31/03/2023

Toby Grimwood-Snook

Logic to Artefact

UBLLW1-15-M - Logic to artefact

Project Brief

This portfolio is a product of the brief; Logic to Artefact.

The academic objective of this module is to understand and connect computational theory to computational design and relative digital fabrication outputs.

The practical objective of this module is to create a shelter for 1 - 3 users.

The learning outcomes for this module are:

- Interpret Pattern Language theory and form-finding methods and recognise the interrelationship with computation design processes in academia and practice.
- Identify and apply the types and fundamentals of visual programming, material testing, and digital fabrication methods to represent micro-scale design in response to a primer brief.
- Apply researched knowledge of pattern and form-finding algorithms in design processes to produce a well-crafted artefact (simulation/fabrication).
- Flexibly and creatively select appropriate computational and fabrication methods by pro-actively undertaking substantial investigations to address an architectural system design.
- Present an interpretation of fundamental Pattern Language and form-finding theories in a crafted artefact, graphical and verbal illustration at a high level of abstraction, arguing from competing perspectives.

Theory - A Pattern Language

The computational theory implemented in this project is A Pattern Language by Christopher Alexander, Murray Silverstein, and Sara Ishikawa.

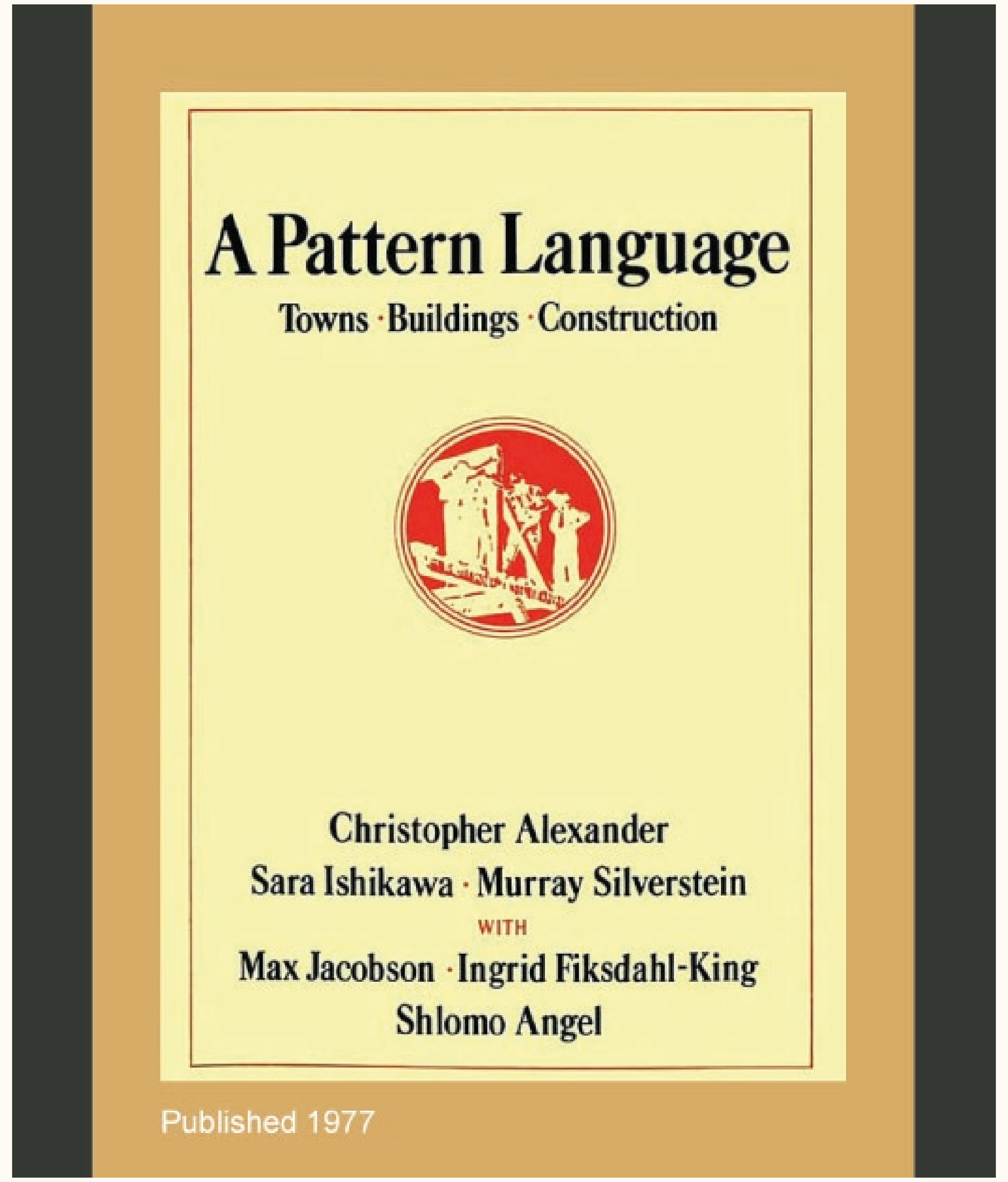
The book, not inherently computational in nature, is a text that presents a set of design patterns for architecture, urban planning, and other related fields.

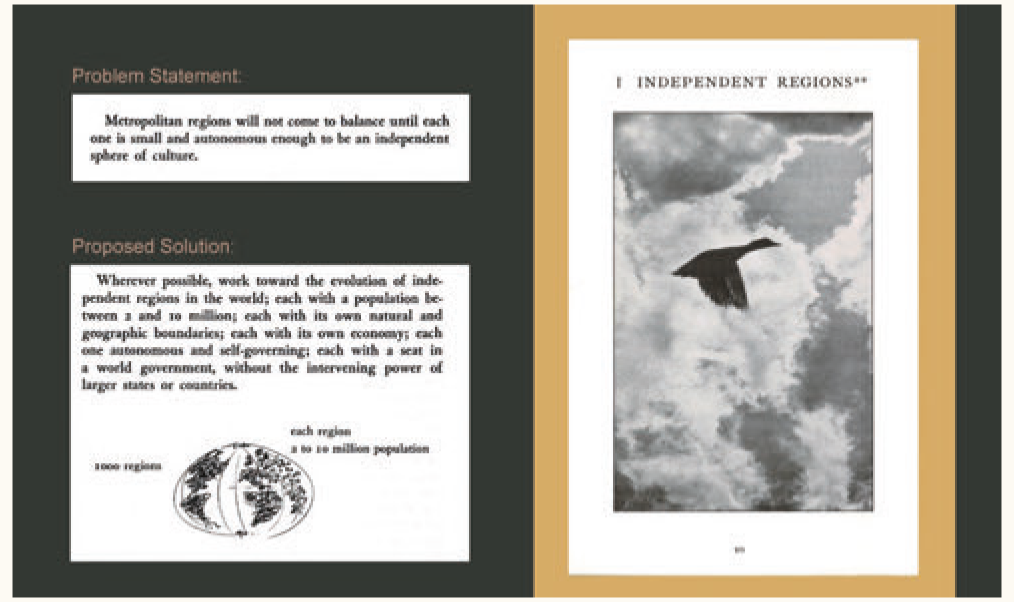
It's approach to design and problem-solving can be seen as computational in some ways, however. For example, the authors propose a hierarchical organization of patterns, where larger patterns are composed of smaller ones, and so on.

This can be seen as similar to the hierarchical organization of functions and modules in computer programming. Therefore, deriving any specific, computational methods from A Pattern Language proved challenging as the books 'computational logic' is just that; - *'computational logic'*.

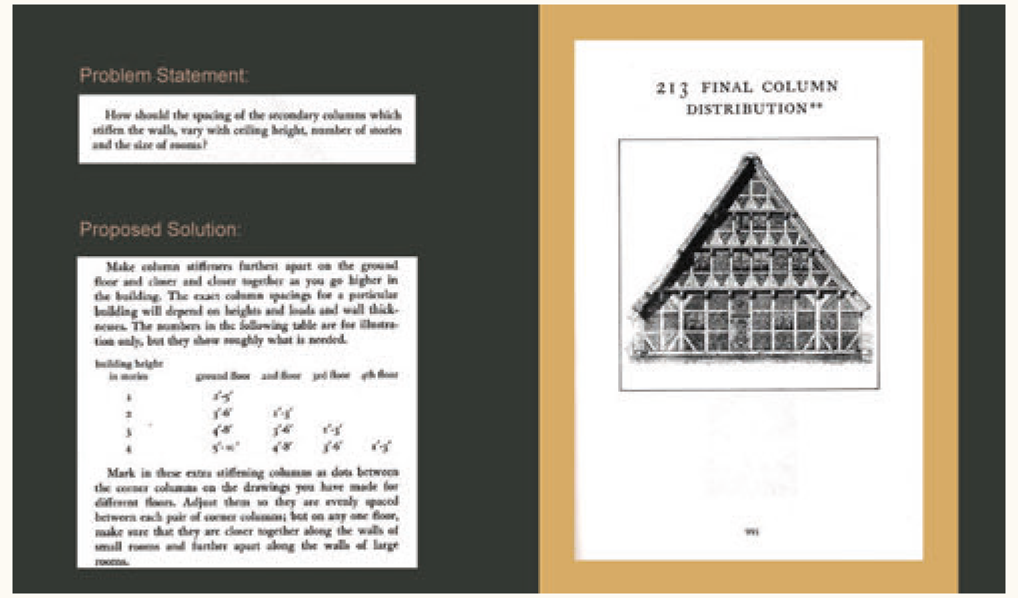
Resultantly, I focused on the other main theme of the book. It's defined *'how to'* quality, providing design solutions using standard elements for anyone to use. This is the 'logic' I'll use in this project.

With Grasshopper and Rhino, I will produce a tool to create playful forms with standard elements that interactively informs the user on it's buildability. The building elements I will test are UK standard bricks. The tool will also be adaptable for use of other types of common, standard construction elements.





Pattern Example:
Pattern 1



Pattern Example:
Pattern 213

Typology

The architectural typology for this project is a shelter.

A shelter is described as a place giving temporary protection or a shielded or safe condition.

The limited number of users restricts the size of the architecture to that of a pavilion. The pavilion typology will be considered throughout this project.

The 'pavilion' has been generated though coming up with a overarching theme, in this case 'play'.

The formal generation of the shape grammar will be made by manipulating a key structural, material, mechanical or textural component from a non-planar starting point.

Each conceptual stage within the formation of the end product will all be influenced by the theme, 'play'.

The deliverables of the brief mention the *interpretation of fundamental Pattern Language and form-finding theories*. The theory for this project is *The* Pattern Language.

A Shelter for Play

Within the group work stage of this project, we produced the following introduction:

"Our aim is to produce a shelter for play. We will abstract computational methods from our relevant theories that promote responsive, playful engagements with our built form."

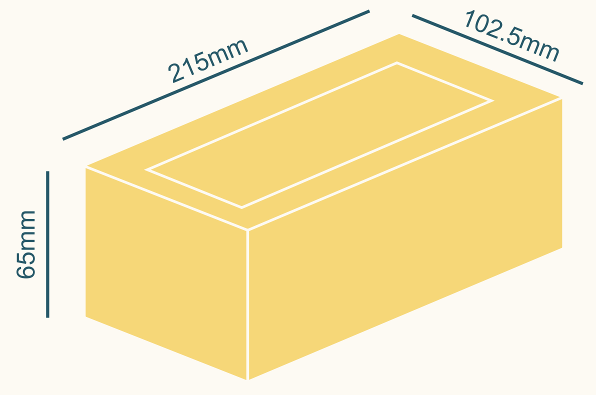
The four themes we decided to be explored in a playful way are: sight, touch, navigation and light.

Individually, I was to investigate how 'sight' might inform playful engagements with a built form

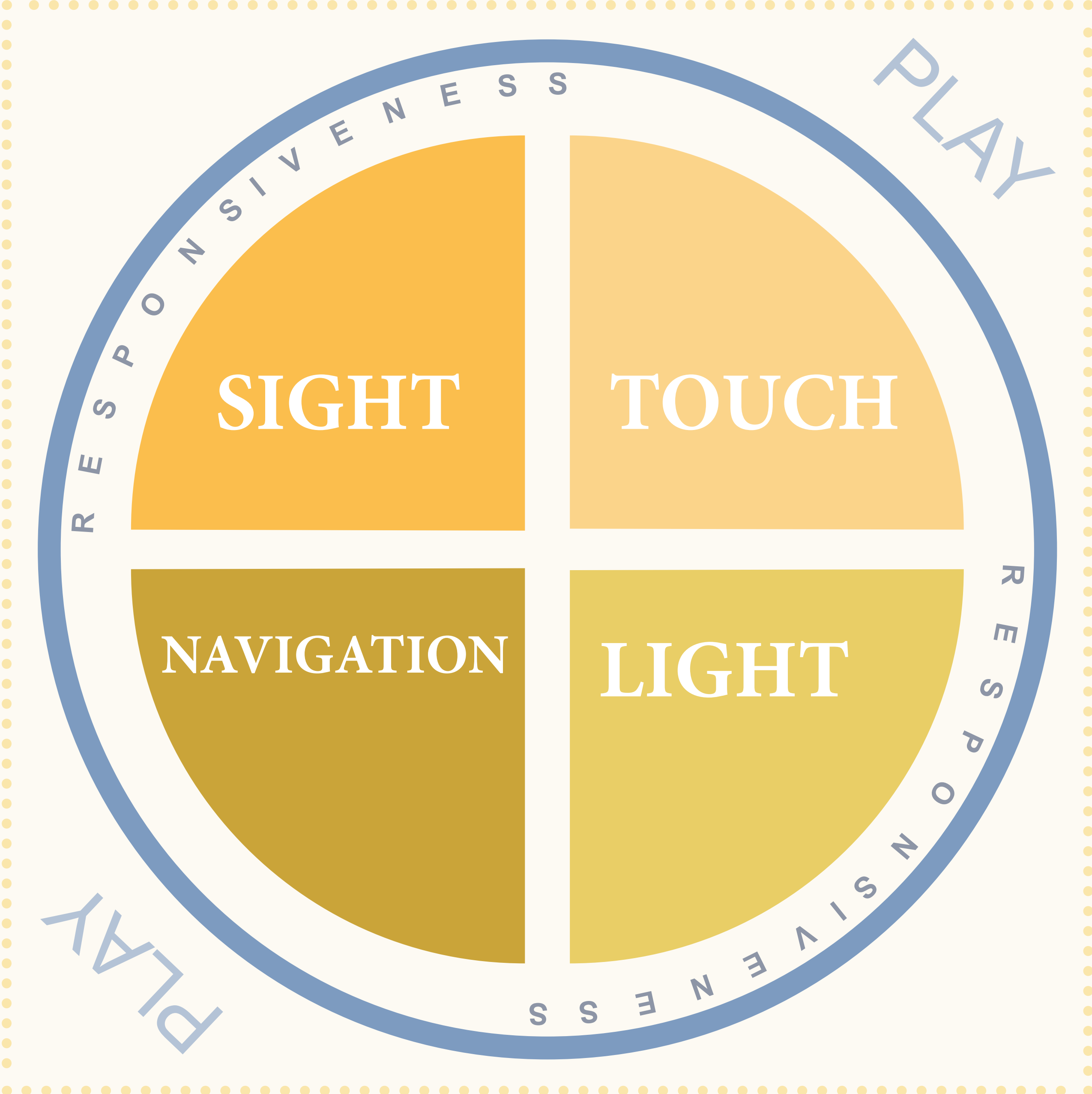
A Standard Element - Brick

The standard brick (215 x 102.5 x 65mm) is that exact size because of user-centred design. This size optimises building efficiency and cost.

They are also widely available and inexpensive - both of which are attributes that A Pattern Language encourages users to seek when building.



UK Standard
Brick Size



Emergent Logic

The emergent logic derived form A Pattern Language seeks to rationalise the main objective of the text - *user centred design*, and how this could dictate shape grammar and form generation.

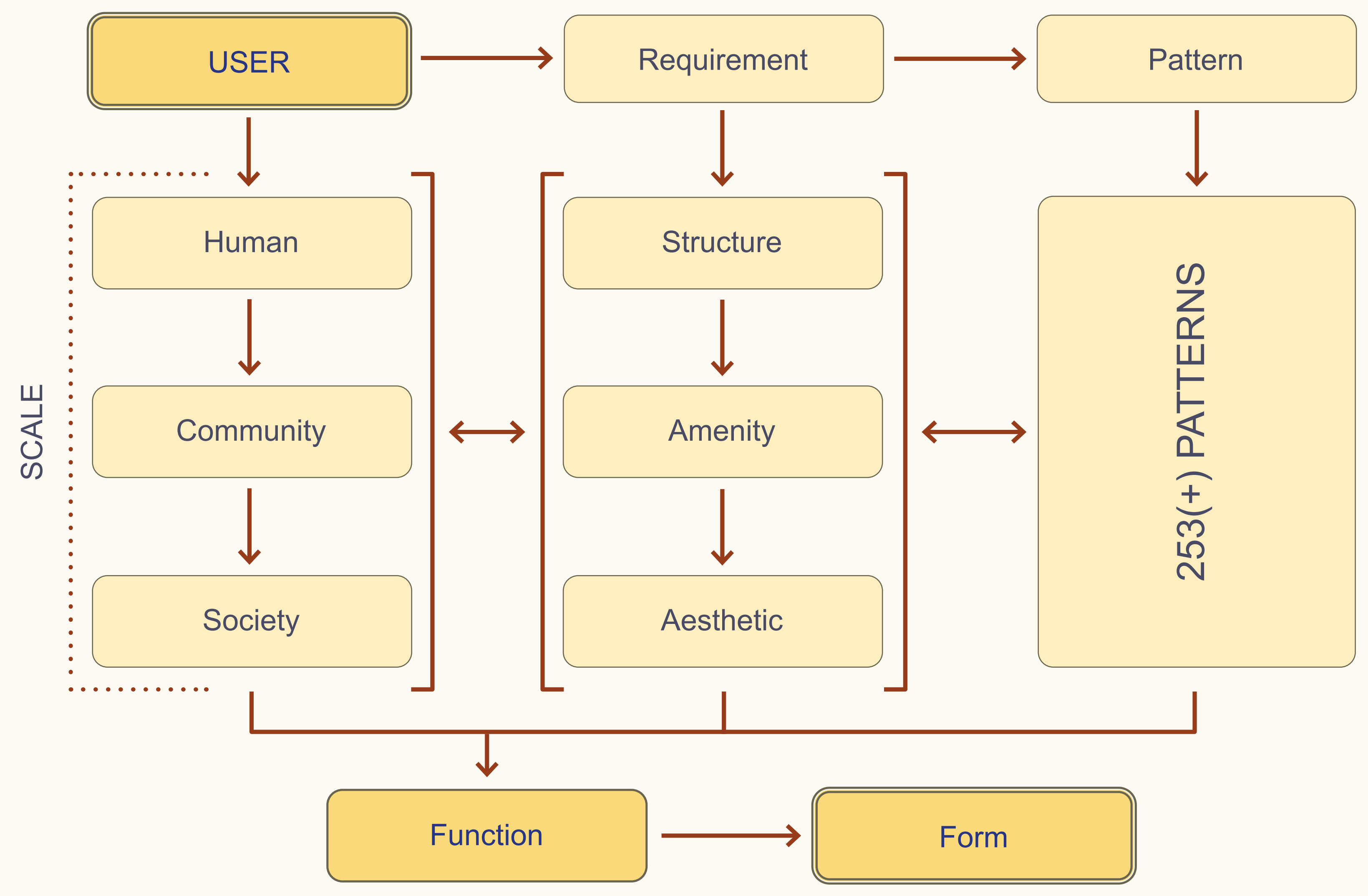
Starting with the 'user', we look at scale in one axis, and 'Requirement' in the other. All categories and variables are connected and interchangeable.

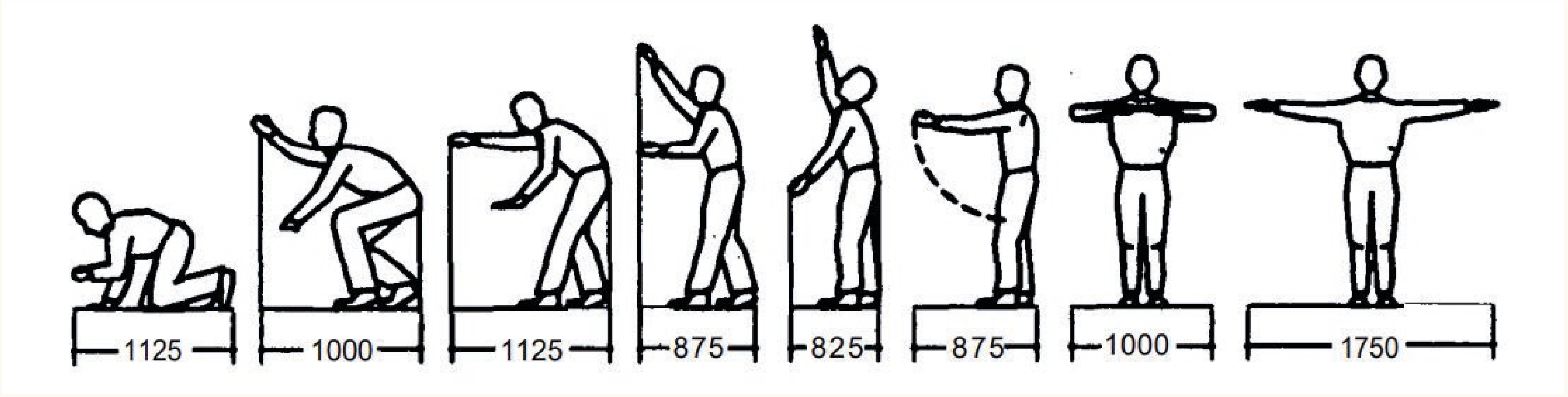
However, each route - from User to Form - can represent any form of building architecture.

In the case of this project, because of the limited user base - it will naturally focus on the 'Human' scale with my primary 'Requirement' looking at structure.

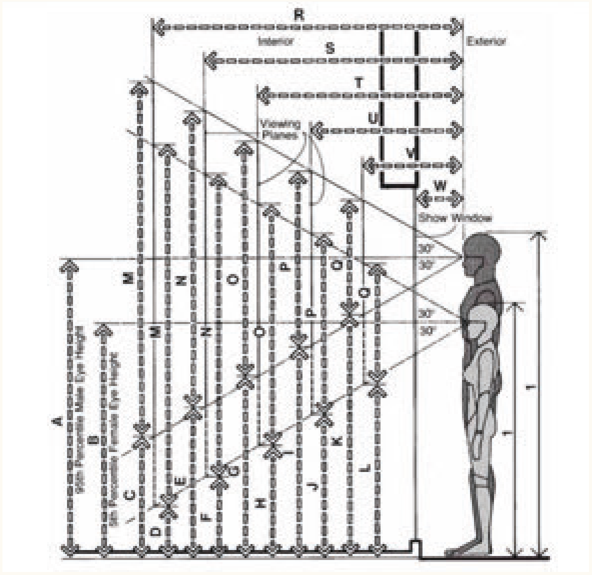
To generate a playful response from the user, the interact-ability of human scales and features are considered.

Individual experiences of 'sight' of users of different scales and acceptabilities will yield varied, but generally homogeneous results. In that, any user will be able to interact with the pavilion.





The Metric Handbook



Human Dimension and Interior Space

Brick Tool

This brick tool's primary function is to test parametric and non-regular brick wall designs. The tool will also convey to the user where the geometries become too extreme, resulting in the brick-building logic graphically displaying the unbuildable parts of the structure.

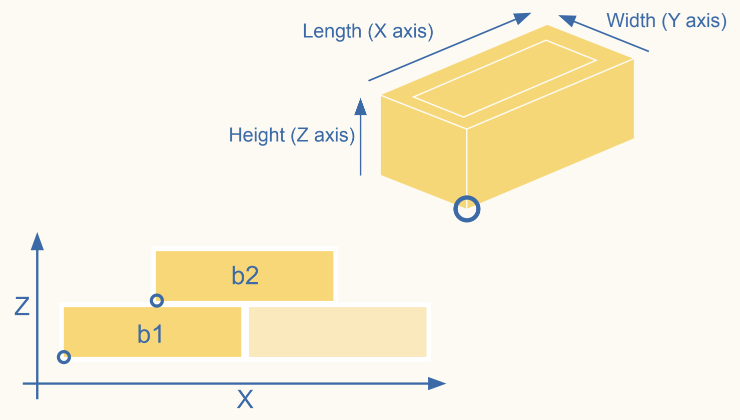
Similar to A Pattern Language, the Grasshopper tool will allow 'anyone' to generate otherwise incredibly complex geometries and subsequent structures using standard cuboid blocks. This project will be focusing on using a UK standard brick.

In regards to the how this relates to 'play', and my theme of 'sight', the nature of the brick pattern generating along a irregular, non-planar surface between two lines of varying length, produces varying gaps between each bricks in each row. These gaps allow light to pass through but also users to look through the structure.

As the structures become more complex, the gaps between the bricks become more dynamic and irregular.

Brick Logic

In a regular brick wall, the construction logic is defined in 2 dimensions, XZ:

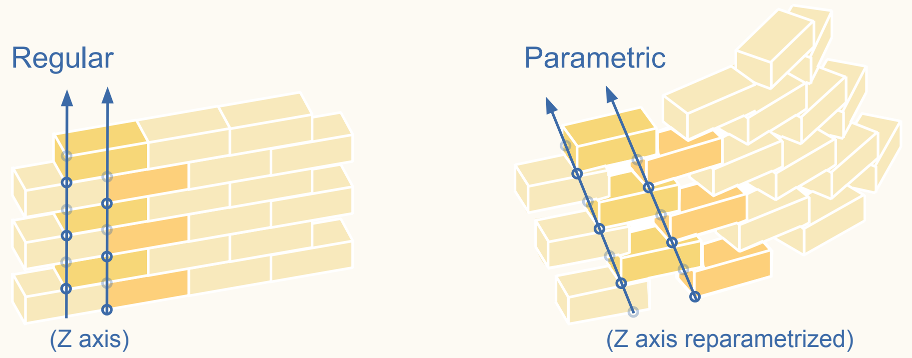


Where, b1 is distributed along the X axis by the length of b1. b2 is then added above in the Z direction by the hight of b1 and along the X axis by a value of half of b1.

In short the resultant vector for a **regular** brick wall:

$$\text{The position of b2} = \frac{\text{Length of b1 (X axis)}}{2} + \text{Height of b1 (Z axis)}$$

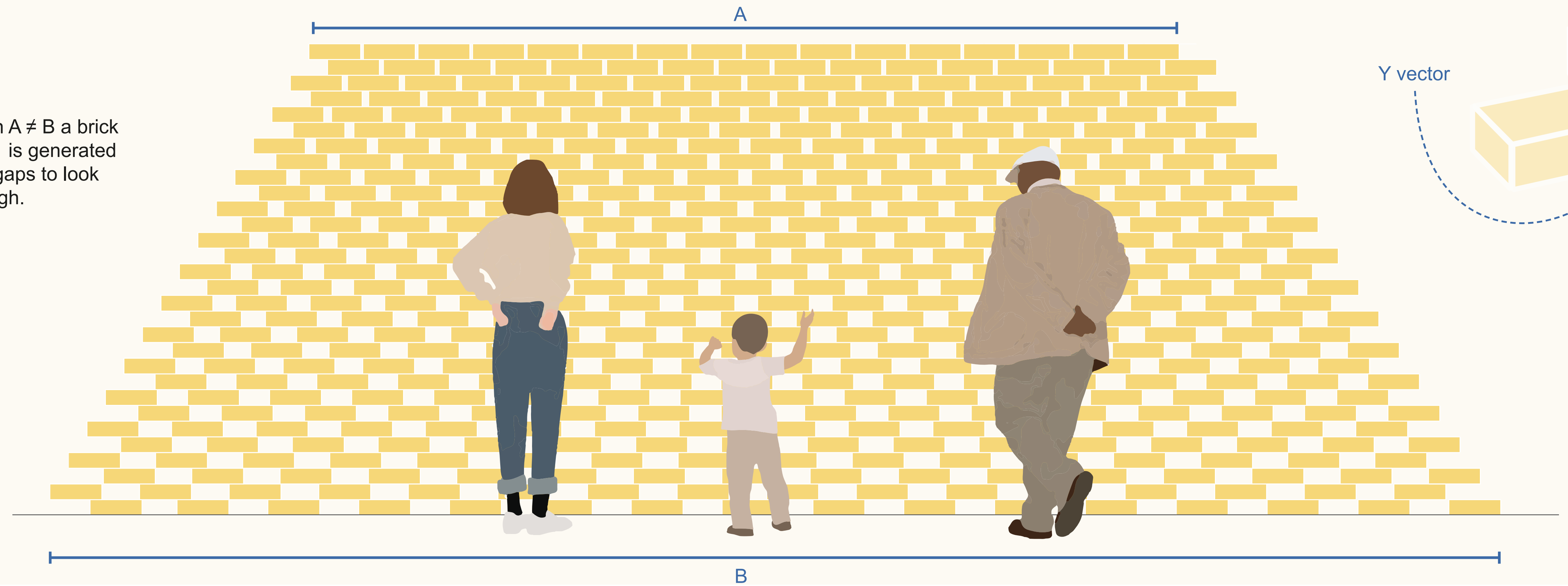
When the value on the Y axis increases for each row, the brick wall is no longer regular and a parametric structure is formed.

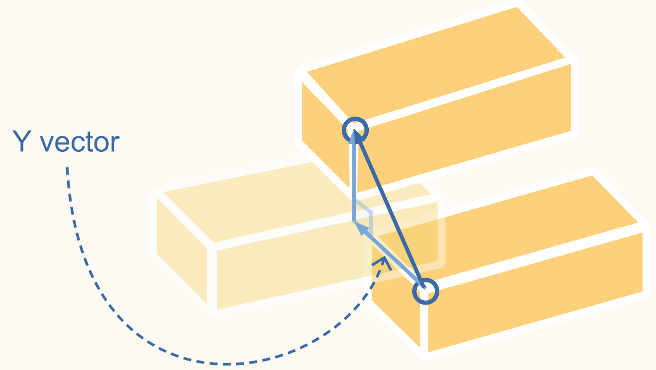


The logic will use the vectors generated between the (0,0,0) points of each brick and the brick above it. Because of the 'either/or' logic implemented to select every other point in the row, the next brick up is two rows in the z direction.

By deconstructing the vector between the two bricks and extracting the value of the vector in the Y axis, the brick overlap limit can be defined. This is how the Grasshopper tool will decide whether the parametric brick wall is 'buildable'.

When A ≠ B a brick wall is generated with gaps to look through.





Location

Location: Greville Smyth Park

A pavilion in the park, next to an existing playground, paths and football pitches.

The pavilion created will sit comfortably within this context, producing an interesting and playful structure.

Site

Site Plan

Site Location - Aerial View

Playground

Football Pitches

Precedents

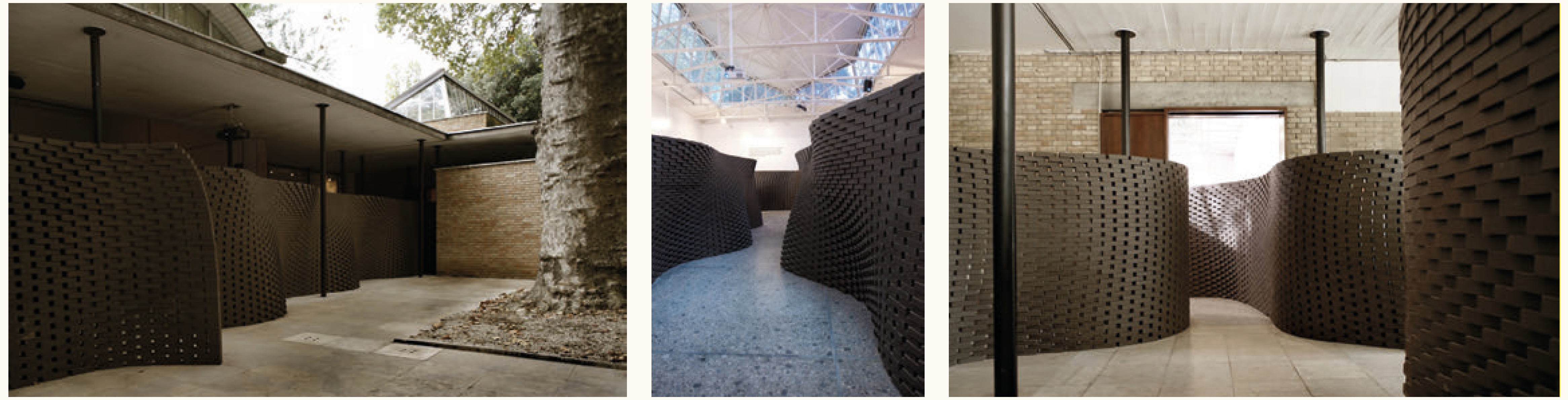
These are the precedents initially looked at to inform the logic of this project.

The qualities exhibited by them are useful to understanding how 'brick logic' is implemented within parametric design.

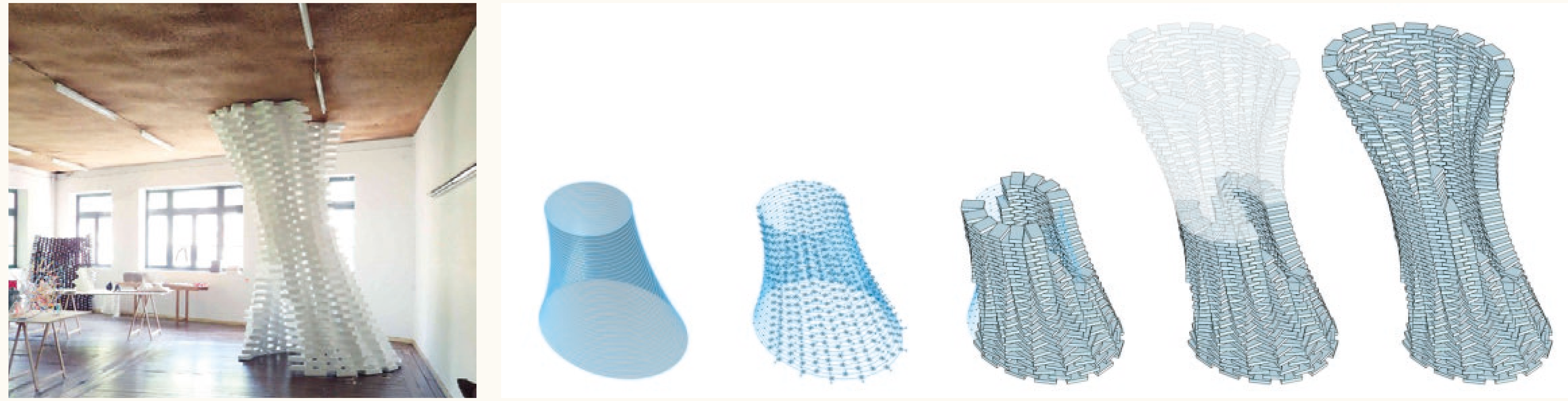
In the case of the Brick Tower by the Digital Fabrication Laboratory, they provide a useful digital work-flow diagram, showing how their complex brick tower is generated from a surface, to points mapped along it, to individual bricks mapped to each point.

While each example uses different materialities and scales - they all use much smaller, human-scale, standard elements to generate complex forms.

HKU's robotic fabrication LAB 3D prints terracotta brick enclosure at UABB shenzhen



A 100 meter long brick wall - The Swiss contribution to the 11th Venice Architectural Biennale 2007–2008



BRICK TOWER by DFL (Digital Fabrication Laboratory)

Methodology

I propose to test the Grasshopper brick tool by trying to understanding it's geometric limits in wall generation.

I will test different pavilion forms, starting with simple, irregular non-planar surfaces, advancing to more complex and intricate with each brick test iteration.

Each wall-generating, non-planar surface is formed from two poly-lines. The complexity of these lines will produce more intricate brick generations as they become less linear.

Once the brick tool shows they are 'buildable', they are suitable for fabrication.

Will it build?

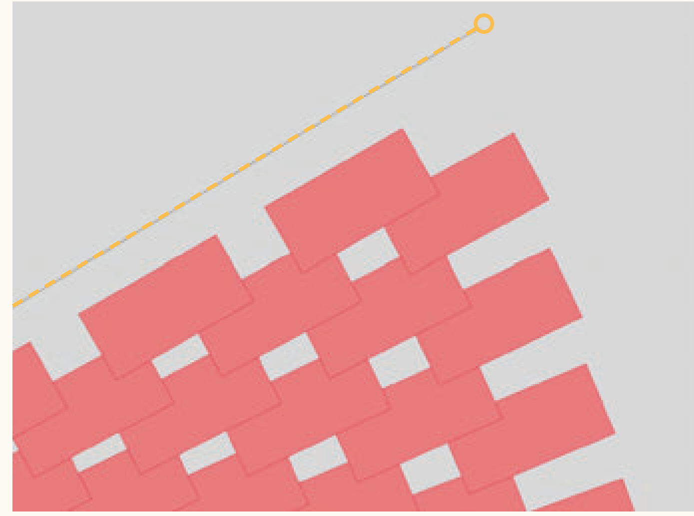
Test - Control

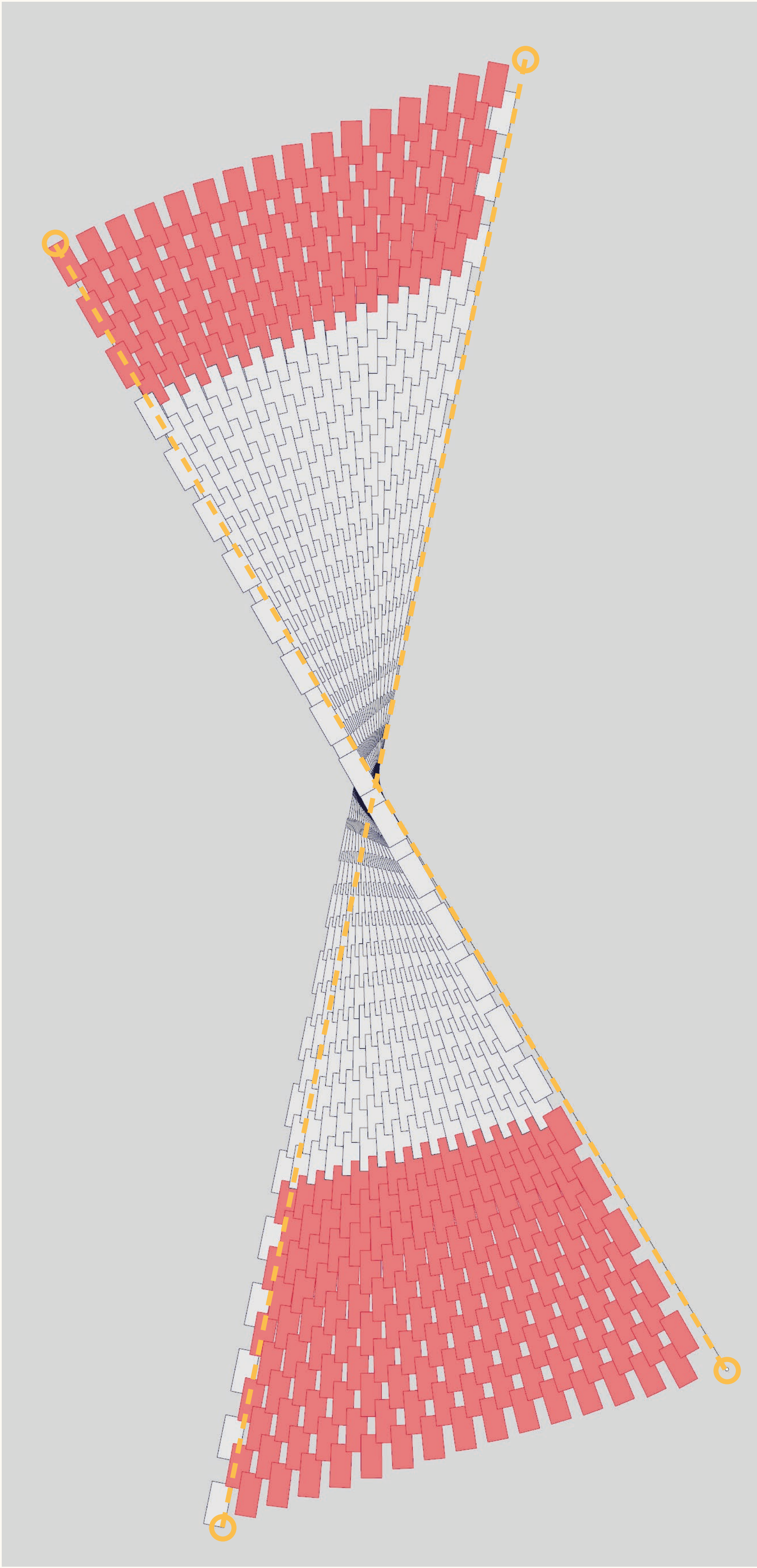
The first example of the tool depicts a 'control' brick wall, generated from two simple straight lines of different lengths. This tests whether the grasshopper logic is functioning.

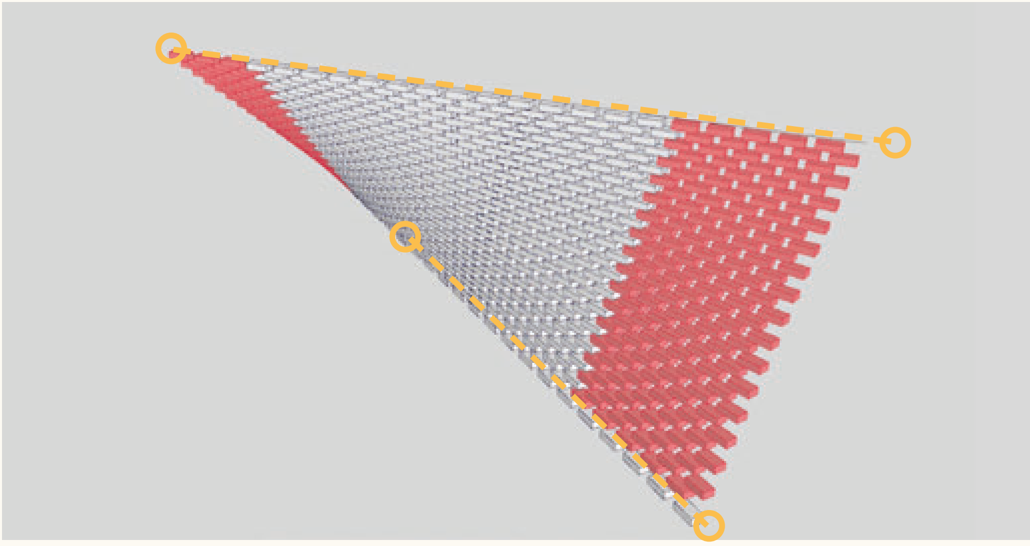
Demonstrated in the Rhino screen-shots, the wall is generating two different colours of bricks. The white bricks indicate the 'buildable' part of the proposed wall, and the red depicts where the overlap distance is too large and therefore alerts the user that it's unbuildable.

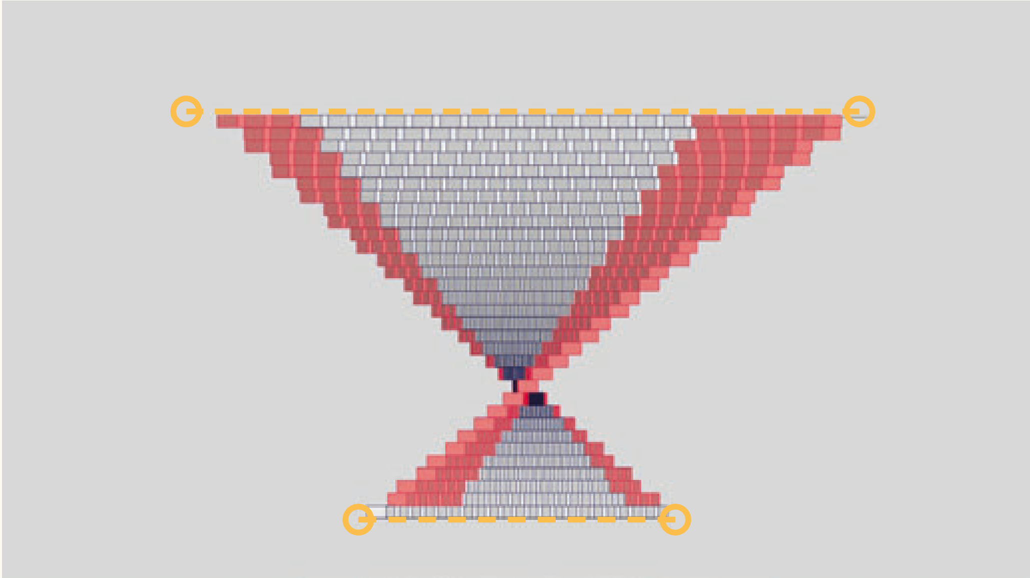
The first layer of the bricks on the 'ground' will always appear white as there is no brick row below it.

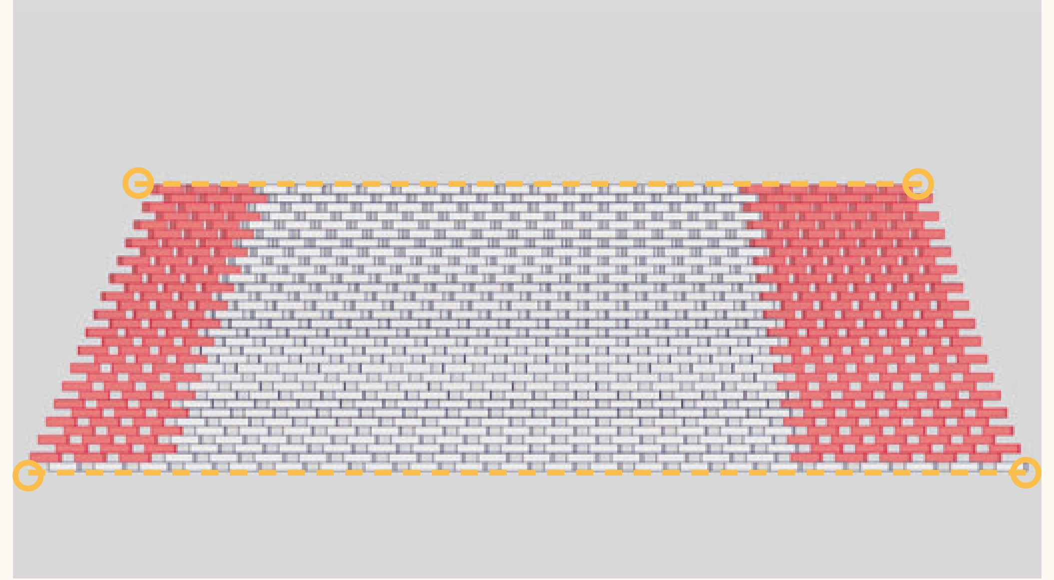
By adjusting the nodes that generate the shape of the wall to a less extreme distance from the other, the amount of unbuildable wall will reduce.


Close-up plan showing non-buildable bricks

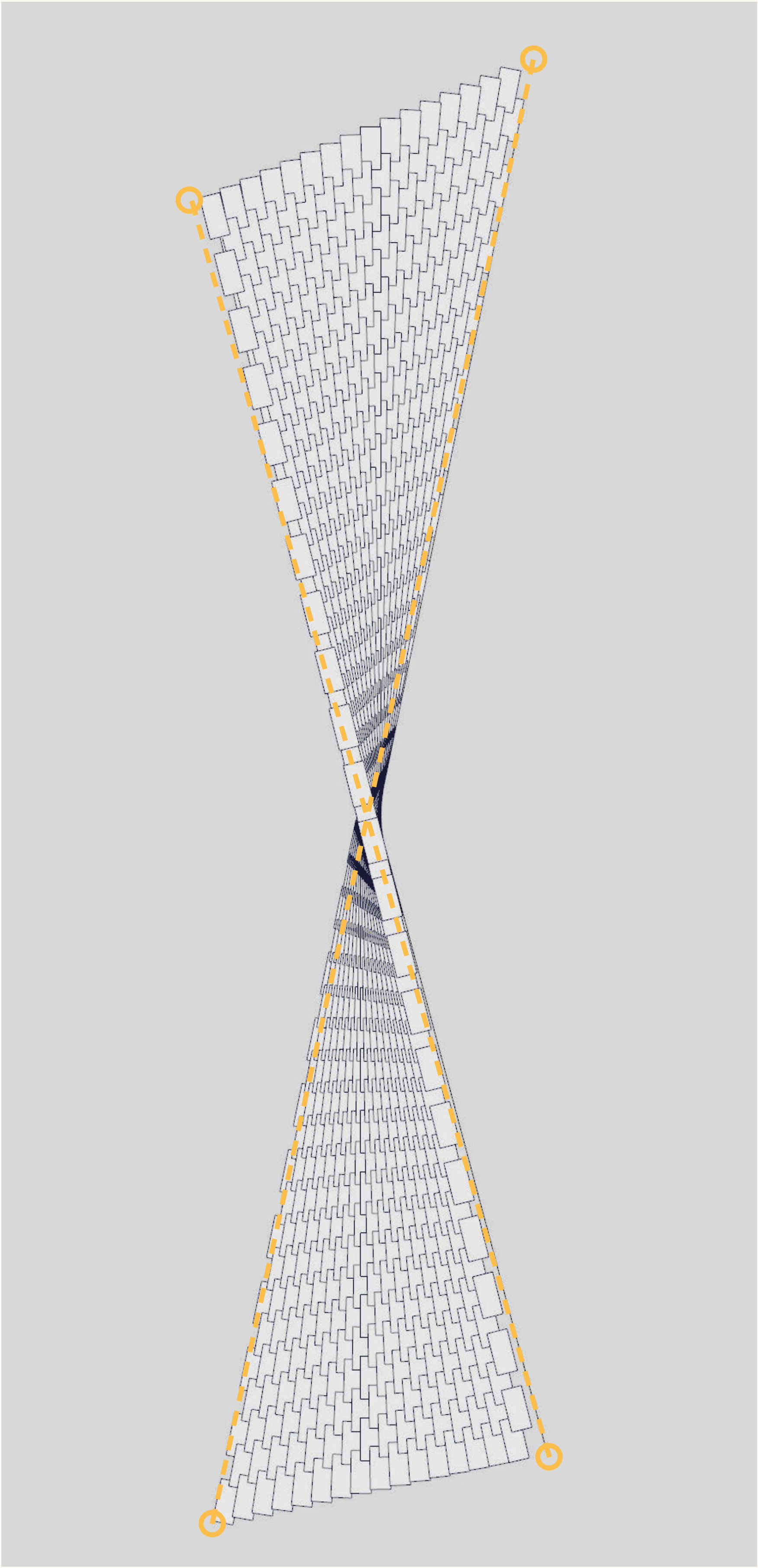

Control 1 - Plan

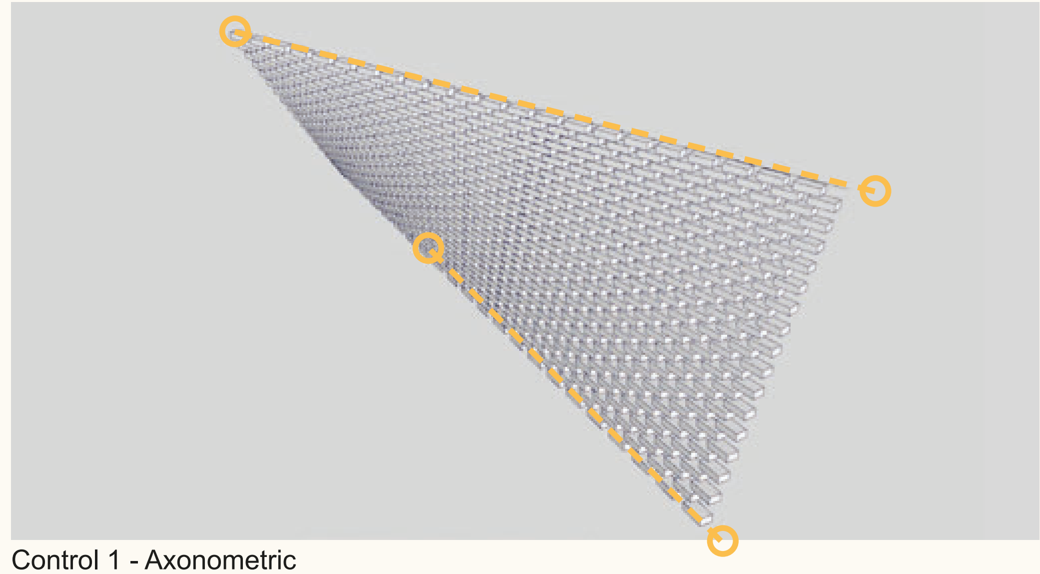

Control 1 - Axonometric

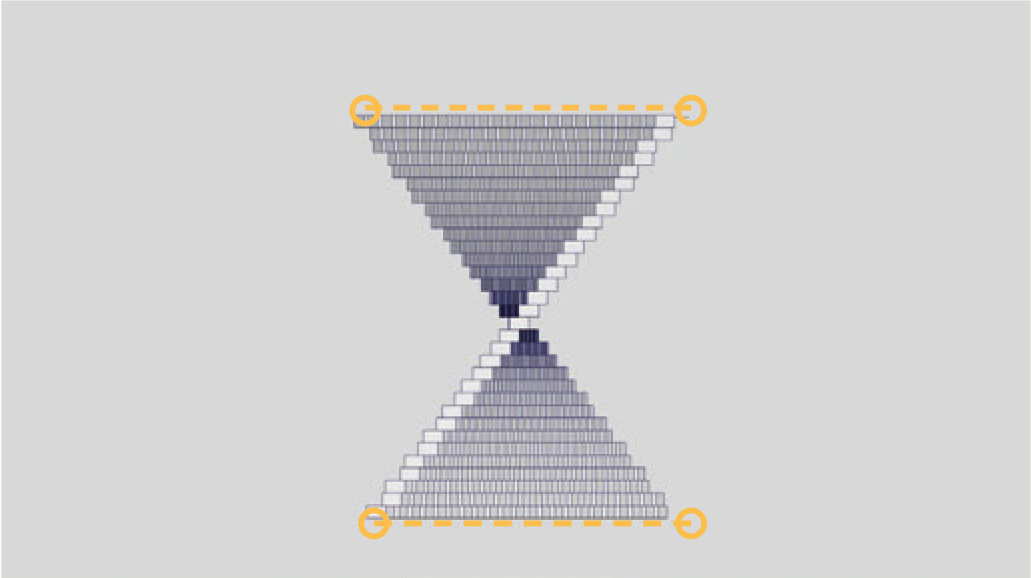

Control 1 - Side Elevation

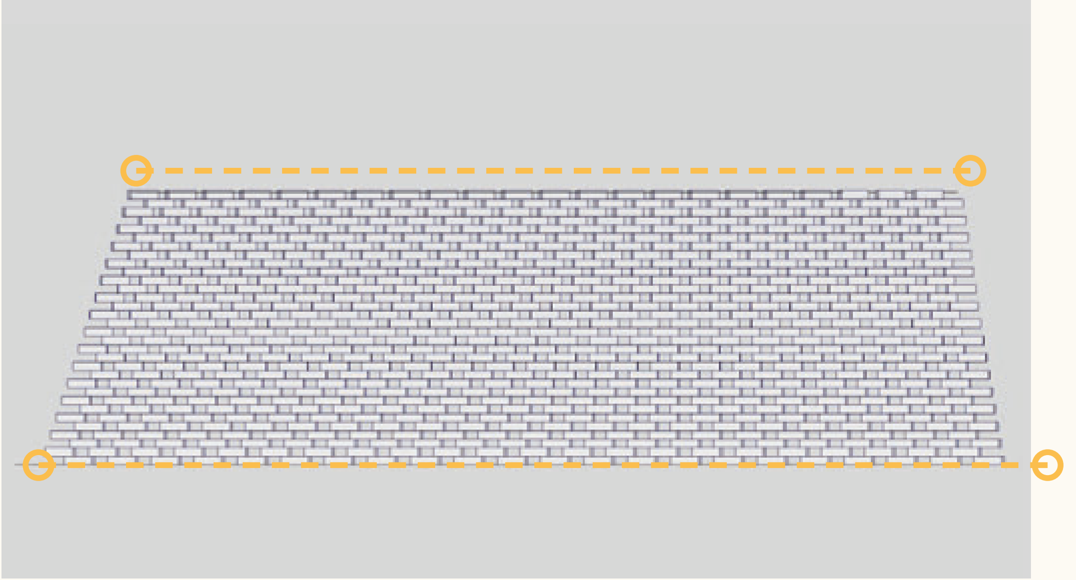

Control 1 - Elevation

Now that the nodes of the polylines have been altered to less extreme positions, the tool depicts a white wall - indicating the structure is now constructable.


Control 1 - Plan


Control 1 - Axonometric

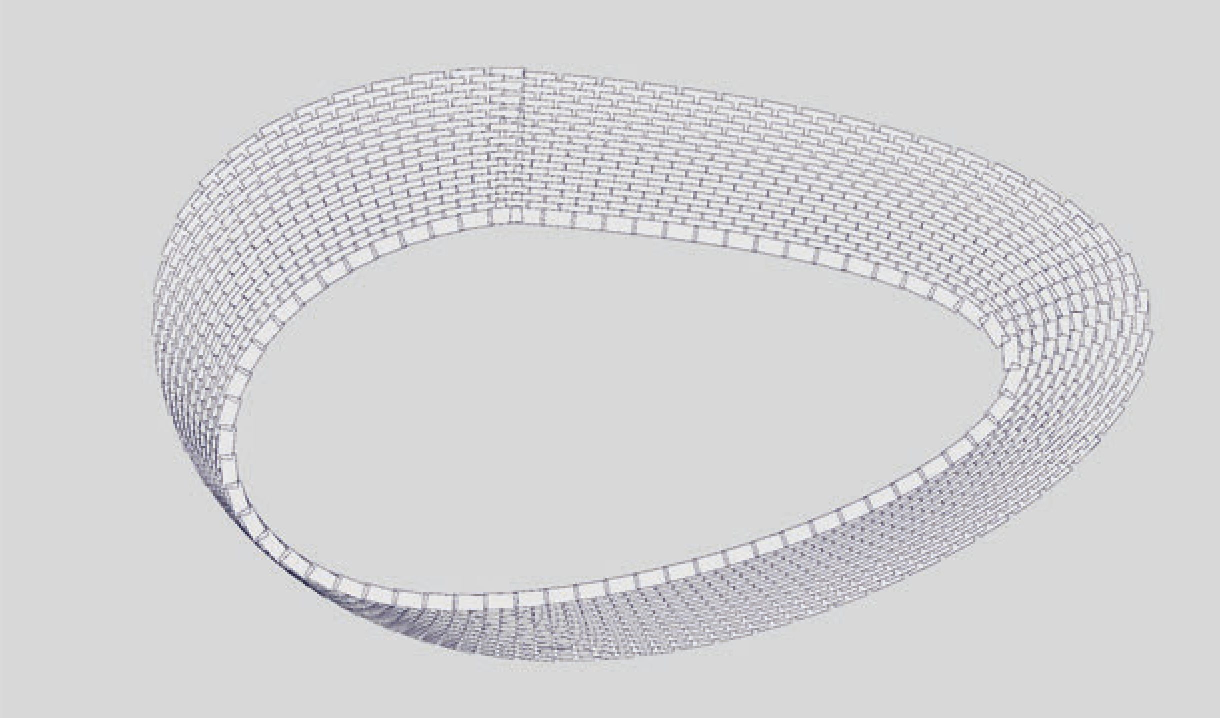

Control 1 - Side Elevation


Control 1 - Elevation

Other Block Type Test

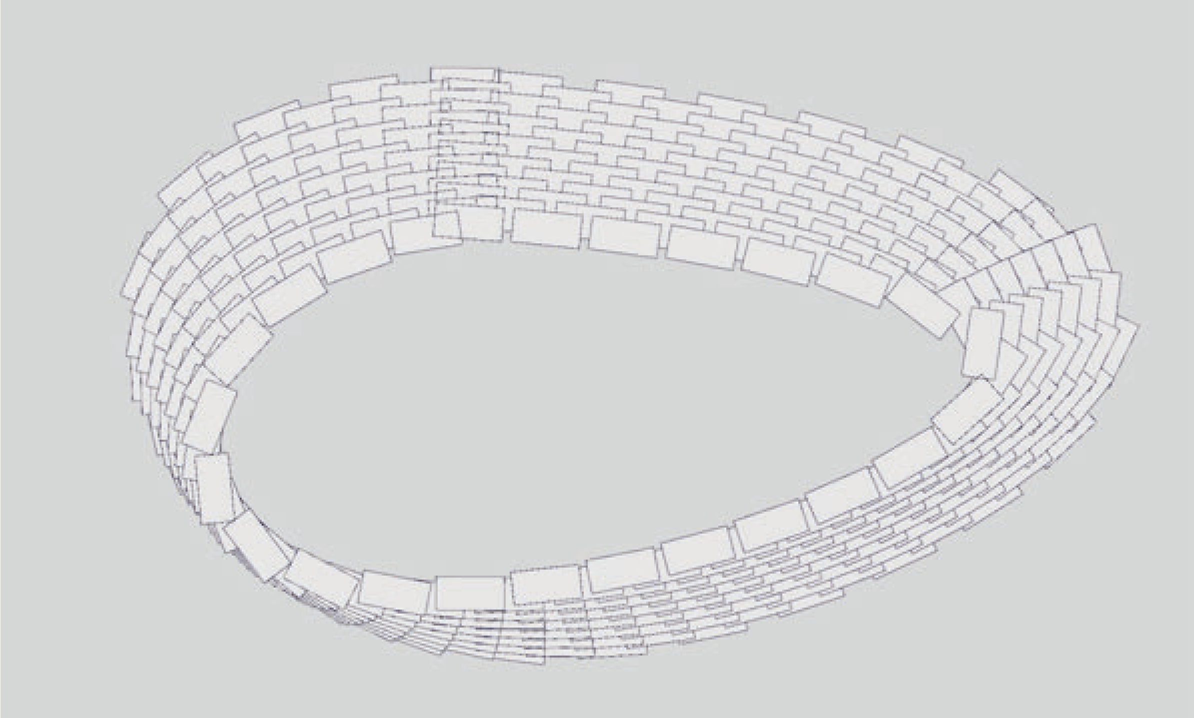
The tool is designed so that adjusting the block dimensions to satisfy a different standard element - in this case a 'Breezeblock' - the Grasshopper logic will continue to generate the structure.

The block forms below have the same 'base plain' but the standard element has been changed.

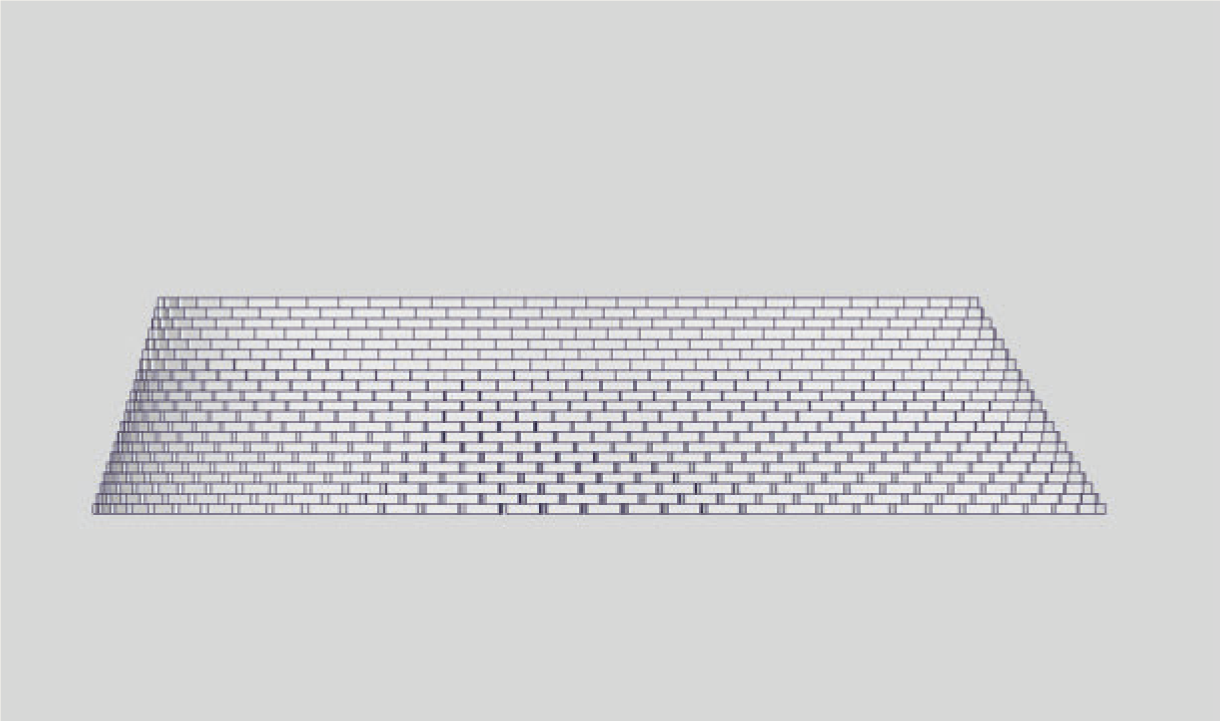


Uk Brick - Plan

Standard UK Brick
(215 x 65 x 102mm)

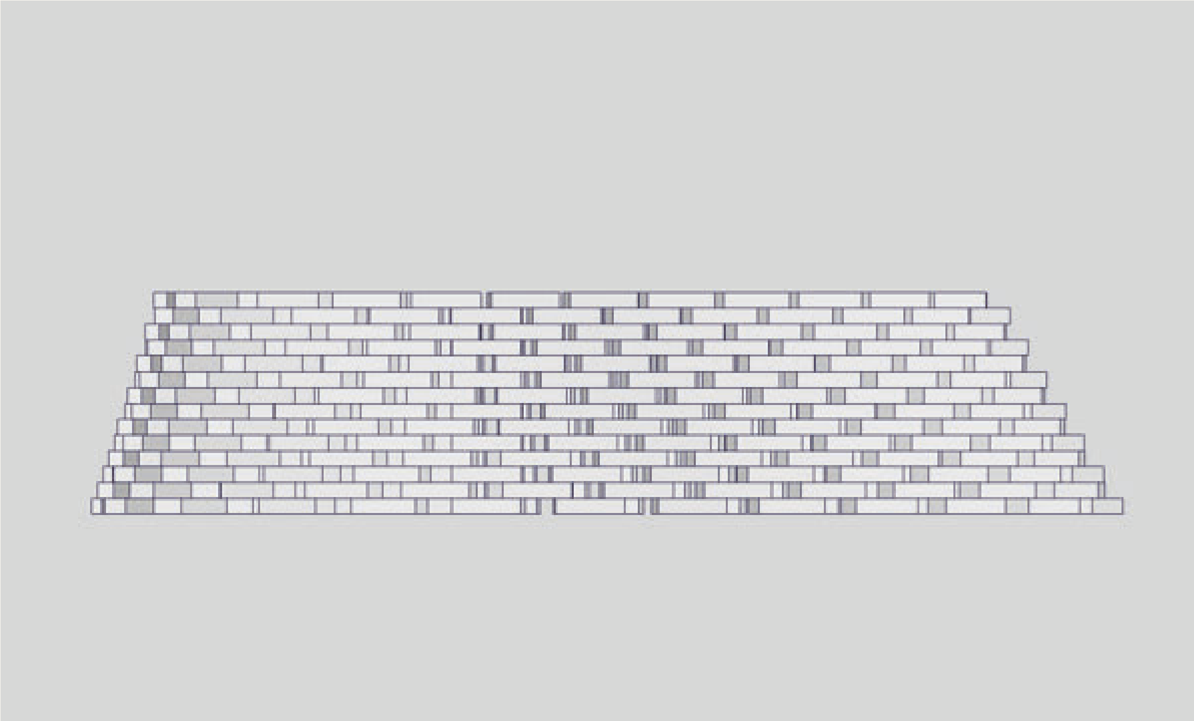


Concrete Breeze Block - Plan



Uk Brick - Elevation

Concrete Breeze Block



Concrete Breeze Block - Elevation

These tests have been produced using the geometry from 'Brick Test 1' - The form finding for Brick Test 1 can be found on page 12 and 13.

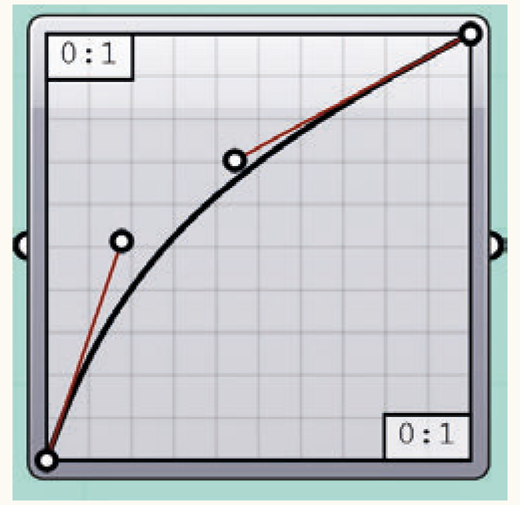
Threshold Tool

It was important to add the ability to create thresholds within the generated walls while also being able to move the position along the wall as needed.

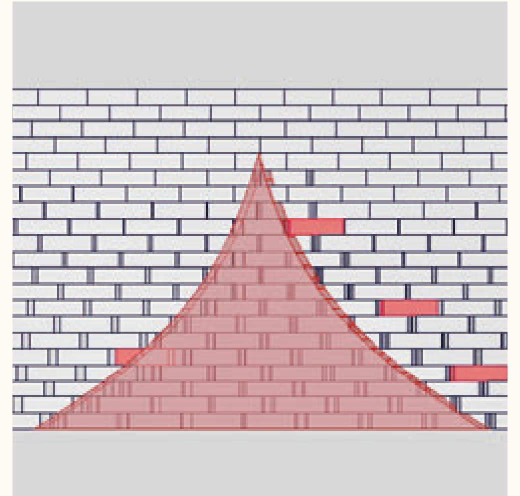
A regular cylinder was used initially in order to test the Grasshopper logic, ensuring it worked. While the logic generated red bricks above it, it was apparent that this method wouldn't be constructable without further structural intervention.

In order to create a threshold that works within a parametric brick wall, a more triangular geometry required.

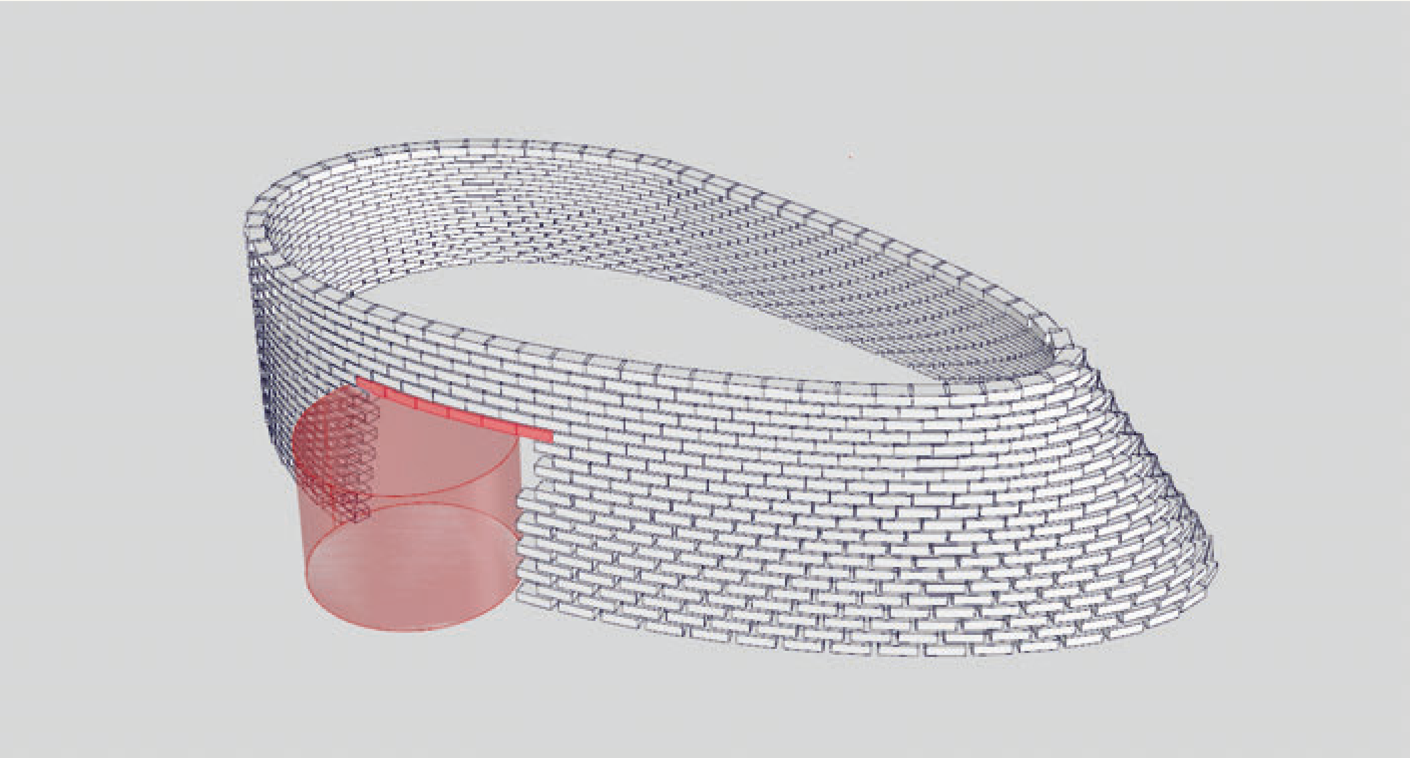
Implementing the 'Graph Mapper' component in Grasshopper, the user can manipulate the sides of an equilateral triangle to form sweeping arcs that adjust to the irregular distribution of bricks.



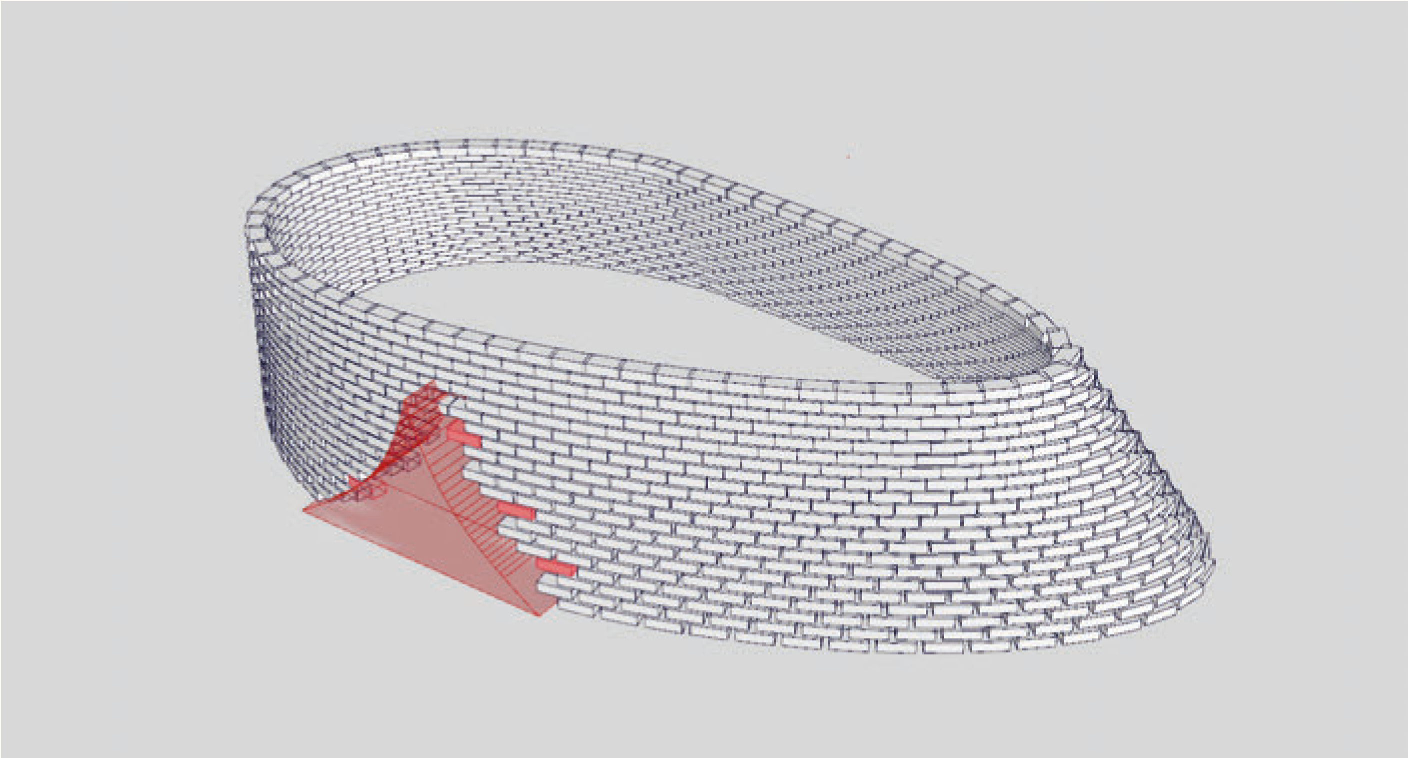
Graph Mapper



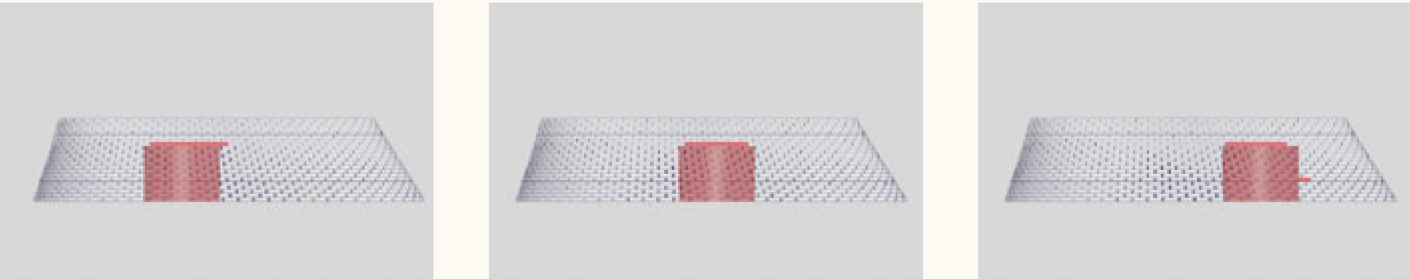
Resultant threshold geometry



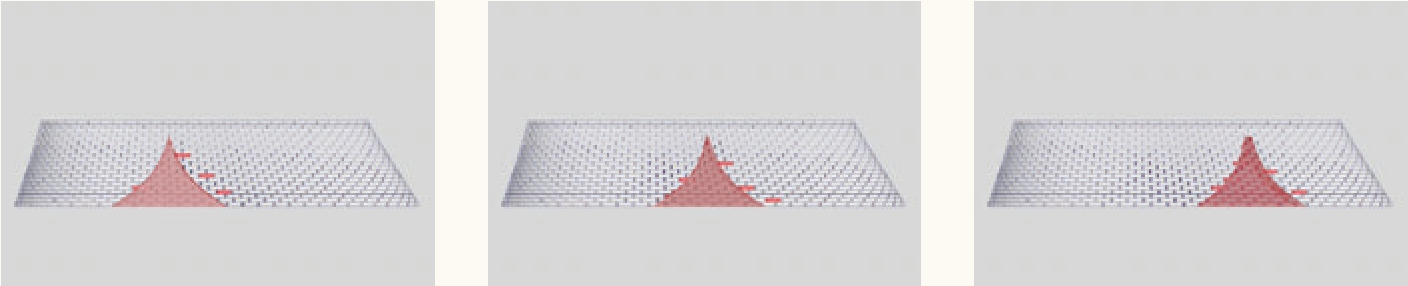
Cylinder 'Control' Threshold - Axonometric



Threshold - Axonometric

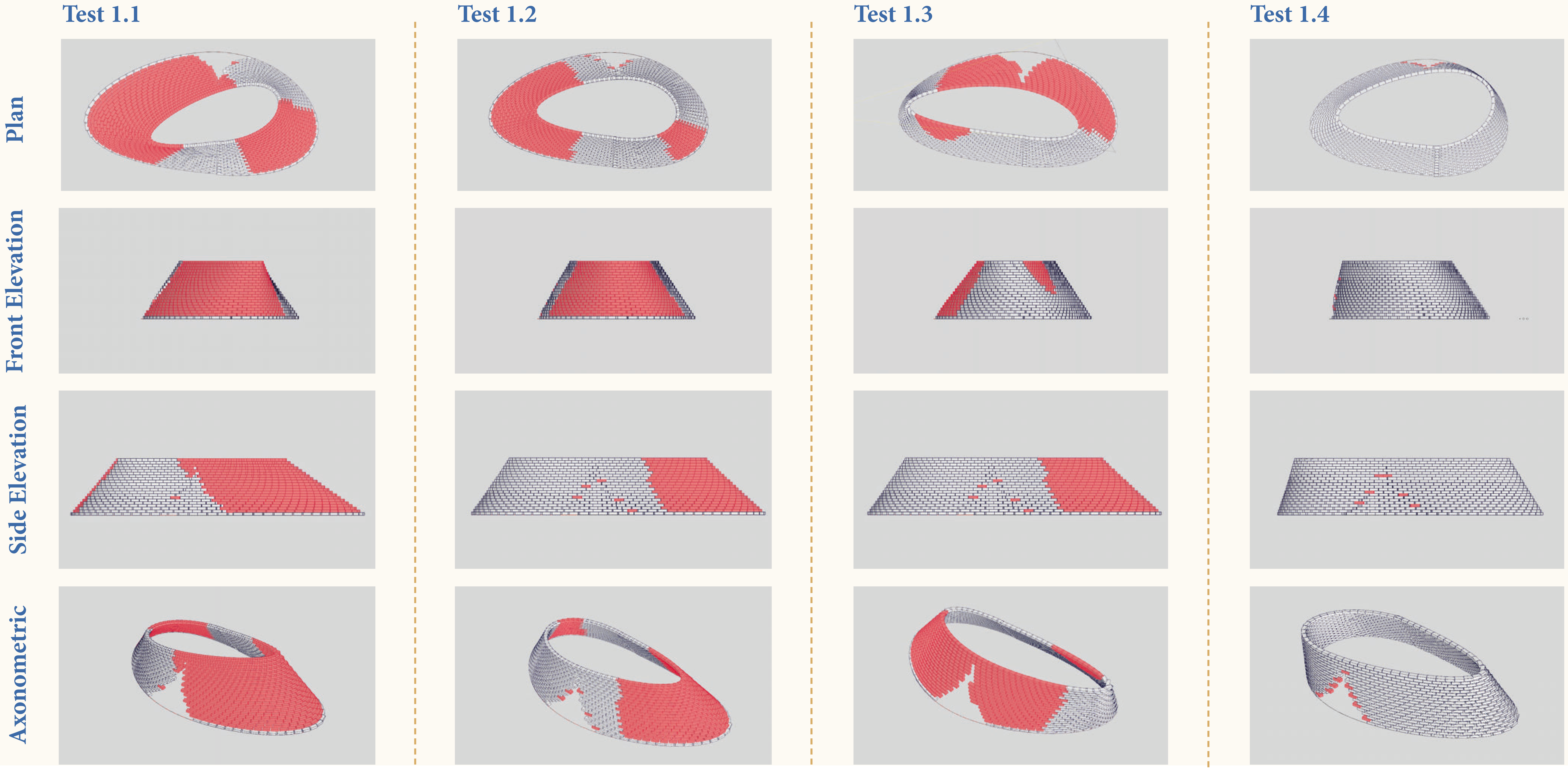


Cylinder 'Control' Threshold - Movements



Triangular/Arc Threshold - Movements

These tests have been produced using the geometry from 'Brick Test 1' - The form finding for Brick Test 1 can be found on page 12 and 13.



Brick Test 1

Each test begins with 'free-form', non-planar surface generated from two polyline curves.

The first test began with two simple, irregular, yet smooth curves.

Much like the Control Test, this was to see how the geometry responded to curved lines, while also enclosing a space.

Iteratively adjusting these curves to create a 'buildable' structure was straight forward as graphically it was evident where geometries were too extreme.

The first test was not necessarily very 'playful', but it was a good place to starting point in testing the viability of the tool.

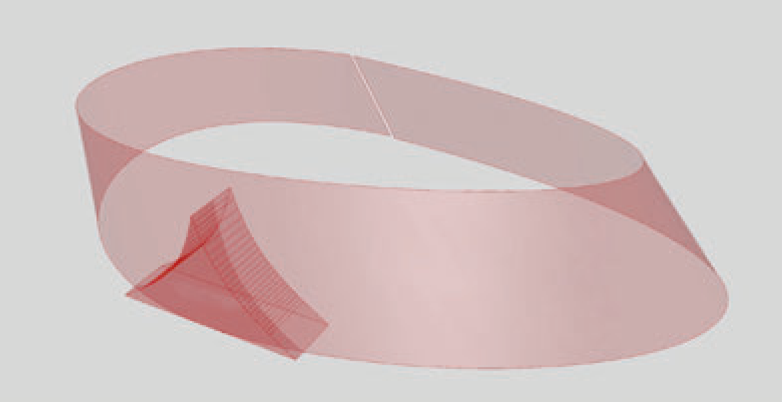
Brick Test 1

Outputs

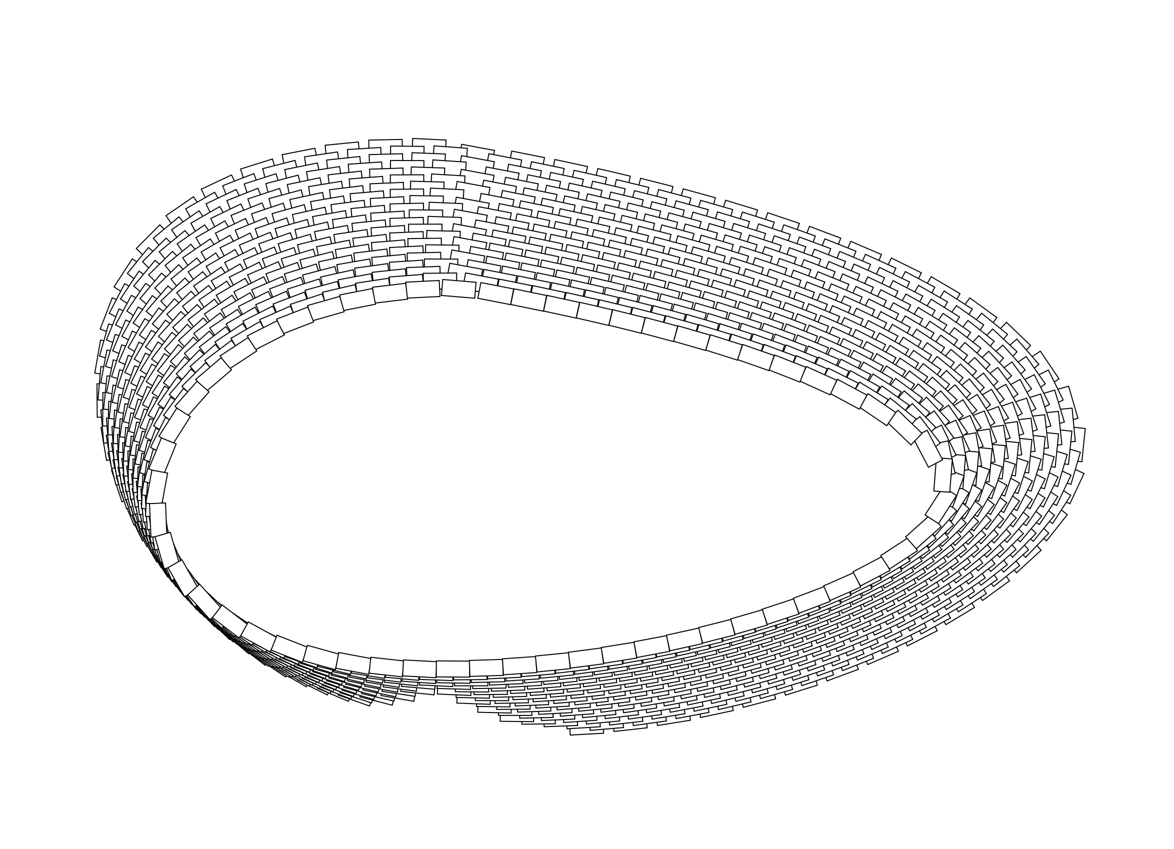
Once a 'buildable' outcome has been found, outputs can then be generated.

These can include 2D line drawings, 3D digital and 3D printed, scaled models.

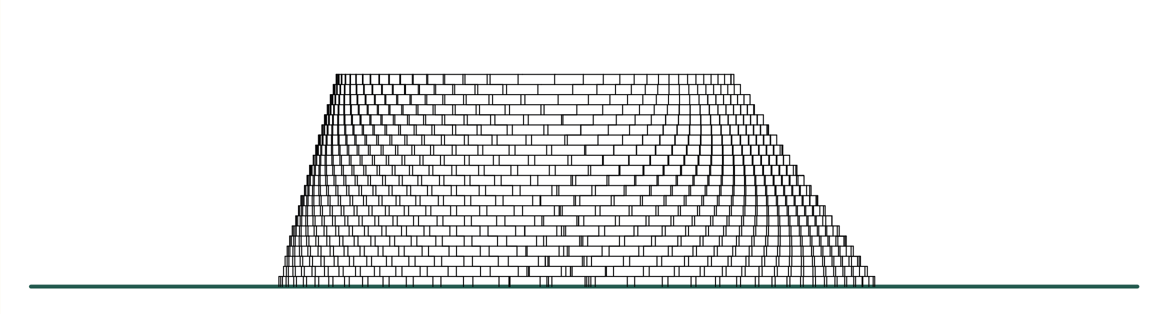
The 3D printed models offer a good example of how the pavilion responds to light.



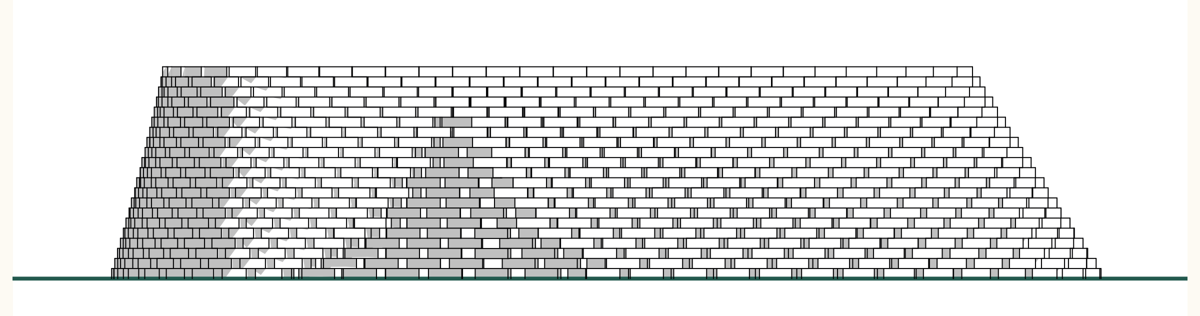
Plain and threshold - base geometry



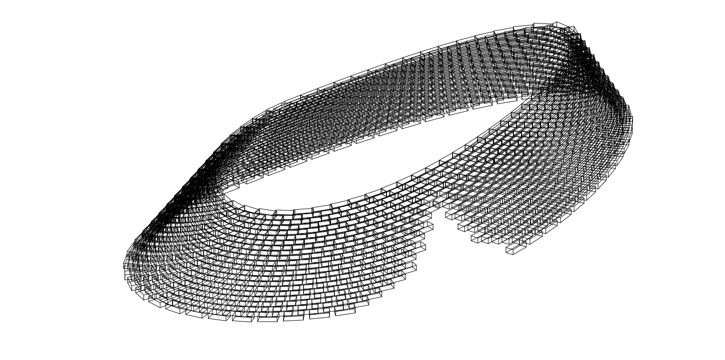
Line Drawing - Plan



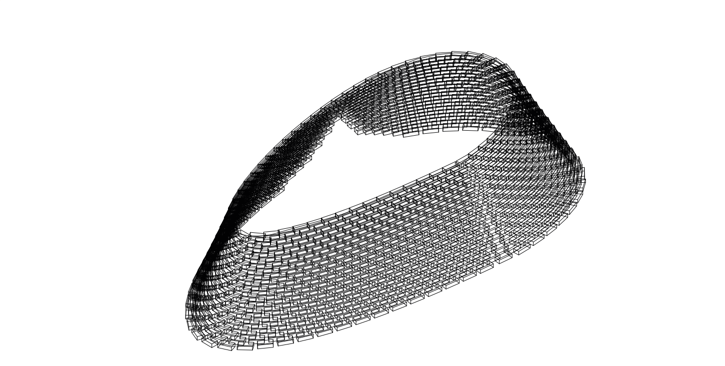
Line Drawing - Front Elevation



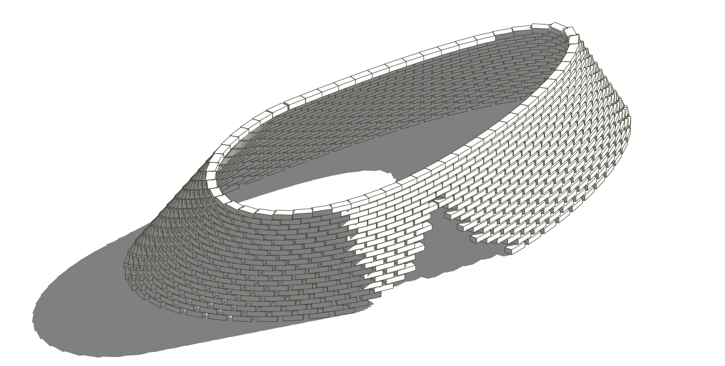
Line Drawing - Side Elevation



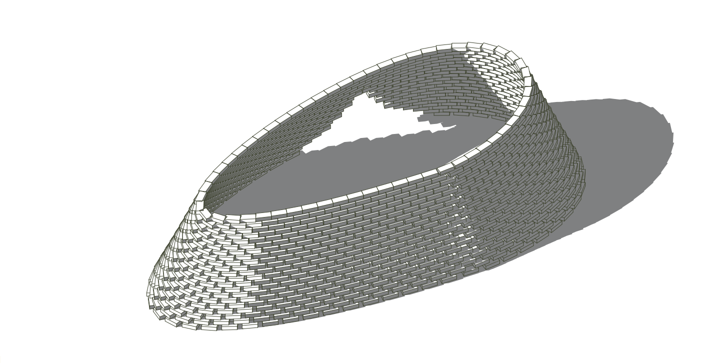
Wire-frame Axonometric 1



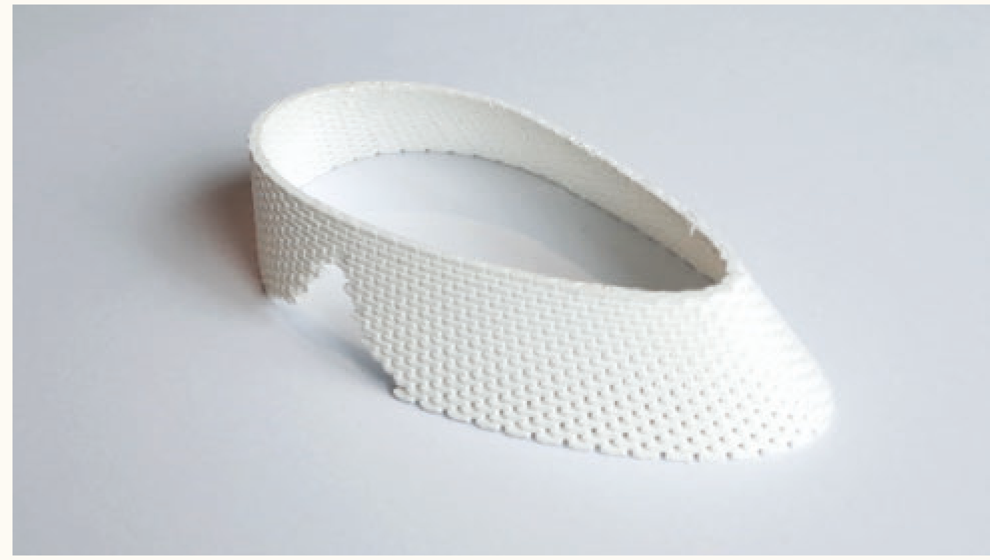
Wire-frame Axonometric 2

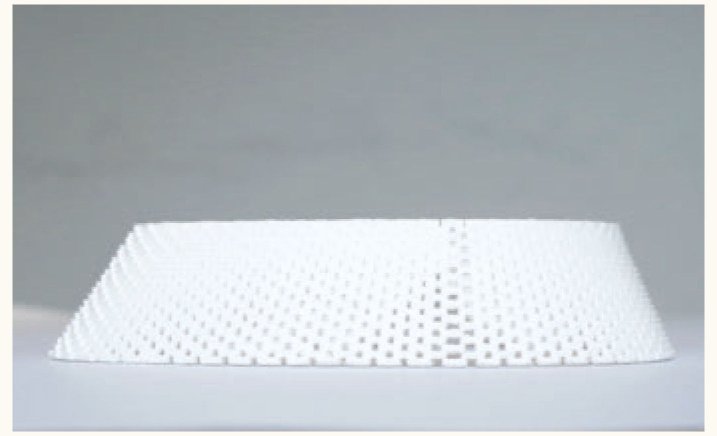


Axonometric 1



Axonometric 2





Brick Test 1 -
3D Printed Model

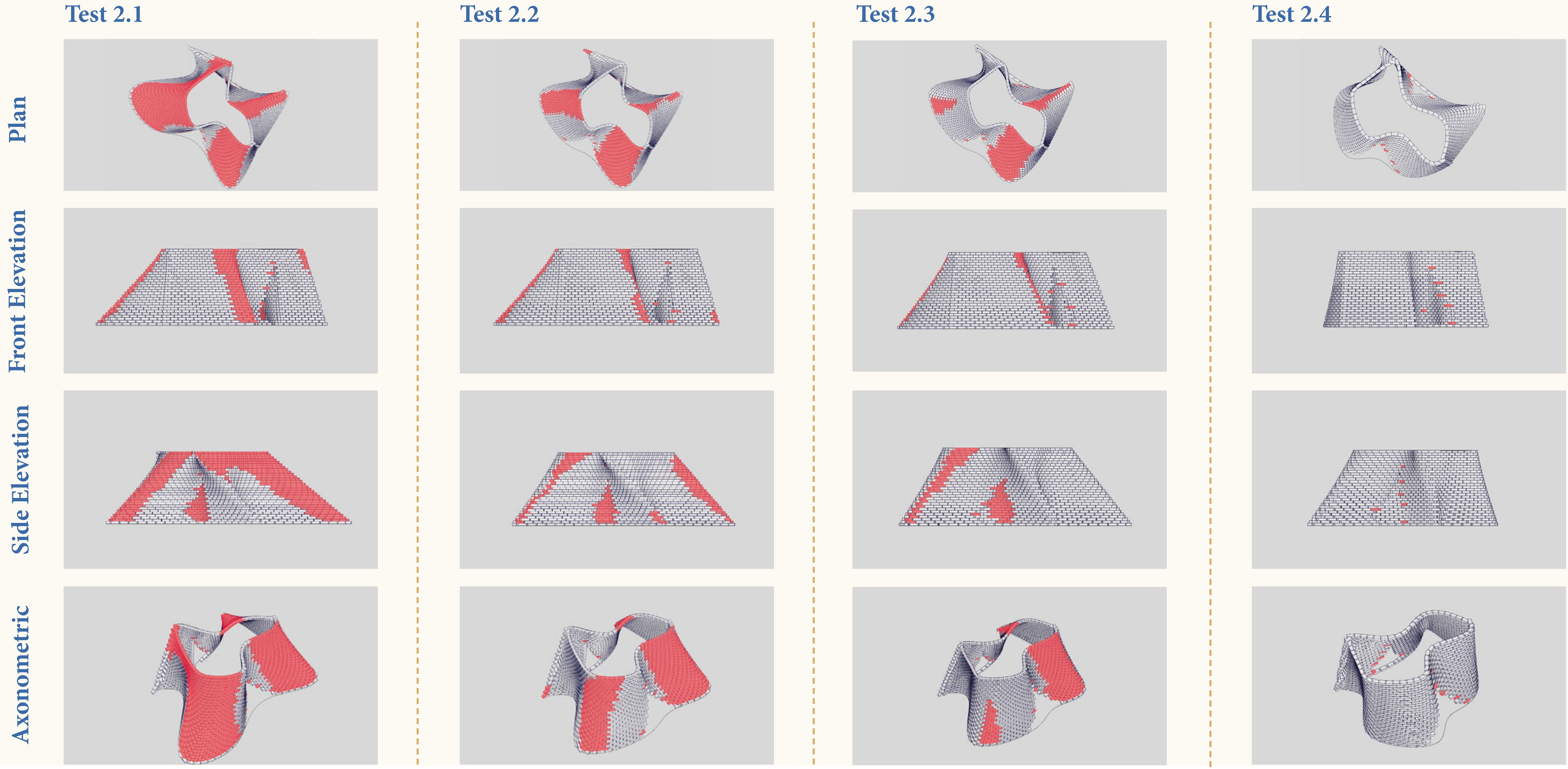
Brick Test 2

Brick Test 2 starts to explore the limits of the tool, implementing curves that differentiate from one-another, featuring more exaggerated geometries.

The idea here was to test a pavilion that produces a more dynamic shape than Brick Test 1. The threshold was carried all the way through the pavilion so it created an opposite opening through to the other side.

Another factor explored was how intricate the generated geometry could become in a relatively small space and still be 'buildable'.

The 'unbuildable' areas in these tests appeared in less predictable areas of the geometry here, however the 'buildable' end result still strongly resembled the original form.

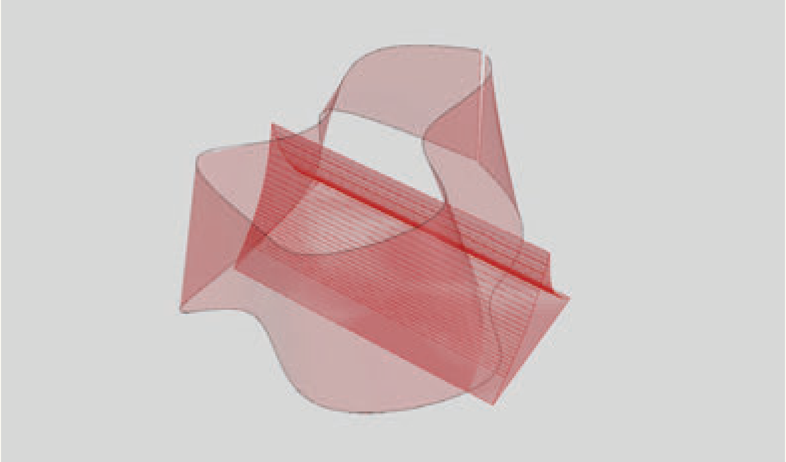


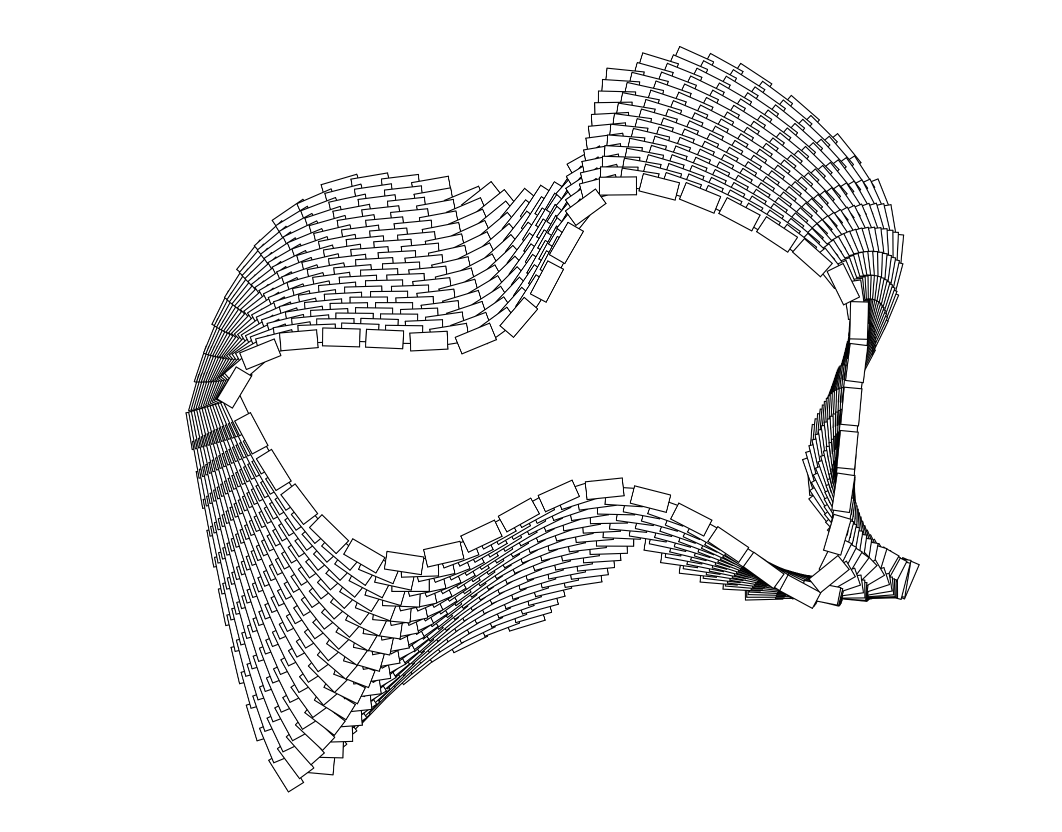
Brick Test 2
Outputs

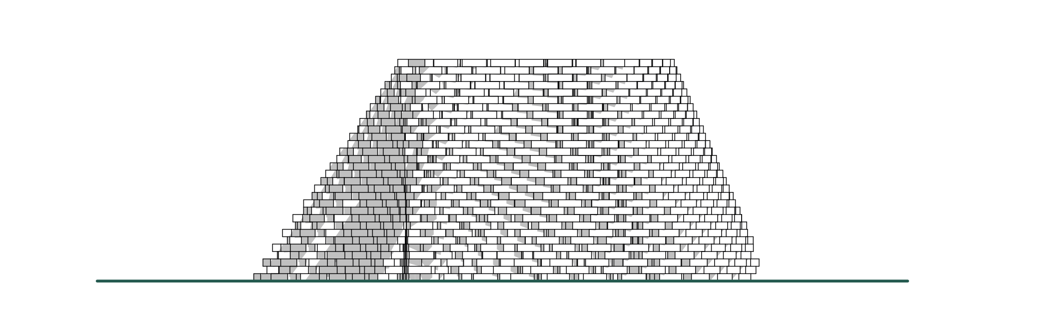
Brick finalisation of brick test 2 provided a much more interesting form than Brick Test 1.

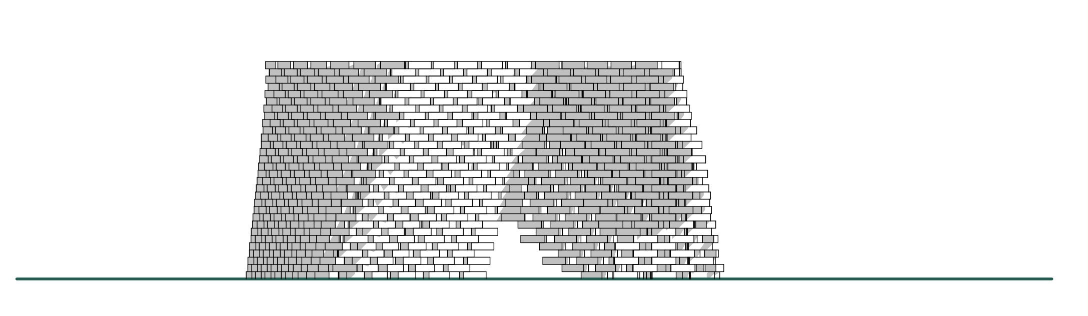
When 3D printed, it was evident where the limits of the tool began to show, especially where the generating surface became more angular and less curved.

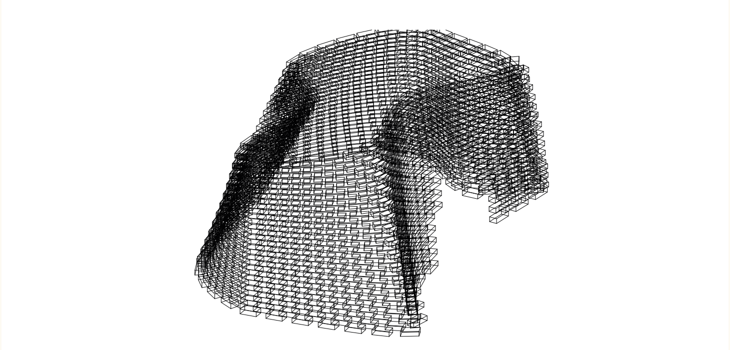
This iteration is more playful however, providing more to explore as a user and more dynamic ways to experience light in the pavilion.

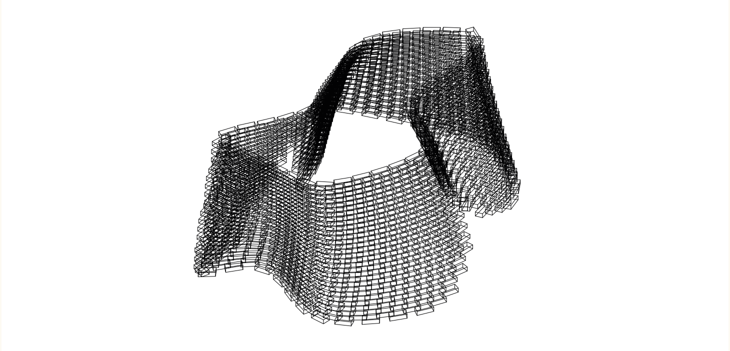

Plain and threshold - base geometry

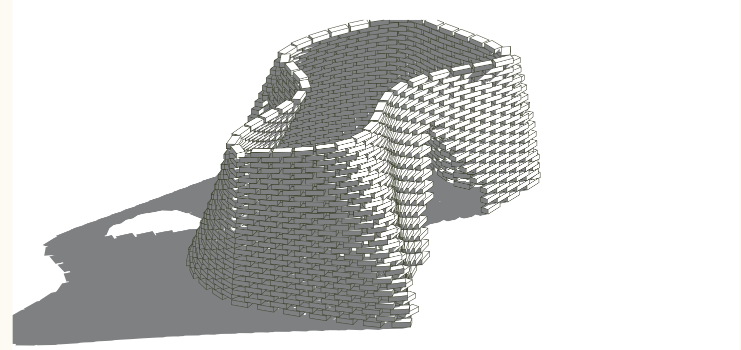

Line Drawing - Plan

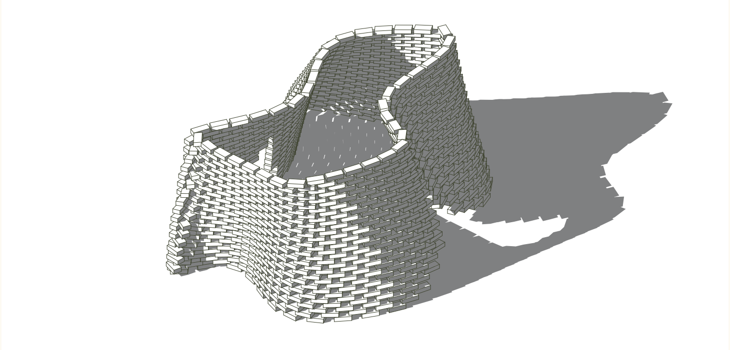

Line Drawing - Front Elevation

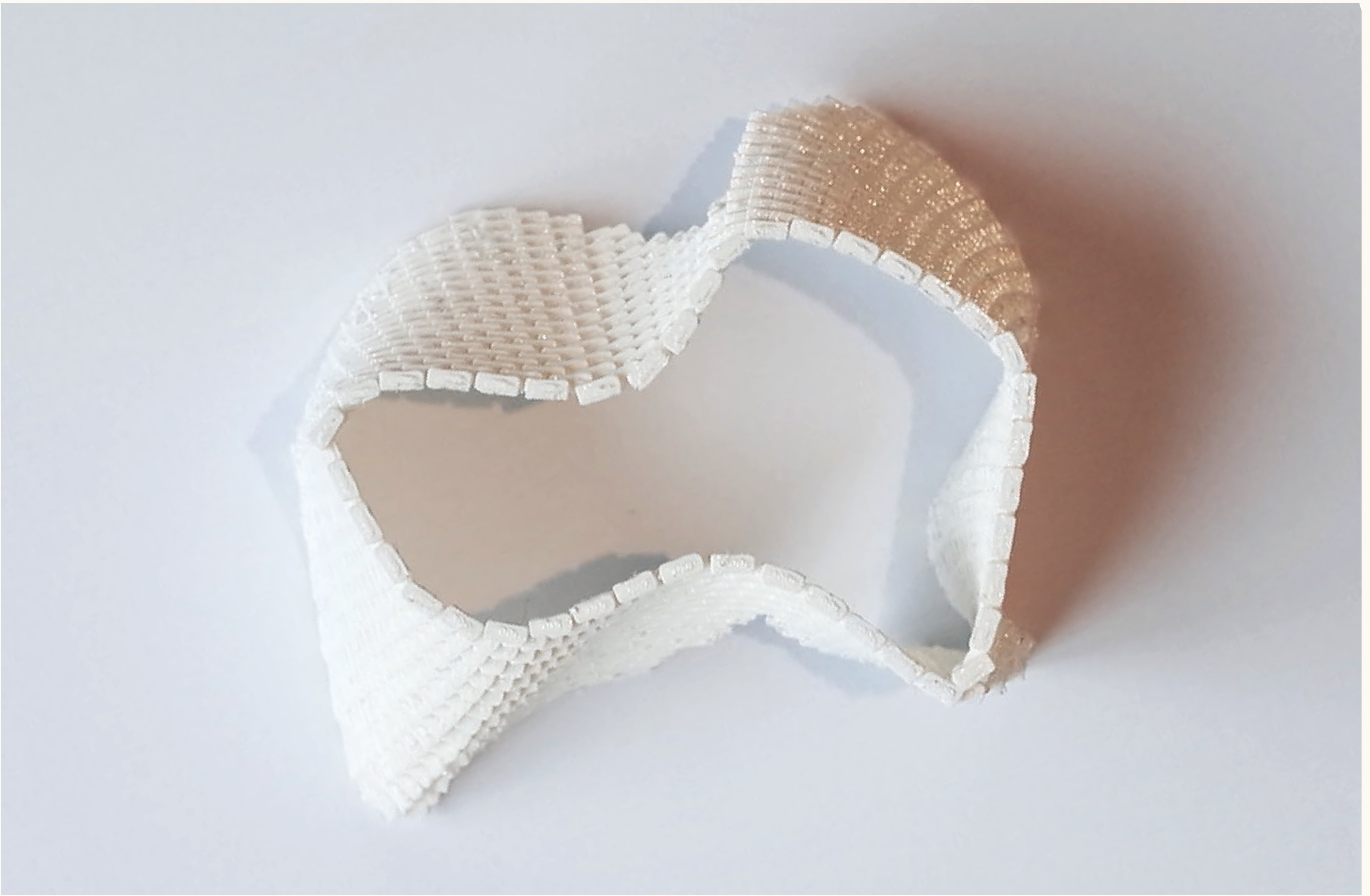

Line Drawing - Side Elevation


Wire-frame Axonometric 1


Wire-frame Axonometric 2


Axonometric 1


Axonometric 2

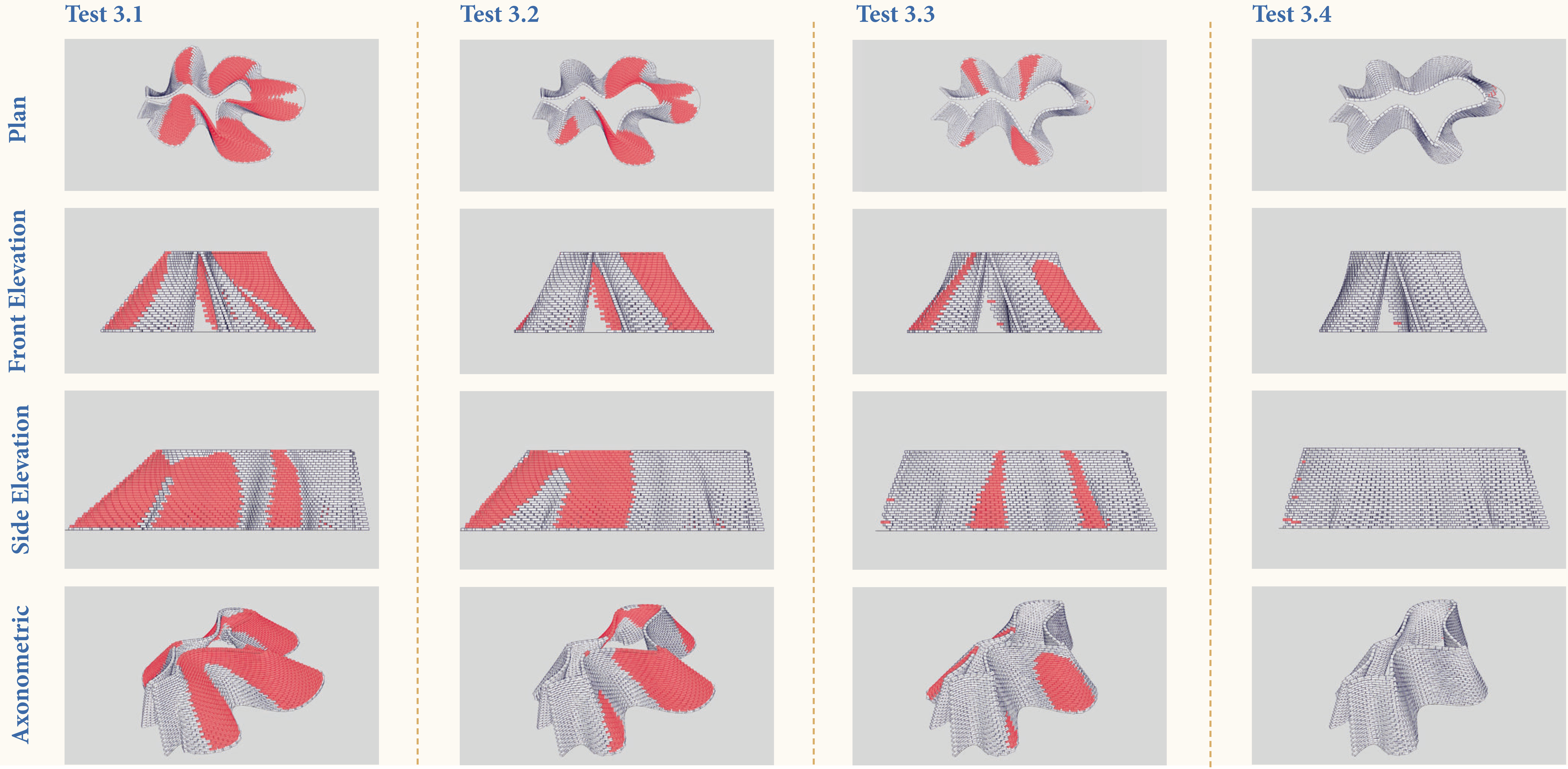


Brick Test 1 -
3D Printed Model

Brick Test 3

With the final Brick Test, I wanted to create a space that the user could playfully explore. Taking aspects from Brick Test 2, while still respecting the scale of a pavilion typology, Brick Test 3 began by trying to create 'pockets' of space for users to discover and enjoy.

As expected, the initial resultant brick forms were largely unbuildable. Through careful alterations, a brick structure was found that could be built while satisfying its' aim.

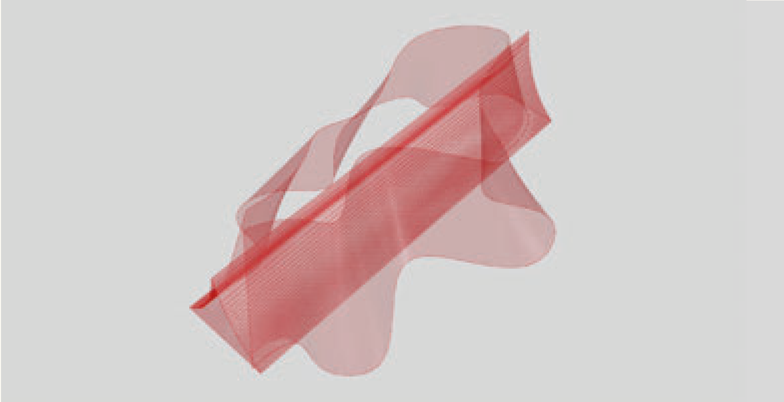


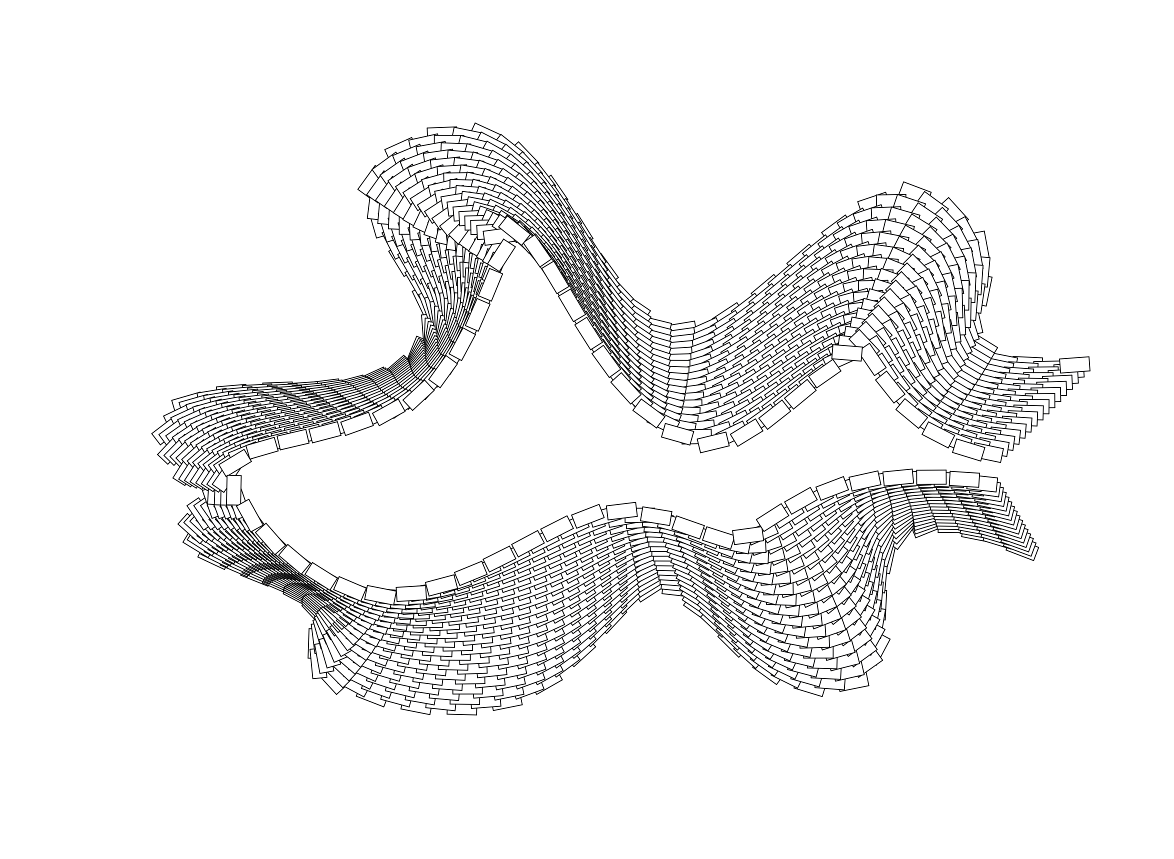
Brick Test 3
Outputs

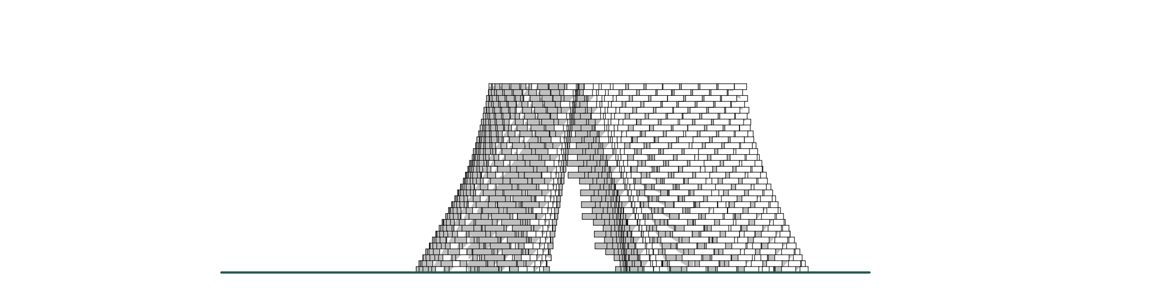
As shown in the 'Plain and threshold' geometry below, this brick pavilion has a threshold at each end with pockets of structure in between.

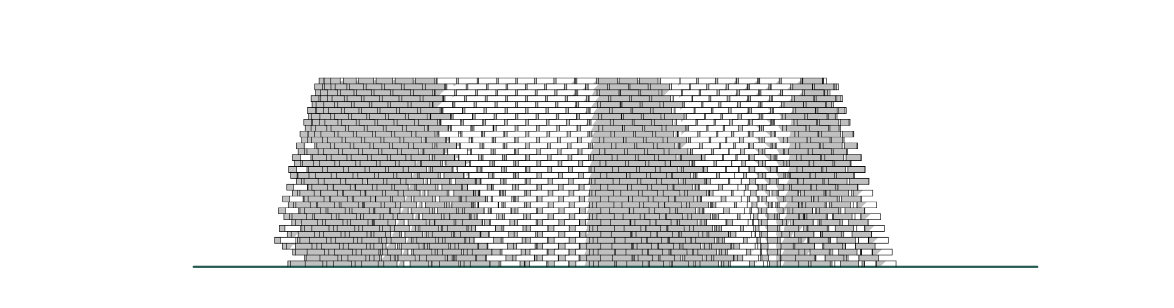
This iteration is certainly the most playful, making the most use of the limited space available.

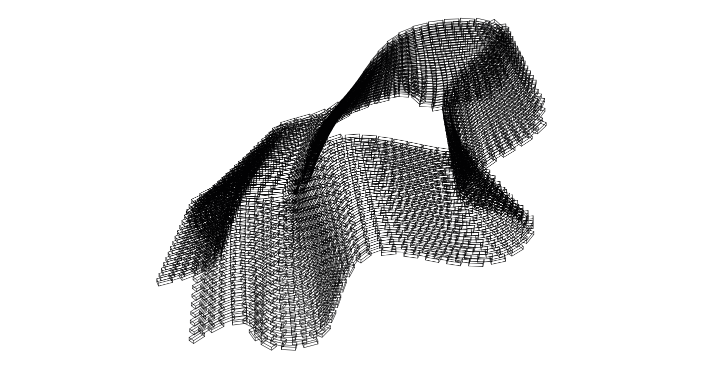
The 3D printed fabrication of this test strengthens its playfulness by exhibiting its handling and distribution of natural light in space.

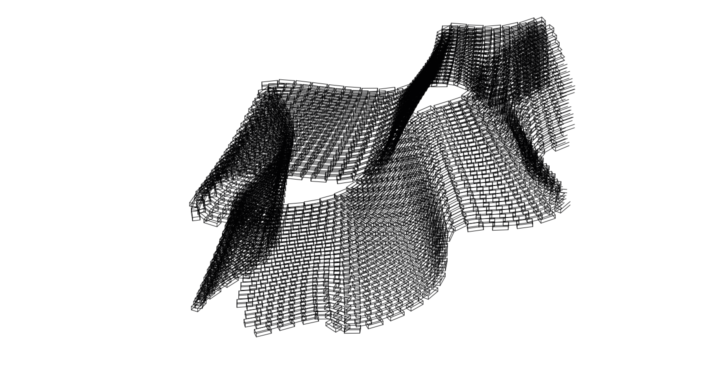

Plain and threshold - base geometry

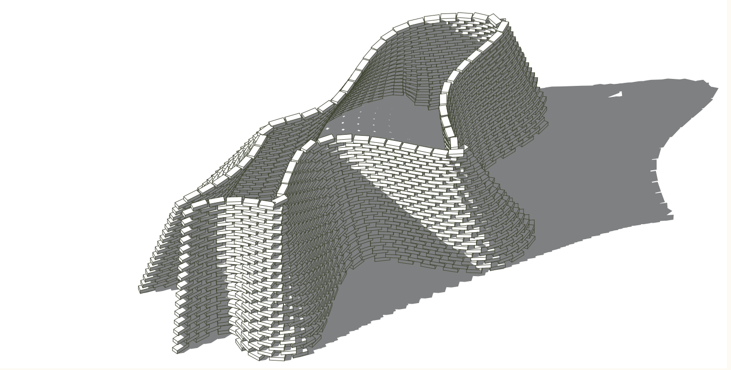

Line Drawing - Plan

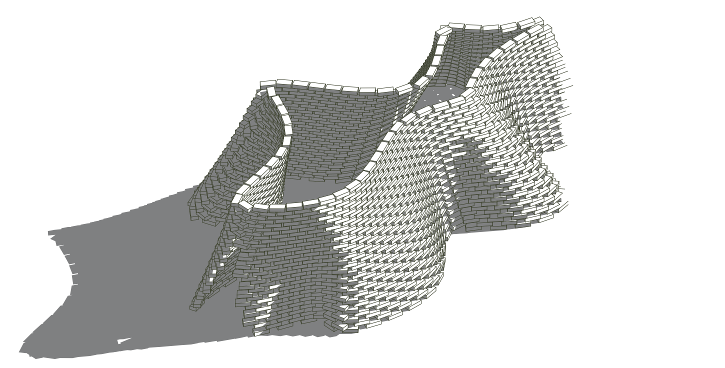

Line Drawing - Front Elevation

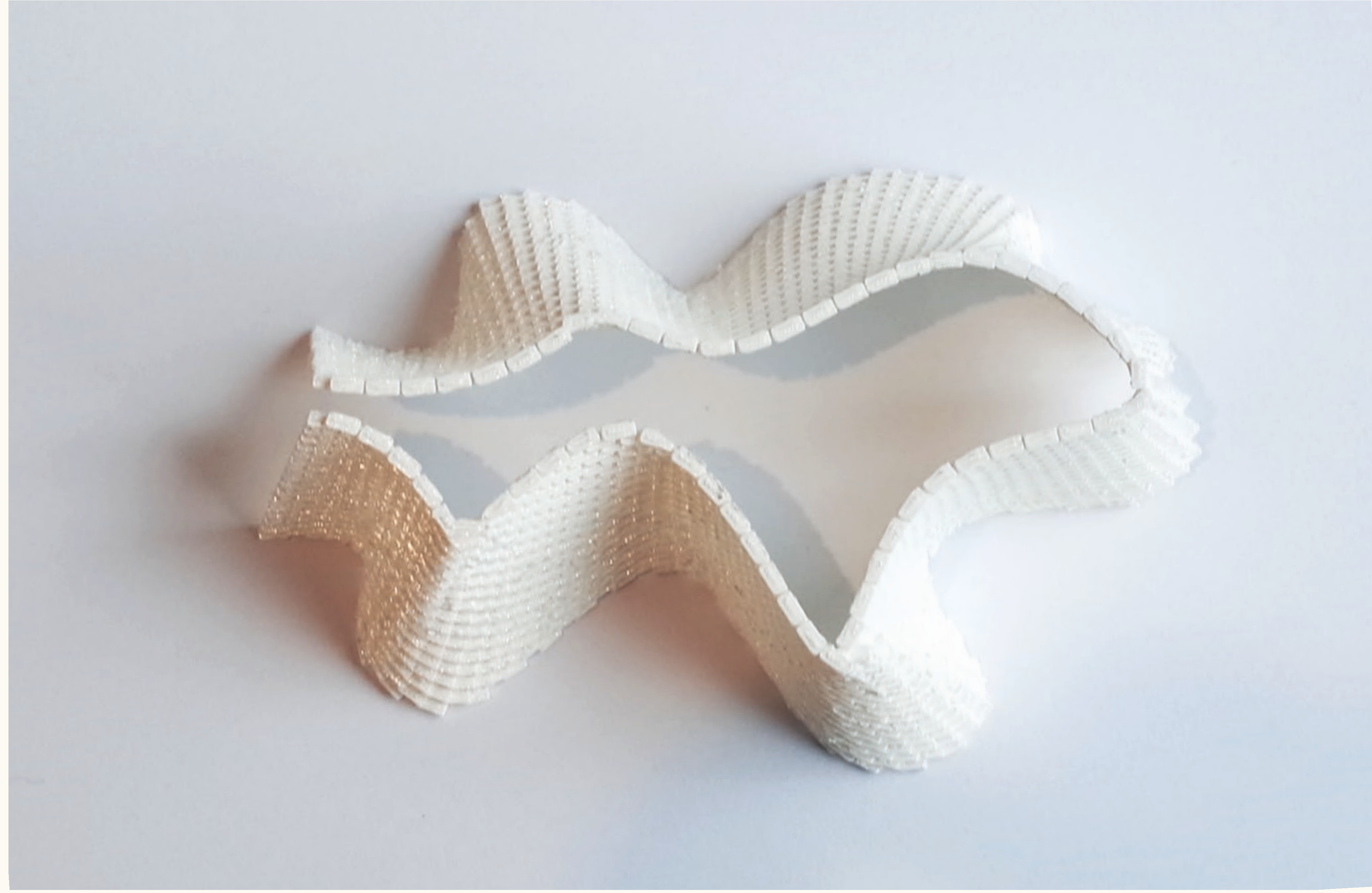

Line Drawing - Side Elevation


Wire-frame Axonometric 1

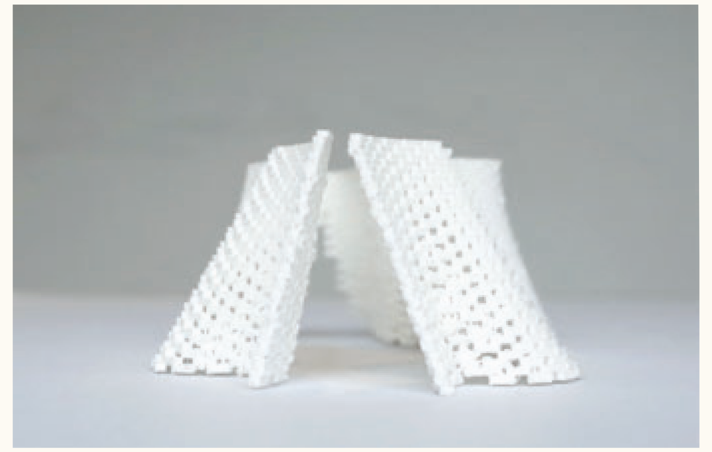

Wire-frame Axonometric 2

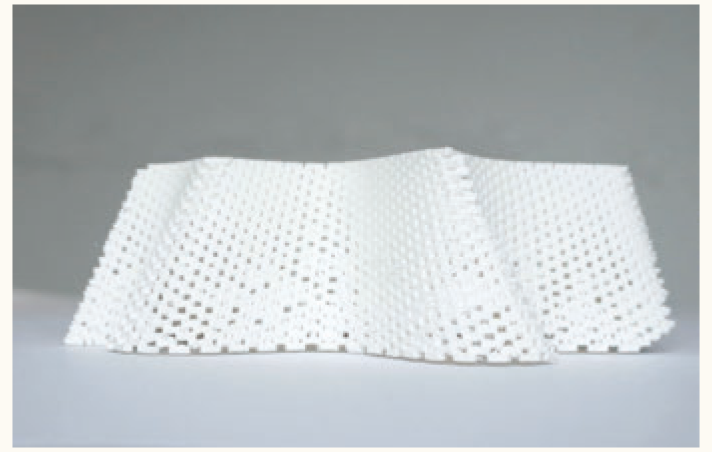

Axonometric 1

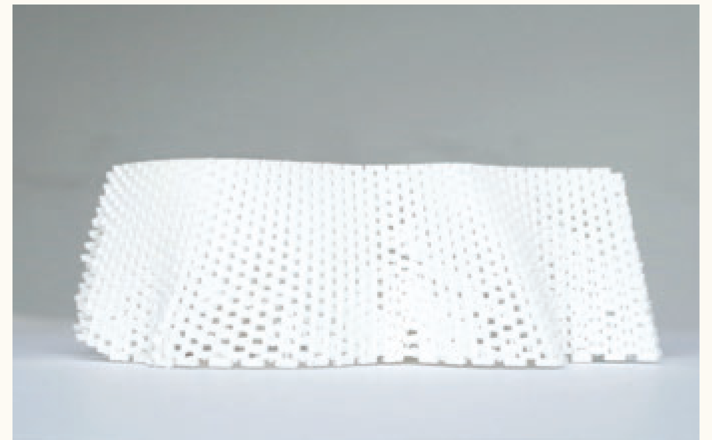

Axonometric 2



Brick Test 1 - 3D Printed Model







Conclusion

This project produced an artefact that responded to the brief in the following ways:

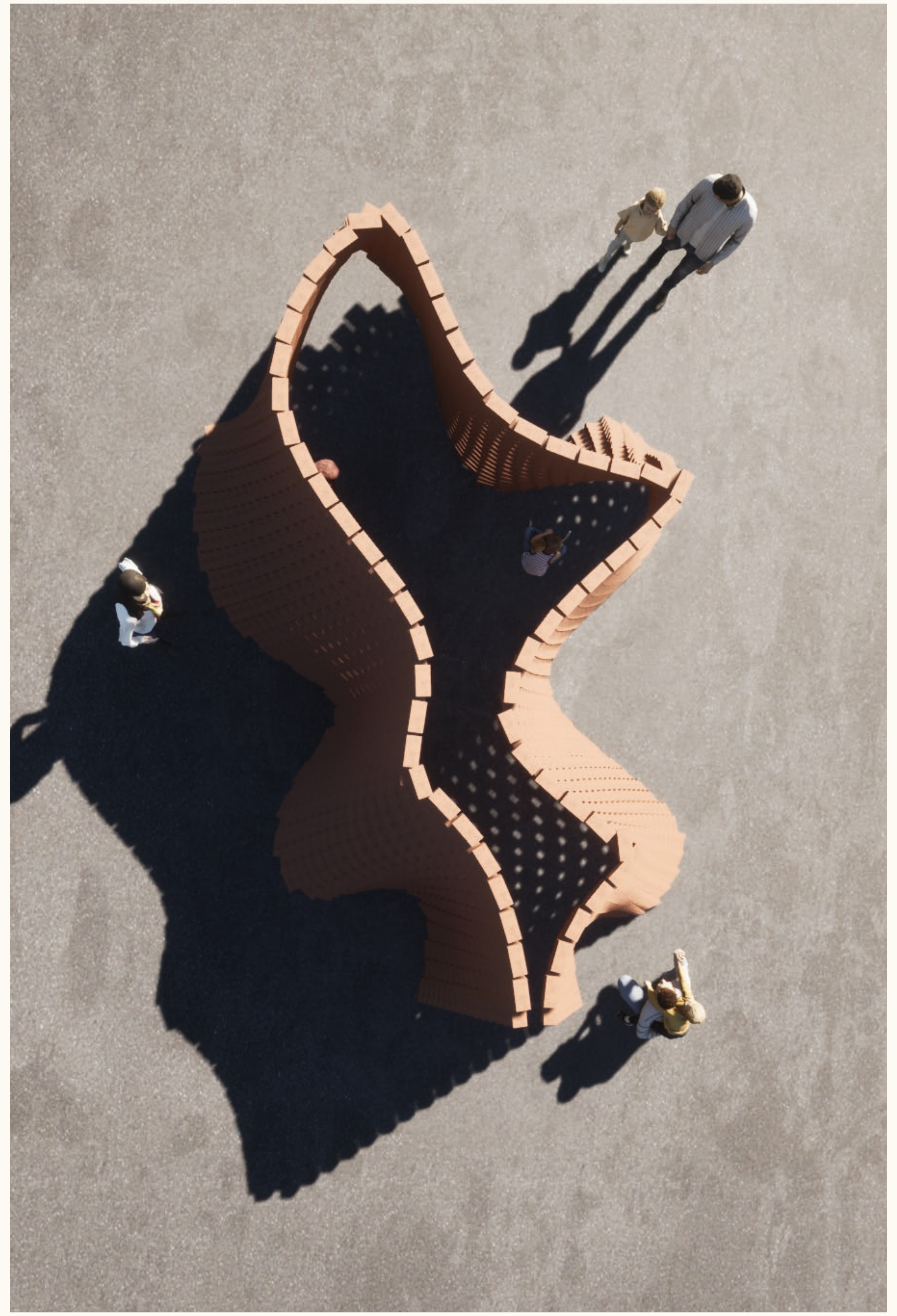
By interpreting a challenging but integral theoretical work in computational architecture, I have produced a pavilion that is geometrically playful and user-centred.

Within this project, it could be considered that the Artefact is the logic itself - the Grasshopper brick tool.

The logic of A Pattern Language gave way to much of the founding logic of computational computing and design, and was itself a design guide of sorts.

The brick tool made for this project sought to reference this factor, producing something people can use to inspire, create and build with.

Brick Test 3 in context




2023/ 3/29 21:32


2023/ 3/29 21:38


2023/ 3/29 21:32

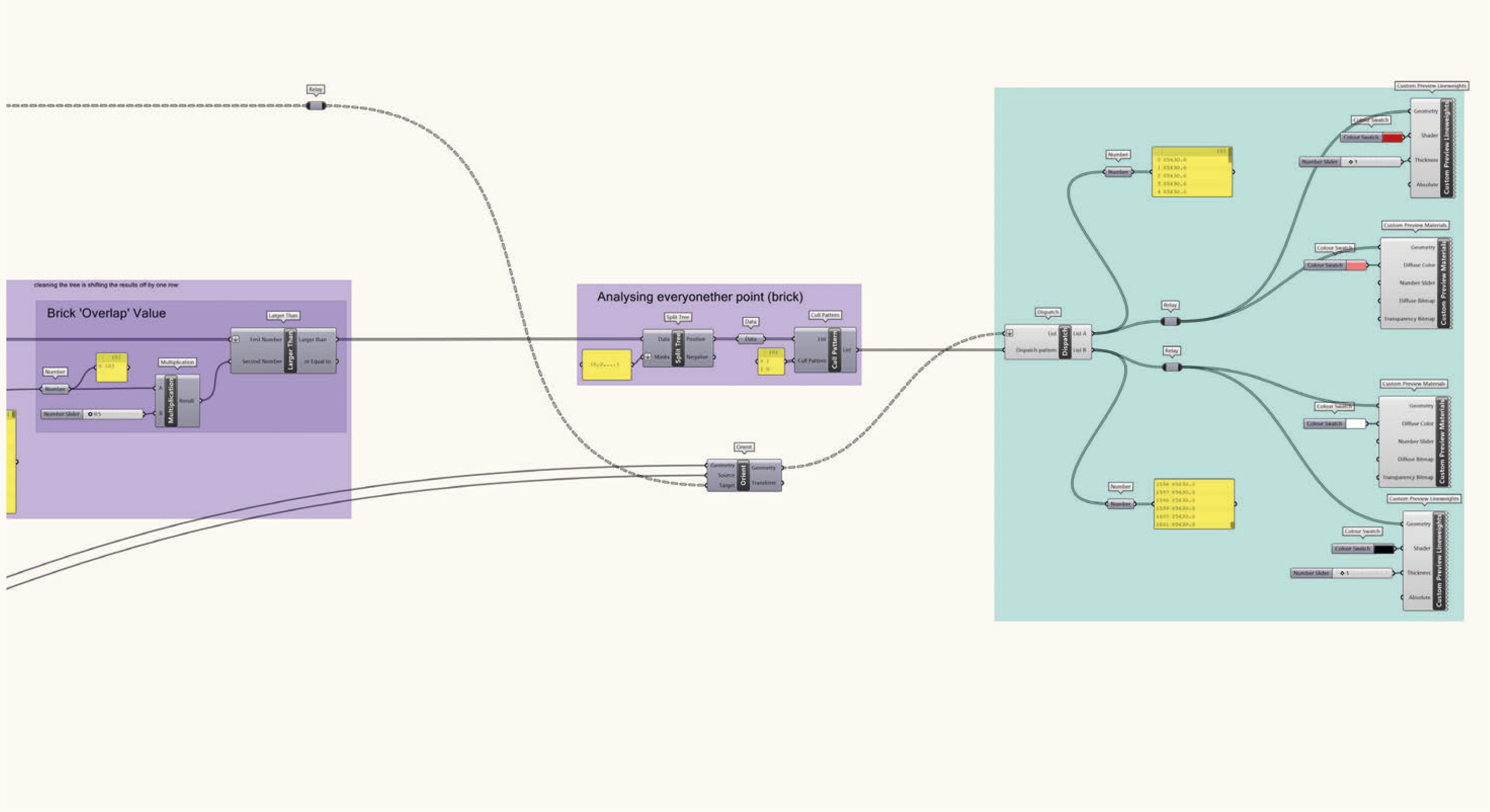
Appendix

Grasshopper Components 1

Grasshopper Components 2

Grasshopper Components: 3

Grasshopper Components: 4



Grasshopper Components: Complete