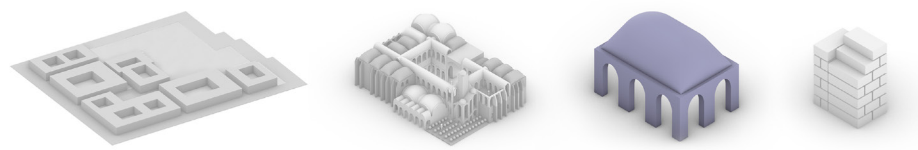


Figure46: Diagonal symmetry, mirroring and alignment per structural element



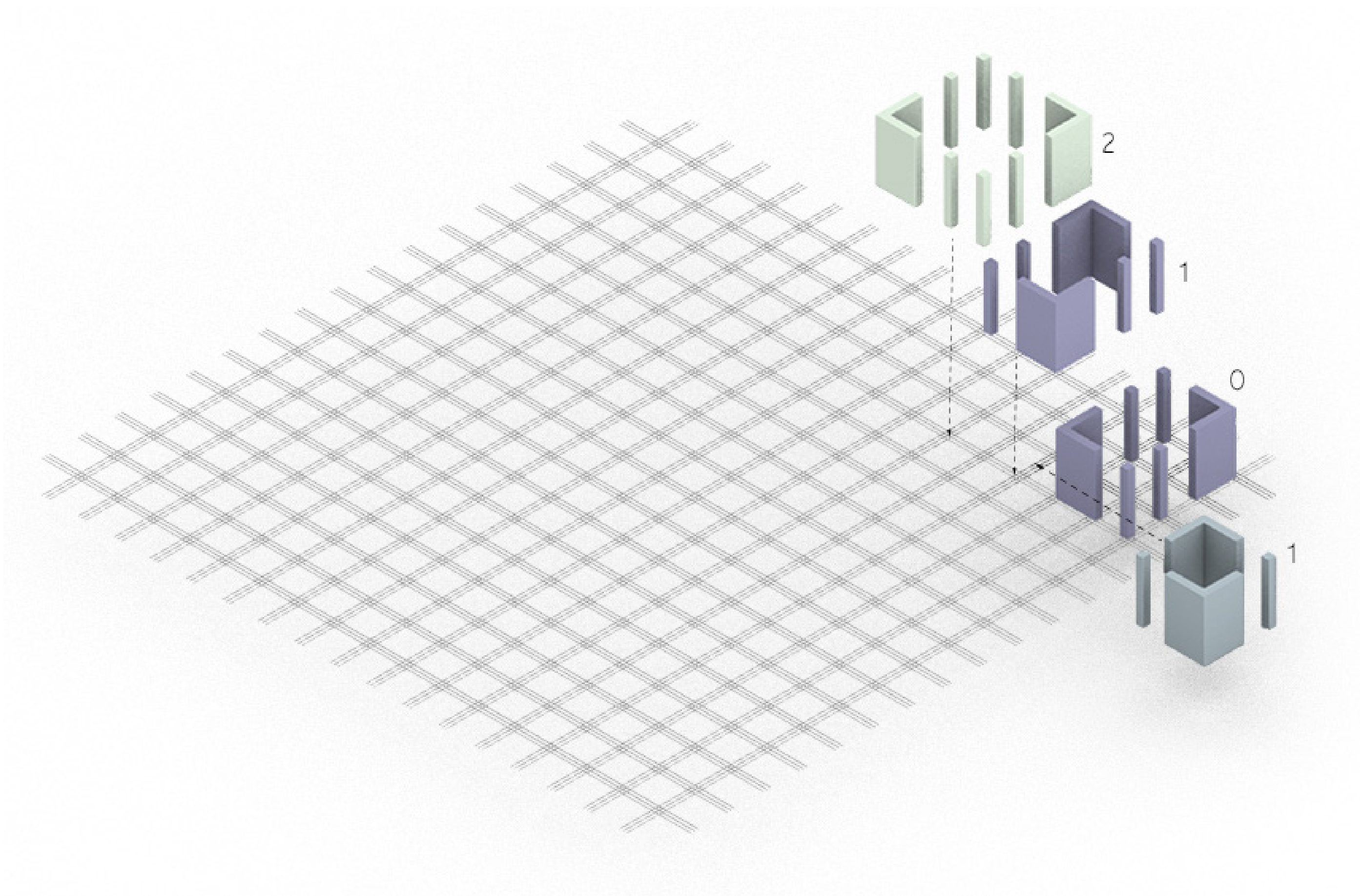


Figure47: Sequence of room type placement on the grid

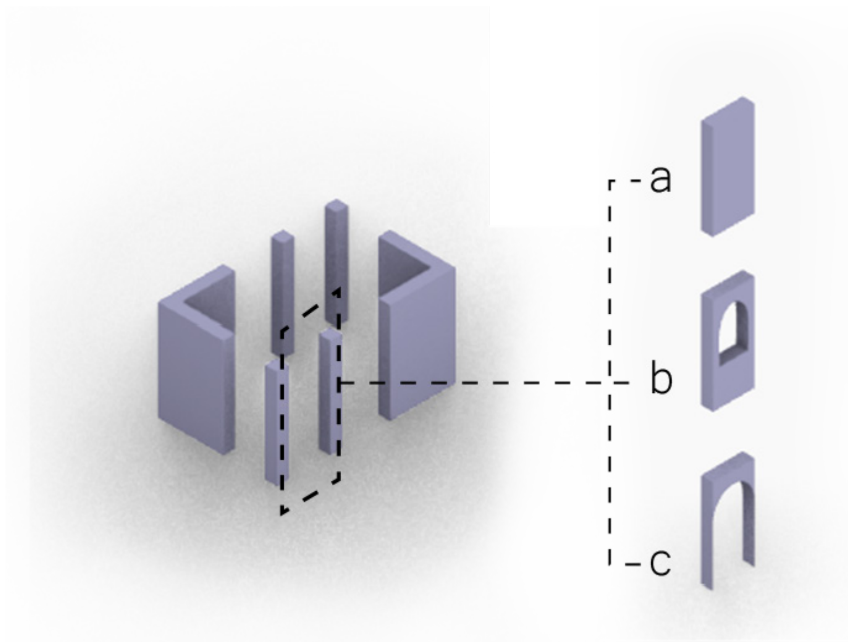


Figure49: User's preference and freedom of choice for the rest of the opening and construction elements based on the room needs

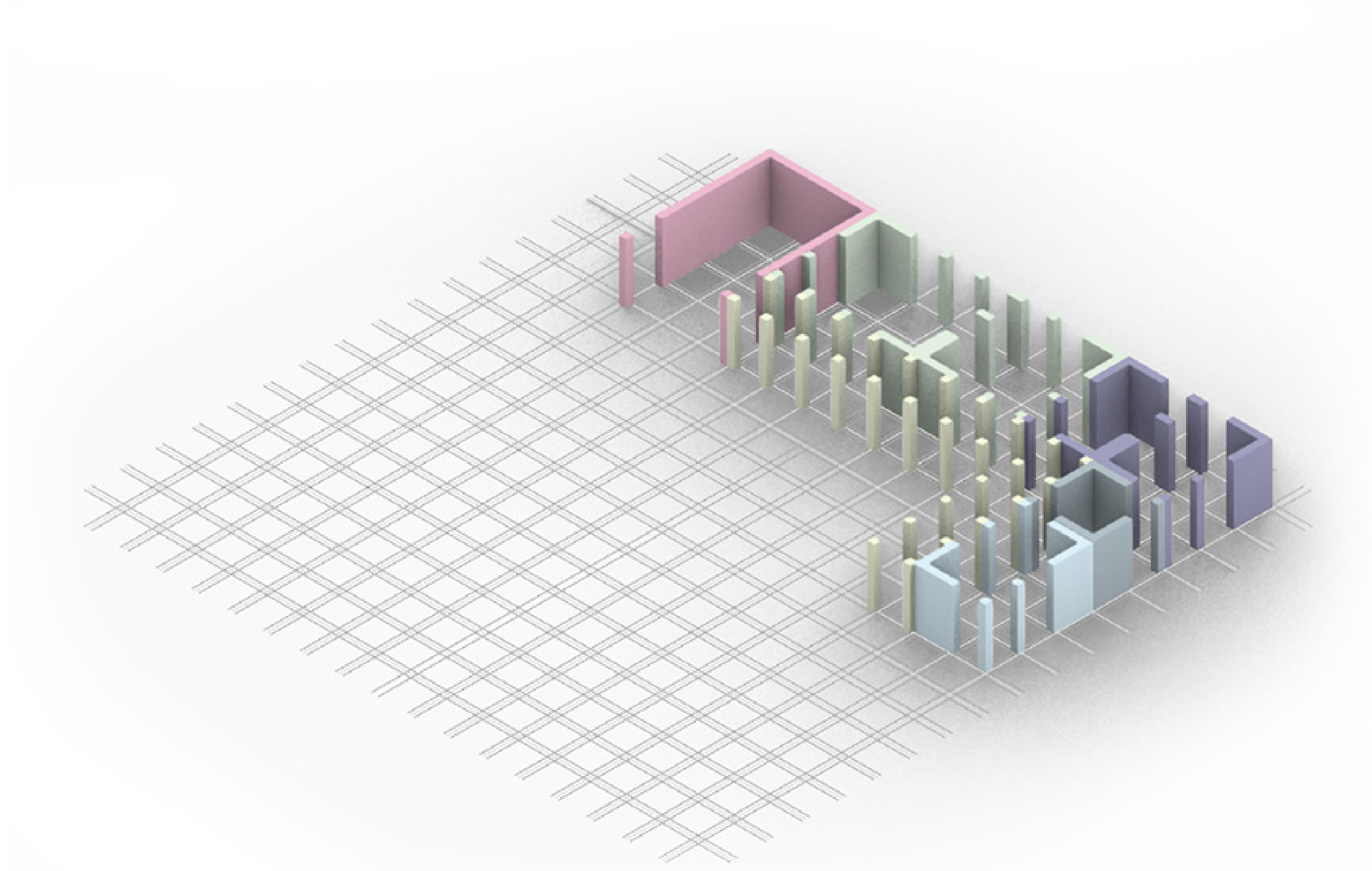


Figure48: FInal main structure configuration in relation to room placement

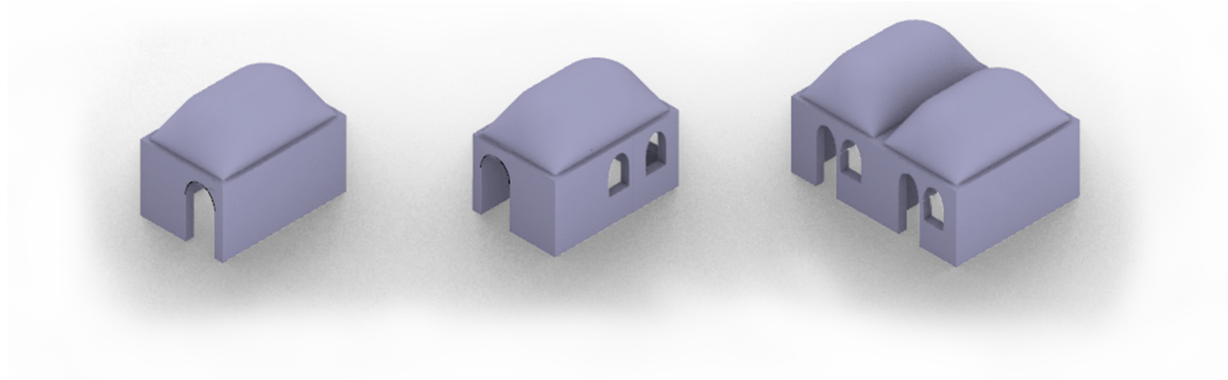
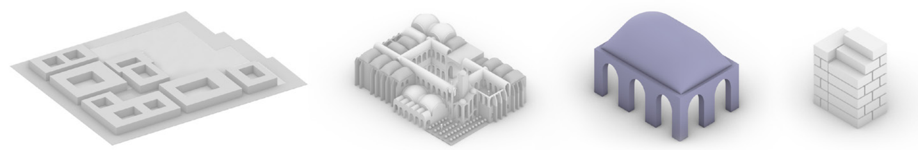


Figure50: Different user's choices lead to different room unit results and a variety of results and room connections



FIRST TESSELLATION APPROACH

In the initial form-finding process, the shapes were tested with having just columns as a base structure, to be able to get a better impression of the roof shaping and the overall volume. This was also done due to the fact that the walls provided more limitations during the dynamic relaxation process using the Kangaroo plugin in Grasshopper, because they were set as anchor elements, which would not allow a desired relaxed form and added a certain level of stiffness during the relaxation process.

For the tessellation, it was decided to study the different units independently from each other, since this is also the base of the entire system logic and not to align them next to each other from this stage. From the initial trials of tessellation, the goal was to create smooth surfaces, straight corners and with side openings. In the beginning, starting with the 2x3 room unit, a tessellation for the shape similar to cloister vault was tried as a desired roof shape, which would also be efficient to construct. Different mesh subdivisions were tested, using the Constant Quads Split, the Catmull-Clark or the Triangulation Subdivision. In the case of the other units, simpler vault types and domes were tested, since they are larger spaces and some, like the iwan and living room, were thought of as double height ones.

After defining the tessellation pattern for each type and subdividing the mesh, a dynamic relaxation process through Kangaroo was carried out each time to visualize the volume and shaping of the room. In this process, the columns were set as anchor edges, by selecting their vertices. By doing so, it is easier to see also the relation of height of roof to the height of doors and to ensure that it can be feasible for the spatial requirements of the room. To the next, some of the first tessellation attempts can be seen:

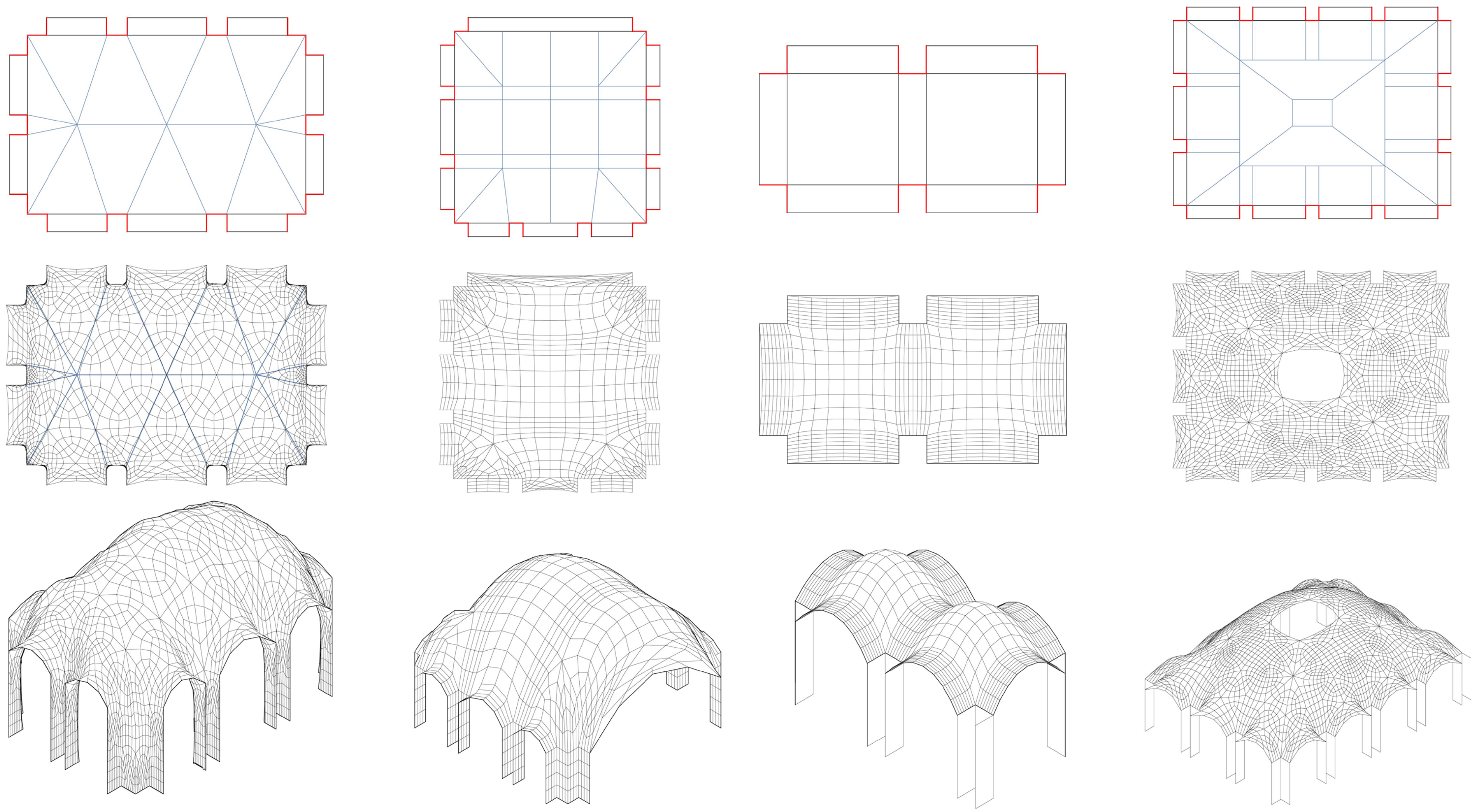


Figure51: First tessellation and mesh subdivision approach and dynamic relaxation

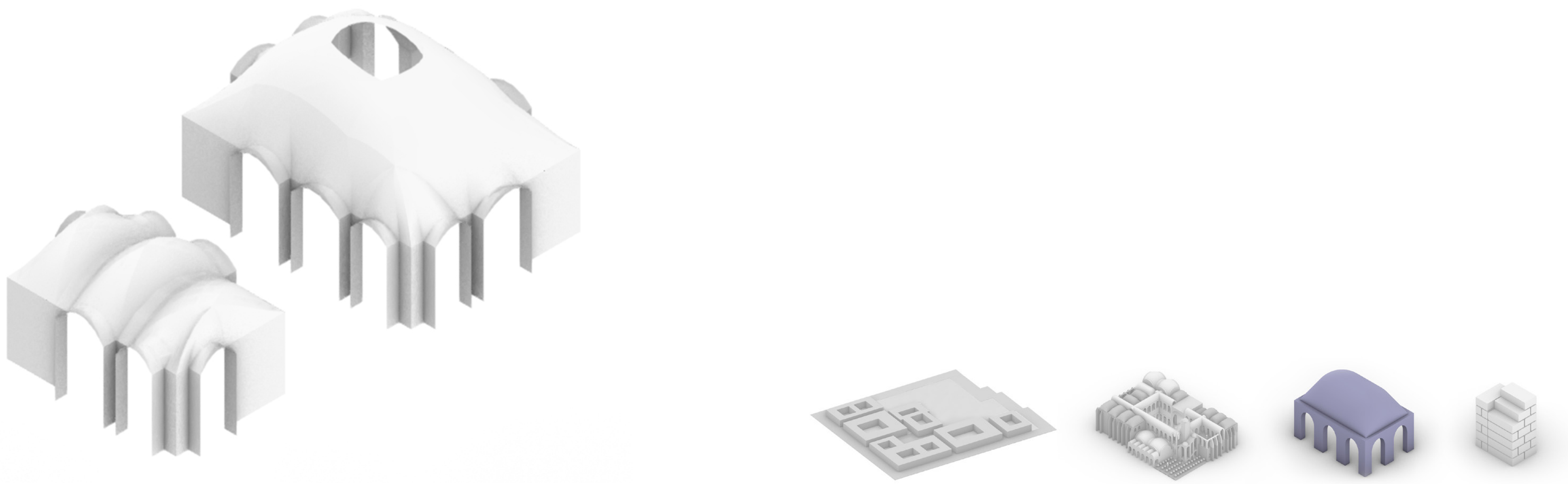


Figure52: Dynamic relaxation tryouts with adding walls as anchor elements

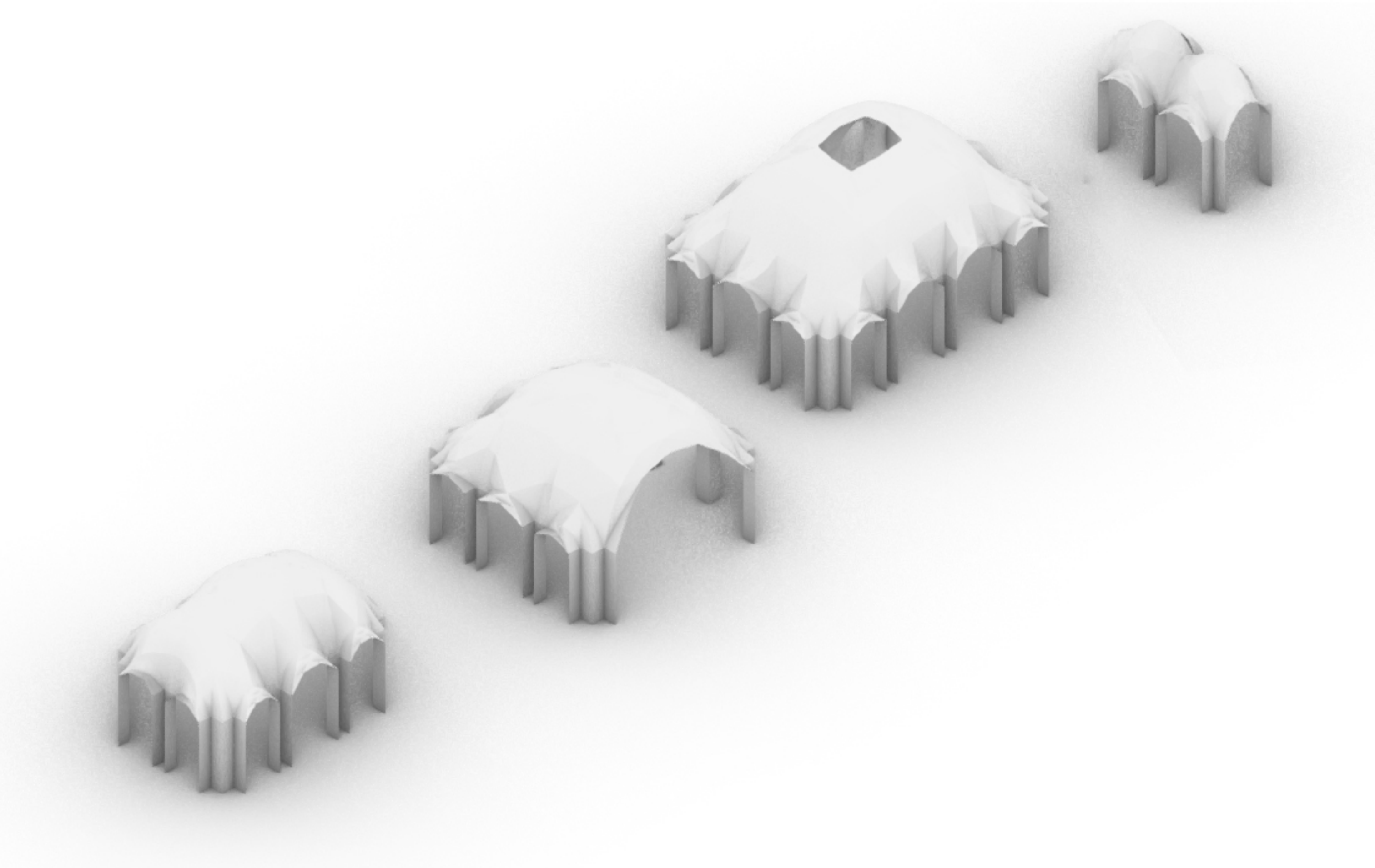


Figure53: Dynamically relaxed forms of the previous examples in the previous figure

Figure55: Dynamically relaxed forms as results of the different tessellations and mesh subdivisions iterations for the iwan unit

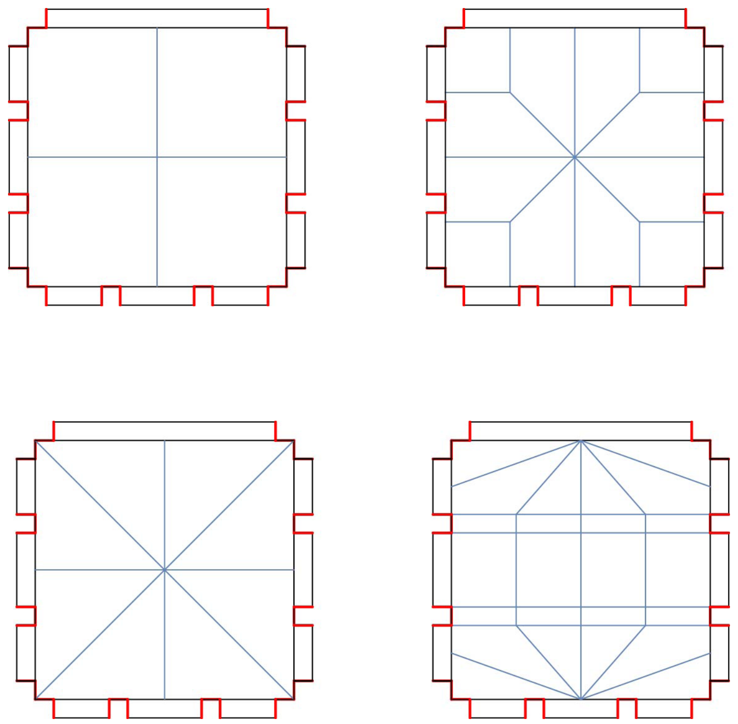
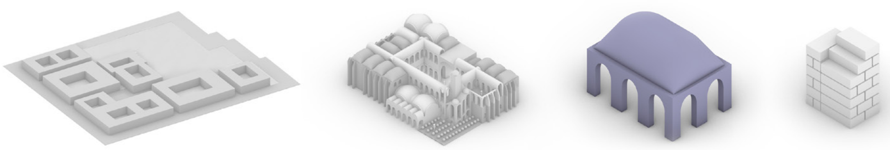


Figure54: Various tessellation pattern explorations for the iwan unit



FINAL TESSELLATION

For the final tessellation and form-finding, a simplification was chosen and a more modular approach, also related to the fact that some spaces are incremental in size. In order to be able to replicate and combine the spaces, this would have to apply also at the roof shaping logic, creating simpler vaulted and domed roofs that can easily adjust to different units and that can increase in size. Also, the mesh subdivision was simplified by using mostly the Triangulation or Quadratic Subdivision, in order to be later easier translated to construction elements. Here, some of the tessellation and mesh subdivision patterns can be seen, before further simplifying the shape to adjust to the roof construction constraints.

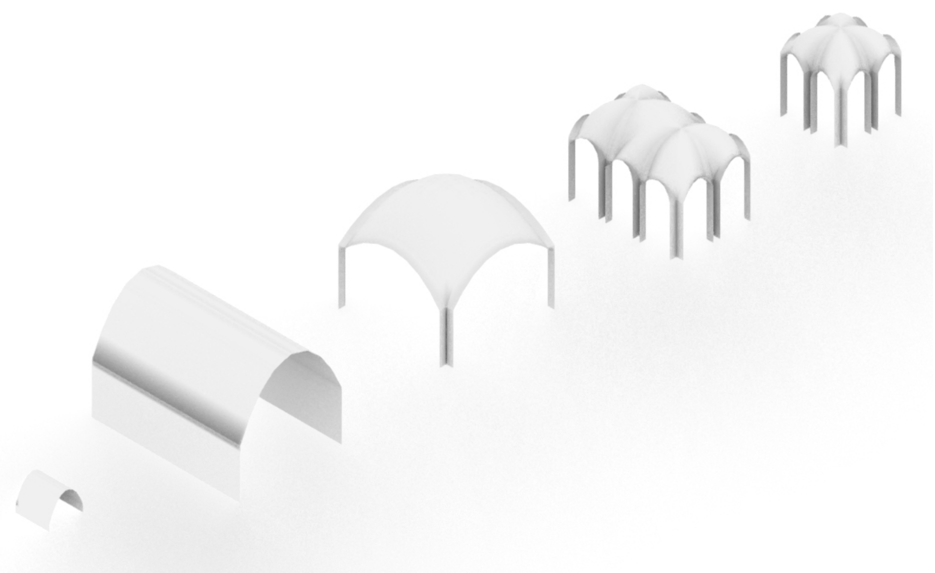


Figure56: Dynamically relaxed forms as results of the later stage of simplified tessellations and mesh subdivisions

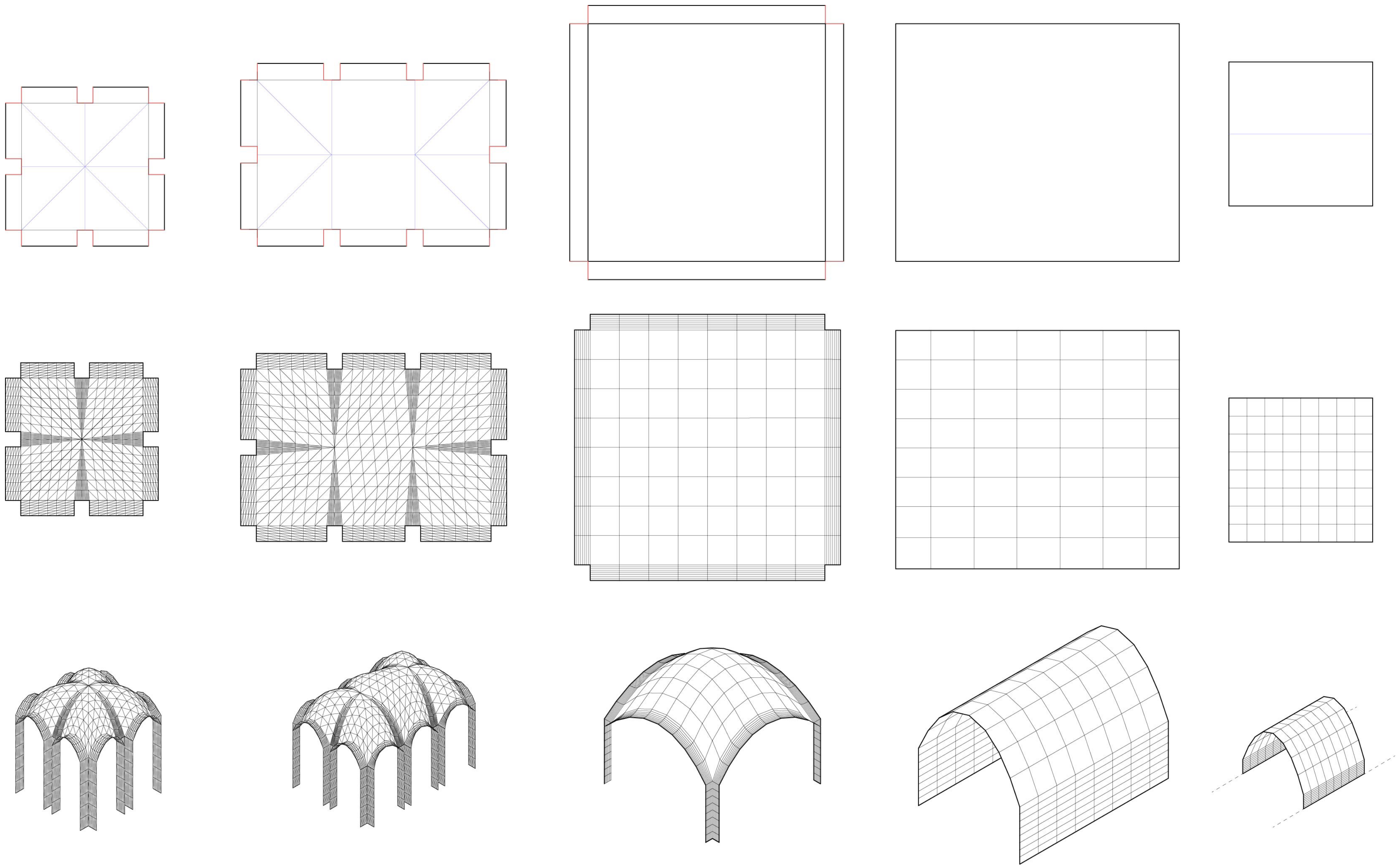
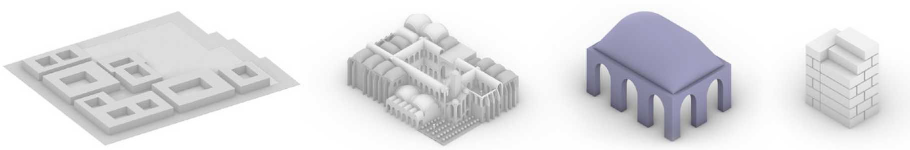


Figure57: Later stage of tessellation and mesh subdivision approach and dynamic relaxation



ROOF FORMING

The roof was defined for spaces by categorization into two - buildable and non-buildable on second level. The spaces which were buildable posed restriction in terms of maximum height permissible for spaces with different proportions. This meant that every space which is buildable on top irrespective of length or width needs to have the same height to avoid additional filling on top. The smallest room module works with 2 units in x axis and can incrementally increase in y axis. For this, the corners of the spaces were first tessellated and maximum height defined so that the roof can increase incrementally in the y-axis through a modular system. Riwaq connecting to the room module was also fixed by defining the maximum height of vault for one unit whose length can be increased with increasing size.
The spaces which are not buildable on second level were defined as per their support conditions - columns or walls, This resulted in vaults and dome which can start at the required height to create double height spaces.

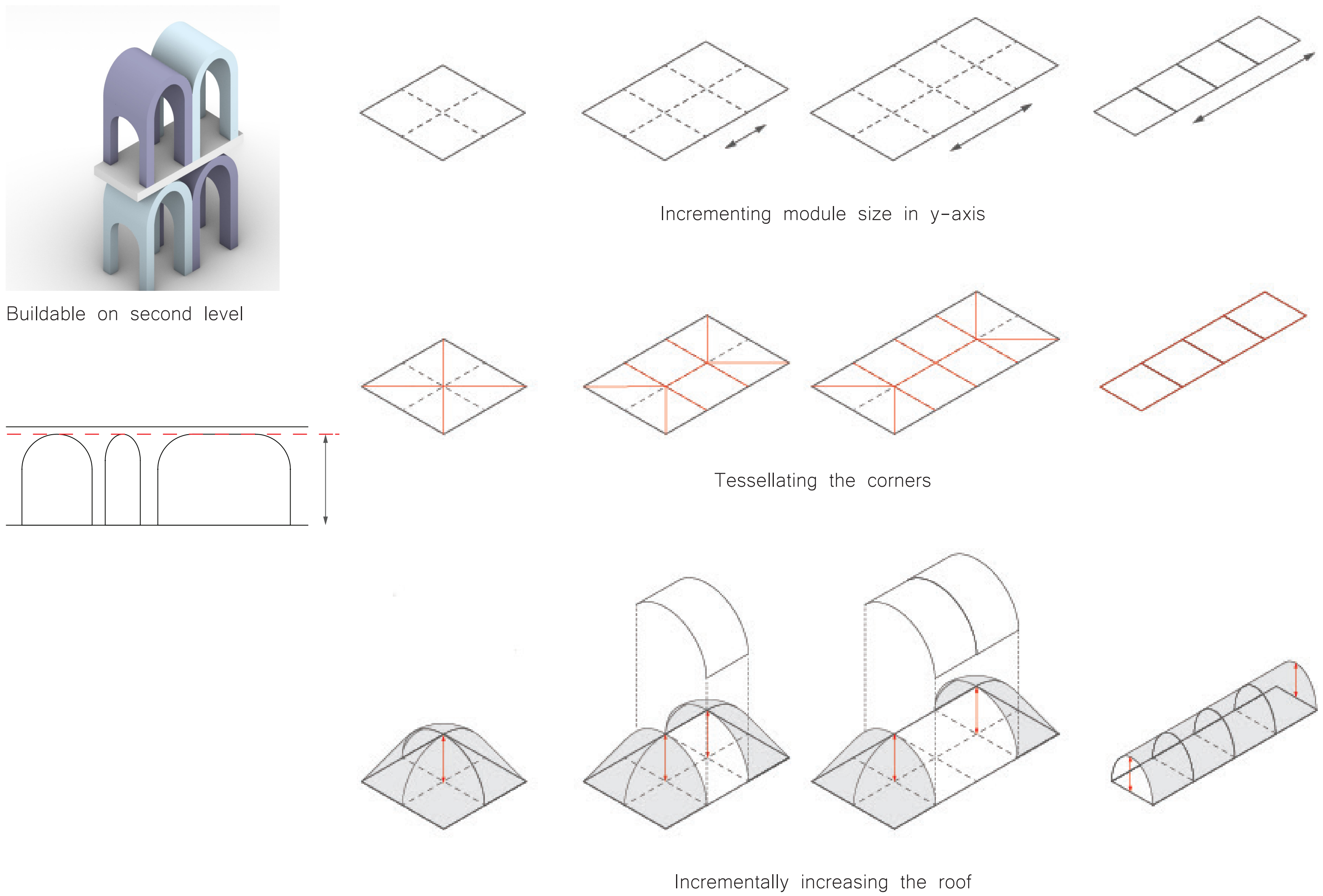


Figure59: Roof configuration for the modules buildable on second level

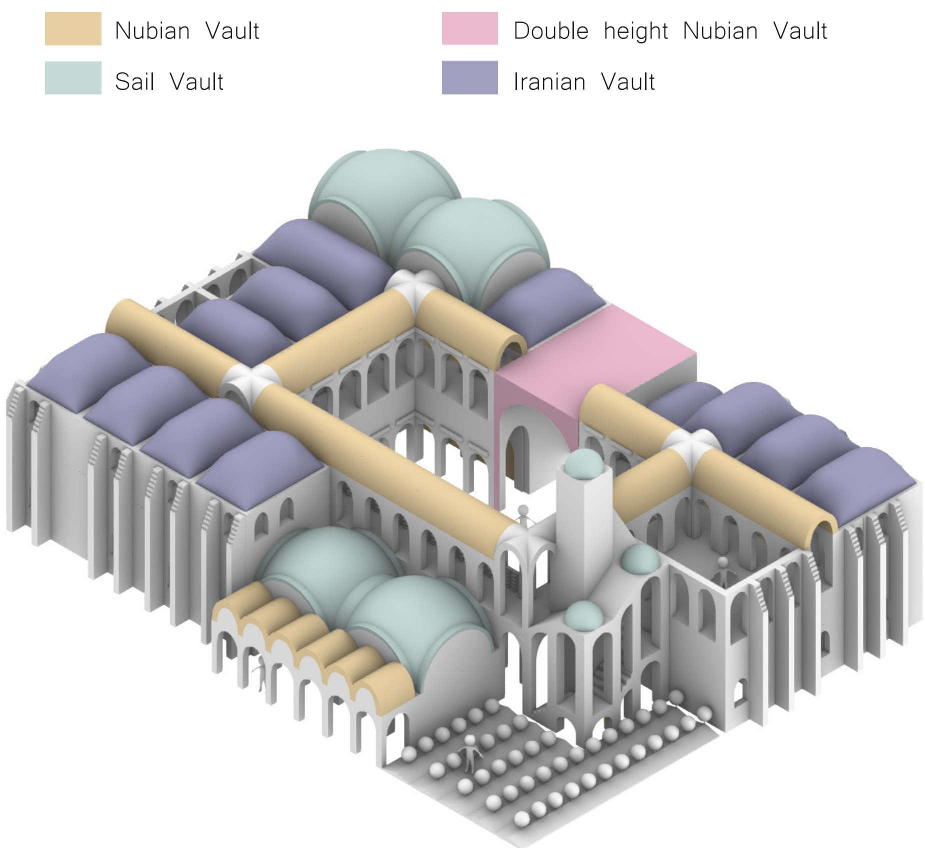


Figure58: Roof configuration for the dwelling unit

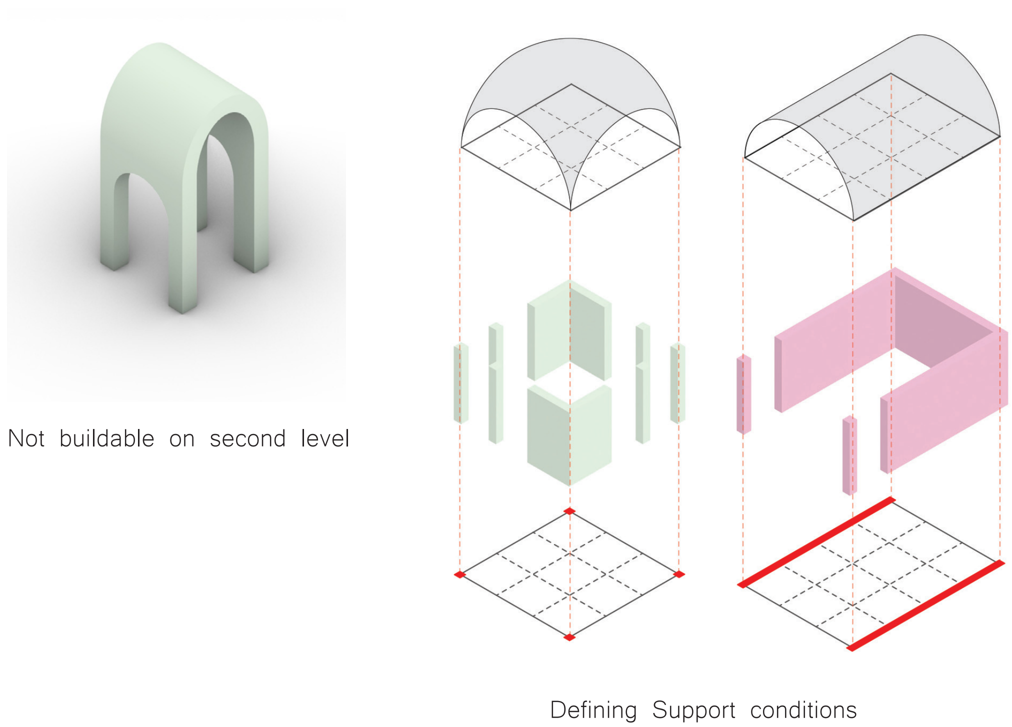
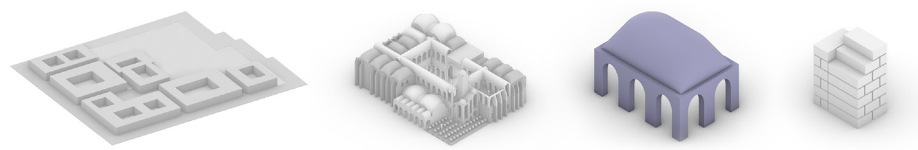


Figure60: Roof configuration for the modules not buildable on second level



SIMPLIFICATION

GEOMETRY SIMPLIFICATION

After the simplification of the dynamically relaxed form into domes and vaults, the form was further refined under a specific geometry in order to ensure accuracy in construction. Each curve of the form was redrawn to the closest ellipse or a circle. Three different ellipses and one circle was made. One ellipse was made for the openings connecting various spaces, one ellipse defined the vault for the Riwaq and one for the Iwan.
Secondly, a circle was also derived for the central barrel vault for the incremental space. Since the height could be increased for this space due to limitation in foor to floor height, the curve was a circular arc and not elliptical. However, being Earth construction, an elliptical curve can provide better compressive strength and transfer of forces. Thirdly, another circular arch is defined for the living room to start the sail vault on the square geometry.
The final form consists of 3 elliptical arches and 2 circular arches to simplify the construction process.

IWAN

The Iwan is a double height space where the arch starts fdrom the base of the second floor level and goes as high as 2.1m to result in an elliptical arch.

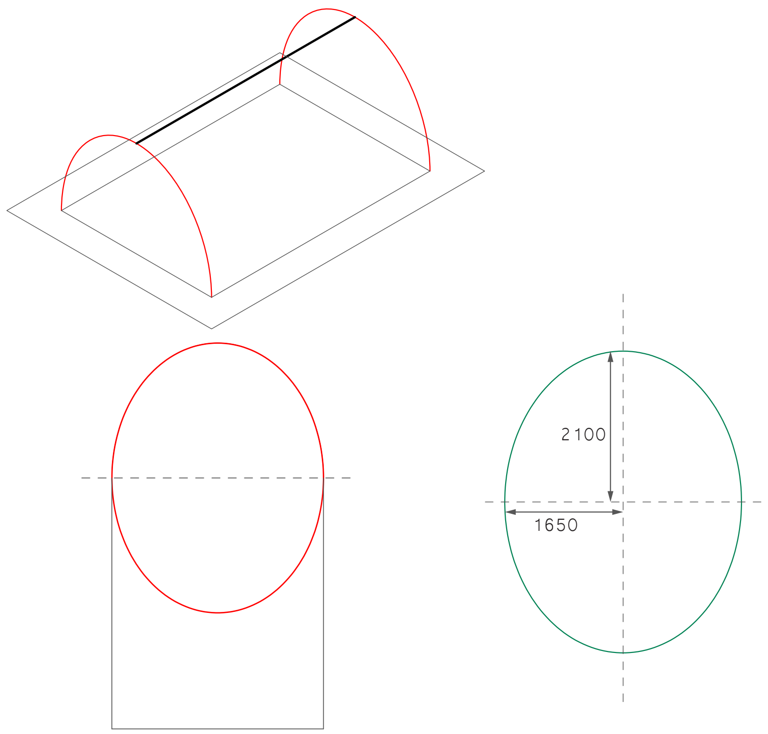


Figure61: Geometry simplification for Iwan

RIWAQ

The module has arches spanning in the two directions and at two heights. Both the arches are simplifies as ellipses.

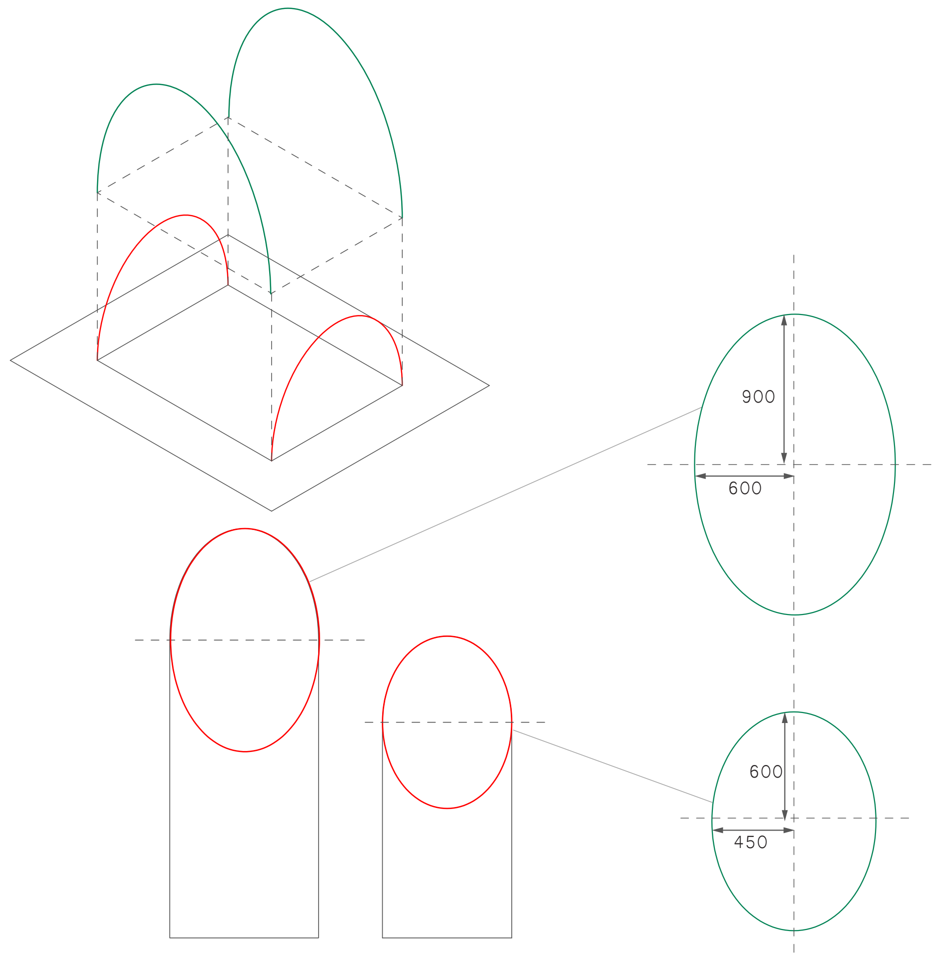


Figure62: Geometry simplification for Riwaq

LIVING ROOM

The dome of the living room is designed as a sail vault cut by 4 rectangle os the sides which defines the arches, resulting in circular geometry.

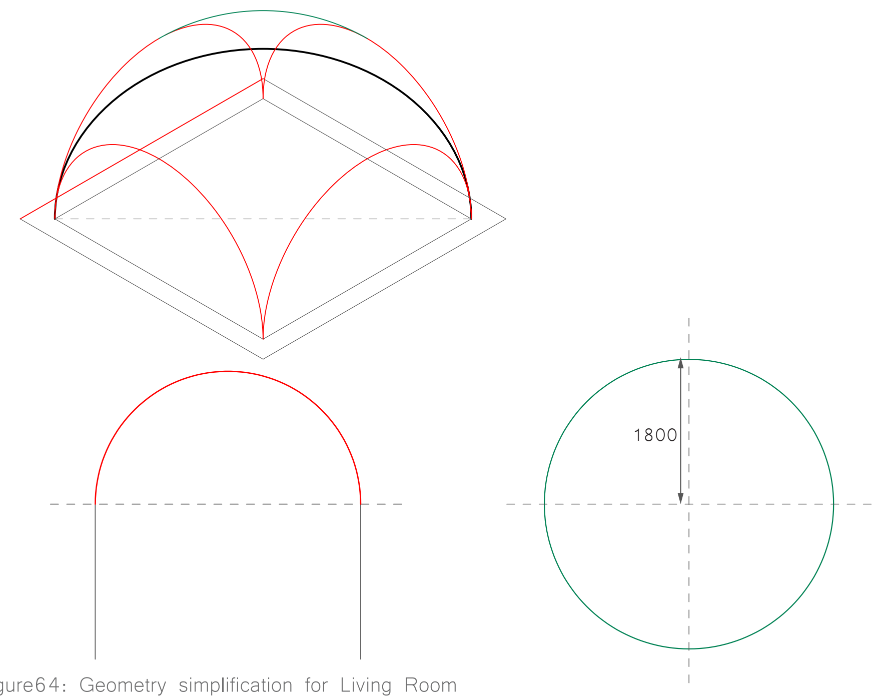


Figure64: Geometry simplification for Living Room

ROOM UNIT

The corner grid of the room unit is defined as the base geometry of a cone to create cove corners. The height defined then defines the central arches. Due to the proportions, the geometry is circular. However, possibility in increase of height can lead to better compressive geometry of an ellipse. The openings are ellipses which are same for all the different spaces to match the opening size.

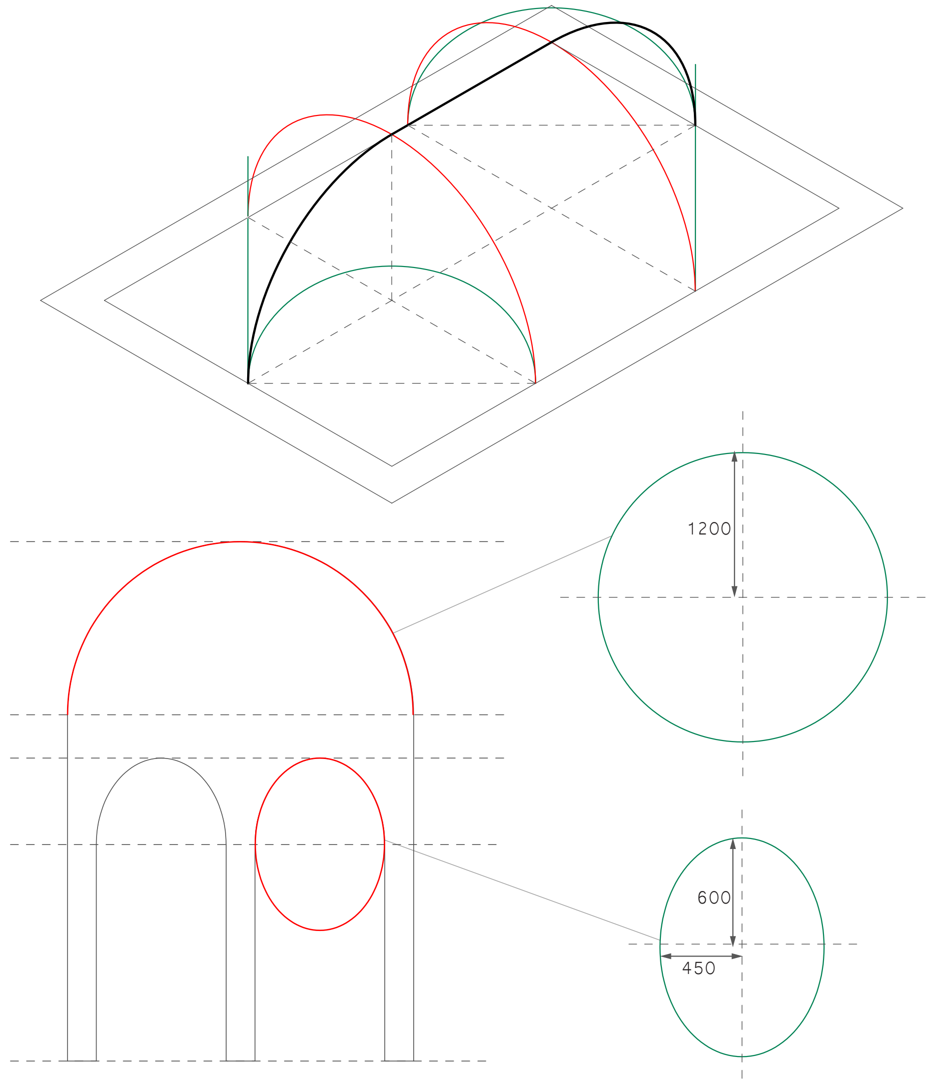
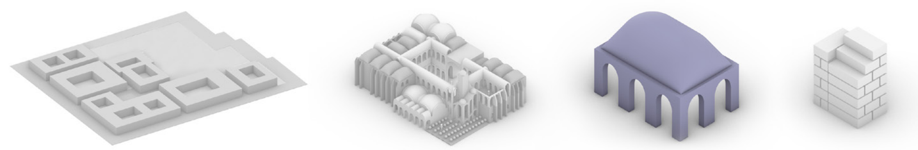


Figure63: Geometry simplification for typical Room Module



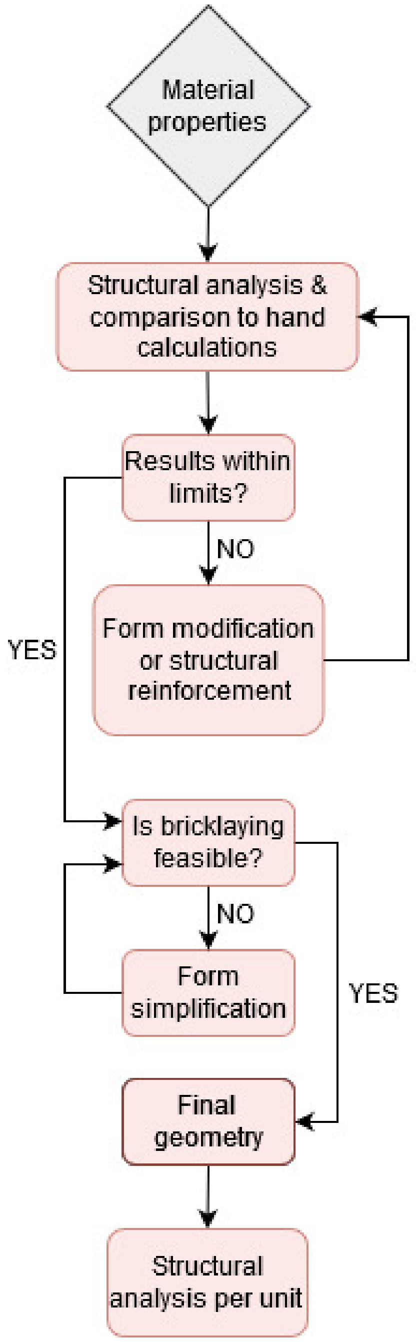


Figure65: Flowchart of the structural analysis methodology and process

04 STRUCTURING

METHODOLOGY

The structural analysis was carried out in parallel with the design of the room types, their form finding and constructability. The goal was to reach a desired solution within the material requirements and following the same modular systematic approach in having easily repeatable parts. In this approach, certain compromises needed to be made through trial and error and the results from each side were used as feedback to enable decision making and possible shaping or structural improvements.

First of all, certain parameters were set as constraints in respect to the systematic modular logic of the whole design. These were, for example, the general plan dimensions, the width of the openings and the height of the arch above, the dimensions of the walls and columns, the spring height of the roof and the maximum height of the modules having a second floor. However, some flexibility is allowed as long as it respects the unit modularity and if it can be accomodated in this logic, by creating multiples or subdivisions of units and measurements. This relates to the roof heights as well, but is influenced also by the constructability and workability at the same time to find a close to optimal trade-off.

Once the parameters were set, a first approach for the shape of the arches, vaults and domes was realised through the use of the Kangaroo plugin for Grasshopper, as explained in the previous chapter, in order to support their self-weight. However, although this provides with an estimation of the shape in a dynamically relaxed form, it still remains an approximation and it cannot be directly used for a more accurate structural analysis. This is also because the constraints required for the different heights cannot be fixed precisely and the roof structure is not as close as it would have been with the actual construction method.

Because of that, a second stage of form-finding and meshing was carried out to optimize the shapes and to use it as an input for the structural analysis, which was realised through the Karamba plugin for Grasshopper. In this process, the final roof shape that was studied for its constructability was used for the analysis. Apart from that, different meshing and mesh subdivisions were tested, in order to find the one that gives more accurate results, since the software interprets the meshing in a specific way and that influences its internal calculation process. In this continuous process to optimize the shapes, the loading conditions were set for each case, the final geometry was structurally analyzed and the results were evaluated according to the limitations of the material taken into consideration.

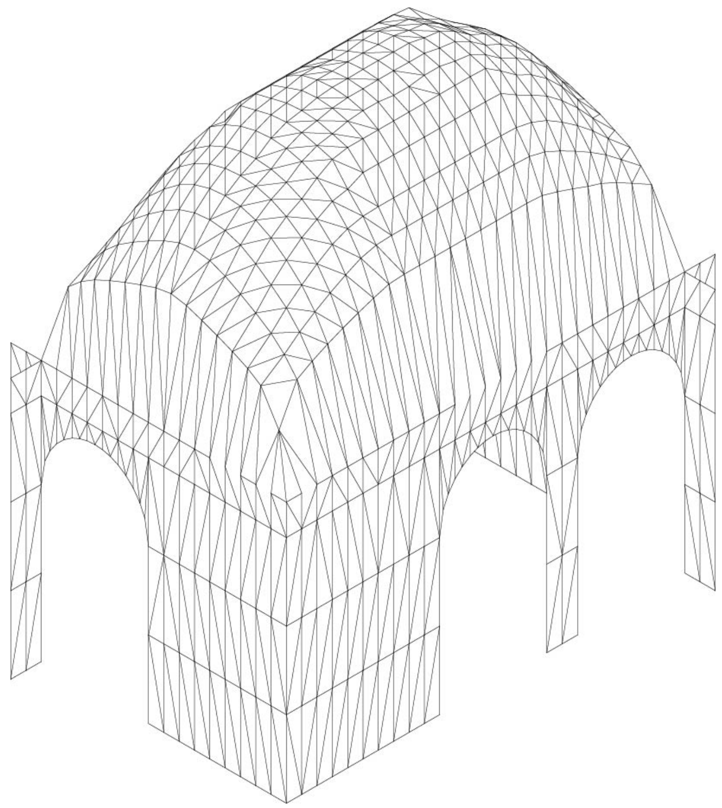


Figure66: Example of the mesh subdivision and refinement in 150mm wide segments, which are the smallest subdivision that fits every other dimension

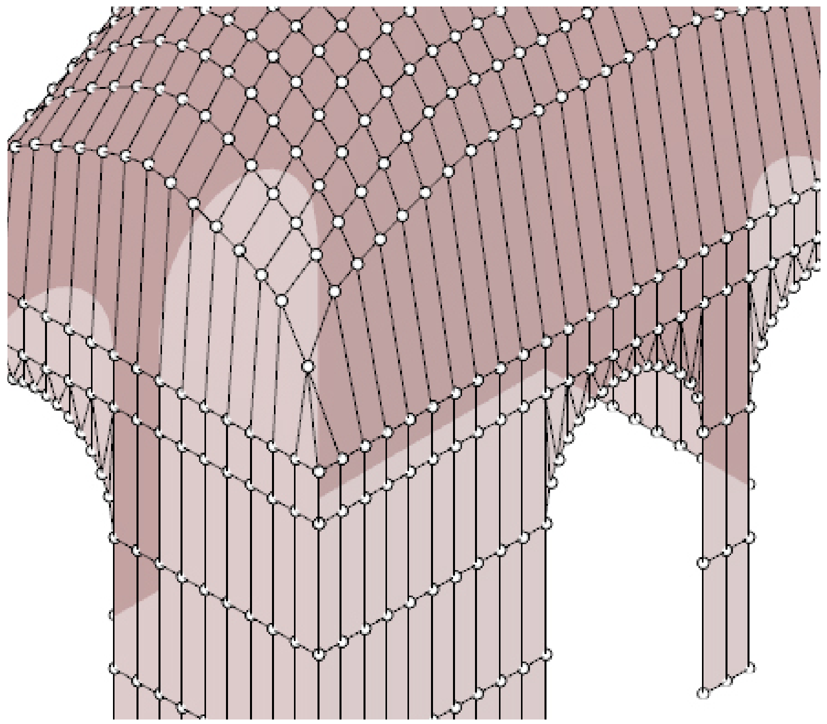
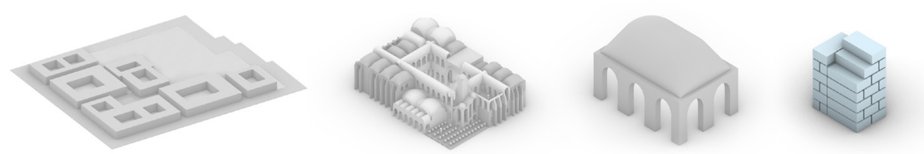


Figure67: Importance of all vertices to be connected to each other during the mesh subdivision and refinement for a correct load distribution



MATERIAL PROPERTIES

ADOBE BRICKS

A literature study has been done on the composition and properties of the adobe. The standard composition of adobe is 30% clay/silt, 30% fine sand, 40% coarse sand. Also, water is added in an amount of 10% of the weight of the mixture. In addition to the standard composition, additional materials can be added to the mixture. Some examples of this are straw, cow dung, lime and wood chips. It is also possible to compress the adobe. All these additions affect the properties of the adobe.

The properties that are relevant for the adobe are the compressive strength, tensile strength and the Young's modulus. The literature mainly contains the maximum compressive strength. However, a lower value must be used to guarantee the safety of the material. This is called the safety factor and is usually 2. A rule of thumb is applied for the tensile strength. For adobe, this rule of thumb is 1/10 of the compressive strength.

To provide an overview of the maximum compressive strength and tensile strength, all the collected data has been combined in the table below. Because some

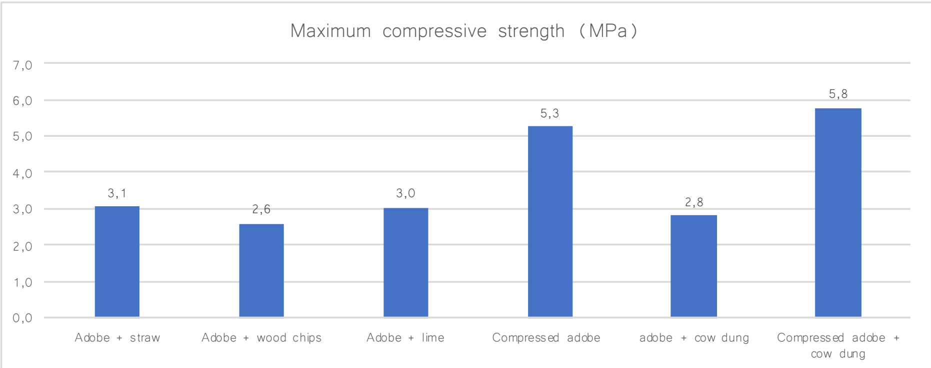


Table1: Maximum compressive strength

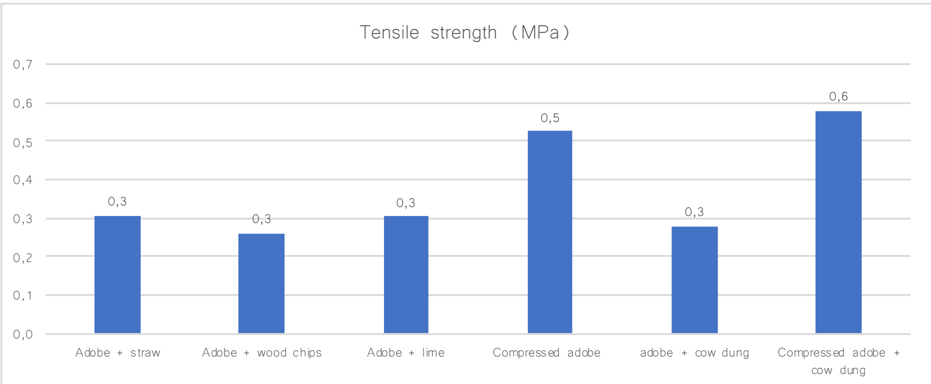


Table2: Tensile strength

Unfortunately, a Young's modulus is not known for every composition. The Young's modulus that were found are listed in the table below.

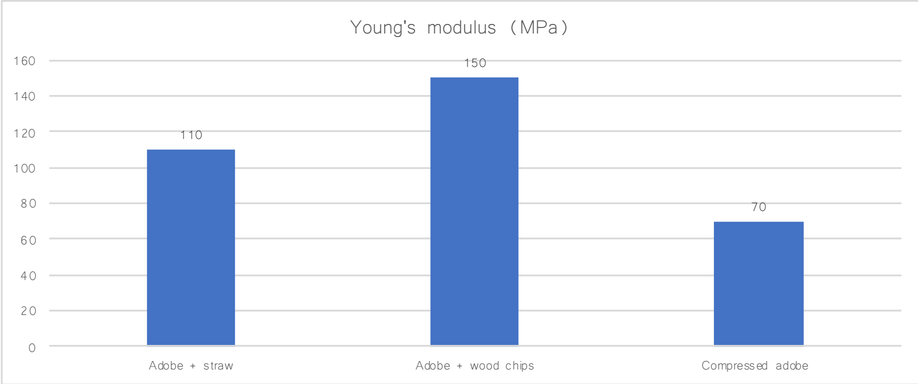


Table3: Young's modulus

The final composition that will be used is the adobe in combination with straw. This composition has a maximum compressive strength of 3.06 MPa. Due to the safety factor of 2, the allowable compressive strength is 1.53 MPa. The tensile strength is 0.153 MPa and the Young's modulus is 110 MPa. The density of the adobe with straw is 1500 Kg / m3.

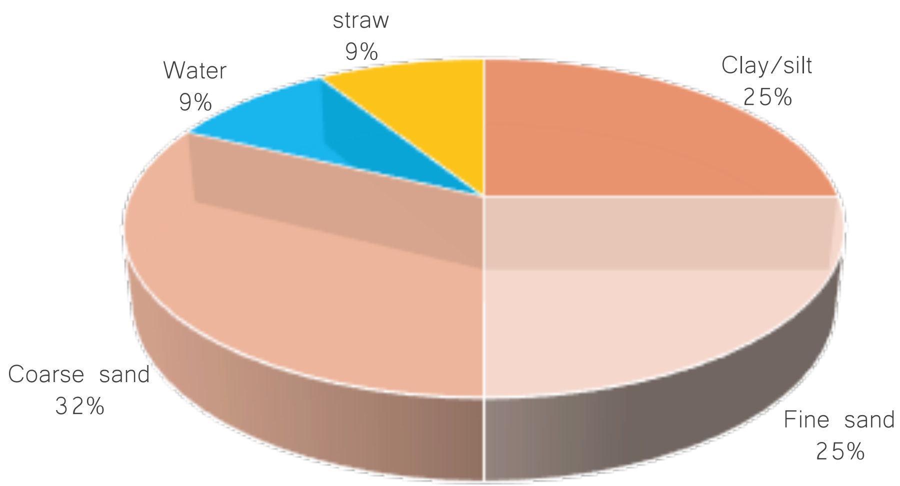


Figure68: Composition adobe

The sources that have been used are for adobe with straw Modul_ab_ity (2019), Bustan (2019), Illampas et all. (2011), Quagliarini et all. (2009) and Li Piani (2019). The source for wood chip adobe is Bustan (2019). The source for adobe with lime is Bharath et all. (2014). The sources for compressed adobe are Aubert et all. (2013) and Guillaud (1985). The source for adobe with cow dung is (Millogo et all. (2016) and the source for compressed adobe with cow dung is Yalley et all. (2013).

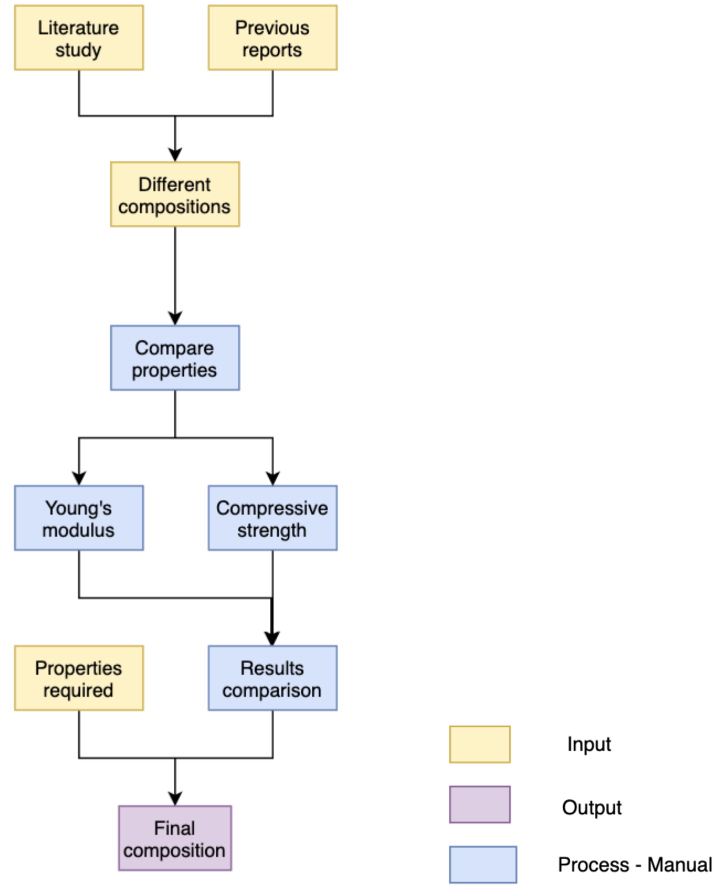
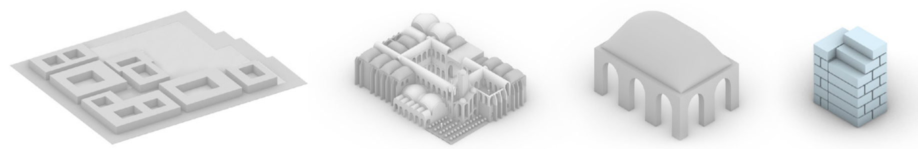


Figure69: Flowchart composition adobe

EARTHEN PLASTER

The roofs will be provided with an extra layer of earthen plaster. This is done to make the roofs waterproof. The composition of the earthen plaster will be the same as the composition of the adobe bricks. In addition, lime is added to the composition. This makes the earthen plaster water-resistant (Owens, 2015). Despite the fact that the earthen plaster is water-resistant, the plaster layer will have to be replaced after a few years due to wear and tear.



STRUCTURAL ANALYSIS PROCESS

GEOMETRY DESIGN AND MESH REFINEMENT

One of the goals for the geometry design of each space is its structural independency. This is related to the modularity concept, the general design freedom and flexibility of the spaces in the way they will be placed on the plan layout, their connectivity and adjacency, as well as the possibility to accommodate a second level on top. To achieve this, a generic structural solution was chosen where a minimum main structure would be fixed as to be sufficient for the structural stability of each space on its own. Because of the tartan grid used, which is 1500mm x 1500mm, two angle-shaped wall structures spanning 1500mm on each side are defined, which are diagonally symmetrical to each other to avoid creating moments in the structure and to achieve better bidirectional stiffness. In this way, every space can be structurally stable on its own and because of the same dimensions and modularity, when two spaces are placed next to each other, they have their similar structural elements aligned to each other, wall next to wall and corner next to corner and this alignment can be achieved by mirroring the space that is following if necessary. By doing so, even though in the end there is a double wall thickness, this provides a better stability and also a better base for a second level structure. A recap of this approach can be seen in the diagrams above.

The decision for non-shared walls comes also from the fact that there are certain double height spaces, such as the iwan and living room, which mean that there would have been a structural inefficiency if a one-storey space would share a wall with a double-storey one. And since most of the spaces are either double-height or can have the possibility for a second level, then it is better to opt for an individual and more stable structure. This wall-and-column layout refers to the minimum structural system that is required for its stability. The rest of the openings can be either left as openings or also closed off as walls, as explained in the form finding chapter.

When it comes to the roof structure, in the first approach the relaxed meshing that was used for the form finding process using the Kangaroo plugin gave approximate geometry, because it wasn't possible to fix the heights and spring levels in their exact measurements. Even though it wasn't precise, it was used initially to get a general estimation of the stress distribution, but was later rejected as an approach to a more accurate one to the actual roof structure, which was generated based on the constructability requirements and brick-layering method.

Once the surfaces defining the structure were obtained, they were translated into mesh to be able to perform the FEA analysis. Since the thicknesses of walls and the roof are different, they are originally drawn as separate meshes, so as to be able to attribute seperate conditions later in the simulation. The subdivisions used were of the Rectangular or Triangulation depending on the case, trying to achieve a good enough resolution in order to have the most reliable results possible. In most of the cases, for the roof subdivision a grid of 150mm x 150mm was finally used as a projection on the roof surface to enable better connection of all the vertices between each other and subdivisions of same sizes. This measurement was used as it is the smallest one that fits in every other dimension, based on the modular concept. The meshing and subdivision were quite important for the analysis, because the stress distribution is projected on each mesh face, so a non-homogeneous mesh subdivision can affect the final result and show local concentrations and unjustified stress peak points. Because of the above constraints, after several tryouts for parametrically generating the mesh geometry and subdivision, a geometry input from Rhino was chosen as a way to approach it. This was realised by using the defined geometries from the Shaping process for each roof unit as a base geometry to extract the wireframe and edges and manually subdivide the mesh edges into segments of 150mm width. Those were then used to generate the mesh in Grasshopper having a smooth and homogeneous mesh subdivision, as seen in the figure in the Methodology section.

In the case of contact between two different elements (for example between domes and walls) it was critical for the FEA analysis through Karamba, to check that they had all the vertices in common in the contact area, as seen in the above diagram in the Methodology section. Having vertices in the same position is translated by Karamba into a single structural node and forces can then move from one element to another without errors dictated by geometric inconsistencies. Therefore, the different meshes are joined in the beginning to create the subdivisions and then are separated again to apply the separate load conditions and element cross-sections. Once the geometries with the respective variable parameters have been defined, there were several tryouts to check the validity of results and their correspondence with the hand calculations for a double check.

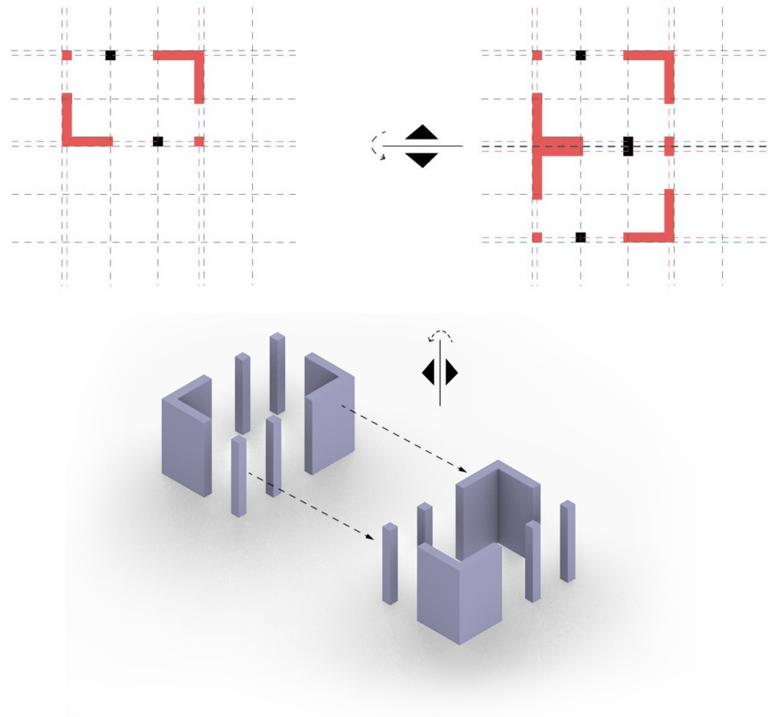
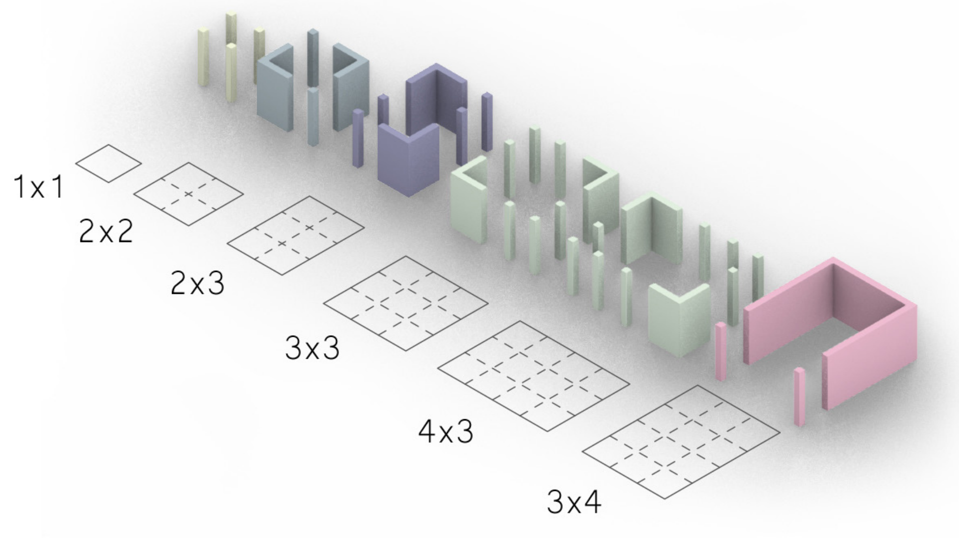


Figure70: Recap of the mirroring and alignment logic of the main structural elements, as explained in the Shaping chapter



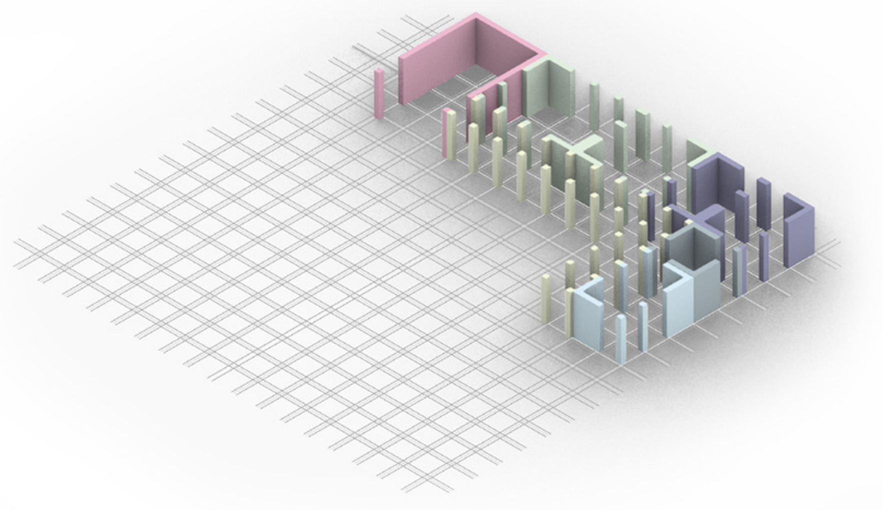
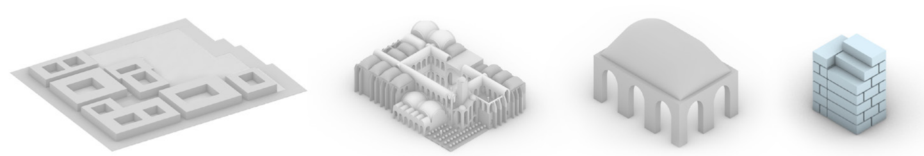


Figure71: Room units with their main structural elements defined, as explained in the Shaping chapter



KARAMBA SETTINGS

INPUTS FOR STRUCTURAL ANALYSIS

Once the geometric model was completed, the structural analysis was carried out. The software chosen for the FEA analysis was the Karamba plugin used in Grasshopper. For the materials settings, the properties of the brick chosen were used as input, paying attention to the different units used in the software, for example KN/cm^2 for the Young's modulus and KN/m^3 which is required for the density. The same applies for the output results, where the stresses are shown in KN/cm^2 and the displacement in cm. The material inputs used are 110MPa for the Young's modulus and 1500 kg/m^3 for the density, which are then transformed to the right units. For the shear modulus and the yield strength, the results from previous teams were taken into consideration.

When it comes to the geometries, these are identified by Karamba as "elements". In this case, the element type of both the walls and roof is classified as shell elements and in each case its own cross section has been assigned. Because of the modular system and brick size, the wall thickness is assigned as 290mm, but the roof has initially a value of 140mm based on the brick size used for the roof, but with the flexibility to change depending on the results. The whole wall and columns have the same thickness assigned which works better as a whole system and during the construction process. The support conditions were then assigned to the vertices of the model having an elevation equal to zero (i.e. in contact with the ground) blocking both translations and rotations in all directions, simulating a clamped support. Lastly, the various loads acting on the structures were set, depending on each case. In the end, the results were translated to KN/mm^2 or MPa for the stresses and to mm for the displacement to have a better understanding and consistency with the hand calculations.

LOADCASES

Two types of loads were used in the Karamba settings, the mesh load and the point load. The first one refers to distributed loads which are acting on the whole mesh and are having a perpendicular direction to the global horizontal reference plane. In these cases, these loads are namely the live loads, with a value that changes depending on the type of structure and the number of floors, as it will be described later. The point load was used for the gravity and for the non-uniform loads, such as the earth filling weight above certain room types, as well as the support reactions of the second floor built on the kitchen/bedroom room type and riwaq which were considered as loads on the ground floor.

Regarding the live loads, these were applied as uniformly distributed loads on the units, such as the sand load and a value of 0.75 KN/m^2 for occupancy load in the case of practicable covers (riwaq) and 2.75KN/m^2 in which the addition of mobile objects was foreseen such as furnishings (bedroom, kitchen).

The earth filling load is a non-uniformly distributed load and it was applied as a point load on the vertices of the affected meshes. The value to be applied to the vertices has been approximated by calculating the area of the mesh faces close together and multiplying it by the distance from the plane passing through the highest point of the dome or vault. The volume of earth above a given vertex is then multiplied by the density of the material (1500 kg/m^3 for the sand), obtaining a point load in KN variable according to the amount of the earth. It is an approximation which might lead to some divergence with the hand calculations and that should be considered in the comparison of the results.

As for the rooms which can accomodate a first floor on top (bedroom, kitchen, riwaq), the loads are defined after calculating the support reactions of the unit as a separate ground floor room. These are then applied with an opposite direction on the top of the containment wall on which the first floor rests. The support reactions of the wall were then assigned again in a second analysis, in the reverse direction to the edge of the unit.

Finally, the wind load is added, which is approximated with a value of 0.75KN (based on the previous works) and then is applied on the vertices facing one direction. This is another load which might give inconsistent results with the hand calculations, since it is not taken into account in that case, but the value is not much to show significant differences.

For the loadcases there was a safety factor added, also considering the different conditions of testing the material and possible discrepancies. These were 1.2 in the case of the self-weight and 1.5 for the case of the live loads. As explained before, these were also divided in non-usable and usable spaces, where for the first case the values were 0.75 KN/m^2 for roof load used just for maintenance, whereas in the second case it is increased to 2.75 KN/m^2, as in combined maintenance roof and floor use. These values are then increased by adding the safety factor.

Material inputs	
Young's modulus (N/mm2)	110
Yield strength (N/mm2)	1.3
Shear modulus (N/mm2)	60
Specific weight (KN/m3)	15

Figure72: The material properties that were used as an input for the Karamba structural analysis

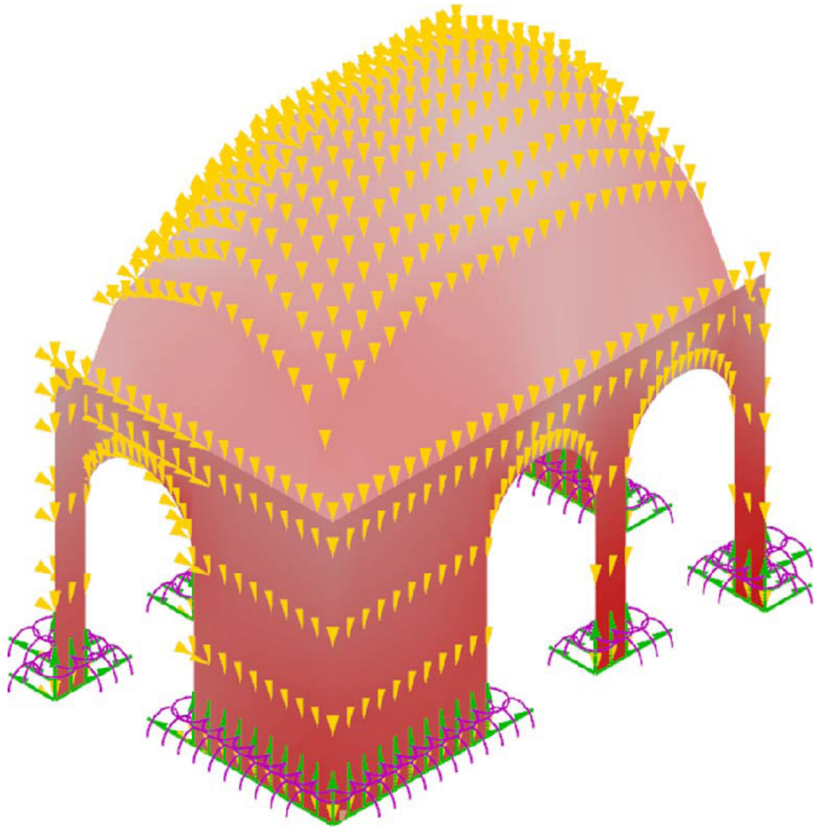
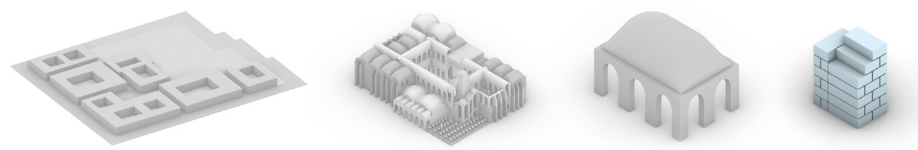


Figure73: The supports and load forces as applied on a room unit



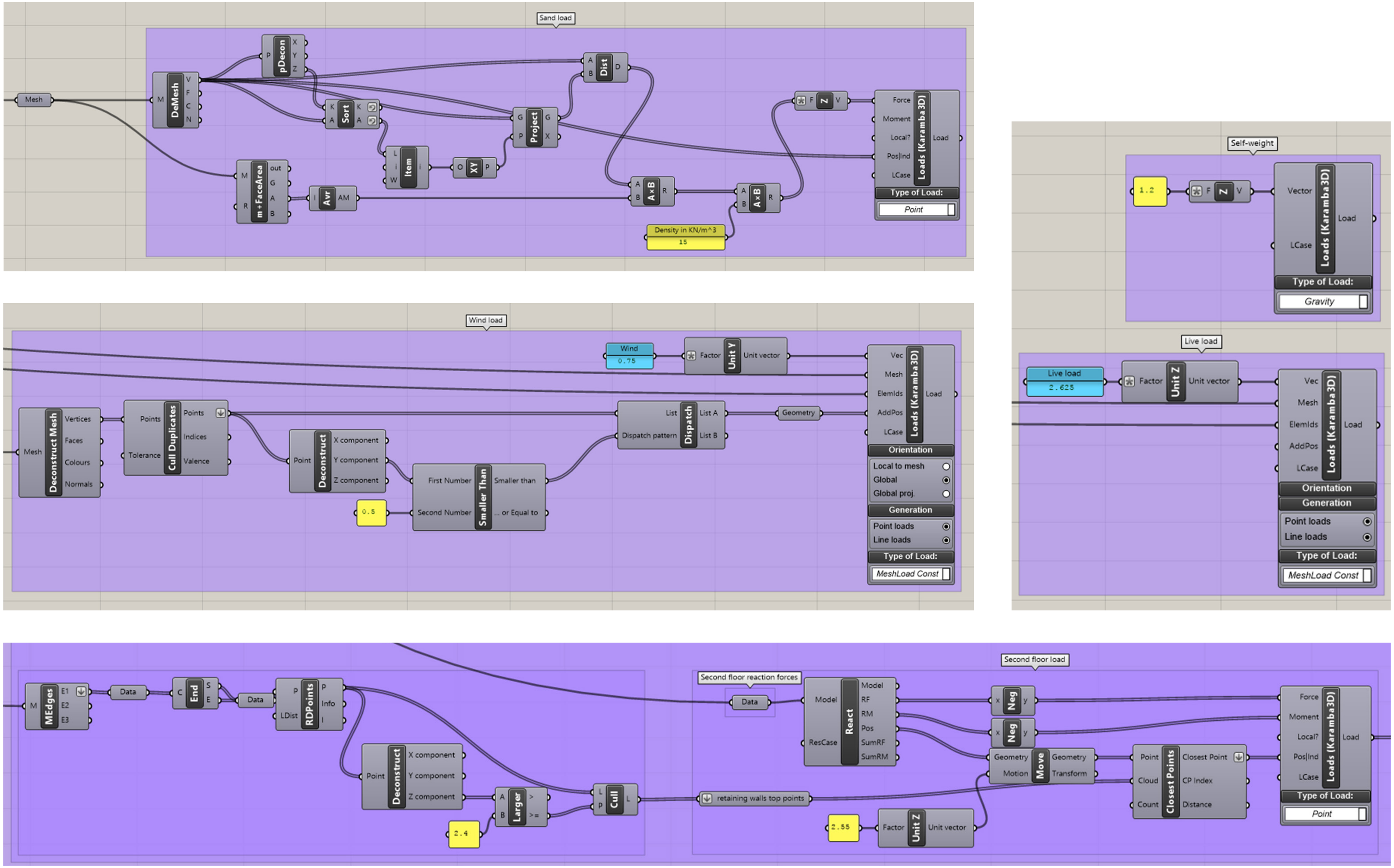
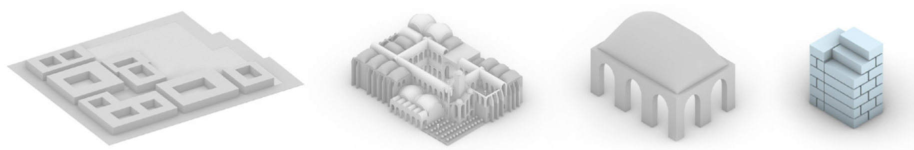


Figure74: The loadcases used as inputs in the structural analysis in Karamba



EVALUATION AND DISCUSSION OF RESULTS

For the evaluation of the results obtained from the analysis, it is important to define the allowable limits for the compressive and tensile stresses, as well as the displacement. These were determined taking into consideration the properties and the limits of the material itself. Based on the brick type chosen and the tests from previous works, the compressive strength was set to the value of 1.53 MPa, while the tensile strength was determined as 1/10 of the latter one, which gives a value of 0.153 MPa. The maximum displacement was decided to not exceed a value of 0.002 (or even 0.003)*span, based on the previous works. If this limit is exceeded, the mechanical properties of the material are reduced and the deformation reaches a critical point, which is not structurally safe. After defining these limits, the analyses of various room types are explained, showing for each case the values of displacement, the maximum values of the first (for tensile stresses) and second (for compressive stresses) principal stresses, as well as the material utilization. In most of the spaces, also hand calculations were made, as can be found in the Appendix, and the results were compared for validation. Important notice is that in the hand calculations the wind load was not taken into consideration and that the approach for the sand filling load is slightly different between the manual and the computational method. For this reason, a first comparison was realised without adding the wind load and then certain deviations were also considered acceptable.

BEDROOM / KITCHEN UNIT (BARREL VAULT CLOSED WITH COVES)

The bedroom unit is one of the room types that can also accommodate a second floor on top. In this case, two analyses were realised, one for just a ground floor unit and one for a double floor unit. The roof on top is a barrel vault closed with coves. In the case of the ground floor type, the loads considered were that of the self-weight, live load of a value of 1.125 KN/m^2 and the wind load. the maximum compressive stress was 0.205 N/mm^2, while the maximum tensile stress is 0.126 N/mm^2, both within the allowable limits. The maximum displacement is 6.53mm, which doesn't exceed the value of 7.2mm which is the limit for this case (0.003*2400mm). The maximum stresses are observed at the base of the columns and some tensile stresses peaks can be seen on top of the arched openings. The utilization reaches a value of 16.5% for compression, which is more than the one for tension, since the material and structure works mostly in compression. However, it is a quite low value and this is up to a point due to construction constraints.

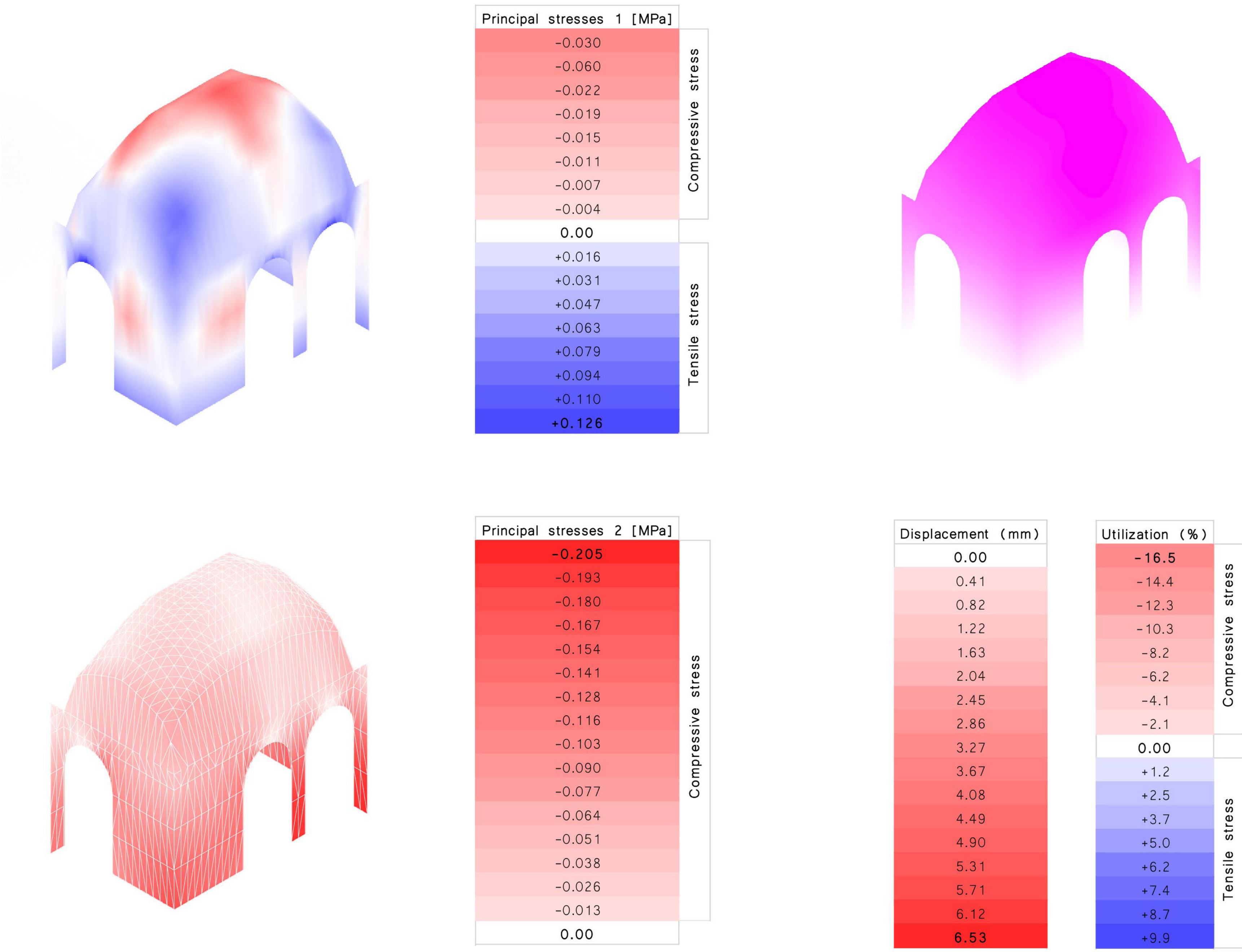


Figure75: Structural analysis results

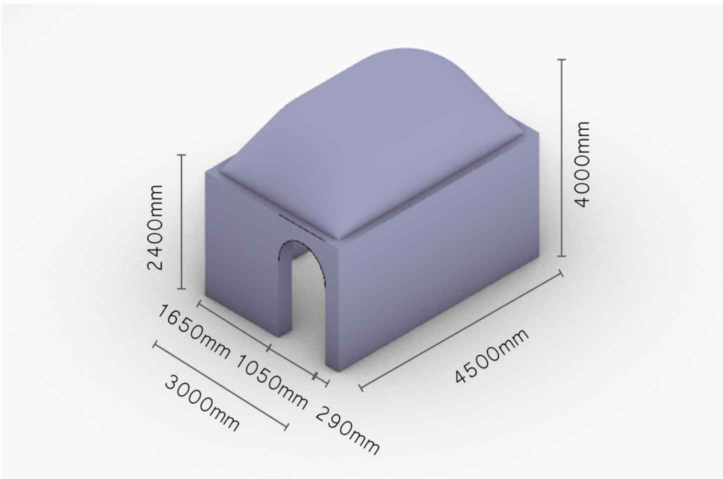
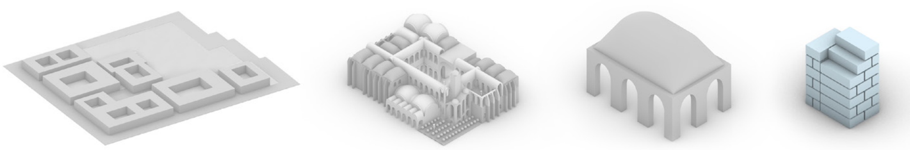


Figure76: General room dimensions



When it comes to the double floor unit, some additional loads were added such as the sand filling in between the two floors and the load from the second floor, which has a value same as the support reactions from the previous calculation but an opposite direction. Also, the live load in this case has a higher value of 4.125 KN/m^2, because it becomes a usable space. In this case, even though the compressive stress is within the limits with a value of 0.768 MPa, the tensile stress reaches a value of 0.321 MPa and the diplacement is 20mm (maximum limit 17.4mm, as in 0.003*5800mm). In this case, since we cannot change the wall or column thicknesses due to construction and modularity constraints, different improvements are tested. For example, if the brick type is replaced with a stronger one with higher tensile stress, such as the compressed adobe, which has a tensile strength of 0.55 MPa and increase the roof thickness to 200mm, the results lead to values of 0.275 MPa for the tensile stress and 17.1mm for the displacement. Even though the compressive stress is increased to 0.772 MPa, it still remains below the limit of 1.5 MPa. The utilization in this case is 61.8% working for compression, which compensates to the lower utilization for just a ground floor, meaning that a one-floor unit has already a structure as a provision that works better if an additional floor is constructed on top. And since it is an open and flexible system, some overuse of material can be expected.

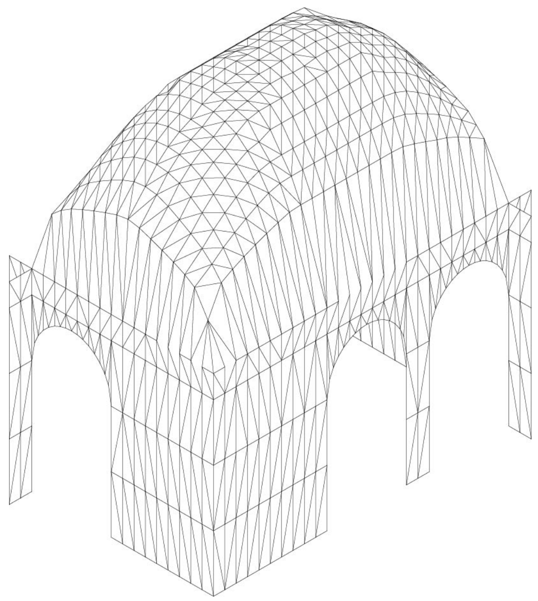


Figure77: Geometry and mesh subdivision

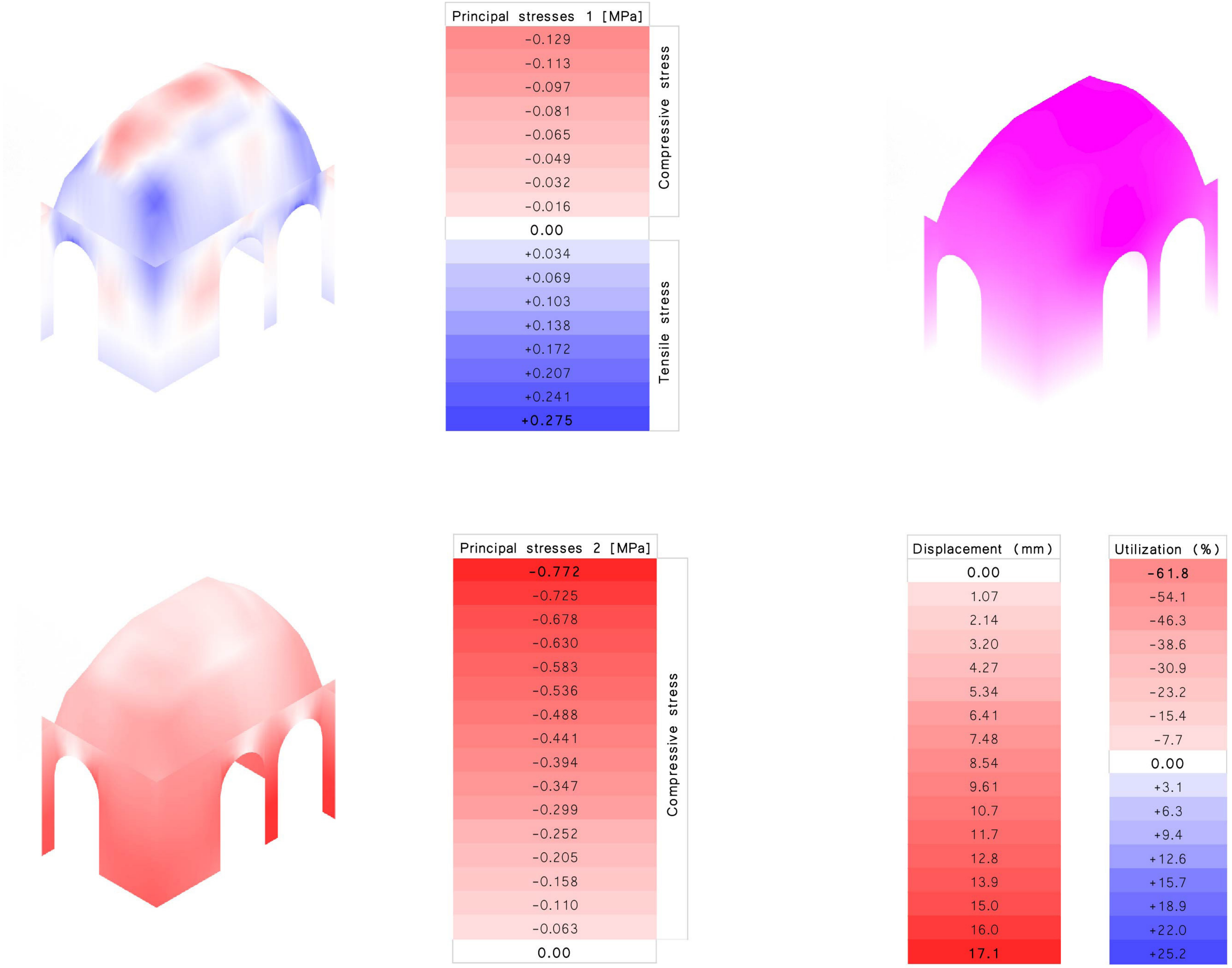
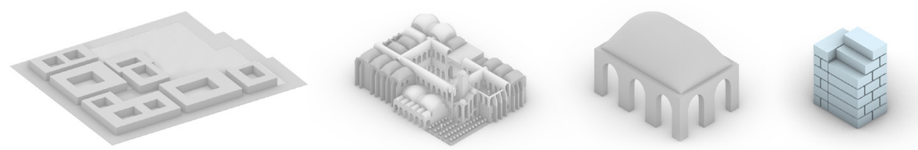


Figure78: Structural analysis results



RIWAQ UNIT (NUBIAN VAULT)

The riwaq unit has a nubian vault structure, however not an optimal one. because of the necessity to align the heights to the adjacent structures and arched openings. The fact that the columns span every 1500mm helped in part for the force distribution. Thea analysis here is based on a ground floor unit. The loadcases considered are the self-weight, the wind load, the sand filling on top, as well as a live load of 2.625 KN/m^2 for maintenance. In this case, the maximum compressive stress found is 0.173MPa, the tensile stress 0.057MPa and the displacement 2.73mm (allowable 6.3mm). All the values are still acceptable and below the allowable limits, but also in this case the utilization value is 14% working for compression, which is also justified as in the previous example.

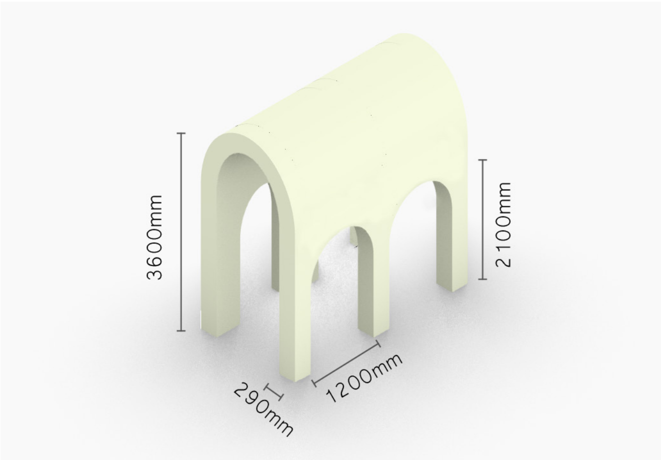


Figure79: Gerenal room dimensions

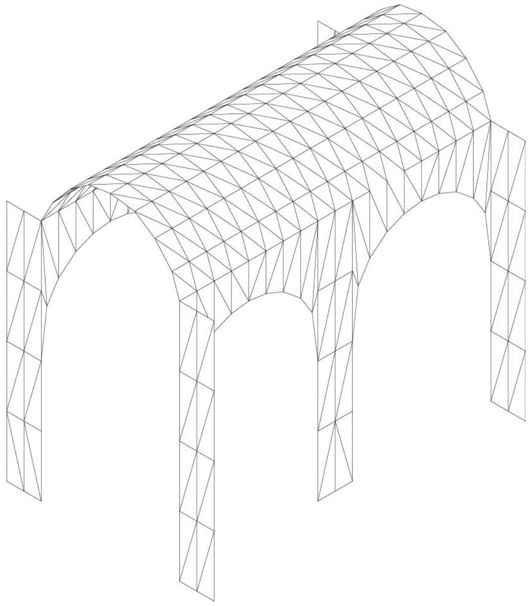
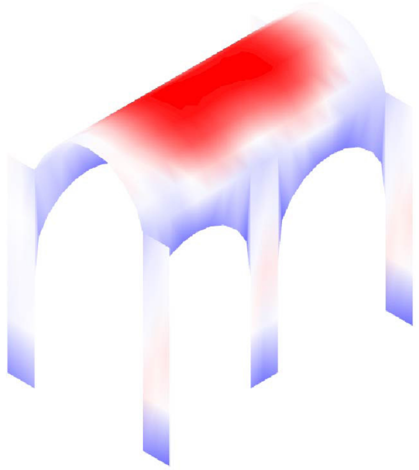
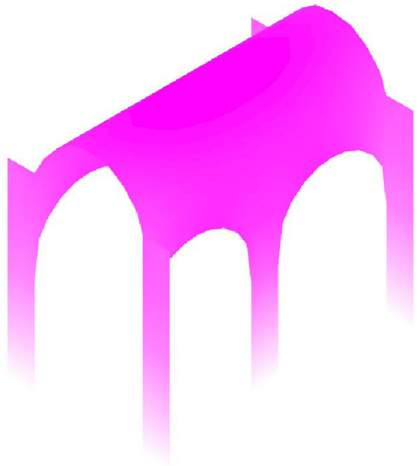
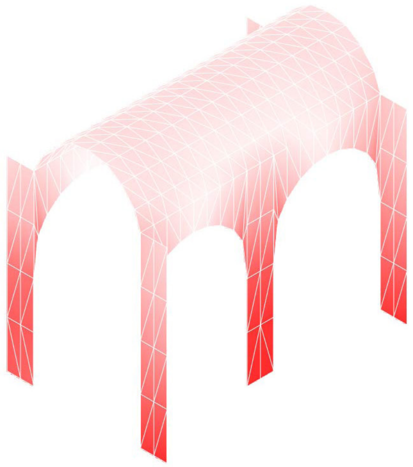


Figure80: Geomtery and mesh subdivision



Principal stresses 1 [MPa]	
-0.012	Compressive stress
-0.010	
-0.0087	
-0.0072	
-0.0058	
-0.0043	
-0.0029	
-0.0015	
0.00	
+0.007	Tensile stress
+0.014	
+0.022	
+0.029	
+0.036	
+0.043	
+0.050	
+0.057	

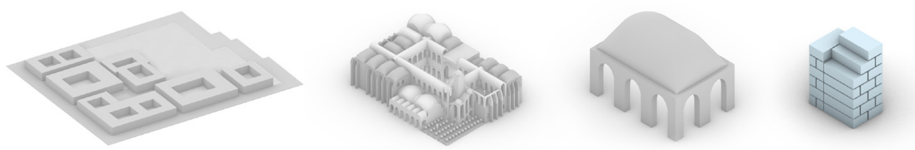




Principal stresses 2 [MPa]	
-0.173	Compressive stress
-0.151	
-0.129	
-0.108	
-0.086	
-0.065	
-0.043	
-0.022	
0.00	
+0.0009	Tensile stress
+0.0017	
+0.0026	
+0.0035	
+0.0043	
+0.0052	
+0.0060	
+0.0069	

Displacement (mm)	Utilization (%)
0.00	-14.0
0.17	-12.2
0.34	-10.5
0.51	-8.7
0.68	-7.0
0.85	-5.2
1.02	-3.5
1.12	-1.7
1.36	0.00
1.53	+0.6
1.70	+1.2
1.87	+1.7
2.04	+2.3
2.22	+2.9
2.39	+3.5
2.56	+4.0
2.73	+4.6

Figure81: Structural analysis results



LIVING ROOM UNIT (SAIL VAULT)

The living room has a double sail vault roof and is a double height room structure. In this case, the load cases considered were the self-weight, the wind load and the live load of 1.125 KN/m^2 fro maintenance. In the first calculation round, with the default settings for brick type and roof thickness, the results gave a maximum compressive stress of 0.479 MPa, which is within allowable limits. However, the tensile stress was 0.338 MPa, while the displacement reaches a value of 20mm (allowable value 15.3mm derived from 0.003*5100mm). In this case, as in the double floor units, the values exceed the limits. By increasing the roof thickness to 280mm this time and changing the brick type to the one with higher tensile strength, the results are again within limits. After the changes, the maximum compressive stress is 0.457 MPa, the tensile one 0.427 MPa and the displacement is reduced to 15.1mm. This was selected after several tryouts of roof thicknesses, so as to have a quite satisfying trade-off between tensile stresses and displacement and to set the last one within the limits, without increasing the tensile stresses above limits. These stresses are accummulated mostly above the arched walls and openings and on the arch between the two vaults and are justified because of the large spans. Lastly, the utilization is 35.1% for compression, but in this case the utilization for tension reaches a higher value of 36.8%, which is not optimal or desired, but is a result of this compromise.

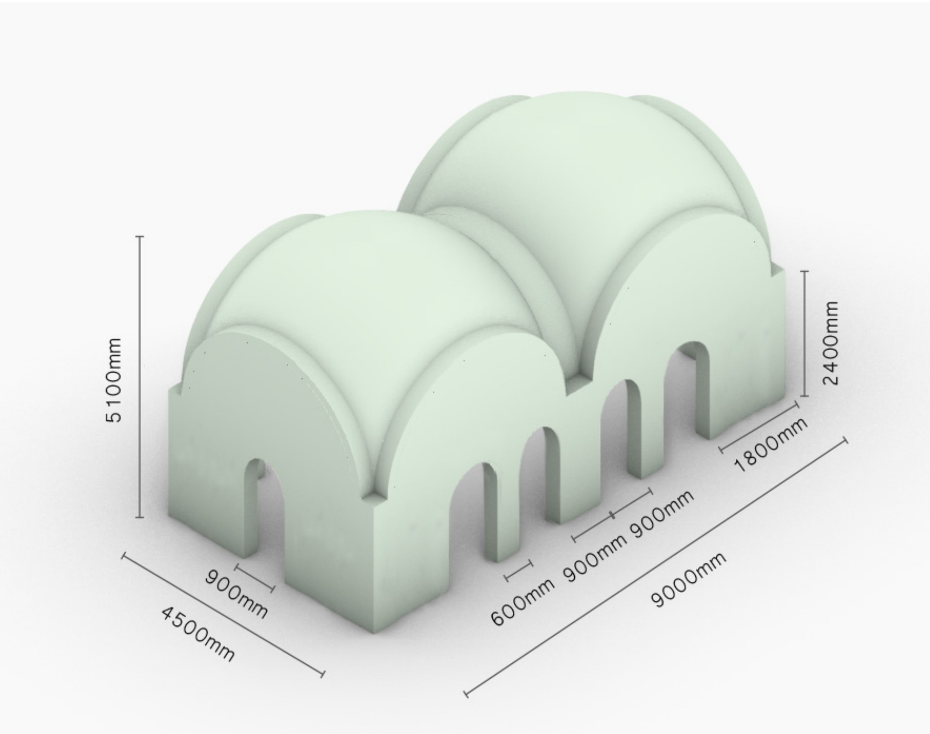
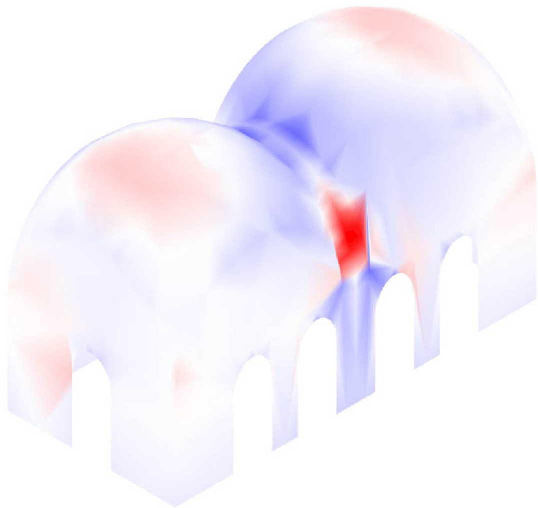
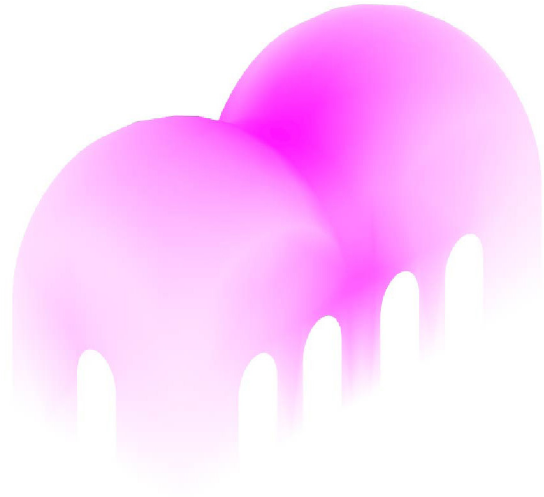
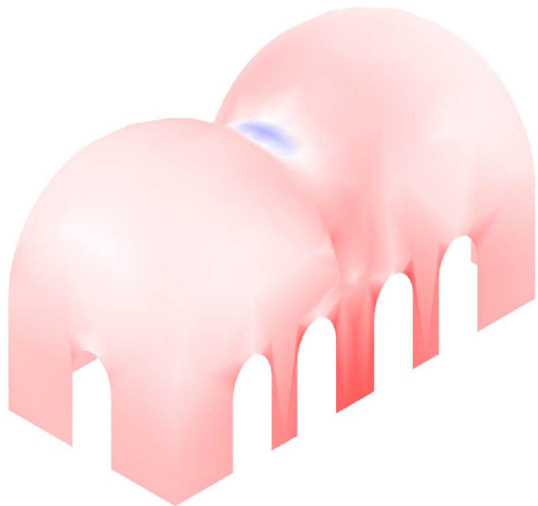


Figure82: General room dimensions



Principal stresses 1 [MPa]	
-0.152	Compressive stress
-0.133	
-0.114	
-0.095	
-0.076	
-0.057	
-0.038	
-0.019	
0.00	
+0.054	Tensile stress
+0.108	
+0.162	
+0.216	
+0.270	
+0.324	
+0.378	
+0.432	





Principal stresses 2 [MPa]	
-0.462	Compressive stress
-0.404	
-0.346	
-0.288	
-0.231	
-0.173	
-0.115	
-0.058	
0.00	
+0.015	Tensile stress
+0.031	
+0.046	
+0.062	
+0.077	
+0.093	
+0.108	
+0.123	

Displacement (mm)
0.00
0.87
1.74
2.62
3.49
4.36
5.23
6.11
6.98
7.85
8.72
9.59
10.5
11.3
12.2
13.1
14.0

Utilization (%)
-34.2
-29.9
-25.6
-21.4
-17.1
-12.8
-8.5
-4.3
0.00
+4.5
+9.0
+13.5
+18.0
+22.5
+27.0
+31.5
+36.0

Figure83: Structural analysis results

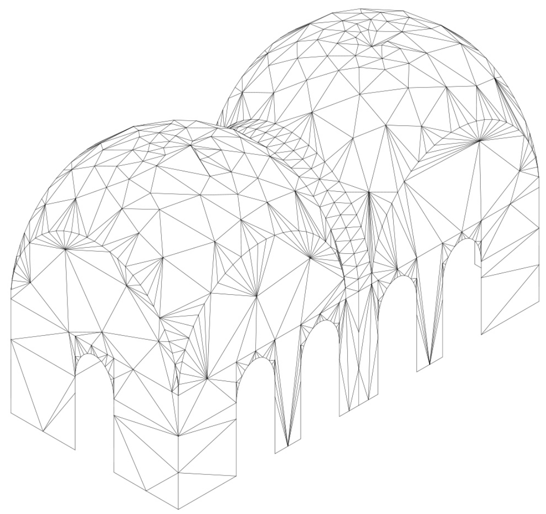
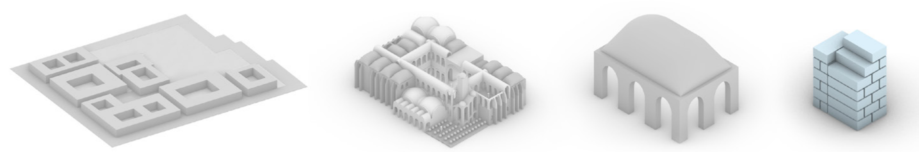


Figure84: Geometry and mesh subdivision



IWAN UNIT (BARREL ARCH WITH NUBIAN VAULTING)

The last unit analysed is the iwan which is also a double height unit with a barrel arch with nubian vaulting filled with sand on top and is supported mainly by walls and two column on the front corners. This is due to architectural reasons because of the nature of the room type, which is closed off on three sides, with just a connection to the riwaq on the sides and a large one to the courtyard in front. Therefore, the side arched openings are of the same height as the riwaq ones. The loadcases applied here are the self-weight, the wind load, the sand-filling and a live load of 1.125 KN/m^2, again for maintenance purposes. This room type showed the worst stress and displacement values due to the large height and spans in the openings. Because of that, from the first round of calculations, it was decided to proceed to similar improvements as before, but in this case even more extended. After several iterations for different roof thicknesses, the one with 300mm was chosen, a change of brick type and because it doesn't interfere with other room types or with the modular system, it was decided to add a second wall and column layer to the inside, so to increase the thickness to 580mm (double wall and column modules). By doing so, the final results of the analysis show a maximum compressive stress of 0.745 MPa, tensile stress 0.178 MPa and displacement of 15.6mm, which is below the allowable limit of 18.9mm (0.003*6300mm). The utilization in this case reaches a value for compression of 60.3%, which is still satisfactory. One observation for the iwan is that the analysis through Karamba gave a larger deviation from the hand calculations. The reason for that might have been the simplifications while defining the geometry and the mesh subdivision, but also the small deviations in the load calculations and how they are applied on the mesh faces.

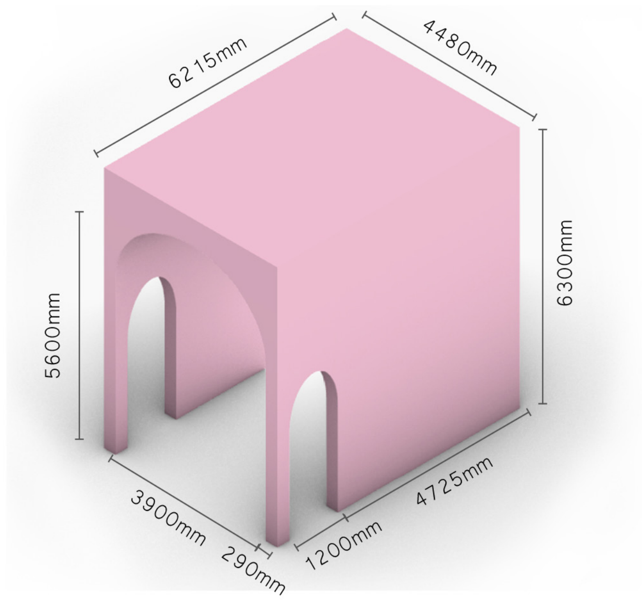


Figure85: General room dimensions

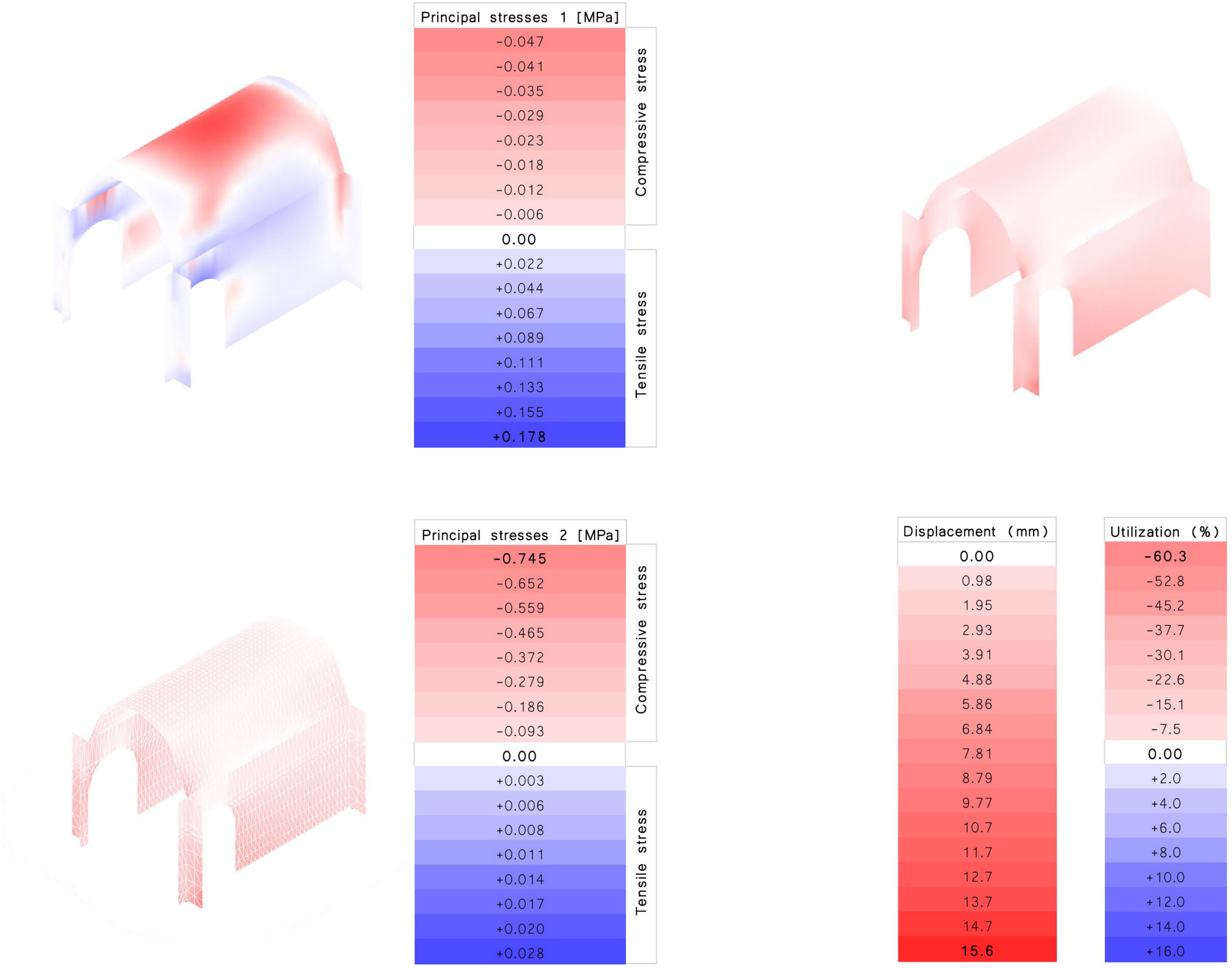


Figure86: Structural anaysis results

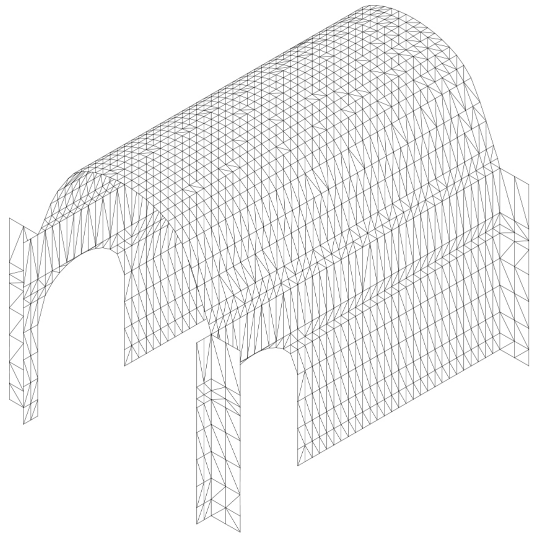
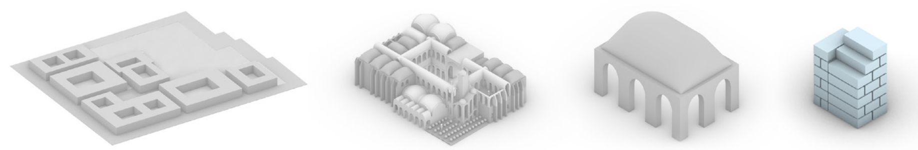


Figure87: Geometry and mesh subdivision



CONCLUSIONS AND LIMITATIONS

As a conslusion, it can be considered that most of the ground floor room structures show sufficient structural performance within the allowable limits and using the “adobe + straw” brick type, which is easier to make. However, when building either double height units or adding a second floor on top of smaller units, the tensile stresses and displacement values exceed the limits in some cases, which can cause critical deformations making the structures unstable and unsafe for use. As explained in the separate cases, due to architectural and construction limitations, it was not always possible to change either the room spans and heights or the wall and column thicknesses. Other solutions needed to be considered as increasing the roof thickness or locally changing the brick type. The last one could be used only where it is needed and keep the low strength brick type as a basic option for the simple single floor structures. However, one additional solution that can lead to certain structural reinforcement without changing the above, can be to add external buttresses for additional support on the walls that are not adjacent to other rooms, which might be more in favor, in order to keep the rest universal for most of the types.

Further limitations were imposed by the meshing subvisions and the Karamba software. Because of the way the software interprets the meshes, various meshing subdivision and simplifications were tried in order to reach one that is closer to the desired result and stress distribution on the mesh faces. Especially, since the roof and wall thicknesses were different, even though the values would be more or less similar with the ones when joining the meshes, the contact edges would not have a smooth transition when visualizing the principal stresses. So, even though the structure is composed by different elements with different cross sections, this distinction is not always acceptable by Karamba. Also the meshing had to be done as a whole, because all the vertices needed to be connected to each other for the force distribution. In the end, the separation was done after the mesh subdivision, however most probably there should have been a more optimal way to approach it, which there was not enough time to try further. Because of this, in some cases the mesh refinement was not uniform everywhere and there might have been some peak stresses concentrated in certain areas where the mesh was subdivided in a higher detail.

Another limitation concerns the analyzed geometry, where Karamba does not allow to assign a direction to the thickness assigned to the elements. This was a problem in this case as the ceilings rest on half of the wall below and not in its interaxis as analyzed by the software, whereas the cross-section of the column is not always correctly assigned and partly falls in the opening, since it is offset symmetrical to the assigned mesh face. A solution to this problem can be to create a two-dimensional model by transforming the applied loads into linearly distributed loads.

Some discrepancies, in the end, might have occured in comparison to the hand calculations, both by simplifying the geometry used, in order to be easily calculated by Karamba, and by the additional wind load and slightly different approach in calculating the sand load.

In conclusion, the results of the analysis carried out, considering some unavoidable approximations, were considered satisfactory. The design can, however, improve with less use of the material in some cases, if the limitations described here are taken into consideration and overcome, while some structural improvements are realized and customized locally per case unit.

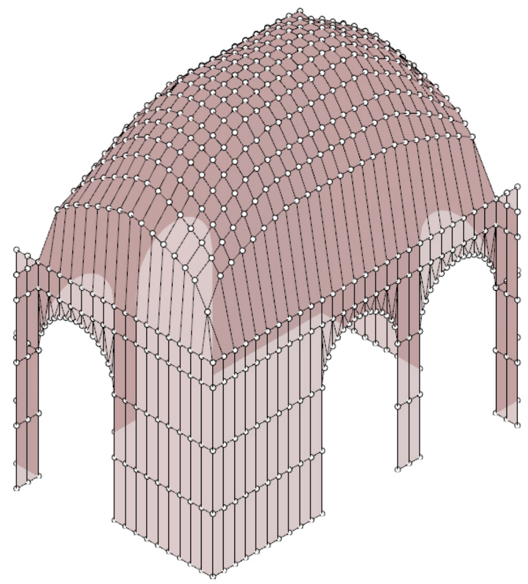
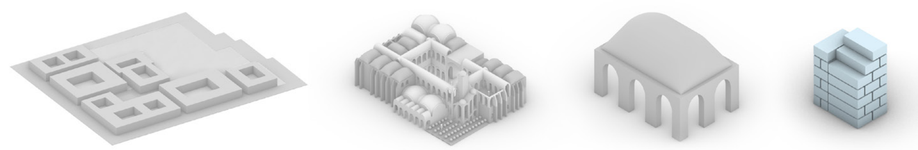


Figure89: Connection of all vertices with each other and importance of not having a vertex unconnected for the force distribution from node to node, which would lead to errors

Figure88: Non-smooth stress distribution gradient when the meshes are seperate elements in Karamba

Figure90: Symmetrical assignment of cross-sections to the elements by Karamba leads to problematic connections from roof to walls and not proper volume placing of the columns, conflicting to the openings



KIT OF PARTS

The kit of parts contains the adobe bricks, the columns, the walls and the roofs. The walls and roofs are composed of different parts. Finally, there are exploded views, where all components are merged to form the different rooms.

BRICKS

The kit of parts starts with the smallest pieces, namely the adobe bricks. The size of the standard brick is derived from the grid size. The grid size used is 1500 mm. The standard brick has a length of 290mm and the joint has a width of 10mm. Together this is 300mm. This means that 5 bricks and 5 joints fit in 1 grid module.

In addition to the standard brick, there are also various other dimensions. There is a 3/4 brick for connecting the walls to the columns. The half brick is used in the columns. The 1/4 brick is used as end brick for the openings. For the roof have a brick that is 50mm slightly narrower than the normal brick. This gives extra flexibility. Finally, there is also have a brick for the arches where the thickness is 75mm on one side and 57mm on the other.

All other sizes of bricks fit in the standard brick. This means that the standard mold can always be used for the other brick sizes, by filling the mold with some filling material.

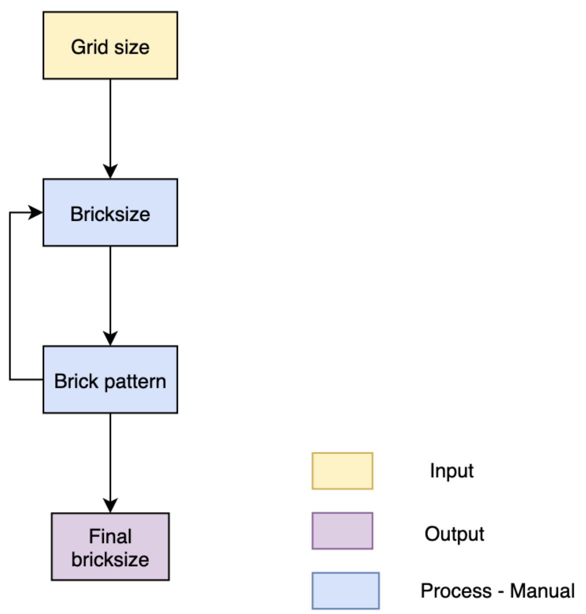
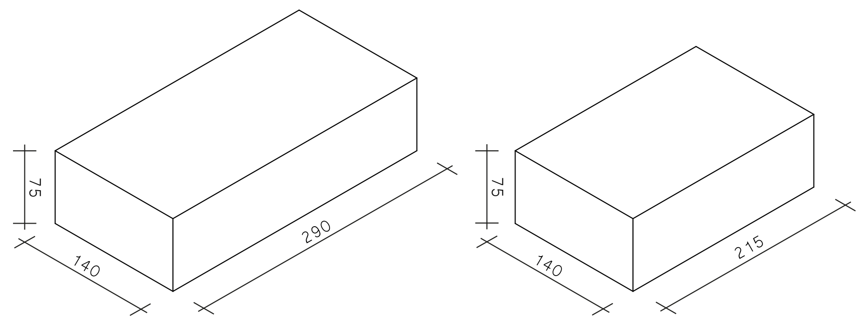
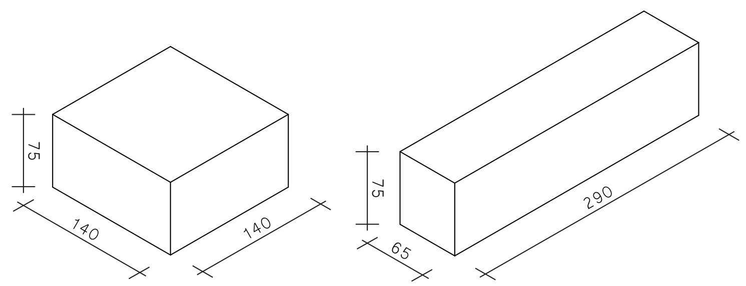


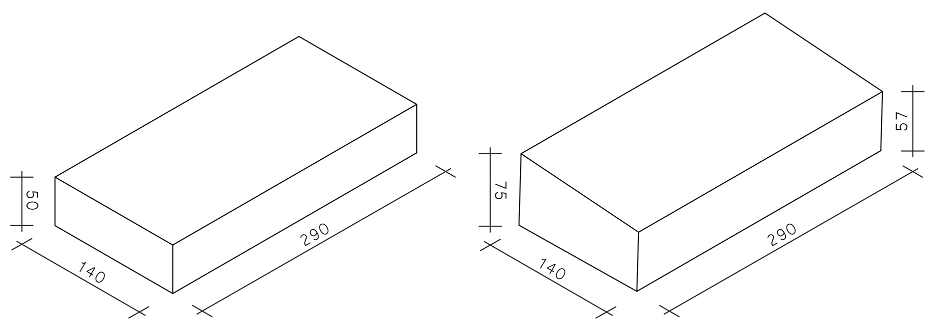
Figure91: Flowchart bricksize



Bricktype A	Bricktype B
Normal brick	3/4 brick
Walls & columns	Walls & columns
4,6 kg	3,4 kg



Bricktype C	Bricktype D
1/2 brick	1/4 brick
Columns	Openings
2,2 kg	2,1 kg



Bricktype E	Bricktype F
Roof brick	Arch brick
Roofs	Arches
3,1 kg	4,0 kg

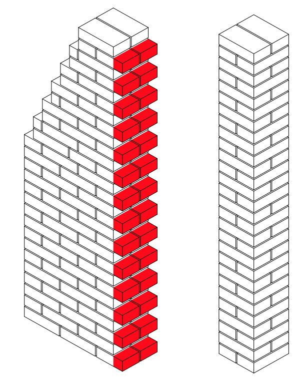
Figure92: Bricktypes

COLUMNS

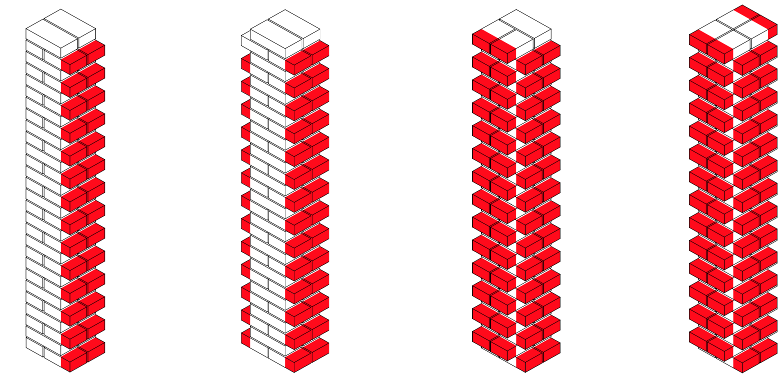
Several columns have been designed. The columns with a size of 290 x 290 mm are for the normal modules and the columns with a size of 440 x 440 mm for the adobe 2.0.

Some columns have protrusions of 75 mm on one or more sides. This is to connect the columns and the walls with each other.

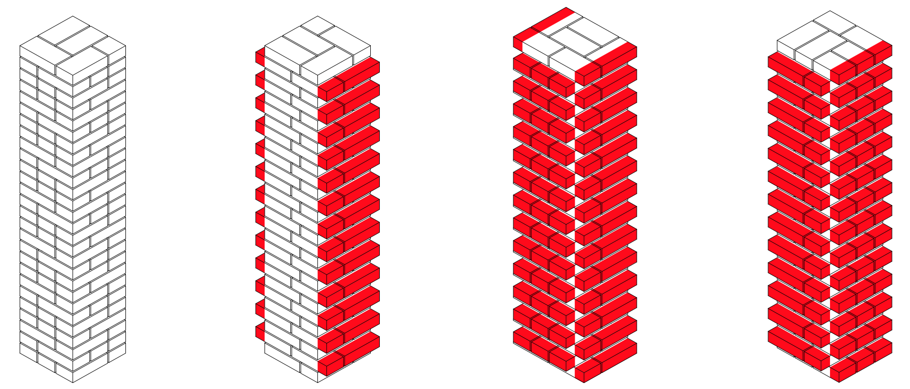
A buttress has also been designed to be used when the construction needs extra stability.



Buttress	Column
750 x 290 x 2380	0 x 290 x 2380
One connection	No connection

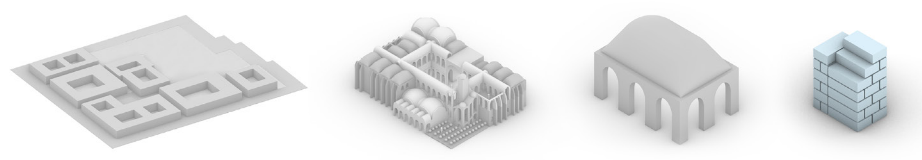


Column	Column	Column	Column
290 x 290 x 2380	290 x 290 x 2380	290 x 290 x 2380	290 x 290 x 2380
One connection	Connection two sides	Corner connection	Connection three sides



Column	Column	Column	Column
440 x 440 x 2380	440 x 440 x 2380	440 x 440 x 2380	440 x 440 x 238
No connection	Connection two sides	Connection three sides	Corner connection

Figure93: Overview columns



WALLS

Just like the columns, the walls have protrusions of 75 mm for the connection of the walls with the columns. There are several wall components. The wall components have a width of 1050 mm & 1200 mm, depending on the location in the grid. In addition to the standard wall components, there are also components with an opening. These components are composed of various parts. There is also a component for filling the openings.

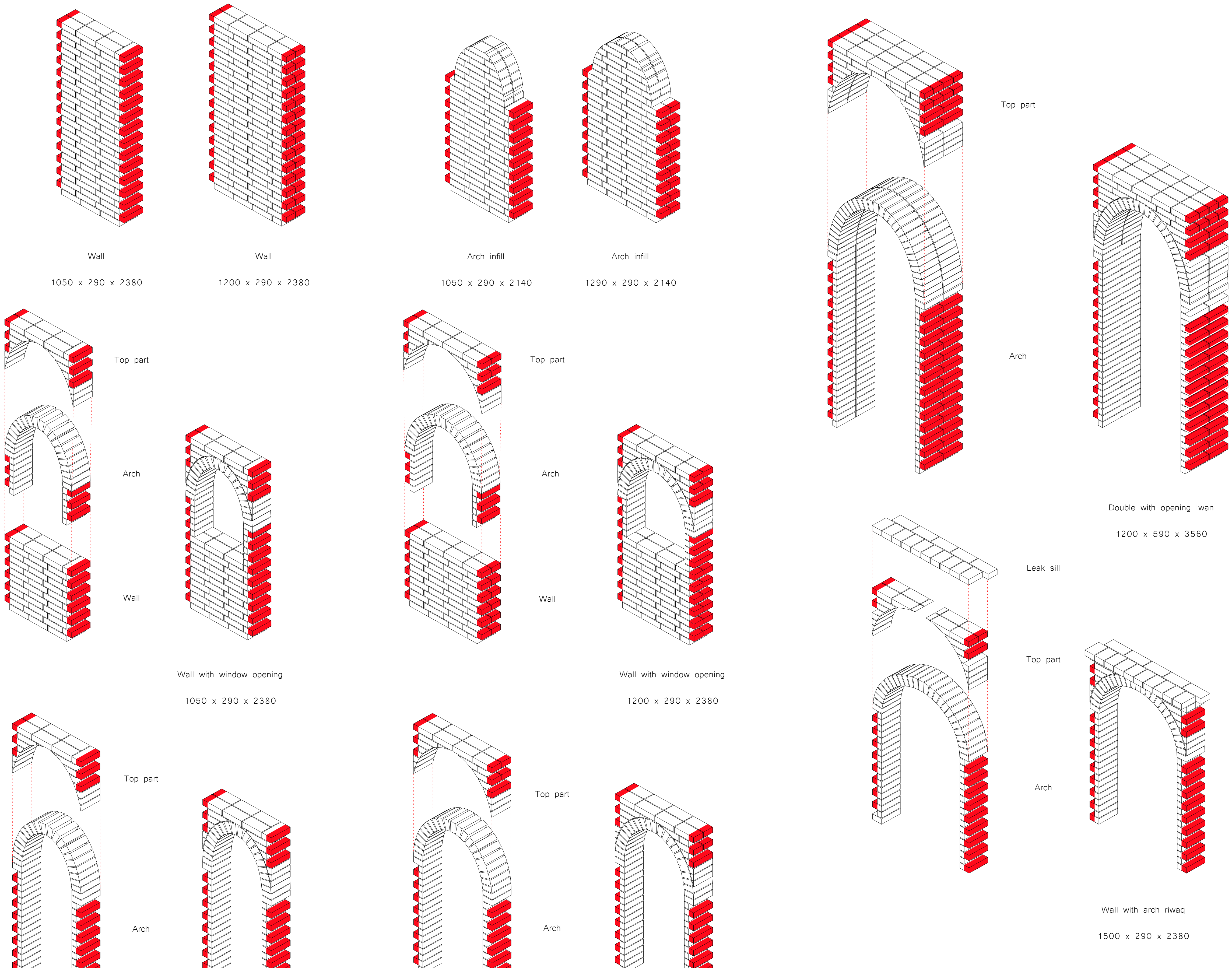
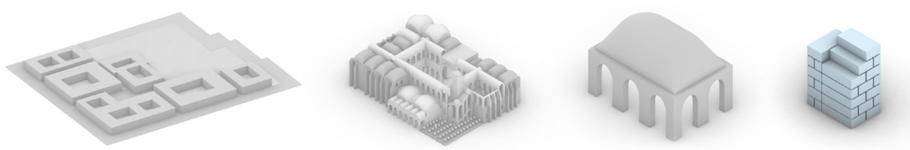


Figure94: Overview walls



ROOFS

Just like the walls, the roofs consist of different parts that together form a roof component. There is a roof component for the living room, riwaq, bedroom/kitchen, iwan and the 2x2 module.

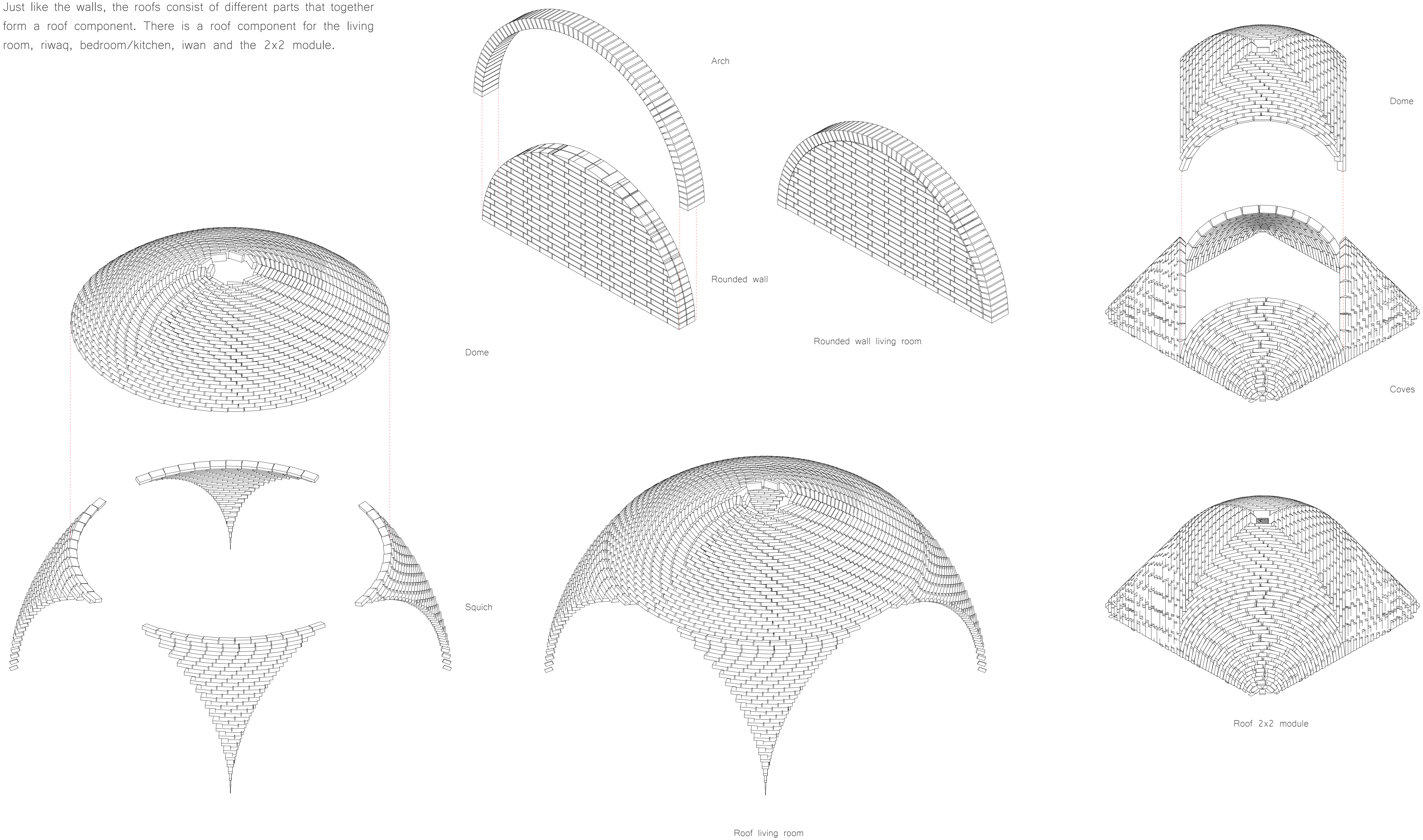
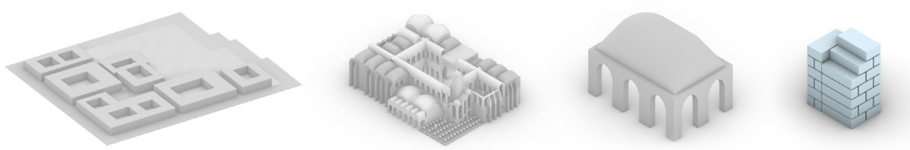


Figure95: Overview roofs



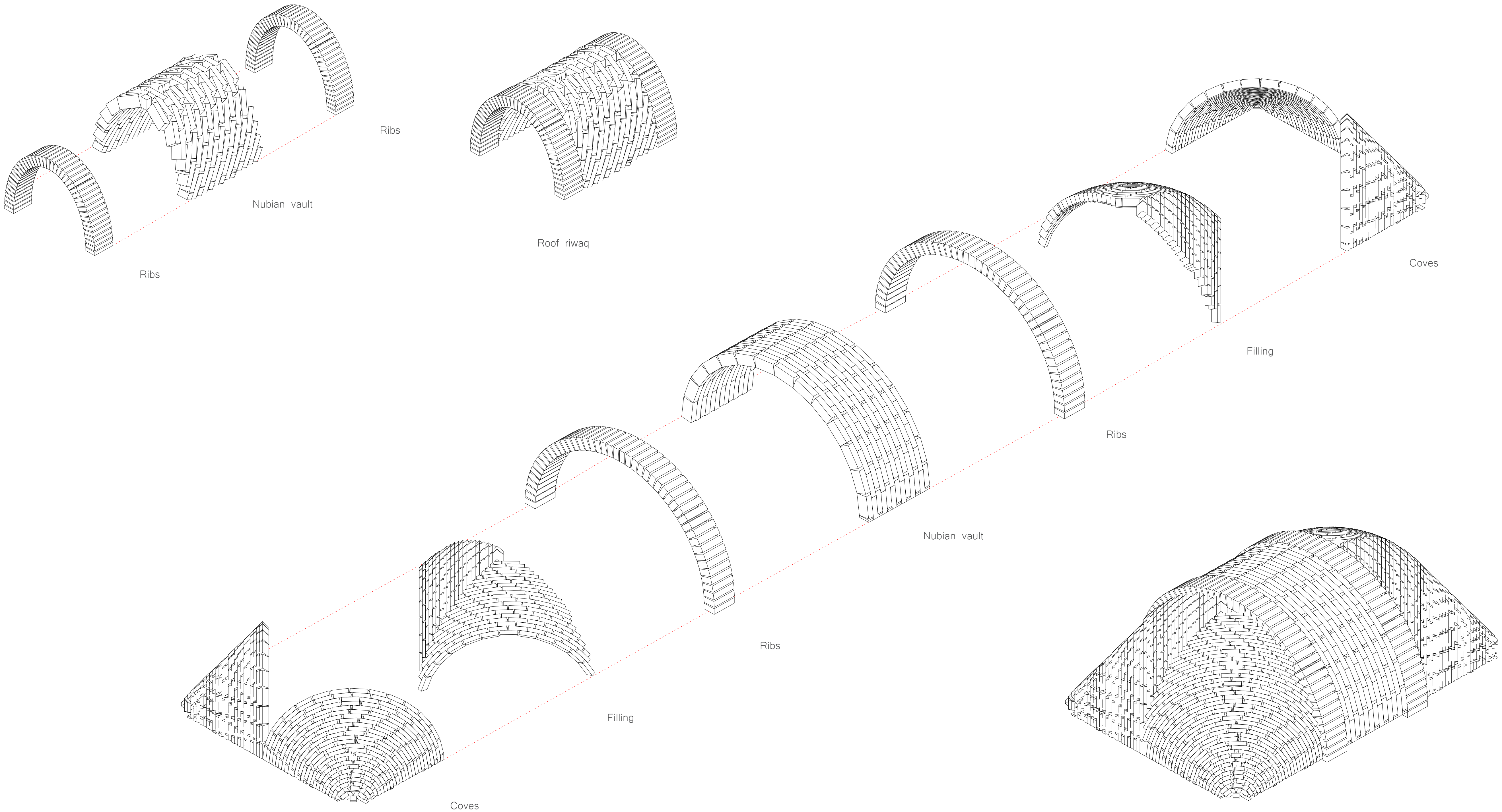
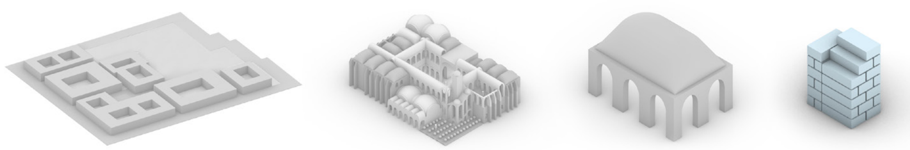


Figure96: Overview roofs



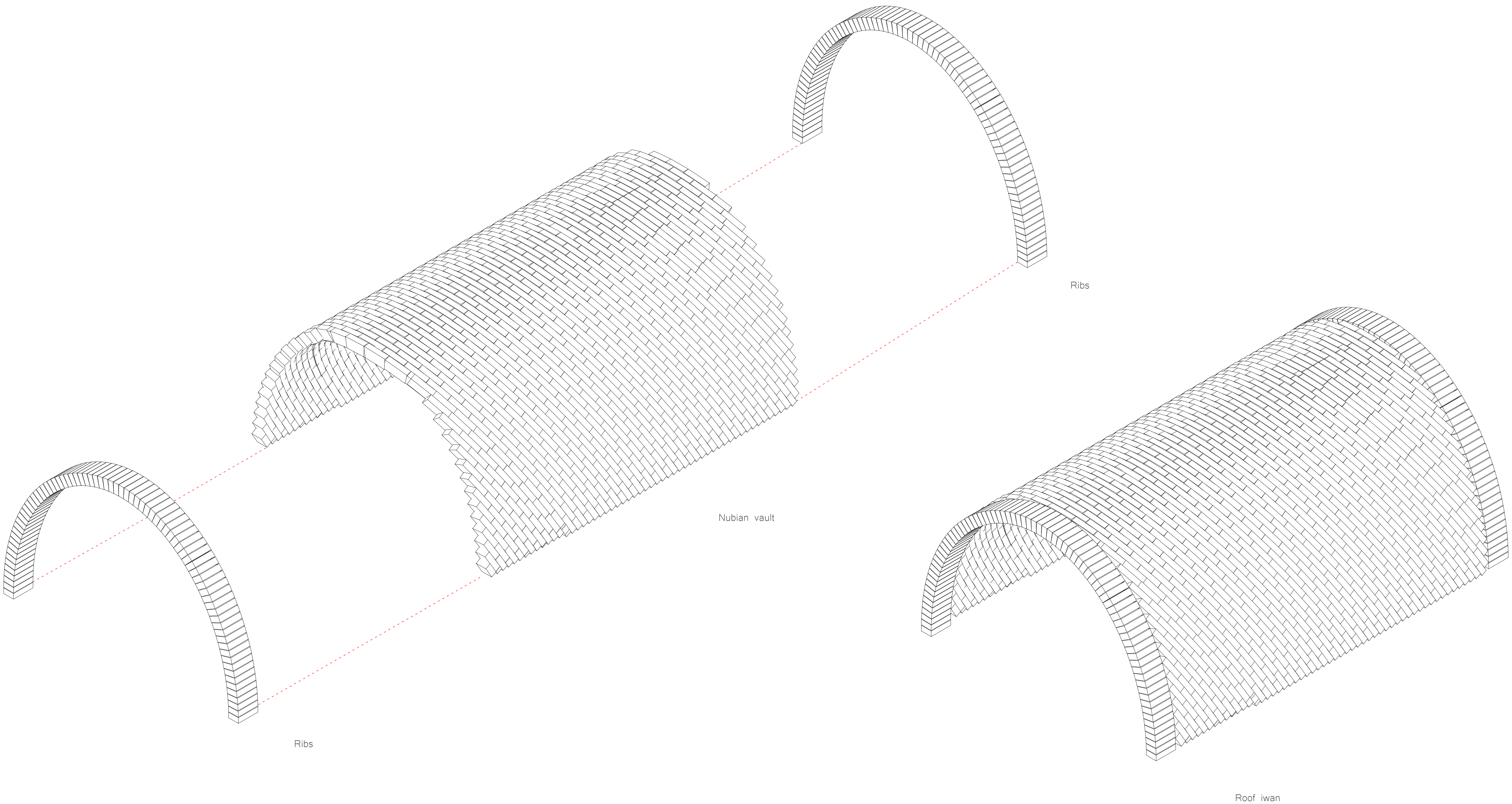
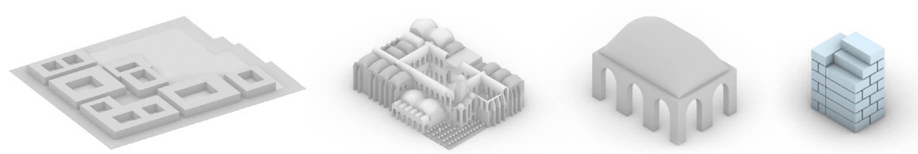
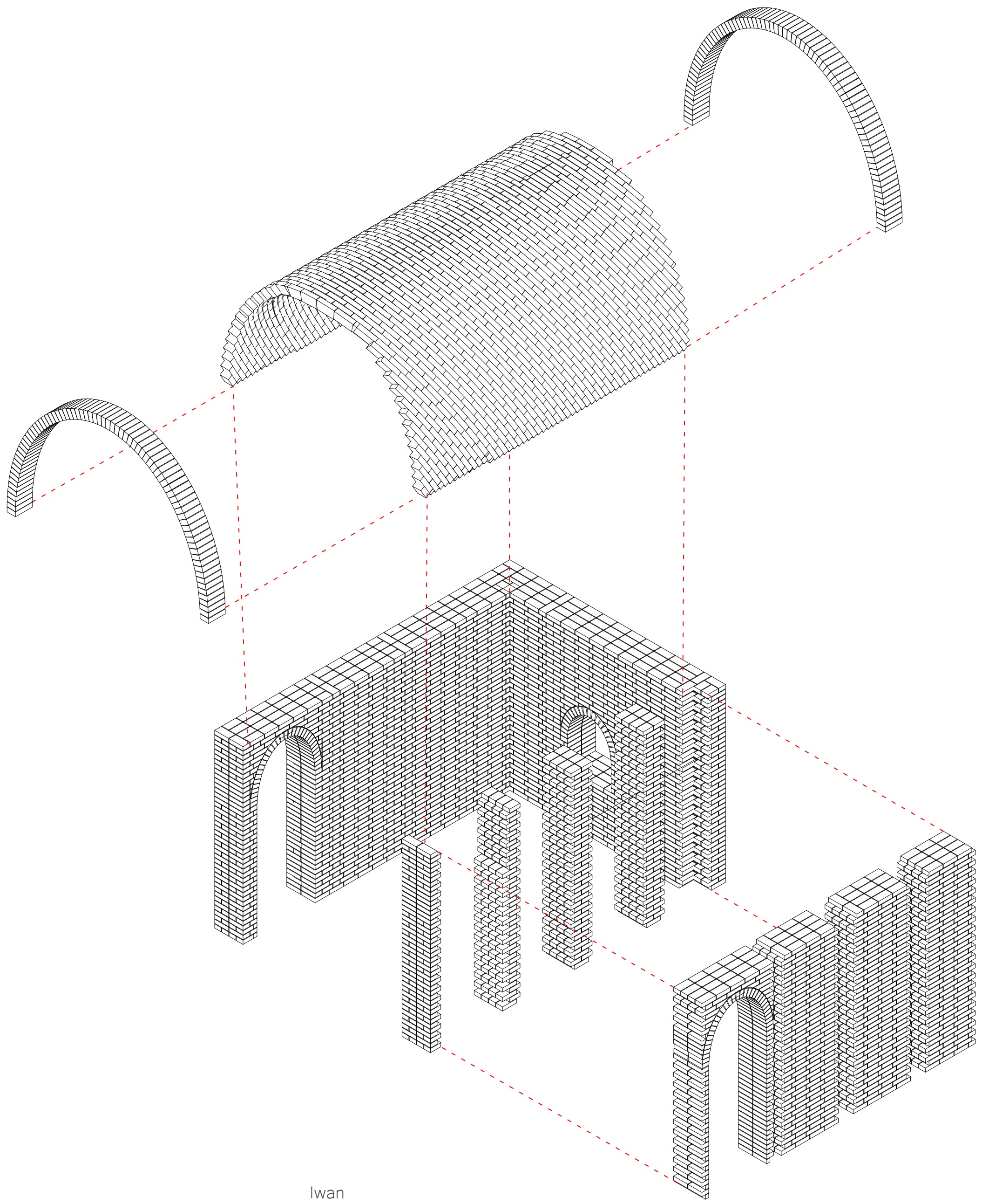


Figure97: Overview roofs

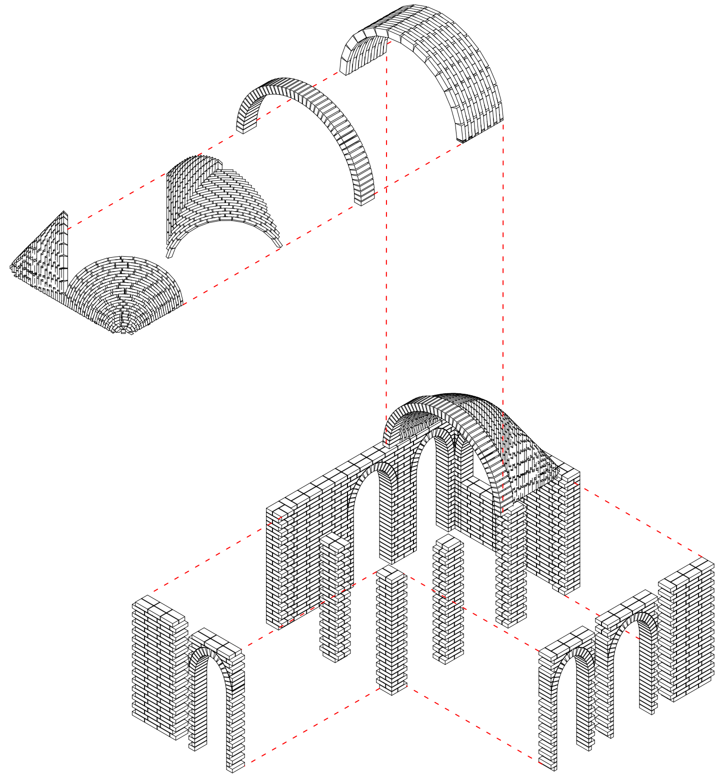


EXPLODED VIEWS

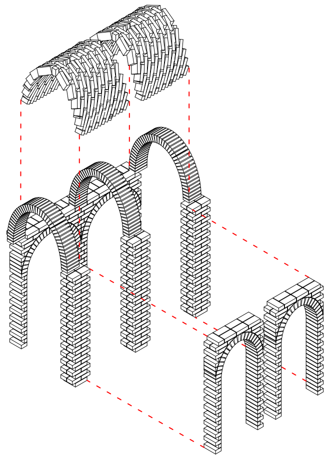
In the exploded views, the final results can be seen when all columns, walls and roofs are connected.



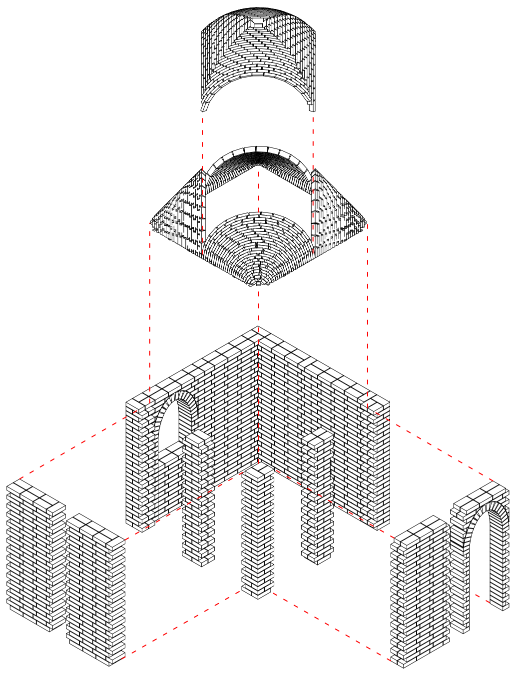
Iwan



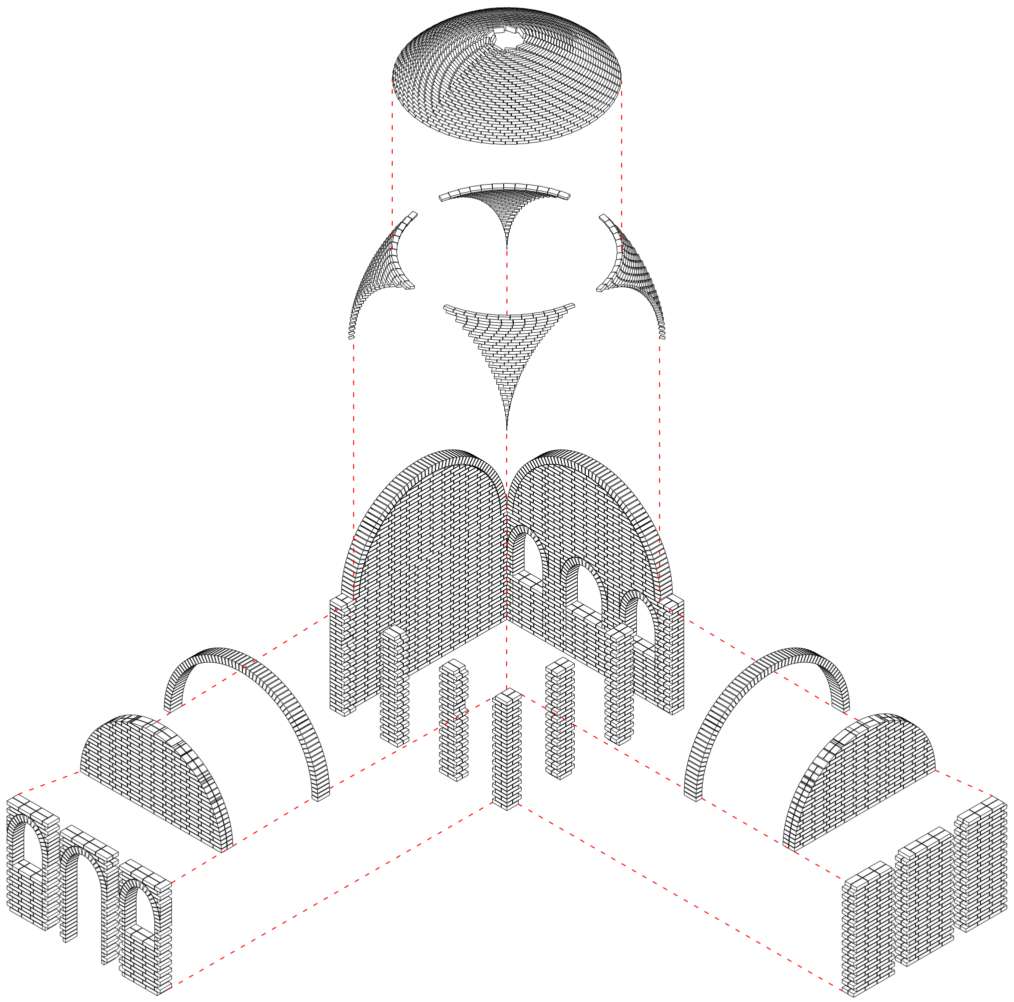
Bedroom/kitchen



Riwaq

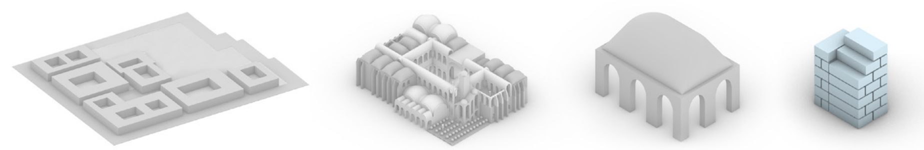


2x2 module



Living room

Figure98: Exploded views



CONSTRUCTION

After analyzing the roof shapes structurally, the construction phases of the spaces were studied. The bricklaying for the units is done with a grasshopper script to generate different modules that are structurally stable by themselves and can be configured to make the whole room. The brick sizes and shapes are designed to make the construction process simple for the mason.

The context of the project provides limitations in terms of resources available for construction of any sort within the Al Zaatari camp. The construction of the housing units should be done with minimal or no supporting material and in techniques that can be easily replicated throughout the camp by the unskilled labour. Simple and modular tools are designed which can be easily created on the site.

Since the different spaces have different requirements such as a second level on top or a double-height spaces, all the units are designed to be structurally independent with a wall thickness of 300mm. Any additional transfer of force because of the second level is dealt with addition of buttresses on the outer perimeter of the housing unit.

CONSTRUCTION SYSTEM

Various construction systems were studied to construct domes and vaults with minimal or no form work. Iranian vault on squinches or a trompe d'ángle vault system is looked at for the construction system of the room module. The process allows construction of individual components which can be combined in different ways to create the desired roof geometry. The design of the squinch corner can support itself on the two adjacent walls in a nubian way and can be used for construction of both, a dome on a square plan or a vault on a rectangular plan.

The vault is built with arches courses staring from the four corners. The surface of the masonry plane are diagonally inclined towards inside. In this way, every single course describes the curve. Every new course is a stable arch as soon as it is completed, and the units of the next course can be set on top. The vault of this type can be built free hand without any form work or centering (Wendland, 2007).

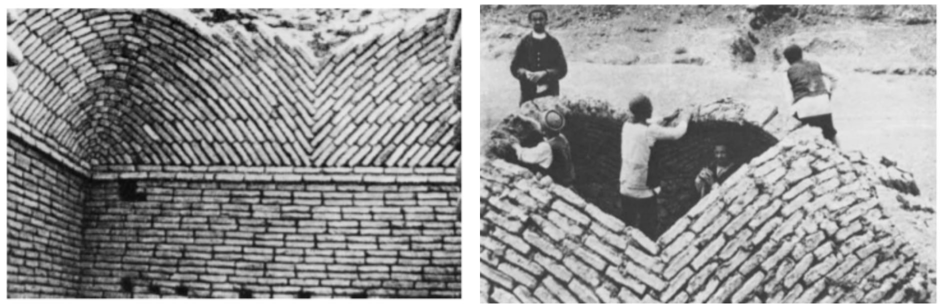


Figure99: Trompe dángle vault built in freestyle
https://elib.uni-stuttgart.de/bitstream/11682/64/1/Wendland_2007_Bam_Vaults.pdf

Figure100: Modern day construction of Iranian vault
https://www.jlconline.com/videos/building-a-brick-catalan-vault-no-falsework

To reduce the material during construction of staircase, examples of making staircase on arches were looked at. The arches not only allows spanning of the staircase but significantly reduces earth filling due to the optimized shape.

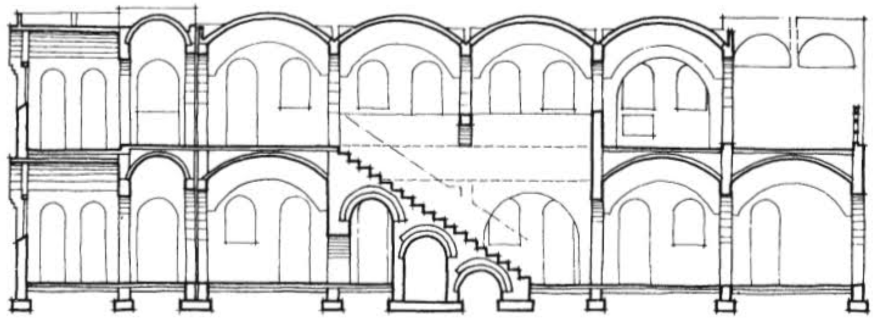


Figure101: Section detail for construction in Earth
http://www.rivendellvillage.org/Building_With_Earth.pdf

Fathy and Fabrizio Carola have both looked into design of a compass to guide the masons for the position of the bricks while making arches or domes. The Fabrizio compass rotates in two directions to even mark elliptical domes. Simple additions to a basic marking tool can result in construction of different geometries easily on site by the unskilled masons.

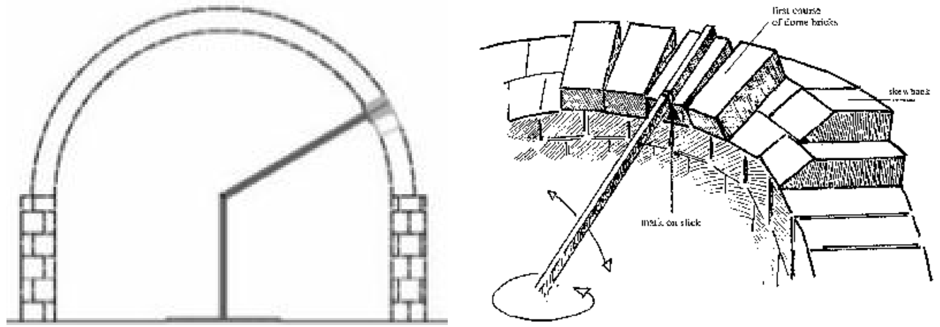
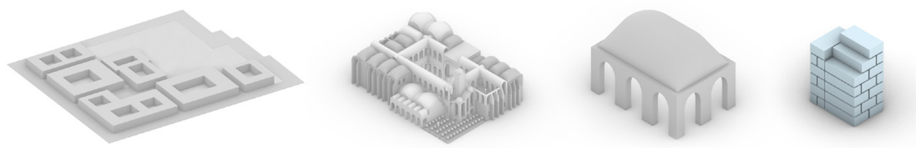


Figure102: Compass design for bricklaying
http://www.rivendellvillage.org/Building_With_Earth.pdf

While constructing a second level, the columns always continue straight and a special connection of piece is designed to span arches at different heights. This acts as guide for column above and also improves structural stability by continuous transfer of loads. Aqueducts are constructed in similar way where columns start on top of each other to avoid any point loads on the arches.

Figure103: Los Milagros aqeduct
Roman architecture and urbanism - From origins to late antiquity



CONSTRUCTION PHASES

A construction manual is created to guide the unskilled labors about the step by step construction phase of different unit. This will also explain the methods to create a tools and guides from locally available materials.

PREPARING THE SITE

STEP 1 - The tartan grid that defined the wall for the units is first marked on the site using chalks and marking ropes. An offset of 1.5 times this width is then excavated up to a minimum depth of 600 mm. The earth dug up will also be used for preparation of the adobe blocks.
STEP 2 - The earth below is rammed and a strip foundation in baked bricks is built up to a height of atleast 300mm to prevent any capillary action in adobe and to provide additional stability to the wall.

INDIVIDUAL UNITS

STEP 3 - The two opposite columns are marked first on top of this foundation. These act as the two points for marking all the openings and columns.
STEP 4 - From the two points, minimum walls, columns and openings are then defined as specified earlier in the structural requirements.
STEP 5 - The walls are raised up to the spring point of the arches for the openings i.e, 1500mm.
STEP 6 - The bricks are then laid for the elliptical arches of the openings using the Archimedes Trammel. The arch is built from the two sides, which each piece resting on top of the previous till the central keystone is placed.
STEP 7 - The walls are then raised up to the spring point of the roof i.e, 2400mm.
STEP 8 - The roof is constructed starting from the four corners as a squinch from the two walls supported in a Nubian style using Brick Type C.
STEP 9 - The angle of the squinch can be checked by a tying a thread at two end and using it as a guide.
STEP 10 - It is built till the point it reaches the next squinch.
STEP 11 - Two ribs are then marked for the central barrel vault using the Archimedes Trammel with the required arm.
STEP 12 - The vault is then filled by supporting every layer on the previous one to close the central part.
STEP 13 - The two coves can then be closed by following the similar bricklaying pattern till the ribs.

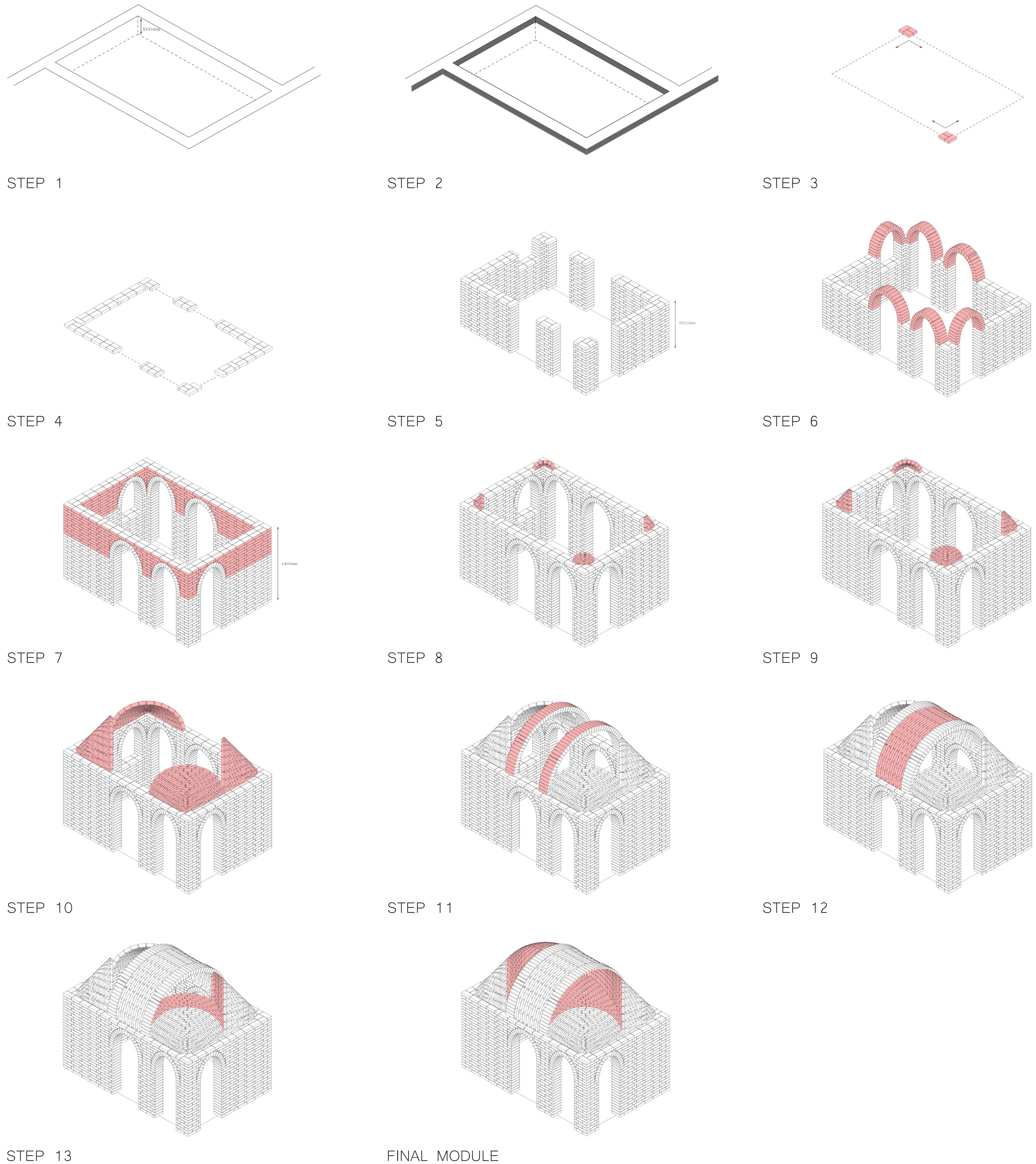
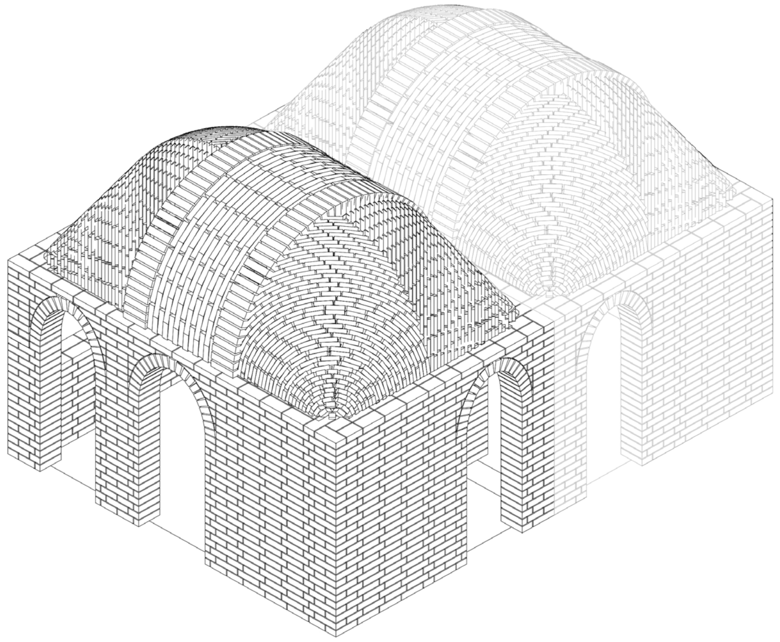


Figure104: Construction sequence for room module

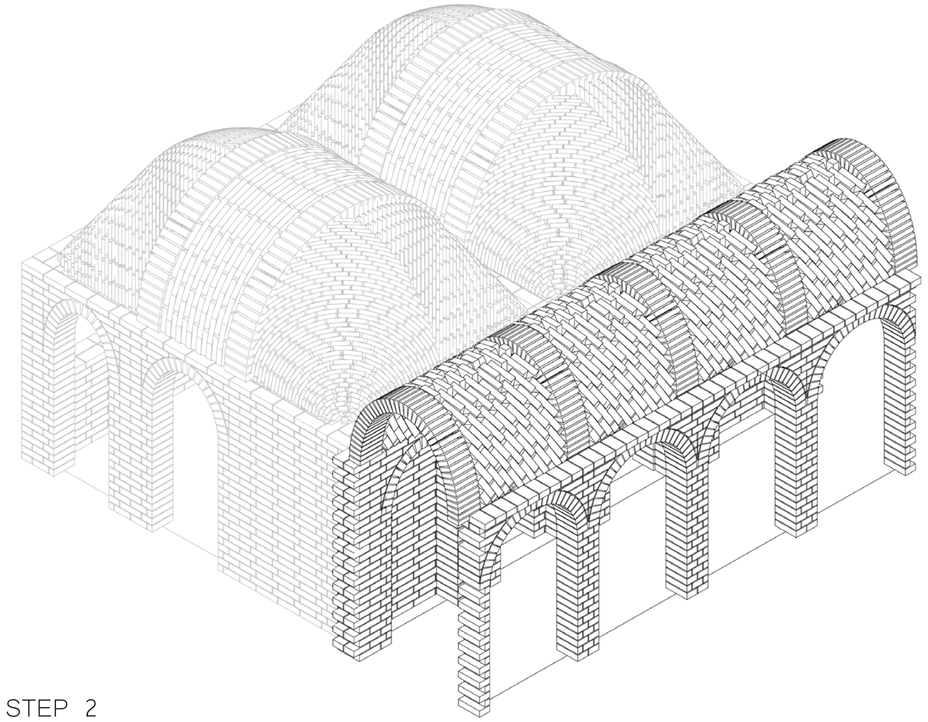
CONSTRUCTION PHASES

SECOND FLOOR ADDITION

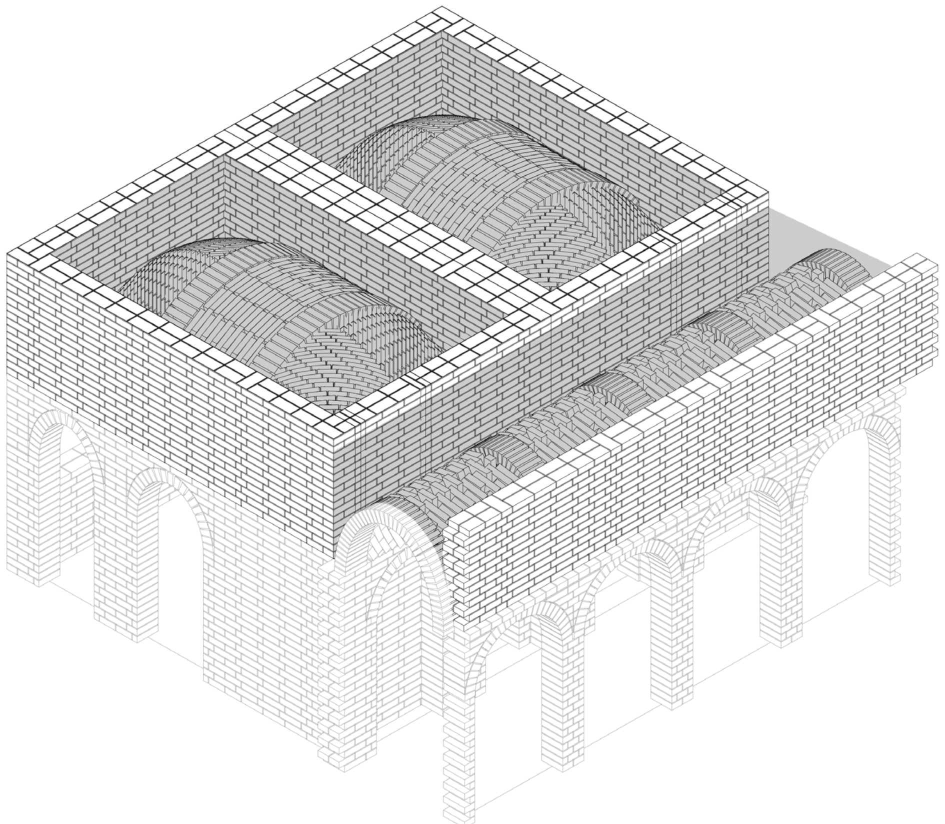
STEP 1 - The room module can be connected to adjacent modules - Room module, Living Room module or Iwan module as per the configuration.
STEP 2 - The rooms can be connected to the Riwaq module to connect to other spaces.
STEP 3 - The walls are raised to support the second for spaces buildable on top. Sand is filled in to add a uniform load on top.
STEP 4 - Second level can be built in the same way.



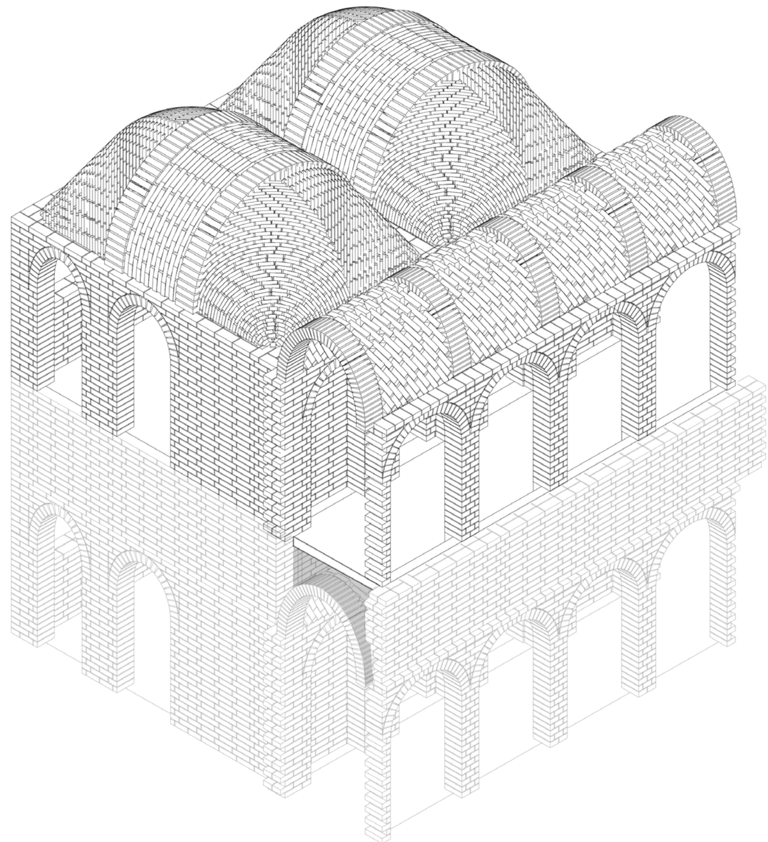
STEP 1



STEP 2

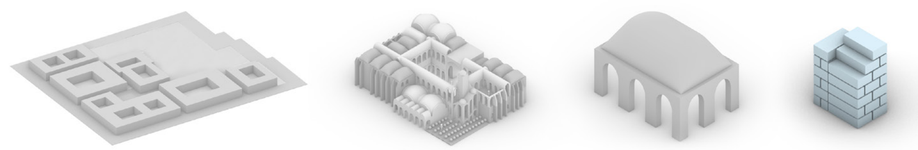


STEP 3



STEP 4

Figure105: Construction sequence for second floor addition



BRICK LAYING

The bricklaying for the various geometries of the roofs is done through a grasshopper script which can be modified for various sizes. Limitation in material strength, sizes and stability guided the bricklaying process.

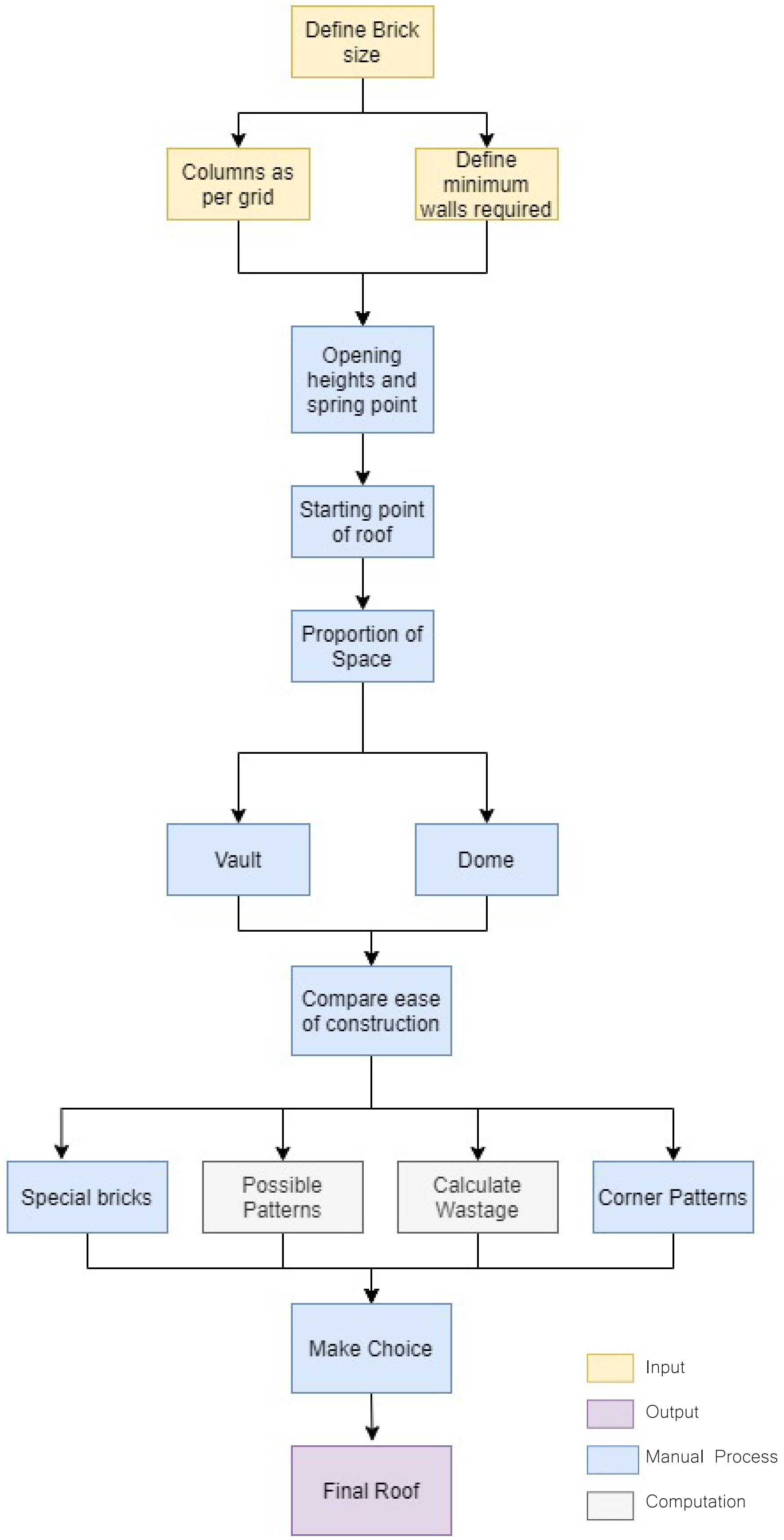
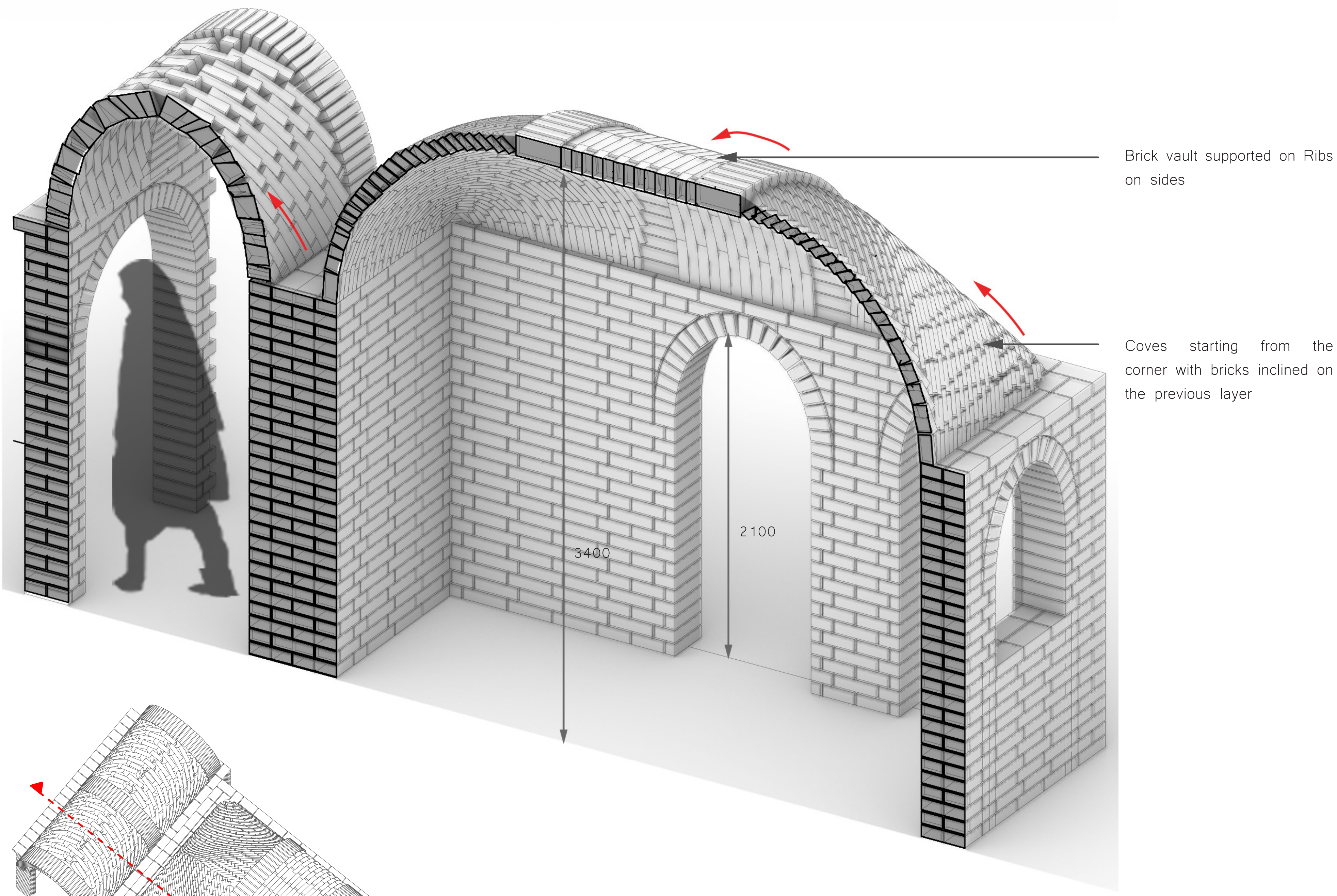


Figure106: Brick laying flowchart



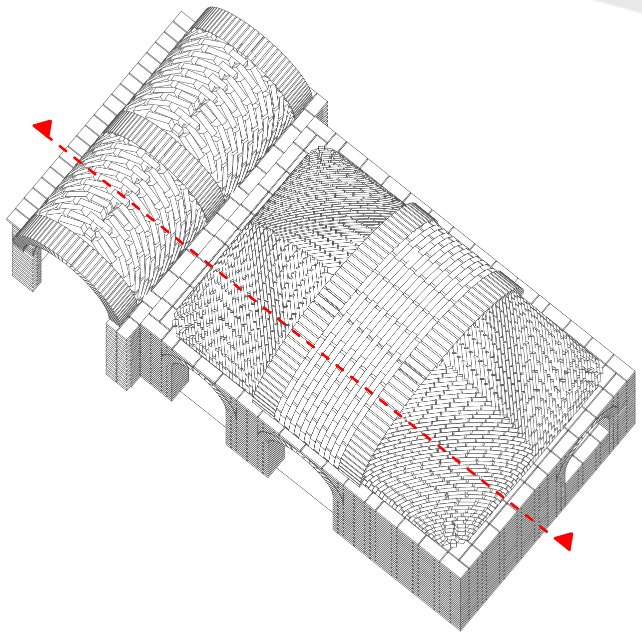
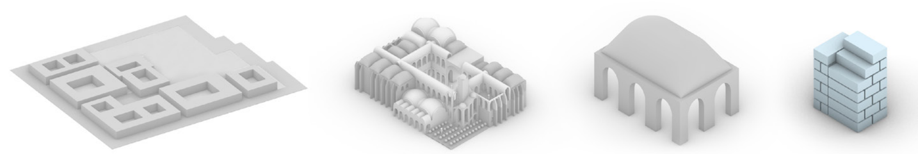


Figure107: Detail showing bricklaying direction for the modules



CONSTRUCTION TOOLS

ARCHIMEDES TRAMMEL

An Archimedes trammel is used for marking elliptical arches on paper. A similar trammel can be constructed on site to mark elliptical arches. The trammel allows movement of tracker arm on two axis of desired length to achieve the desired ellipse. This can also be used for circular arcs by fixing the tracker on the center. It is can be combined with a plate on top to work as a Fathy compass on site to guide step by step position of every brick. The trammel consists of two parts - adjustable and fixed. The adjustable part is the height stand and tracking arm to mark all the elliptical and circular arches on site.

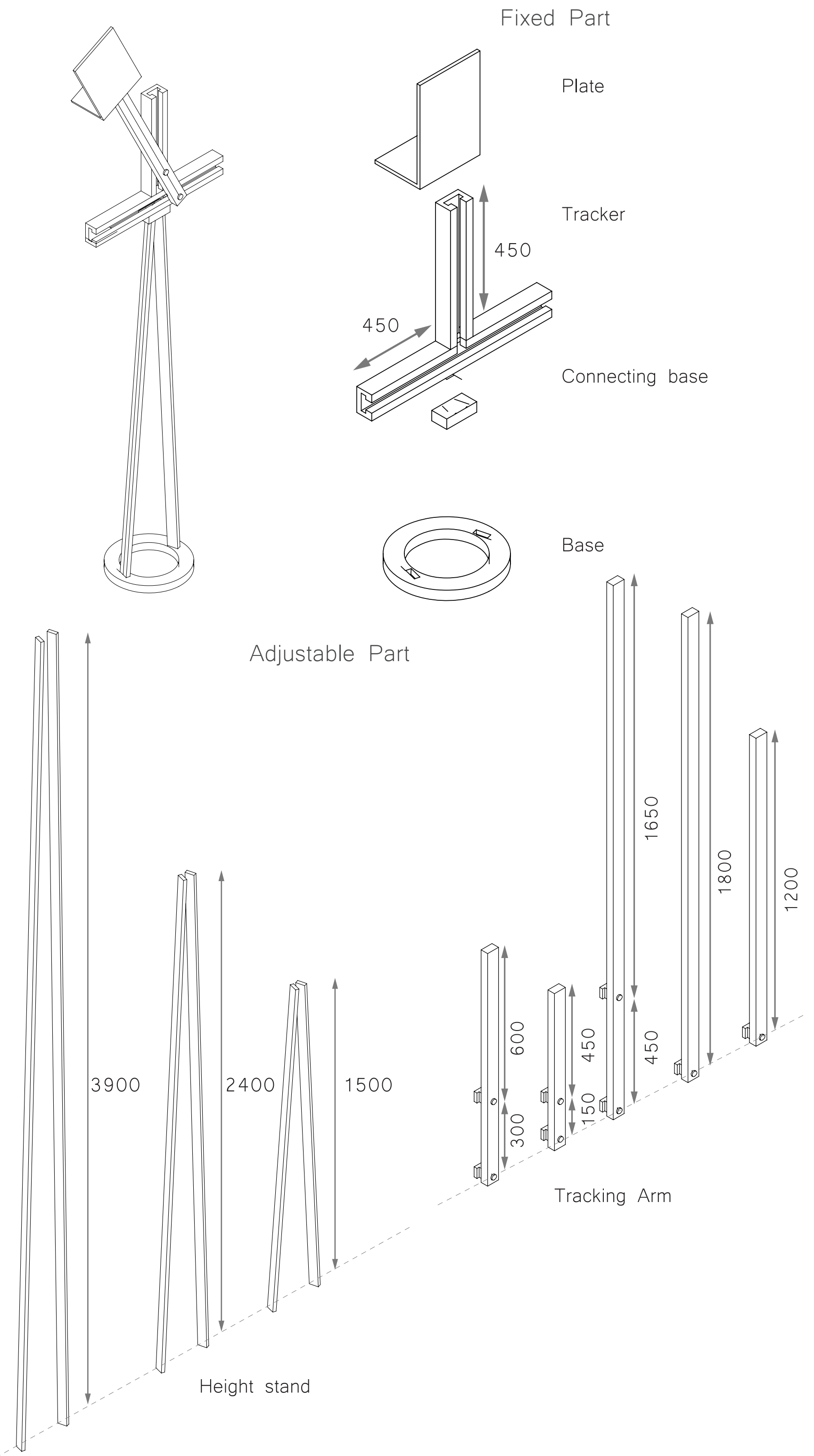


Figure108: Archimedes Trammel constructed on site

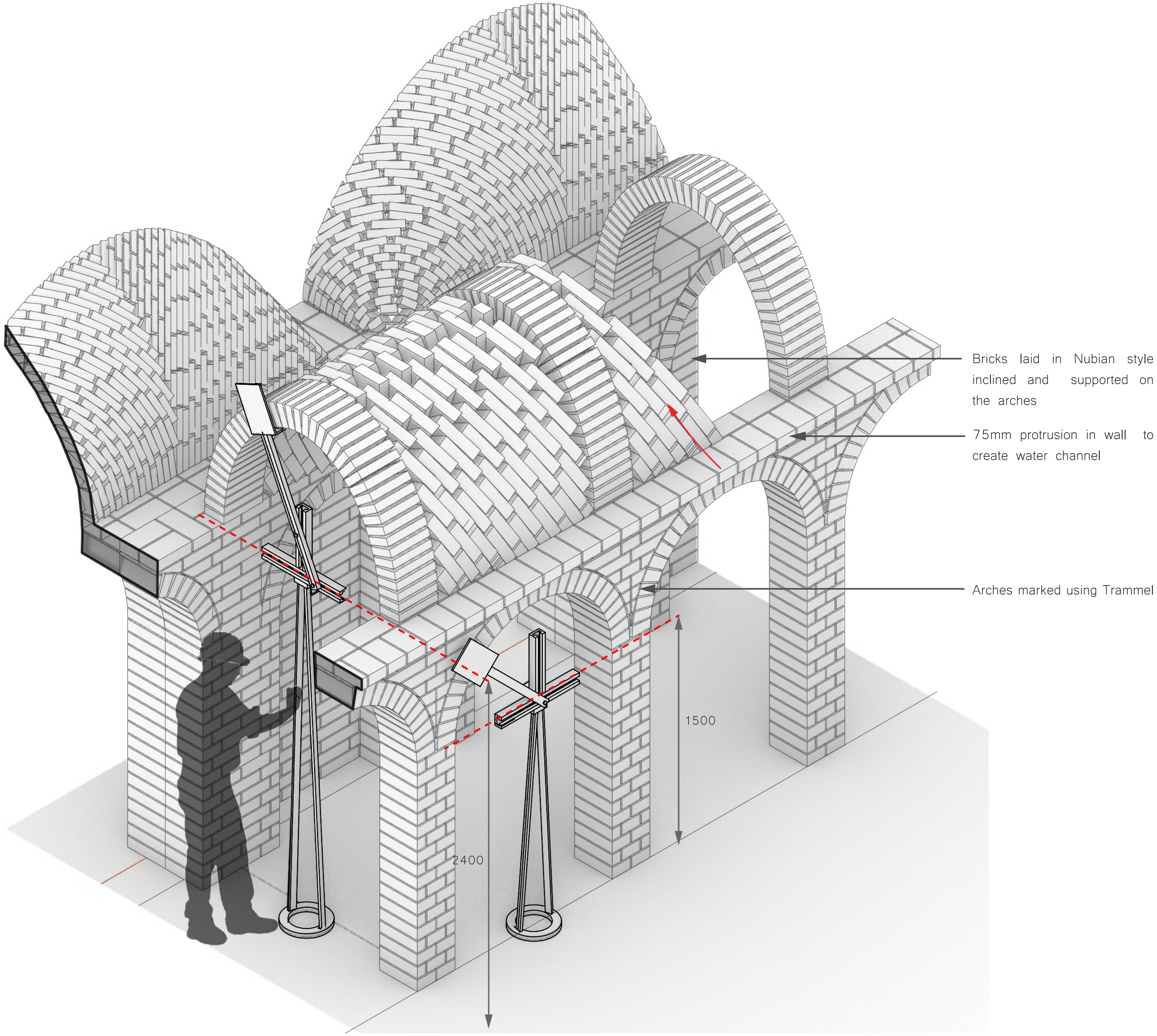
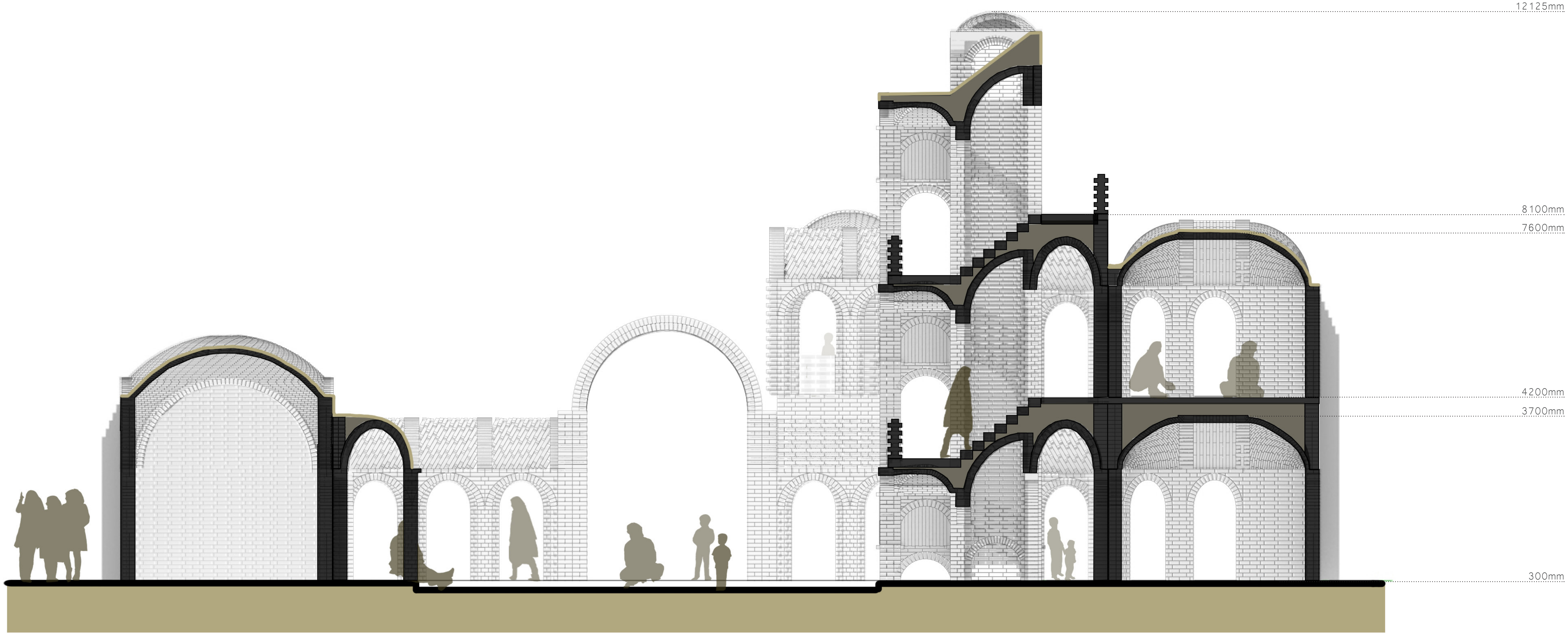


Figure109: Detail showing Riwaq construction next to modules



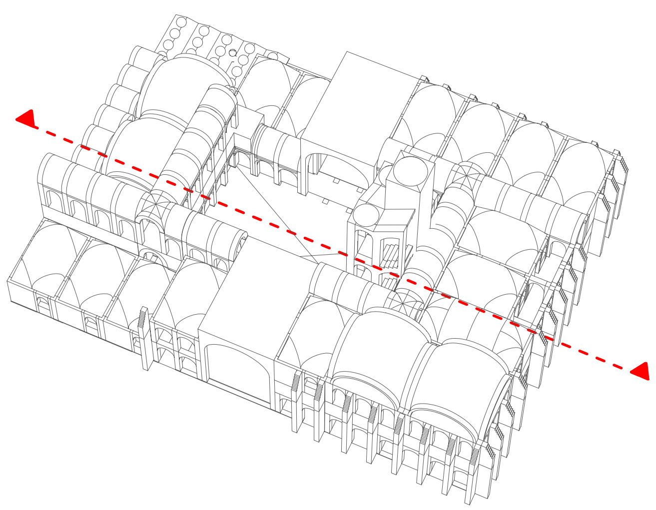
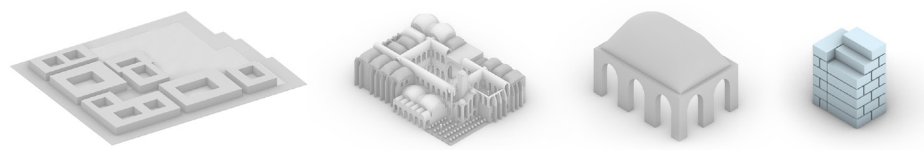
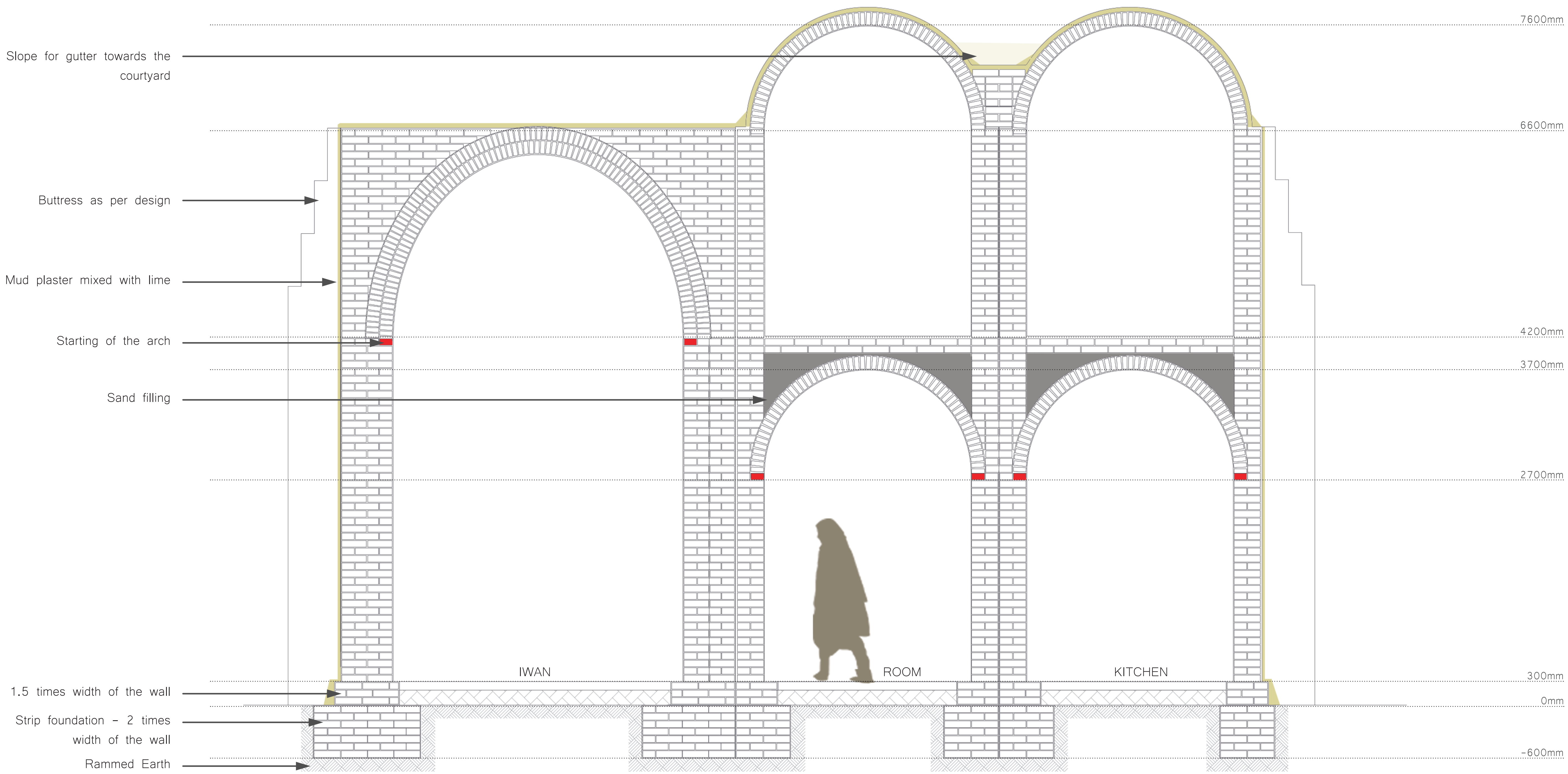


Figure110: Typical section through the dwelling unit





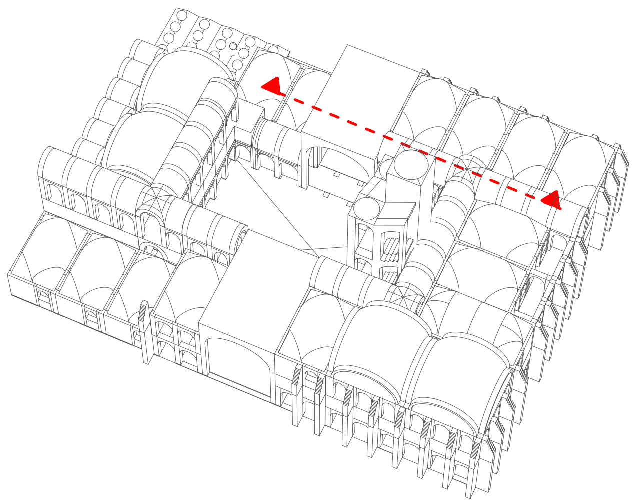
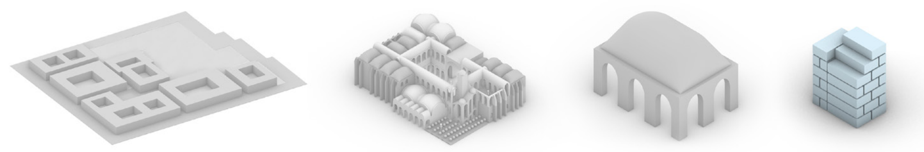


Figure111: Detail section through through Iwan and room module



ADOBE 2.0 - WIND CATCHER

A windcatcher wrapped around by staircase is designed as the challenge for the adobe 2.0. The tower goes as high as 12m and is developed as a system of modular units to ease the construction process. There are arches spanning in the transversal direction and along the perimeter of the tower and a central core which together supports the wrapping staircase.

The wind catcher can be placed in the courtyard to capture the wind at the top level and direct them towards the opening facing the courtyard. A micro climate can be created inside the courtyards with green spaces and water features to cool down this air. The air will then circulate in the spaces surrounding the courtyard through the large size of openings facing it.

CONCEPTUAL DESIGN

STAGE 1 - The tower was initially conceived with incremental arches between the two levels to suit the slope of the staircase.
STAGE 2 - It was then tapered suit the shape for the wind loads and to increase its structural stability.
STAGE 3 - The arches were modified to have the same size and the base plane for its spring point inclined to create a rhythm as the staircase wraps around the tower.
STAGE 4 - Finally, to simplify the construction system and make the arches modular it was designed as a straight tower where buttresses can be added if need be.

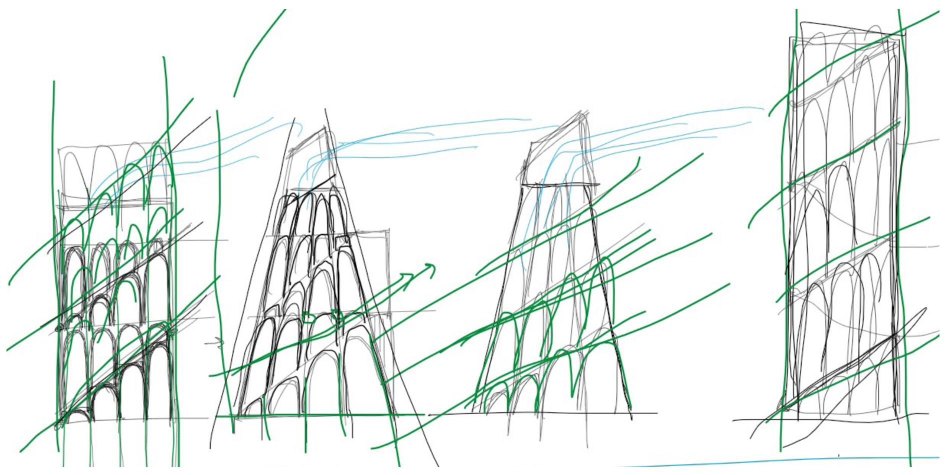


Figure112: Concept design for wind cactcher

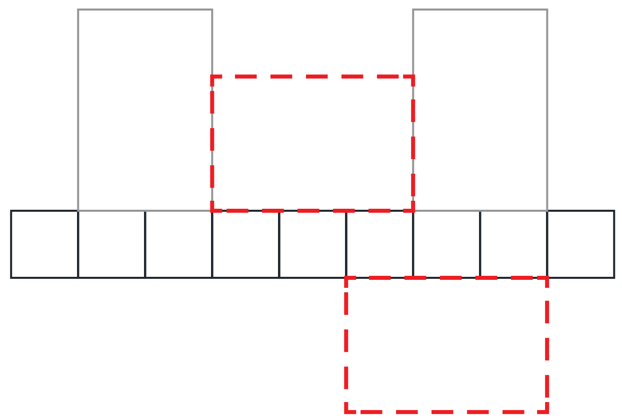


Figure113: Connection of wind catcher on either side of the Riwaq module

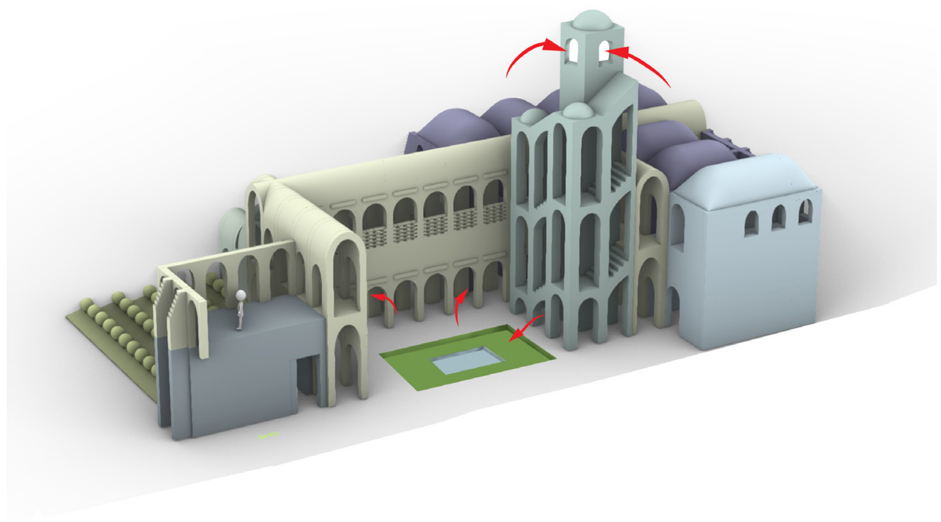


Figure114: Air circulation for wind catcher

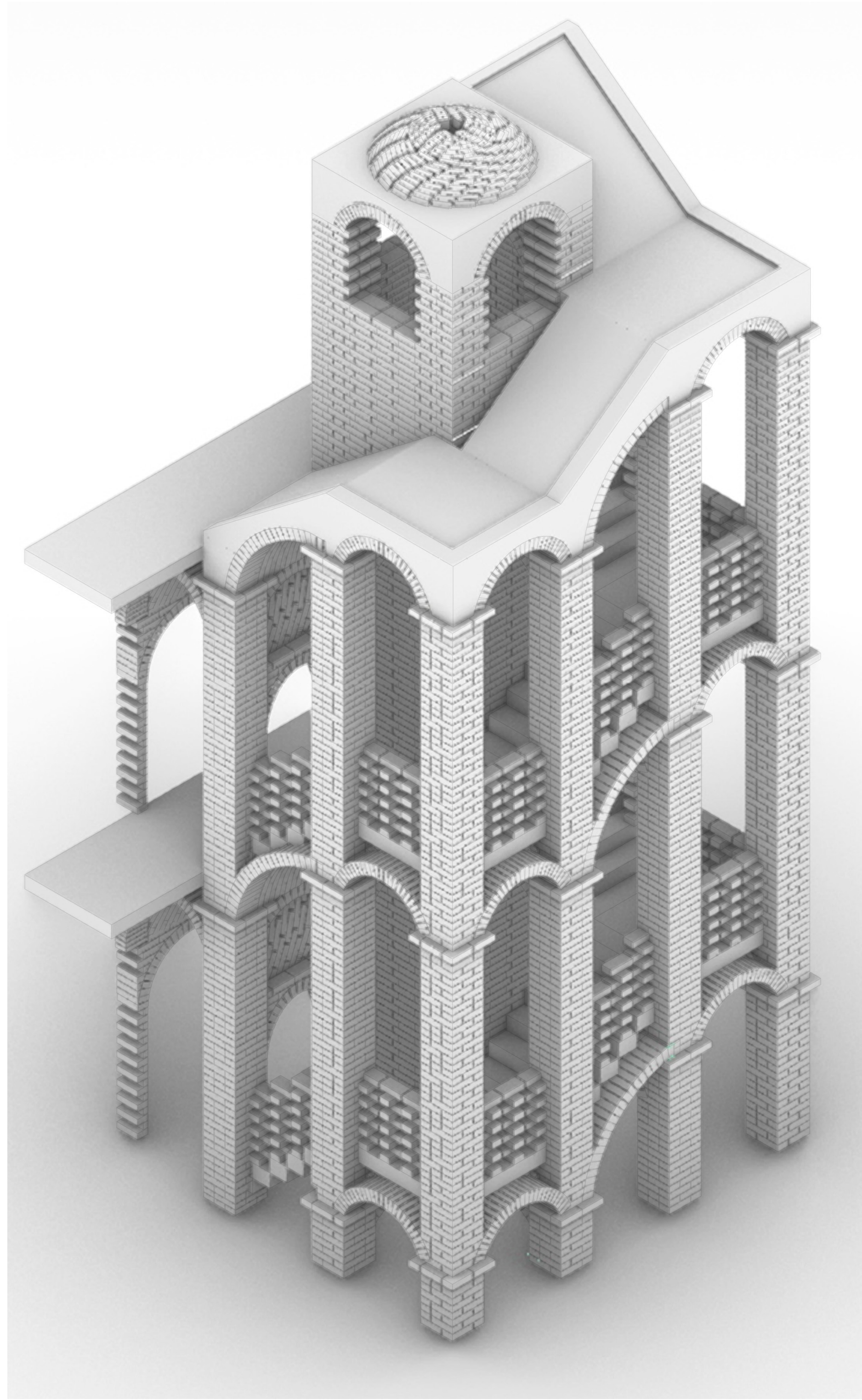
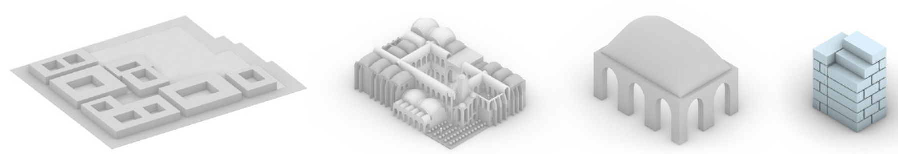


Figure115: Wind catcher



MODULAR DESIGN

The tower is designed as a module of 2x3 units that can fit on either side of the 1x3 Riwaq module.

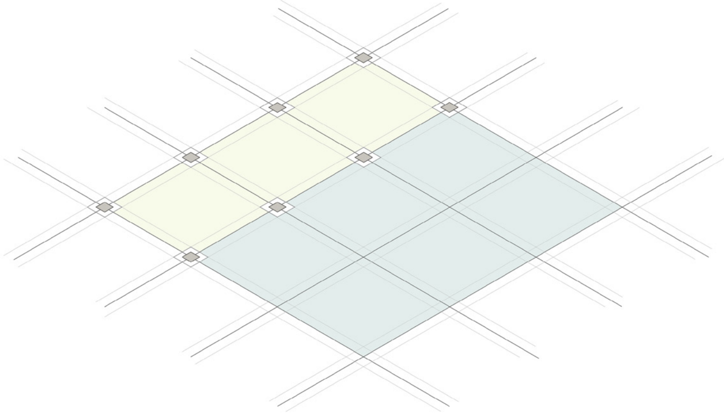


Figure116: 2x3 staircase module connected to Riwaq

The modules are then divided into two categories - Landing Module and Staircase Module. A combination of five of these modules can connect to the next level.

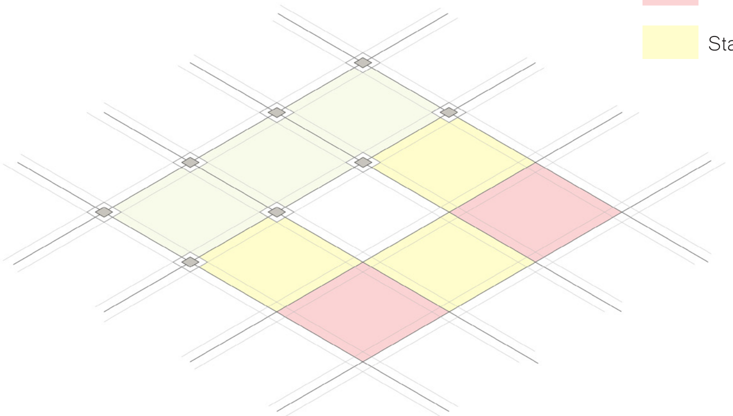


Figure122: Assigning modules into two categories

Columns of 450x450 are designed with a central core of 150x150 that guides the centering for the column above. This allows the brick module of 300 to be placed around in different positions to span peripheral and transversal arches at different levels.

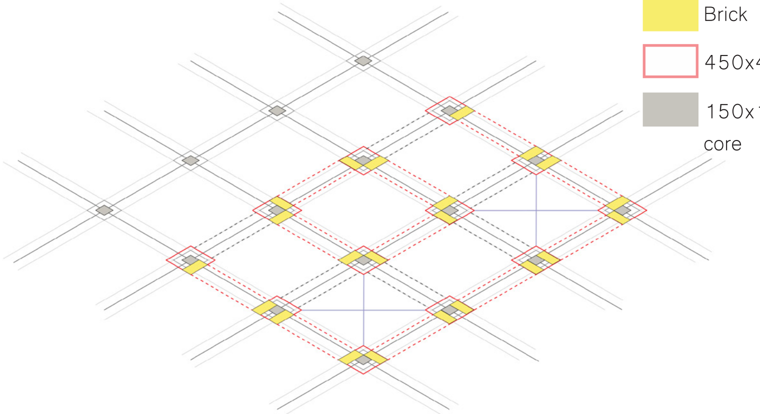


Figure117: Modular arches spanning between columns

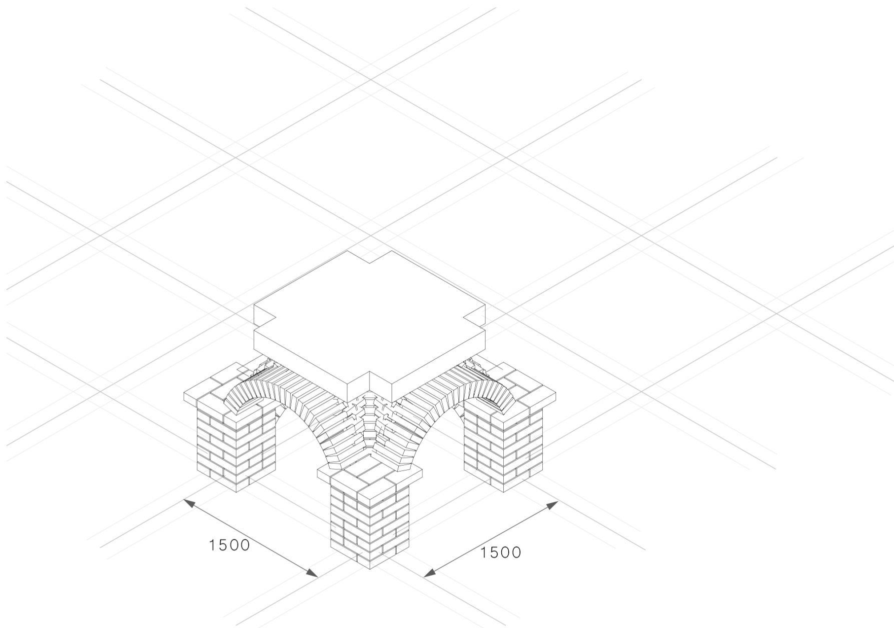


Figure118: Landing Module

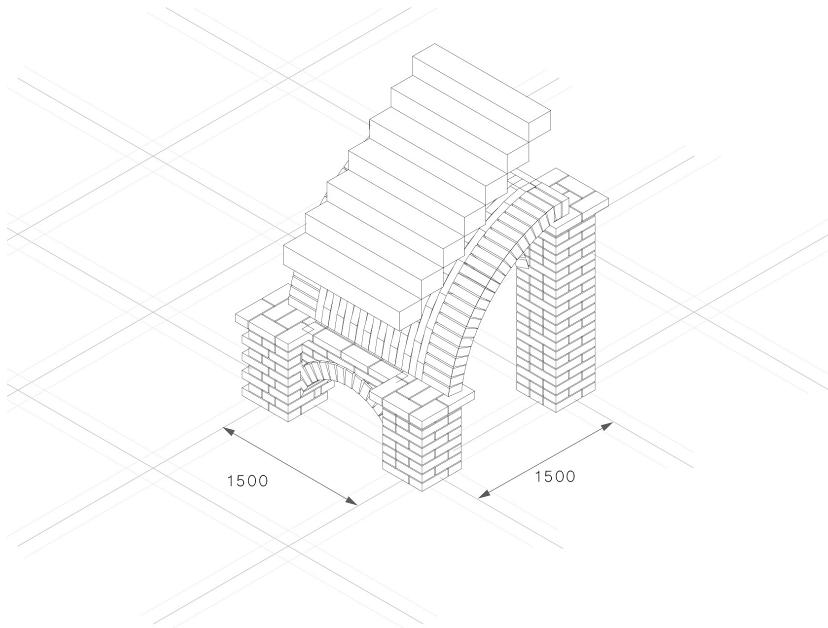


Figure119: Staircase Module

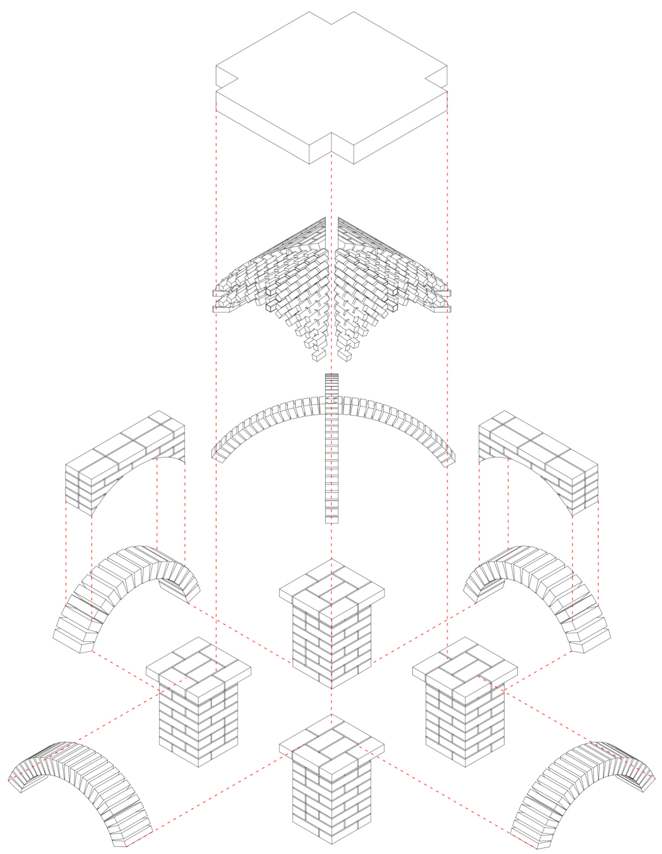


Figure120: Components of landing module

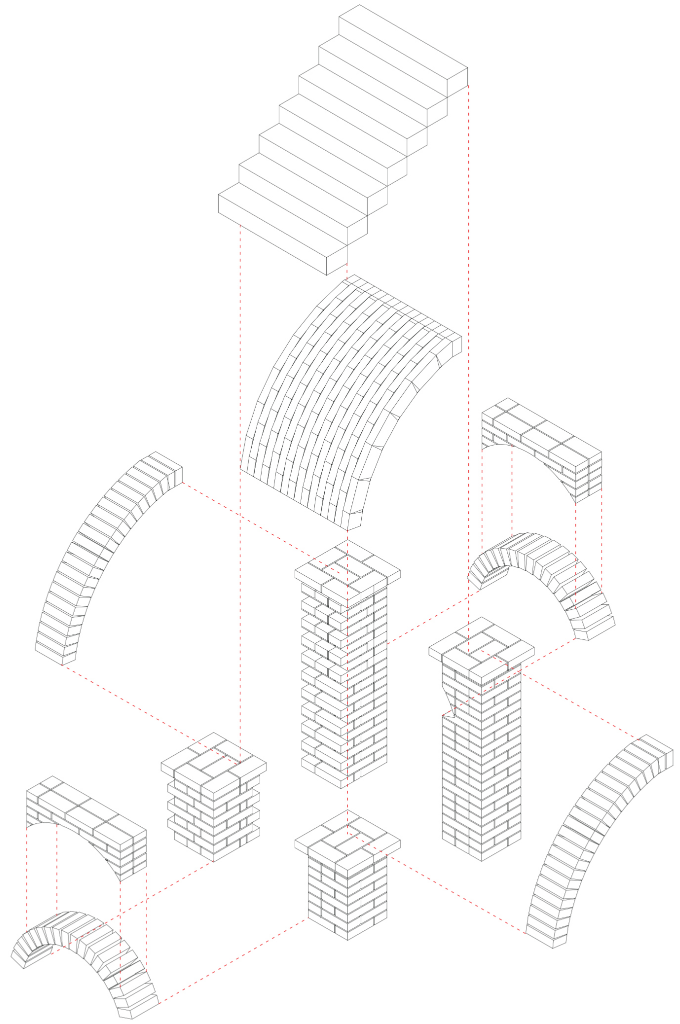
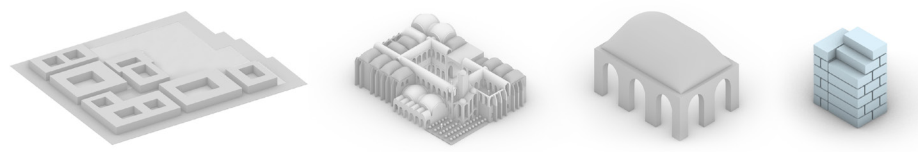


Figure121: Components of staircase module



CONSTRUCTION PHASES

LANDING AND STAIRCASE MODULE

STEP 1 - The first transversal arch in brick type A is marked and constructed using guide type A between the column marked as per the tartan grid.
STEP 2 - The next arch is marked using the same guide.
STEP 3 - The columns and walls on top of the arches are raised till the required height. A 600mm impost is added on the four column to allow the movement of guides.
STEP 4 - Two peripheral arches in brick type A are added on the outer perimeter of the grid using guide type A.
STEP 5 - Ribs in brick type C are added between these arches using guide type C. The guide can rest on the 600mm impost and can be slided out after setting of the ribs.
STEP 6 - The arches and ribs are then used at the form work for brick laying using brick type E.
STEP 7 - The next column is raised up to the required height i.e, 1875, with the same transversal arch supporting the wall on top.
STEP 8 - A staircase arch in brick type A is constructed using guide type B. The guide can be slided out after setting of the bricks.
STEP 9 - Another staircase arch is constructed on the outer perimeter.
STEP 10 - The core wall below the staircase vault is then closed to transfer any loads of the following vault.
STEP 11 - The staircase vault is then filled using brick type A by supporting every layer on the previous one.
STEP 12 - Every layer is added after completing the previous layer to transfer the loads on the sides.
STEP 13 - The staircase vault is then closed between the two staircase arches.
STEP 14 - Landing module and staircase module can be added on the top and any additional space in between can be filled with sand.

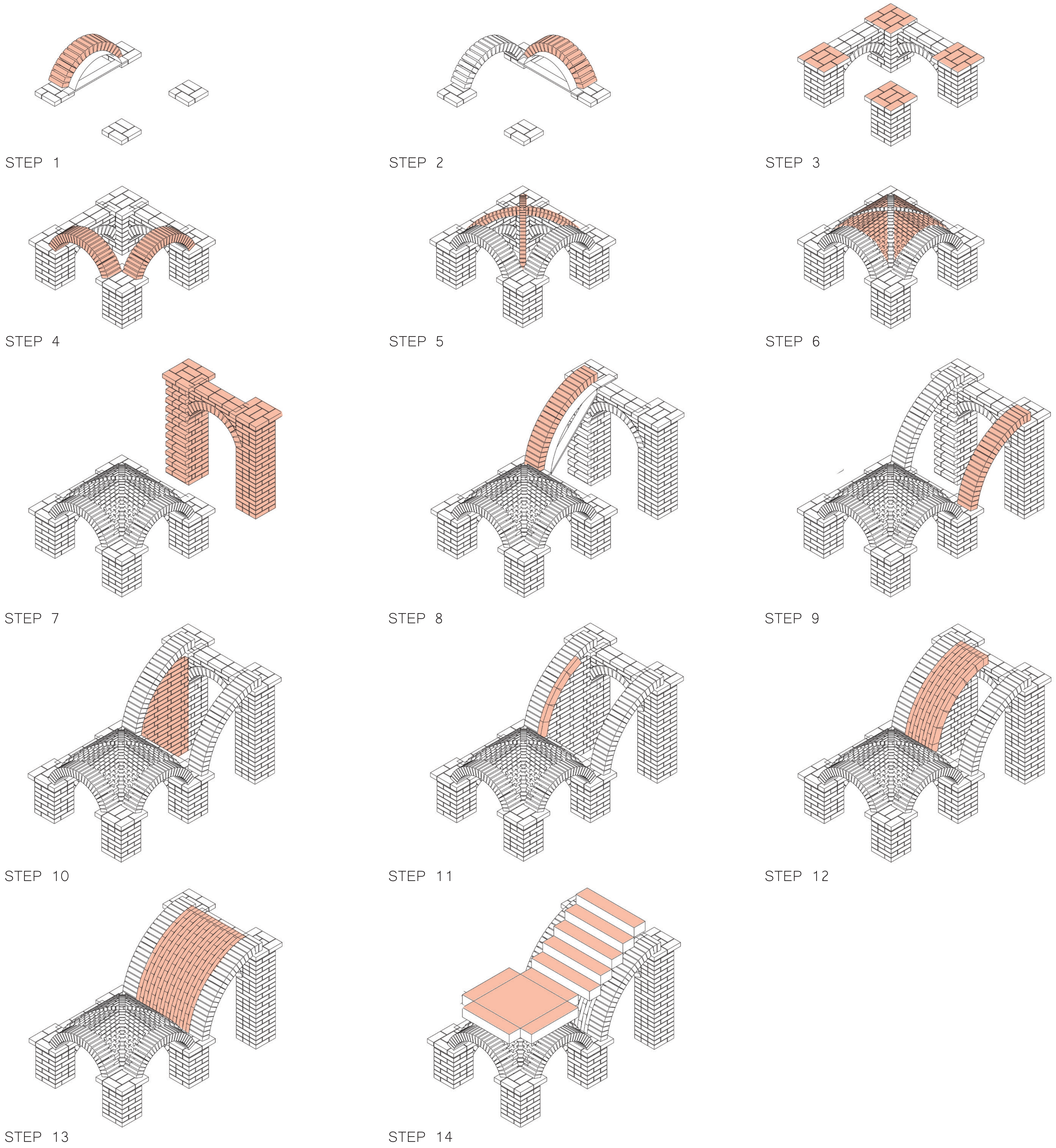
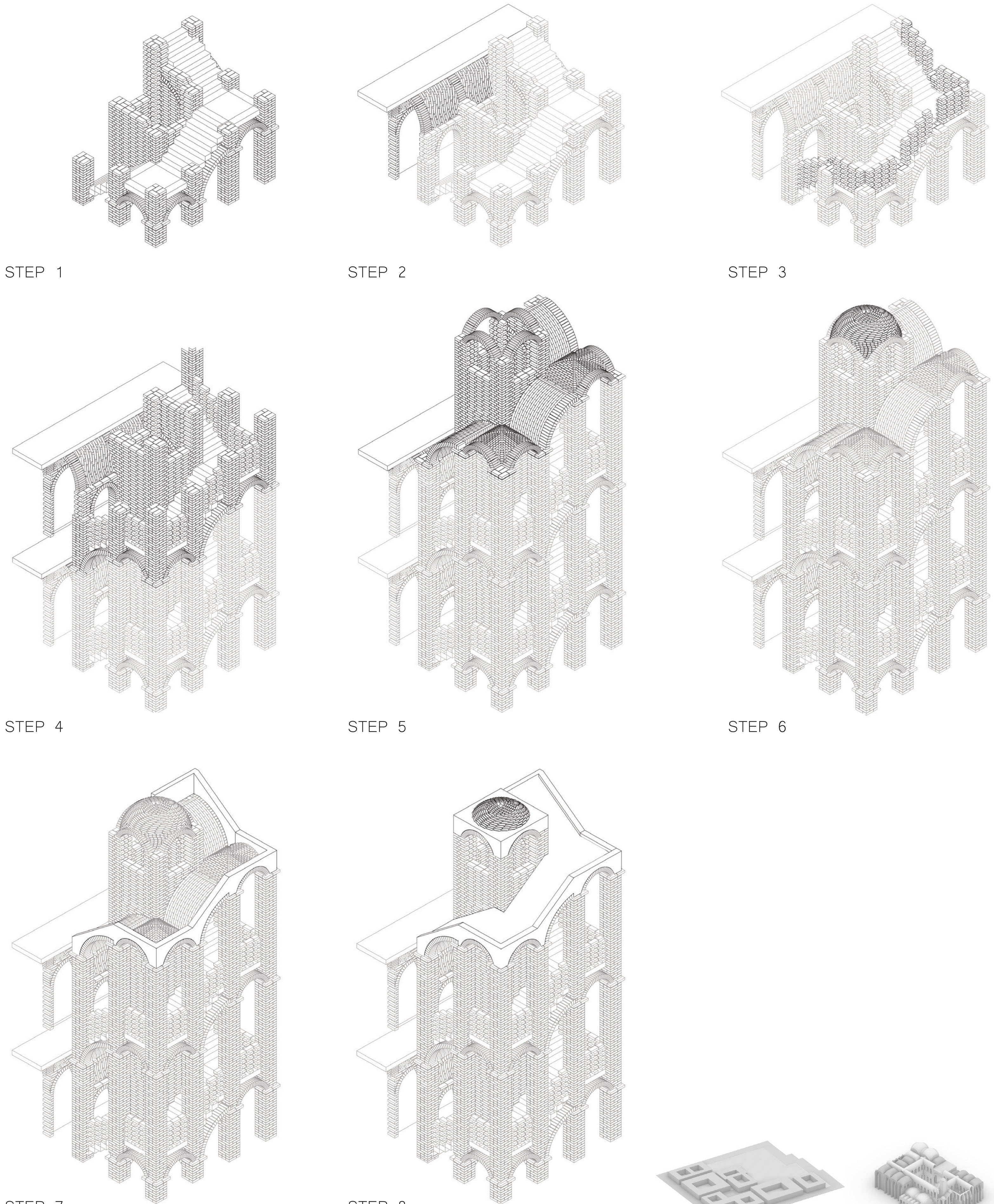


Figure123: Construction sequence for staircase and landing module

CONSTRUCTION PHASES

STAIRCASE SEQUENCE

STEP 1 - A combination of five alternate staircase and landing module can connect to the next level.
STEP 2 - This module is connected to the Riwaq for connection to the next level.
STEP 3 - A brick parapet can be added along the outer parameter.
STEP 4 - Similarly, another floor can be added on the top.
STEP 5 - For closing of the staircase, similar modules are added without any landing or staircase steps on the top. The core of the tower is raised atleast a floor higher than the last built house level for wind catcher to function. Openings are added on the top on the three available sided. Guide type A can be used to span these openings.
STEP 6 - The core is then closed with a dome roof sitting on squinch on the arches.
STEP 7 - Additional wall is raised on top of the outer perimeter arches and columns to add additional load for structural stability.
STEP 8 - The roof is then closed off with mud plaster mixed with lime to create a smooth slope.



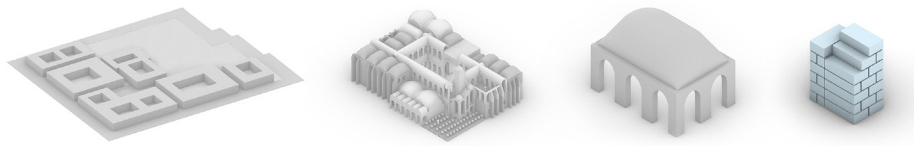


Figure124: Construction sequence for the wind catcher

CONSTRUCTION TOOLS

A simple system of 3 guides can be created to mark and construct all the arches in the tower using the scrap metal and wood available on the site. The guide type A and B can be made of a width of 450mm to support the 300mm brick. After setting of the arch, it can be easily slided out. The guide type C is designed to sit on top of the impost and is made in a width of 150mm. This too can be removed later. The tools can be easily reused for the whole construction system of the tower due to its modular design.

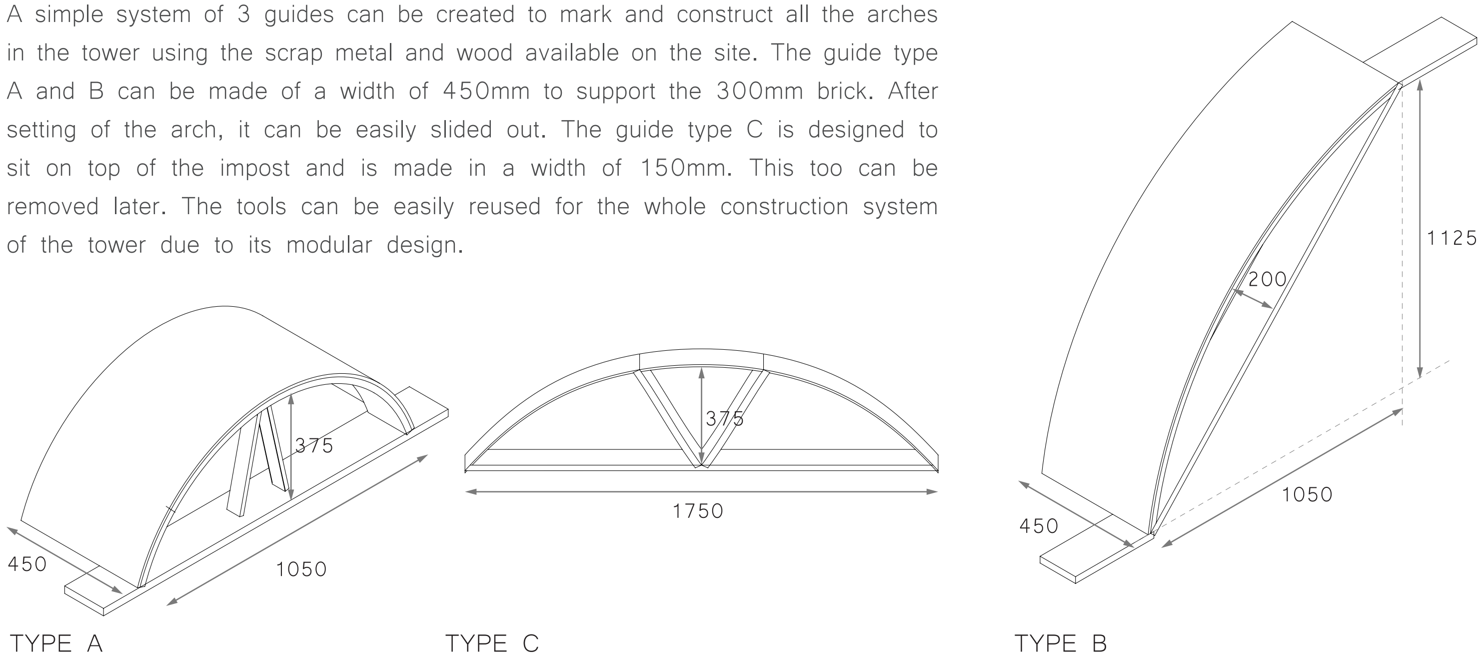


Figure125: Construction guides for arches

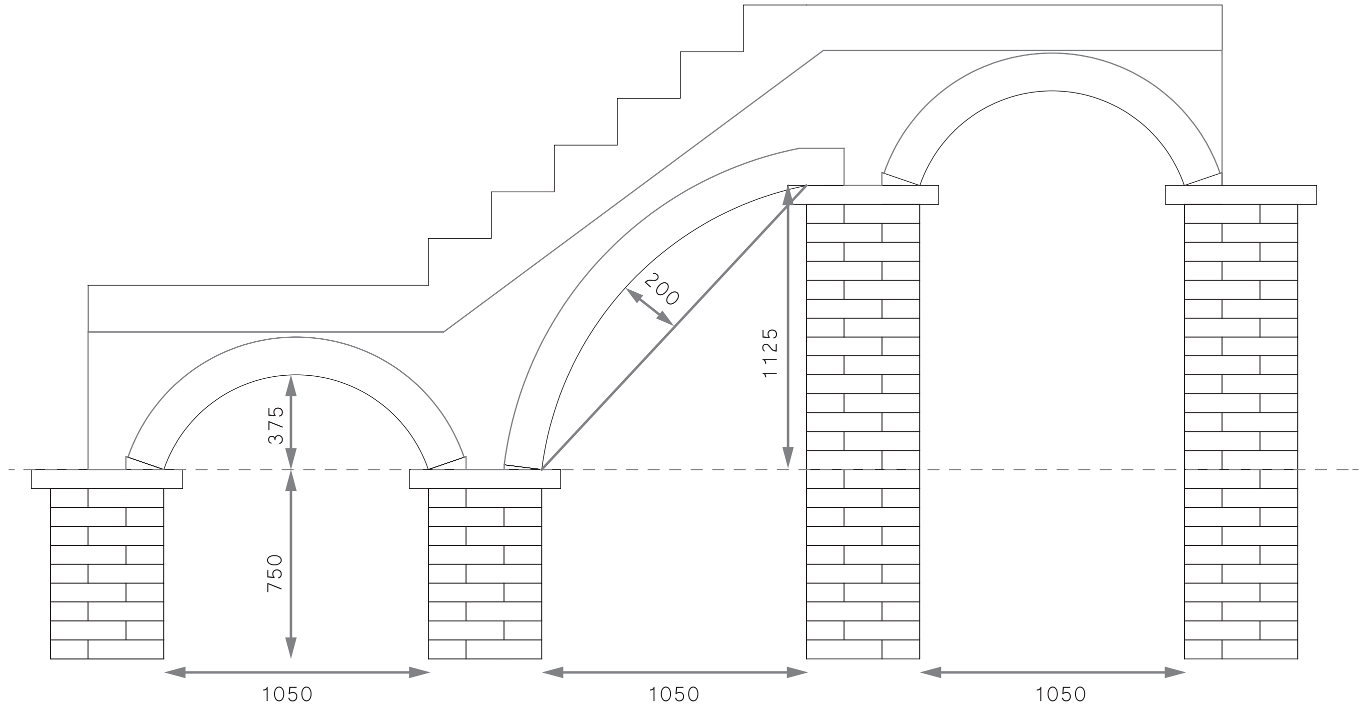


Figure126: Construction of arches using the guide

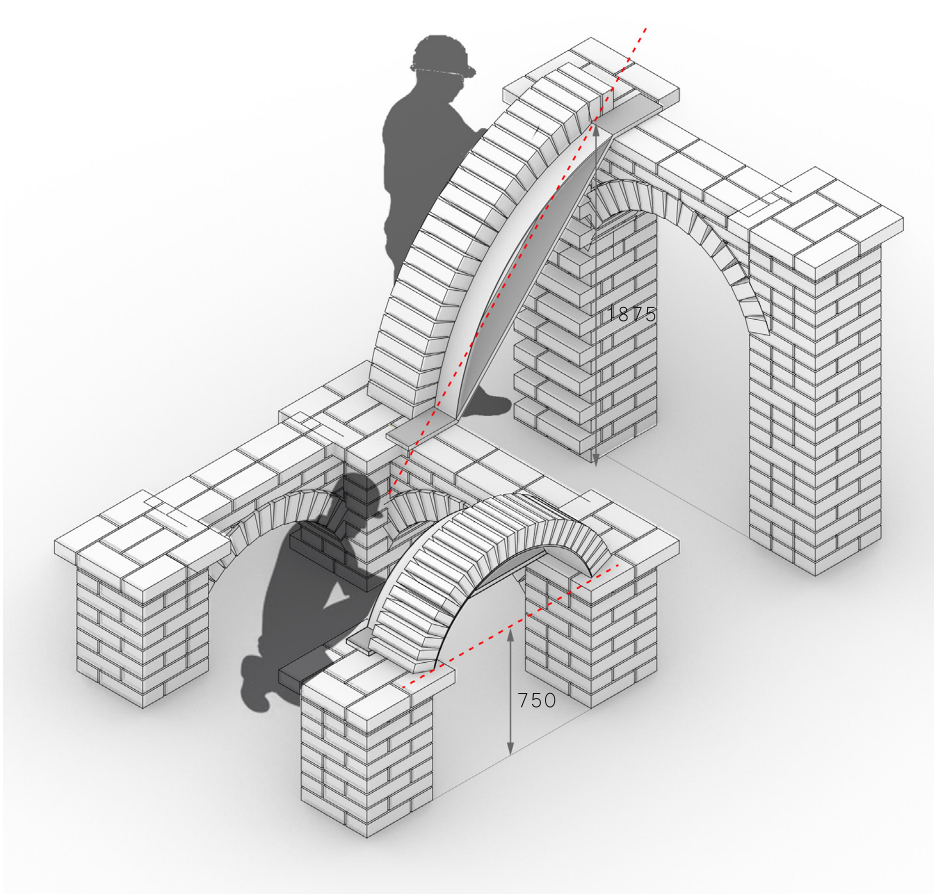


Figure127: Construction of transversal and peripheral arches

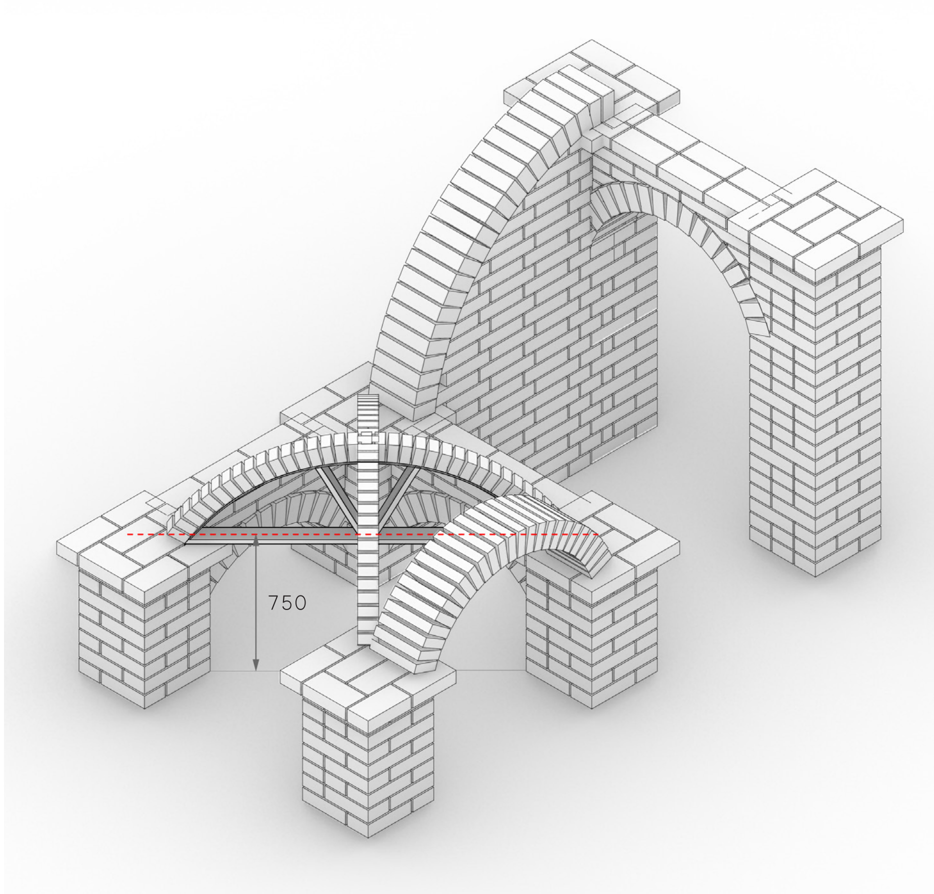


Figure128: Construction of ribs to support the landing

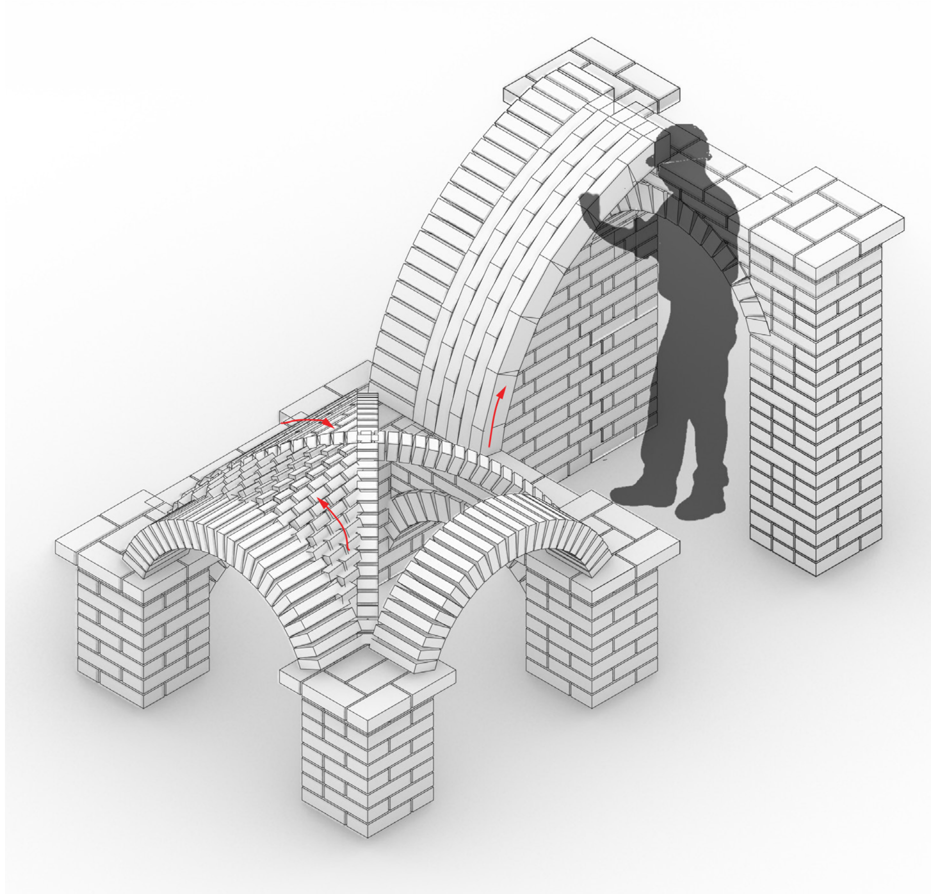
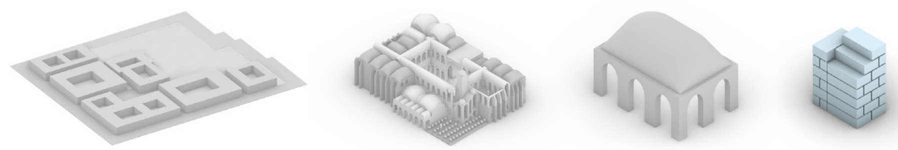


Figure129: Filling of the ribs and vault with brick tiles



2.3) Structural Analysis - Adobe 2.0, Windcatcher

The structural analysis of the windcatcher was done in a parallel with the construction process, updating the geometry and conditions for the analysis according to the requirements for the constructability. A big effort was put into the elaboration for the mesh for the analysis, this allowed to have a grid in which all elements are interconnected and behaving a one whole unit. For this, a manual approach was taken.

Molding the mesh

The initial approach taken for the tower, similar to the one used for the rest of the units, wasn't giving logical results because of a lack of interconnection between the different structural elements. As it can be seen in the construction images and in figure x, the vault for the stairs start on the arch of the landing, resting on the wall along the way and then reaching the arch on the consecutive landing. To level the arches from which this vault starts and ends, and to reduce the amount of sand-filling of the stairs, a small parapet was required. The vault for the landing is also resting on the same elements but at a different height.
Creating this connection with other approaches was creating a different level of refinement along the elements, which was concentrating stresses on areas with smaller refinements and giving inaccurate graphical representations of the results in Karamba.

This led to the final manual approach, in which a mesh was manually modeled for the interconnection of this elements and a logical grid refinement. All arches were simplified in polylines and a triangular grid was created for the internal walls, composed by divisions in x and y axis and the path of the staircase. After the modules were created, they only needed to be copied for the consecutive floors. The mesh elements had to be introduced individually into karamba for the analysis. This allowed to get accurate results, graphical representation of the stress distribution along the components of the tower and show a logical overall behavior of it as a whole unit.

Boundary Conditions

Loads : For the dead wight, an additional load was added to represent the sand-fillings and the final layer of bricks for the stair path. A safety factor of 1.2 was given. A force of 0.78 kN/m2 For the wind load was applied on the faces following the direction of the deformation to simulate the worst case scenario.

Supports: All points in contact with the base were selected as supports and were fixed in all axis of displacement and rotation.

Material Properties and Crossections: The material properties resultant from the analysis were used, and for the cross-section of the different elements, the output from the construction process was used.

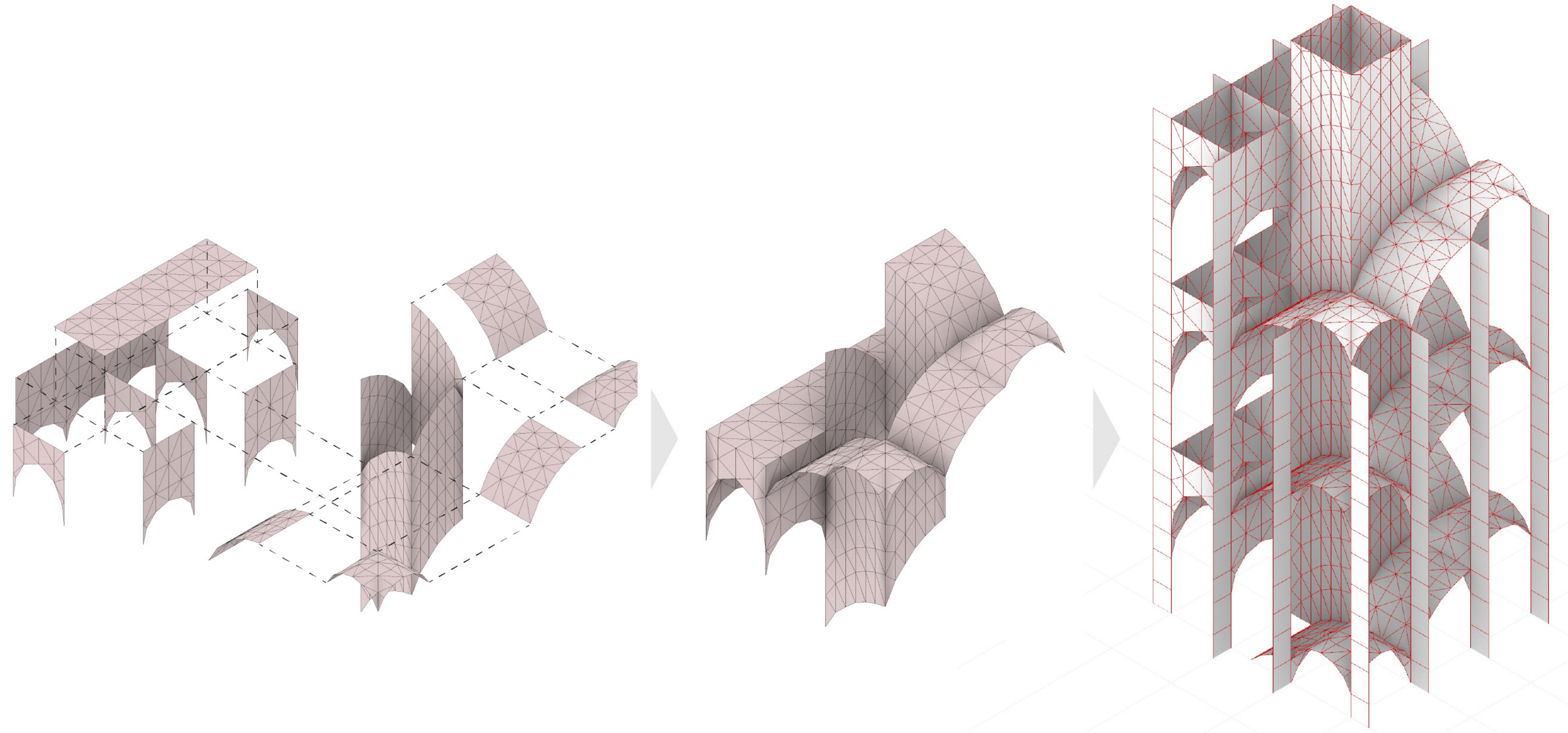


Figure131: Structural Analysis - Widcatcher, Mesh configuration. Source: Group Autorship

Landing:	Area(m^2)	Item	Volume(m^3)	Mass(kg/m^3)	Weight(kg)	Force(kN/m^2)
	2.25	Landing	0.225	1500	337.5	1.47
		Sand-Filling	0.57	1500	855	3.72
		Live Load			450	1.96
		Total Landing			1642.5	7.15
		Safety Factor				8.58
Stairs:	2.85	Landing	0.27	1500	405	1.39
		Sand-Filling	0.51	1500	765	2.63
		Live Load			450	1.55
		Total Landing			1620	5.57
		Safety Factor				6.68

Figure130: Structural Analysis - Table of loads for stair segments. Source: Group Autorship

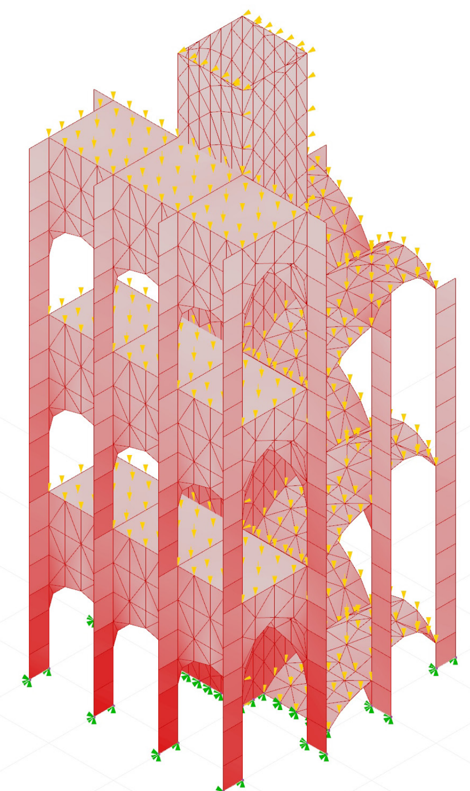
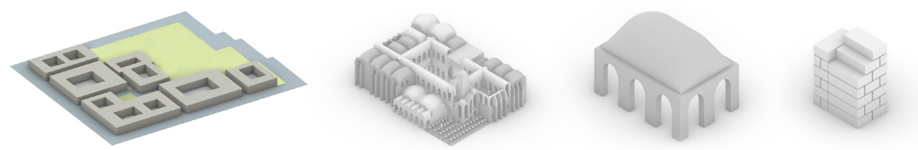


Figure132: Structural Analysis - Widcatcher, Boundary conditions. Source: Group Autorship



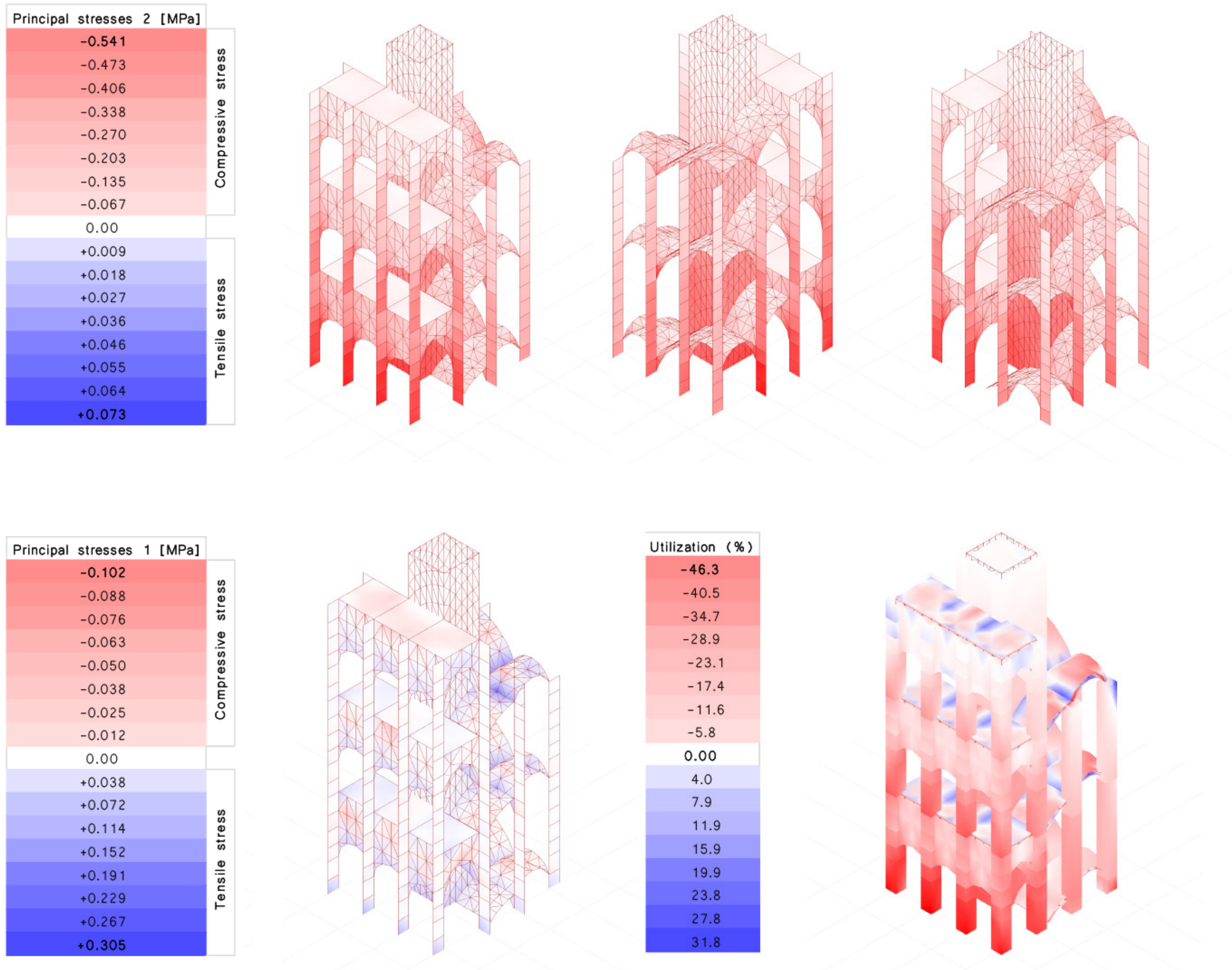


Figure134: Structural Analysis - Widcatcher, Overview of results. Source: Group Autorship

Results

As shown in the results, a maximum principal stress of -0.54 MPa in compression and 0.073 in tension in the vertical direction was obtained from the analysis. This results where corroborated with the hand calculations. As seen in fig X The stresses on horizontal directions show us some pick tension stresses on the two corners that follow the deformation of the tower. Nevertheless, the deformation is inside the allowable displacement limits, reaching a maximum of 3 cm. Utilizations columns was required to rest arches from different directions. Deformation

Max Disp. Allowed
3.72cm
Max Displacement
3.00cm
Max Comp. Stress
-0.54 MPa
Utilization Max
-46.3 %
+31.8 %

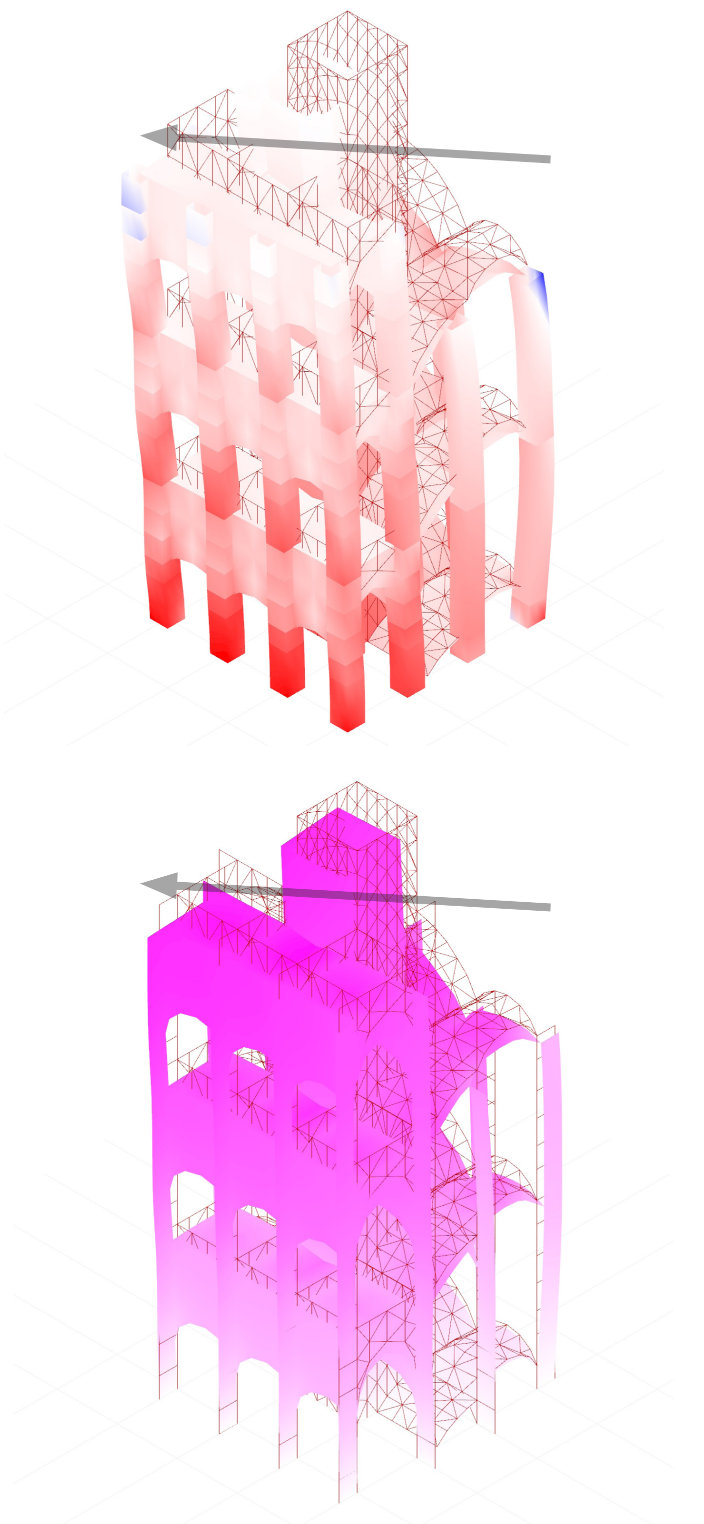
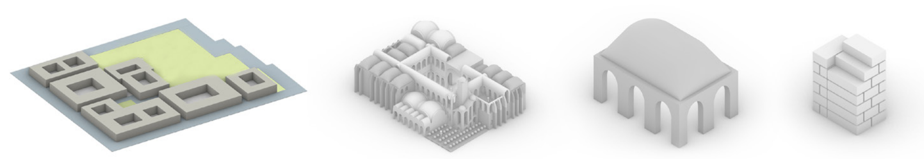


Figure133:Structural Analysis - Widcatcher, Overview of pick horizontal stress and deformation. Source: Group Autorship



LIMITATIONS OF TERRATETRIS

The game logic for configuration stage is not a 100% participatory. Currently for large number of players who are trying to configure the same dwelling the logic favours the players who play it before the rest. For example, not all the players would get their first preference for the arrangement and orientation of a space if all of them or many of them want the same configuration.

The intention in the scripting logic of the game was to replicate the human process as accurately as possible this is not done completely since there are many things which the human designer does intuitively which has a difficult interpretation in terms of coded logic . Those types of intuitive choice-based decisions are not considered completely in the game (If they are considered then they are hard coded and not open for users to decide)

Keeping all levels of the game modular also has its share of disadvantages. Since all the blocks of the house (Spaces) are designed to be independent of each other . Structural optimization for a combination and material optimisation is not possible.

In the Urban level that is scale 03 of the configuration logics a lot of values are hard-coded. Like the width of the road, size of communal areas etc. They should be dependant on the density and the massing of the generated configuration to avoid problems like self-shading and narrow streets with taller buildings.

Some modules in the Kit of Parts for the assembly still need to be figured out to enhance the configuration logic behind it. For example, if a module for wet areas on the first floor is considered then multiple families can aggregate in the same dwelling making an efficient use of the first floor.

Furthermore this logic is based on the design principles and methodology of just the five of us but there could be better ways to design and configure spaces so keeping an open ended system is the ideal solution for this type of a problem.

EXPANSION PACK

To increase the features and utilities of terratetris we propose an expansion pack which indicates the things that we could take up for development further. On the following page there is a diagram which shows the salient features that we intend to achieve with this expansion pack and the ones which we leave for future scope of development.

The expansion pack contains new features which could be relatively easily developed based on the progress we have made so far with the game. The first and foremost feature addition would be the possibility of developing templates for other types of configurations apart from the courtyard house. It could be a courtyard passage house which we have already indicated currently but could become a template. The options for starting the configuration process with just a core, a core + passages or nested passages could also give more design options for the same problem.

Secondly for the urban scale we have defined only a few modules which we could expand to have a more realistic circular scenario for the neighbourhoods. So, modules like Solar power, Solar heating, Roof top farming, and other utility modules could be added.

The kit of parts itself that we have designed can have a lot of options. Currently we chose only the options that were easy to construct but the user should get an access to the whole library of possibilities to choose from.
There should be more factors in a decision tree and a customisable sandbox version for the decision tree could also open up the possibilities for customisation . The sandbox idea should extend to the shaping and constructing process also where users having design knowledge can make their own components which will be compatible with the rules.

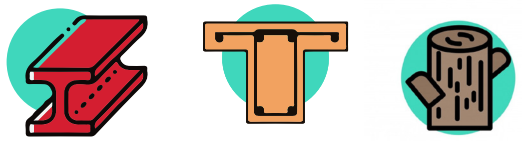
Lastly the idea for terratetris is based on adobe as a material but could very well be extended to Steel , Concrete, Timber and other modular systems . The shaping and structuring libraries would be different, but the configuration libraries can remain the same.

There are a few ideas which could make the game truly a replica of human Architectural designer. The configuration which is not based on rectilinear spaces but more of an organic space could be a new direction to explore. The game should have an option like an optimization plugin where the user can pick the type of algorithm for configuration instead of optimization. Lastly it could also be possible for a reverse process where a 3d model is sliced to contain the required program and configuration. All these process and features cannot be developed on the current base but could be a nice scope for future development .

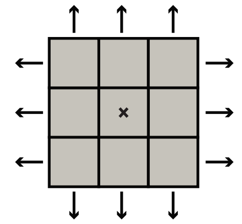
Expansion Pack

1. Courtyard house + Other types
2. More modules for Urban scale
3. More options in Kit of Parts
4. More Decision tree options
5. Timber,Concrete,Steel additions
6. Sandbox modular construction
7. templates for shaping stage

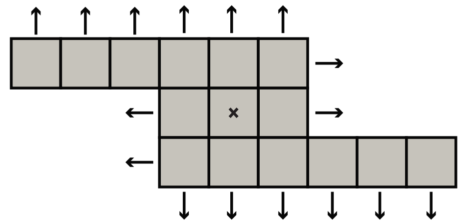
Terra Tetris



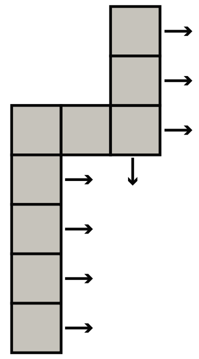
EXPANSION PACK...



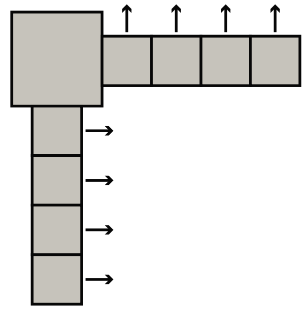
Courtyard Config



Courtyard +Passage Config



Passage / Nested Passage Config



Core + Passage Config

CONFIGURATION

1. Courtyard
2. Passages+ Courtyard
3. Passage
4. Passages + Core
5. Nested Passages

6. Auto selection of the type Finding alternate approaches to directional orientation of spaces

7. Introducing UI where intutive decisions need to be made

8. Indepth mapping of design process by collection of data in terms of decisions made which are converted to procedural rules

FUTURE DEVELOPMENT **
EXPANSION PACK
ALREADY EXISTS

SHAPING

Add more decision making parameters :

1. Climatic orientation
2. Distance from an attractor
3. Distance from entrance
4. Fire escape distance

5. Template generation and User input addition on determinig the sizes of each space and the rules guiding them
6. Sandbox version creating custom decision templates
7. Decision tree generator with a UI for common users

8. Moving beyond rectillinear shapes towards more organic shapes

9. Reversing the process where the user starts by contouring the 3d itself

STRUCTURING

1. Adobe Modular construction
2. Timber Modular construction
3. Steel Modular construction
4. Concrete Modular construction
5. Composite Modular construction
6. Structural Optimization per unit

7. Sandbox version for Modular construction
8. Template for creating custom modular design.

COMING SOON...

REFLECTIONS

CHRISTINA KOUKELLI :

In general, the EARTHY course gave the opportunity to experiment at a great level on compression-only structures and to work with earth bricks and construction techniques that were not familiar to us. What was interesting and challenging was the computational approach taken in all the levels of design and combining the traditional way of building and brick-layering with a more intellectual and sophisticated method. We were introduced to several new computational tools, as well as mathematical logics, and were provided with new ways of approaching housing configurations and topological aspects. It was interesting to experiment with the different computational tools and try to find ways to combine everything under the same logic with the right tools for the right purpose. Another aspect of the course that was intriguing was not only understanding the properties of the material and the construction methods, but also to push the limits, by proposing new ways and building systems and extending its limitations and possible uses. Especially for the TerraTetris project, what was special and worth having worked on was the connection of all levels of design under one universal logic with an intellectual concept behind it. This was an engaging challenge, which would both bring design limitations and flexibility in the design. It would also require several tryouts and adjustments to fit every aspect and decision in all the scales, similar to a problem-solving exercise to find the right balance and trade-offs every time. In all, it was a fruiful experience, which provided with a lot of knowledge and skills to each one of us, with the right guidance and feedback from the tutors to be able to deliver the project.

VICENTE BLANES:

For me, EARTHY was an entirely new approach on the formulation of design. The goal of translating design logic and strategies to rules, functions and behavior of components was a very challenging and enriching approach. Understanding how to open the possibility for people to participate as designers, by inputting very simple information that can change the configuration of their own space and urban context. Although we only reached a small portion of process we initially proposed, understanding the difficulties and how it should be structured and formulated was a very frustrating but encouraging process. The commodity of working with grasshopper played against me on the urban rearrangement strategy, as the script could have been done in python to be simplified and avoid the slow process that difficulted to explore further. Nevertheless, I feel accomplished with the result and with figuring out the logic of how it should work and how it could be further improved. It was a very challenging, time demanding but rewarding course.

ADITYA SOMAN :

The idea of generative design and automating architecture had fascinated me even before enrolling for this course. Through this course we were exposed to a lot of different ideas and ways of thinking not only about generative design but also about earth stuctures and structural design which I really appreciated.
Another intention before starting this course was to learn Python as a programming language since it opens up a lot of different opportunities in computational design ,which was to a certain extent achieved in this course. The workshops provided a sound understanding of basics for us to take it forward. The idea of teratetris was a culmination of a bunch of ideas that were on my mind since some time now but I didnt have the necessary skills to make it work . Through the course I was exposed to even better ideas and introduced to the necessary skills which would enable me to develop it further. I am pleased with the output that we could generate for Earthy via teratetris but there were a lot of things which we couldnt explore or integrate in the given time which I would further like to explore. I was a bit disappointed too in the end since we couldnt really find the time to convert all our ideas into scripts which we could use further. The attitude of the tutors towards the presentations as being another consultataion session rather than a marketing pitch was really good and made the discussions more productive . Lastly I would like to thank all my teammates for working with me on this final group project before our graduation.It was truly an enjoyable experience working with all of them.

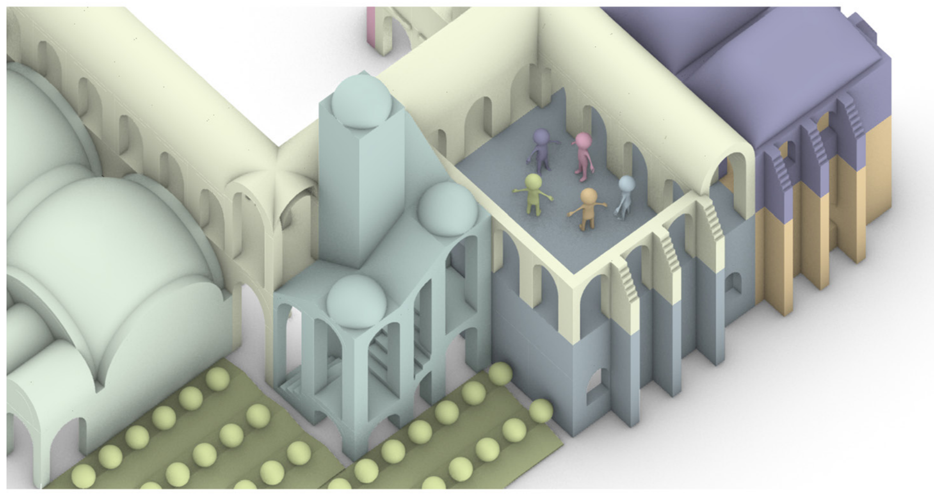
DION VAN VLERKEN:

I have experienced the EARTHY course as very educational. For me this was the first time I came into contact with Grasshopper and Python. The Python workshops at the start of the course were quite intense. The level was immediately very high. It was therefore immediately clear to me that I had to shift my attention to learning Grasshopper first. I did this by making the brick laying. I made a script that is able to generate the entire wall structure from one line. However, it has become clear to the consults that my approach in the script is not the right one. Unfortunately, I was no longer able to adjust this, but I will be aware of this in the future and will apply it in a next script. I also worked on the material properties of the adobe, the kit of parts and the manual calculations of the construction.

I have experienced the cooperation with the group as very pleasant. All have great knowledge and their attitude to work is fantastic. I have learned a lot from them and I am very grateful for it.

NEHA GUPTA:

The process of the Earthy studio is very different from other courses as the process looks at the logic and discretisation of architectural design. As I worked on the project, I realised how generative and modular design approaches can be applied and be extended to spatial configurations in architectural design and can be replicated to solve any design problem. The idea of looking at modular/ unit based design as a game can further help in understanding and simplifying the logics of spatial connectivity and designing. Another aspect that I learnt while working is applying a material's properties to utilize its full potential in construction which can eventually result in sustainable use of materials in terms of efficiency, availability and techniques.



REFERENCES

1.Owens, T. (2015). Earth Plaster For Straw Bale and Adobe Homes. Retrieved from Building with Awareness: https://www.buildingwithawareness.com/earth-plaster-for-straw-bale-homes/

2.Passoni, A., Vigorito, A., Fortich, F., Mousavi, K., & Moumdjian, S. (2019). MODUL_AB_ITY. Delft.

3.Bharath, B., Reddy L, M., Pathan, J., & R Patel, R. (2014). Studies on stabilised adobe blocks. Retrieved from Researchgate: https://www.researchgate.net/publication/273278197_STUDIES_ON_STABILISED_ADOBE_BLOCKS

4.Illampas, R., Ioannou, I., & Charmpis, D. (2011). A study of the mechanical behaviour of adobe masonry. Retrieved from Witpress: https://www.witpress.com/Secure/elibrary/papers/STR11/STR11040FU1.pdf

5.Aubert, J.-E., Fabbri, A., Morel, J., & Maillard, P. (2013). An earth block with a copmressive strength higher than 45MPa! Retrieved from HAL: https://hal.archives-ouvertes.fr/hal-01877125/document

6.Millogo, Y., Aubert, J.-E., Séré, A., Fabbri, A., & Morel, J.-C. (2016). Earth blocks stabilized by cow-dung. Retrieved from Researchgate: https://www.researchgate.net/publication/291554072_Earth_blocks_stabilized_by_cow-dung

7.Quagliarini, E., & Lenci, S. (2016). The influence of natural stabilizers and natural fibres on the mechanical properties of ancient Roman adobe bricks. Retrieved from Researchgate: https://www.researchgate.net/publication/248546225_The_influence_of_natural_stabilizers_and_natural_fibres_on_the_mechanical_properties_of_ancient_Roman_adobe_bricks

8.Changlani, A., Chokshi, S., Vintimilla, E., Zenteno, Y., Wannasawang, P., & Fahriba Mustafa, K. (2019). Bustan. Delft.

9.Li Piani, T. (2019). Experimental-numerical material characterization of adobe masonry: Tests and simulations on various types of earthen bricks and mortar in statics and dynamics. Retrieved from https://doi.org/10.4233/uuid:f2470c09-e8f0-4c1e-b4e4-effc17ecd373

10.Guillaud, H., Joffroy, T., & Odul, P. (1985). Compressed earth blocks: manual of design and construction. Retrieved from http://ndl.ethernet.edu.et/bitstream/123456789/32596/1/6.pdf

11.Yalley, P.-k. (2013). Strength and Durability Properties of Cow Dung Stabilised Earth Brick. Retrieved from https://core.ac.uk/download/pdf/234677669.pdf

12.UNHCR. (2020, February). Relief Web. Retrieved September 2020, from https://reliefweb.int/report/jordan/zaatari-refugee-camp-factsheet-january-2020

13.Yegul, F., & Favro, D. (2019). Roman Architecture and Urbanism - From the origins to late antiquity. Cambridge University Press.

14.Ledwith, A., & Smith, D. (2014). Zaatari: The Instant City. United States of America.

15.Wendland, D. (2007). The vaults in “Sistani’s House” in Bam Citadel (Iran) : Construction principles, shape geometry and design for reconstruction.

16.Minke, G. (2006). Building with Earth - Design and technology of a sustainable architecture. Birkhauser.

APPENDIX

HANDCALCULATION BEDROOM WITH BEDROOM ON TOP

MASS ADOBE BRICKS: 1500 KG/M3
MASS SAND: 1500 KG/M3

PERMANENT LOADS

ARCH GROUNDFLOOR:
ARCHAREA: 2,7 M2
THICKNESS ROOF: 0,14 M
LOAD: 5,7 KN

SAND FILLING:
VOLUME: 1,13 M3
LOAD: 17,0 KN

FLOOR:
FLOORAREA: 2,25 M2
THICKNESS FLOOR: 0,05 M
LOAD: 1,7 KN

ROOF:
ROOFAREA: 2,7 M2
THICKNESS ROOF: 0,14 M
LOAD: 5,7 KN

COLUMN:
LENGTH: 5,8 M
WIDTH: 0,29 M
LOAD: 7,3 KN

TOTAL: 37,3 KN
SAFETY FACTOR: 1,2
TOTAL PERMANENT LOAD: 44,8 KN

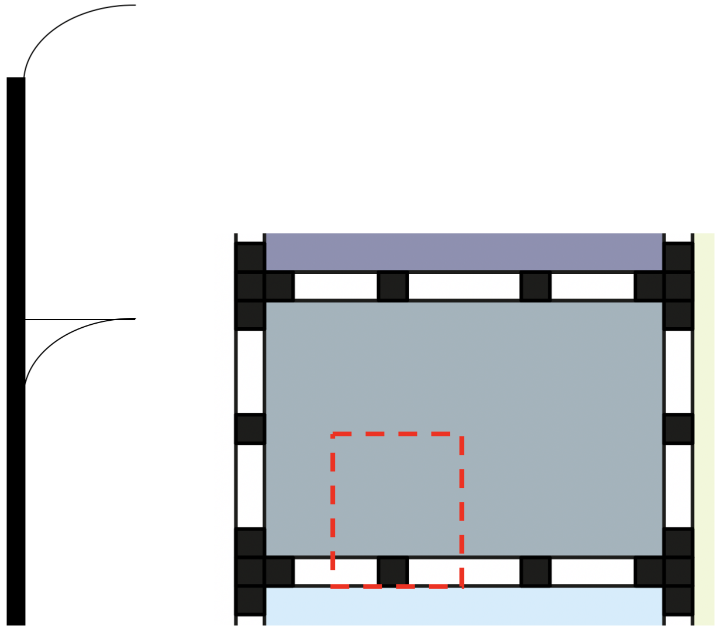


Figure135: Location calculation bedroom with bedroom on top

LIVE LOADS

FLOOR:
FLOORAREA: 2,25 M2
FLOORLOAD: 2 KN/M2
LOAD: 4,5 KN/M2

ROOF:
ROOFAREA: 2,7 M2
ROOFLOAD: 0,75 KN/M2
LOAD: 2,0 KN

TOTAL: 6,5 KN
SAFETY FACTOR: 1,5
TOTAL LIVE LOAD: 9,8 KN

PERMANENT LOAD + LIVE LOAD: 54,5 KN
-> 54541 N

AREA COLUMN: 0,0841 M2
-> 84100 MM2

MAXIMUM COMPRESSIVE STRENGTH: 0,65 N/MM2

HANDCALCULATION IWAN

MASS ADOBE BRICKS: 1500 KG/M3

MASS SAND: 1500 KG/M3

PERMANENT LOADS

ARCH:
ARCHAREA: 4,52 M2
THICKNESS ROOF: 0,14 M

LOAD: 9,5 KN

COLUMN:
LENGTH: 3,56 M

WIDTH: 0,59 M

DEPTH: 0,29 M

LOAD: 9,1 KN

ROOF:
ROOFAREA: 3,38 M2

THICKNESS FLOOR: 0,18 M

LOAD: 9,1 KN

SAND FILLING ROOF:
VOLUME: 1,9 M3

LOAD: 28,5 KN

TOTAL: 56,3 KN
SAFETY FACTOR: 1,2
TOTAL PERMANENT LOAD: 67,5 KN

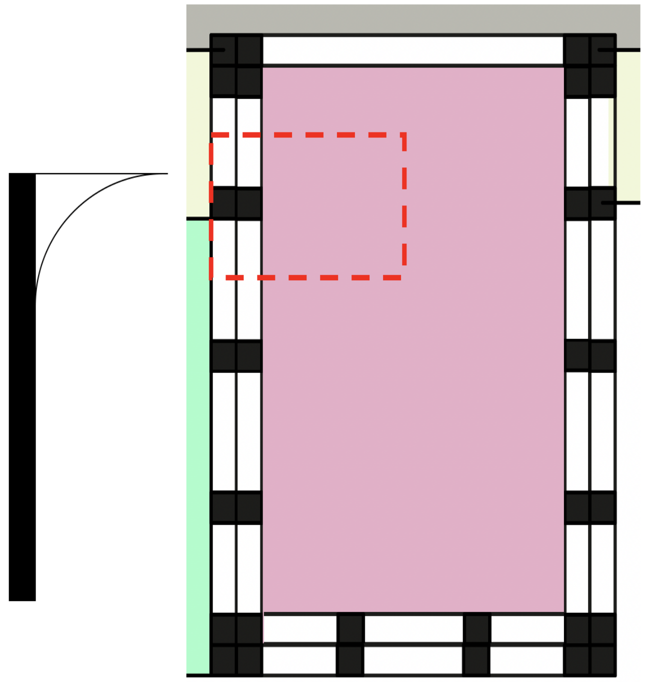


Figure136: Location calculation iwan

LIVE LOADS

ROOF:
ROOFAREA: 3,38 M2
ROOFLOAD: 0,75 KN/M2
LOAD: 2,5 KN

TOTAL: 2,5 KN
SAFETY FACTOR: 1,5
TOTAL LIVE LOAD: 3,8 KN

PERMANENT LOAD + LIVE LOAD: 71,3 KN
-> 71308 N

AREA COLUMN: 0,1711 M2
-> 171100 MM2

MAXIMUM COMPRESSIVE STRENGTH: 0,42 N/MM2

HANDCALCULATION BEDROOM

MASS ADOBE BRICKS: 1500 KG/M3
MASS SAND: 1500 KG/M3

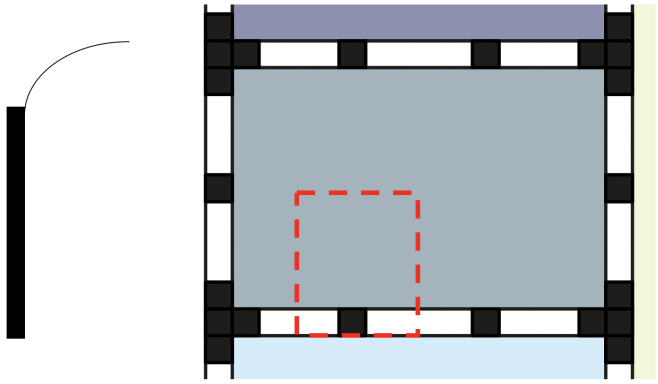


Figure137: Location calculation bedroom

PERMANENT LOADS

ROOF:
ROOFAREA: 2,7 M2
THICKNESS ROOF: 0,14 M
LOAD: 5,7 KN

COLUMN:
LENGTH: 2,2 M
WIDTH: 0,29 M
LOAD: 2,8 KN

TOTAL: 8,4 KN
SAFETY FACTOR: 1,2
TOTAL PERMANENT LOAD: 10,1 KN

LIVE LOADS

ROOF:
ROOFAREA: 2,7 M2
ROOFLOAD: 0,75 KN/M2
LOAD: 2,0 KN

TOTAL: 2,0 KN
SAFETY FACTOR: 1,5
TOTAL LIVE LOAD: 3,0 KN

PERMANENT LOAD + LIVE LOAD: 13,2 KN
-> 13172 N
AREA COLUMN: 0,0841 M2
-> 84100 MM2

MAXIMUM COMPRESSIVE STRENGTH: 0,16 N/MM2

HANDCALCULATION RIWAQ

MASS ADOBE BRICKS: 1500 KG/M3
MASS SAND: 1500 KG/M3

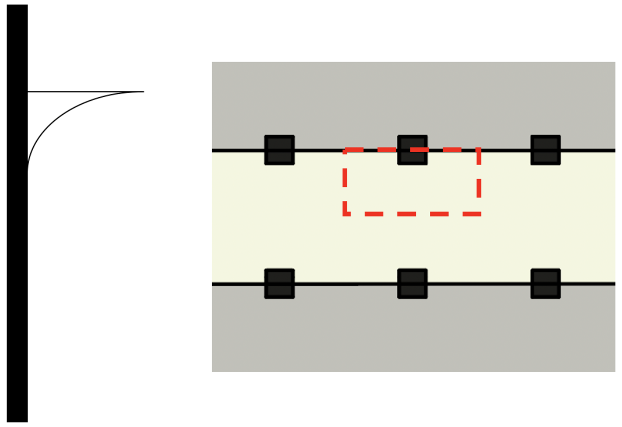


Figure138: Location calculation Riwaq

PERMANENT LOADS

ARCH GROUNDFLOOR:
ARCHAREA: 1,35 M2
THICKNESS ROOF: 0,14 M
LOAD: 2,8 KN

SAND FILLING:
VOLUME: 0,1875 M3
LOAD: 2,8 KN

FLOOR:
FLOORAREA: 1,125 M2
THICKNESS FLOOR: 0,05 M
LOAD: 0,8 KN

COLUMN:
LENGTH: 4 M
WIDTH: 0,29 M
LOAD: 5,0 KN

TOTAL: 6,5 KN
SAFETY FACTOR: 1,2
TOTAL PERMANENT LOAD: 7,8 KN

LIVE LOADS

ROOF:
FLOORAREA: 1,125 M2
ROOFLOAD: 1,75 KN/M2
LOAD: 2,0 KN/M2

TOTAL: 2,0 KN
SAFETY FACTOR: 1,5
TOTAL LIVE LOAD: 3,0 KN

PERMANENT LOAD + LIVE LOAD: 10,7 KN
-> 10743 N
AREA COLUMN: 0,0841 M2
-> 84100 MM2

MAXIMUM COMPRESSIVE STRENGTH: 0,13 N/MM2

HANDCALCULATION ADOBE 2.0

MASS ADOBE BRICKS:	1500	KG/M3
MASS SAND:	1500	KG/M3

PERMANENT LOADS

VAULT FIRST FLOOR:

VOLUME:	0,145	M3
LOAD:	2,2	KN

FIRST FLOOR:

FLOORAREA:	1,35	M2
THICKNESS FLOOR:	0,18	M
LOAD:	3,6	KN

SAND FILLING FIRST FLOOR:

VOLUME:	0,16	M3
LOAD:	2,4	KN

PARAPET FIRST FLOOR:

VOLUME:	0,315	M3
LOAD:	4,725	KN

VAULT SECOND FLOOR:

VOLUME:	0,145	M3
LOAD:	2,2	KN

SECOND FLOOR:

FLOORAREA:	1,35	M2
THICKNESS FLOOR:	0,18	M
LOAD:	3,6	KN

SAND FILLING SECOND FLOOR:

VOLUME:	0,16	M3
LOAD:	2,4	KN

PARAPET SECOND FLOOR:

VOLUME:	0,315	M3
LOAD:	4,725	KN

VAULT THIRD FLOOR:

ROOFAREA:	0,145	M3
LOAD:	2,2	KN

THIRD FLOOR:

FLOORAREA:	1,35	M2
THICKNESS FLOOR:	0,18	M
LOAD:	3,6	KN

SAND FILLING THIRD FLOOR:

VOLUME:	0,16	M3
LOAD:	2,4	KN

PARAPET THIRD FLOOR:

VOLUME:	0,315	M3
LOAD:	4,725	KN

COLUMN:

LENGTH:	12,34	M
WIDTH:	0,44	M
DEPTH:	0,44	M
LOAD:	35,8	KN

TOTAL: 74,7 KN
SAFETY FACTOR: 1,2
TOTAL PERMANENT LOAD: 89,6 KN

LIVE LOADS

FIRST FLOOR:

FLOORAREA:	1,35	M2
FLOORLOAD:	2	KN/M2
LOAD:	2,7	KN/M2

SECOND FLOOR:

FLOORAREA:	1,35	M2
FLOORLOAD:	2	KN/M2
LOAD:	2,7	KN/M2

THIRD FLOOR:

FLOORAREA:	1,35	M2
FLOORLOAD:	2	KN/M2
LOAD:	2,7	KN/M2

TOTAL: 8,1 KN
SAFETY FACTOR: 1,5
TOTAL LIVE LOAD: 12,2 KN

PERMANENT LOAD + LIVE LOAD:	101,8	KN
->	101754	N
AREA COLUMN:	0,1936	M2
->	193600	MM2

MAXIMUM COMPRESSIVE STRENGTH: 0,53 N/MM2

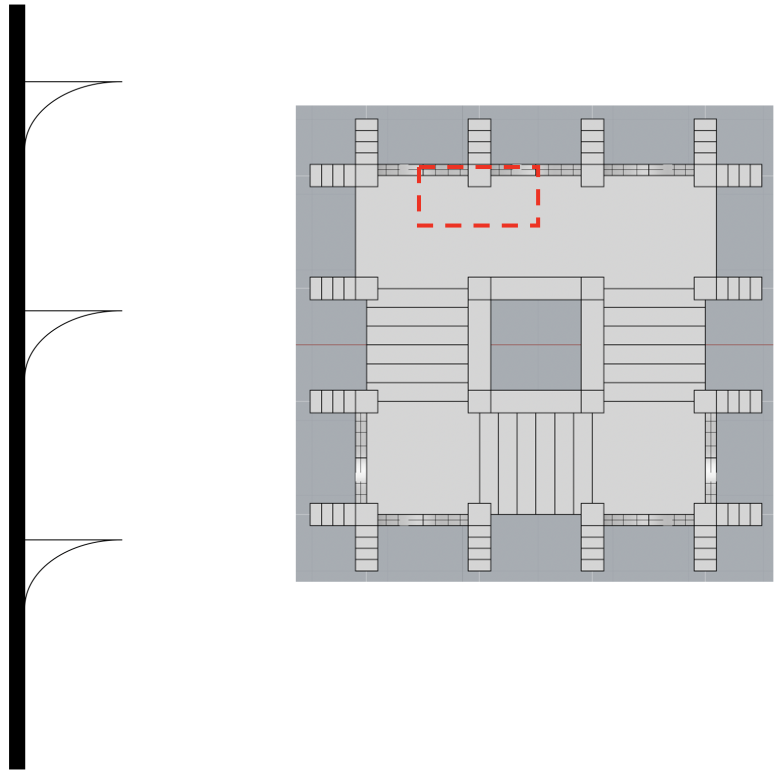


Figure139: Location calculation adobe 2.0

game over_