INTERACTIVE FACADE

by

Tella Irani Shemirani

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs
in partial fulfillment of the requirements for the degree of

Master
in
Architecture

Carleton University
Ottawa, Ontario

Abstract

Today, new digital technologies can play an important role in accelerating information processing, leading to new communication technologies. At the same time, digital technology is rapidly increasing the dynamic interaction between building and users. The overlap of digital technology with the field of architecture has resulted in the development of Interactive Architecture, architecture that evolve beyond parametric and generative design, incorporating new technological advances, absorbing, processing and exchanging data in real time, proposing configuration adapted to the ever-changing needs of its users. These new technologies are changing architecture from a static art toward a dynamic environment. For instance, new computing technologies are developing an intelligent architecture, allowing buildings to sense and analyze their surroundings, and make decisions to evolve an appropriate response to environmental stimuli through adaptation. Interactive Architecture designs space capable of communicating and interacting with the changing.

This thesis is focused on investigating the question of: How can a space develop an understanding of its users (through their body gestures) and respond accordingly? Based on the thesis question this paper investigates the process of creating interactive spaces and explores architectural form generation through behaviour-based AI approaches. The thesis concludes with a proposal for a dynamic responsive architectural system, a through behaviour-based AI approaches.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Professor Johan Voordouw and my co-supervisor Dr. Mojtaba Ahmadi for the guidance, encouragement, and advice they have provided throughout this work. I have been extremely lucky to have such supervisors who cared so much about my work and responded to my questions and queries so promptly Without their patience, support, and immense knowledge, I would never have been able to finish my thesis. I would also like to thank the Ontario Ministry of Training, and all the Faculty of Azrieli School of Architecture and Urbanism Carleton University, specially Professor Jill Stoner, the director of Azrieli School of Architecture and Urbanism and Dr. Federica Goffi, the associate director of graduate programs for their research support.

Last but not least, I am heartily thankful to my father, my mother, my husband and my son for all their love, sacrifice, patience, and understanding all along.

CONTENTS

CONTENTS

CONTENTS

CONTENTS

CONTENTS

List of Figures

List of Figures

INTERACTIVE

FACADE

TELLA IRANI SHEMIRANI

PROF. J.V O O R D O U W

ARCN 5909 2017.

100952209

Introduction

Keywords: Intelligent Spaces, Interactive Architecture, Agent, Environment.

My initial interest in this topic came from participating in three online courses from MIT: Artificial Intelligence (AI), Robotics and Machine Learning. These courses have been augmented through literary sources and a final design project in collaboration with Carleton University department of Computer Systems Engineering. This research explores the potential of sensing computation network and interactive technologies in the behaviour and interaction of a space and its inhabitant to build a new space of performance and interaction, a self-initiated project, where geometry, robotic, mechanisms and interaction collide. Architecture is a technology intensive discipline, where technologies used both in the design process and its products, new changes brought by the computer such as computer-aided design in design process, constructed automation, virtual places and intelligent buildings. Technologies that allow spaces to sense, think, and respond to their occupants and surrounding environment.[1] The term interactivity, in general, is about human experience. In this thesis, the term interactivity relates to human experience with an intelligent responsive building system. Interactive Architecture involves an input (from a person) that is given will change an architectural element and, as a result, people will have interaction with a part of the building. In this experience, the architectural form is no longer stable, it is ready to accept change. This is not an architectural product, but a process-based architecture. The form, therefore, is defined by its users' dynamic behaviour, by changing demands, and by the changing external and internal conditions. Through Artificial

Intelligence and Machine Learning this Interactive Architecture causes an intelligent response. This thesis aims to study how architecture might build a reciprocal, beneficial relationship of people with architecture and its internal and surrounding environment.

Scope

The scope of this thesis aims to understand the broader context of Interactive Architecture. The final design project constructs one module from the future facade system to show the mechatronics (mechanical & electronic) integration. The module is a prototype that can be improved and integrated in future intelligent system. While the original intent was to construct a more complex system, as the thesis developed questions of AI, Robotics and Human Computer Interaction (HCI) were reduced. This project, however, is a starting point in which these issues might be further developed as an intelligent system in later research.

My contribution in the design and implication:

- Realizing the multidisciplinary interactive design
- Design, analyze, and propose the multi mechatronic system.
- Parameterizing and simulating the concept of the design with Grasshopper to facilitate the design.
- Constructing two patterns by arranging frequent use of a modern geometry (a star) in a Islamic pattern

Design Process

COLLECTING INFORMATION

Intelligent Interactive Spaces

> "Intelligent Spaces are rooms or areas that are equipped with sensors, which enable the spaces to perceive and understand what is happening in them." [2]

> "Such intelligent environments are able to watch what is happening in them, build a model of them, communicate with their inhabitants and act based on decisions they make." [3]

Generating architecture through writing algorithms and software have opened the possibility of defining new interactive mechanisms in architecture. These algorithms include, live algorithms: computer algorithms that can help solve common decisions and find the best answer from many possibilities. An example of a live algorithm is a genetic algorithm, such as those used in the algorithm editor Grasshopper, a plug-in for Rhinoceros computer modeling software, to generate a live stream of data, simulating and evaluating in software, to change the design behavior during runtime.[4] Using the knowledge of Informatics and Artificial Intelligence (AI), as an interdisciplinary approach, allows us to create an intelligent space with embedded computation and communication technologies. Having intelligent spaces, inhabitants can being to interact with a computing empowered architecture.[5] This thesis considers space as a social product. Such intelligent spaces have the power to transform people's experience, perception, and the way they interact with space and the people around them.[6] Professor Carlo Ratti from MIT describes this a relationship as making an equation and putting together people, space, and technology.[7] Designing intelligent interaction in architecture where architectural design integrates and negotiates the digital to consider subjects from urban social to practical sustainability issues.

Interactive Architecture

Interactive Architecture is a new approach in architecture at the intersection of interaction design, architecture, computer science, and behavioral and social studies. An approach Michael Fox defines as:

> "Interactive architectural environments are built
> upon the convergence of embedded computation
> and a physical counterpart that satisfies
> adaptation within the framework of interaction.
> It encompasses both buildings and environments
> that have been designed to respond, adapt, change,
> and come to life."[8]

The result creates an intermediate space between humans and their environment. A space where there is intelligent transformation and adaptation in architecture. A responsive space that can sense, think, and understand its users / natural environment, absorb information and respond to that information accordingly.

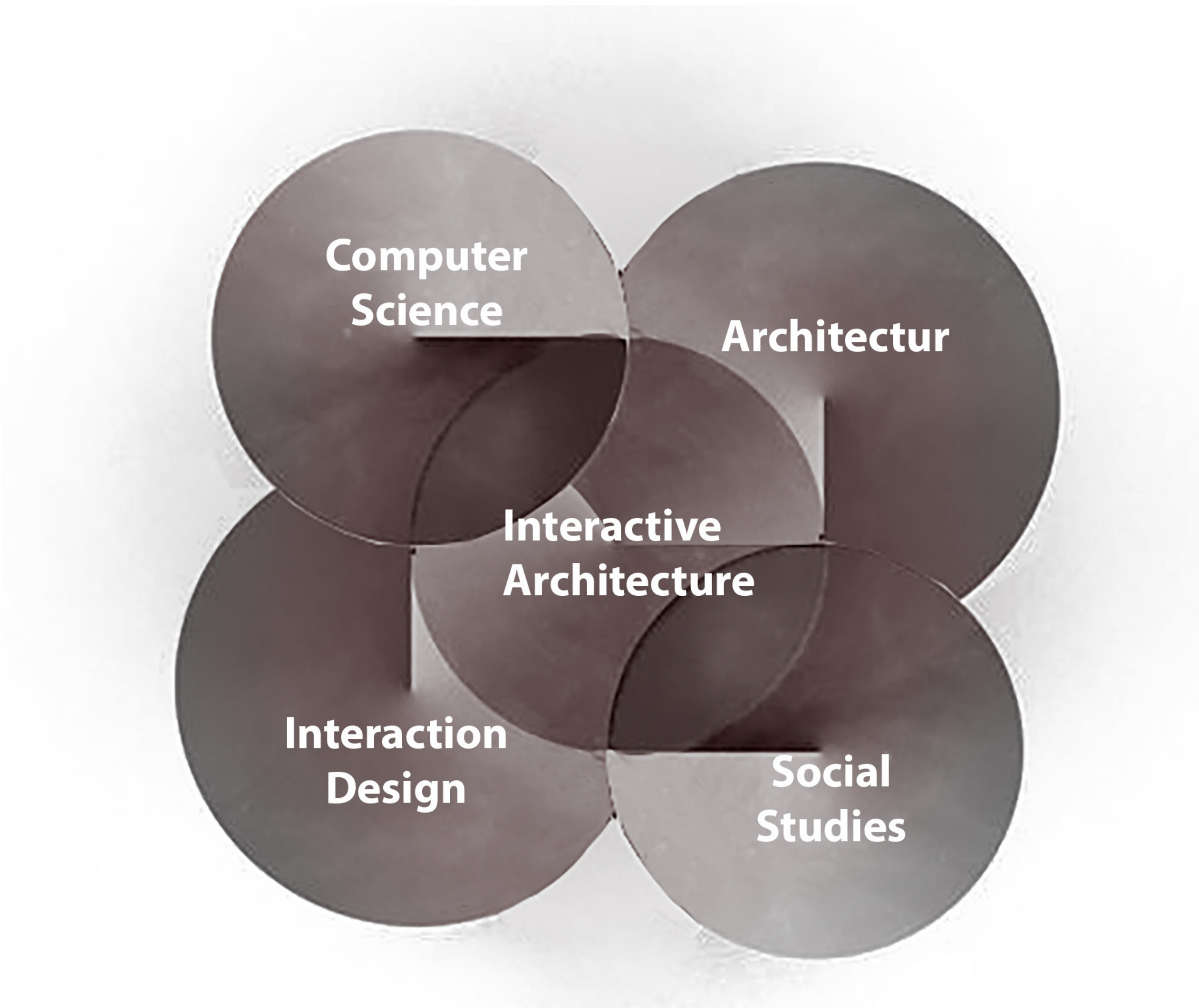


Figure1. Interactive Architecture is seen at an Intersection of:
Architecture, Computer Science, Interaction Design, and Behavioural and Social Studies

Interactive Architecture and Agency

The interactive architectural system is a multi-agent system in which both people and buildings have agency.[9] This means that multiple kind of agents can be identified: inhabitant agents (social agents), intelligent artificial agents, physical agents/Advanced embedded agents/robotic agents), software agents, and virtual agents.[10] Different models of interaction in Interactive Architecture define a user-centered experience and shape the experience of the inhabitant. It has potential to empower inhabitants in participating in the continuous formation of space. The inhabitant-centered approach for design in Interactive Architecture and the use of established user-centered techniques from interaction design provide evidence for the plausibility of Interactive Architecture's claim regarding inhabitant agency. If we consider 'agency' from the social sciences, it is defined as the capacity of a system to make decision on its own and act independently. Therefore, the integrated robotic system becomes the physical agent of the intelligent space - the architectural agent.[11] In addition, a software agent (a computer program) has encoded in its program to decide which, if any, action is appropriate, and finally, the virtual agent is the virtual architectural model that automates interactions between the inhabitant agents and Physical agents (building/building element). The virtual agent is built on a machine learning platform (using Genetic Algorithm). It learns from its experience, detects the human agent's behaviours and offer outputs based on those behaviour. In doing so, it quickly develops the intelligence it needs to understand and offer outputs.[12] From this points of view, four agencies will be identified in my design: inhabitant agents, physical agents, software agents and virtual agents (the architectural model in Grasshopper/Rhino). Moreover, the virtual agent, software agent, and artificial agent are intended to interact with human users.

Methodology

This thesis proposes a methodology for interactive architecture, which includes initial research in the domain of Interaction Experience Design, architecture, computation, and behavioral and social studies.[13] Designing interactive systems is in the field of Human–Computer Interaction (HCI) and offers the potential for Artificial Intelligence in architecture.[14] There are four methods for designing an interactive system: analysis, concept generation, simulation, and assessment.

Design methods

Analysis
Concept generation
Simulation
Assessment

Figure2: Basic Stages of Design Method

There are three Stages of design method for the development of interactive and tangible computing system in a public place: Data Analysis, Data Synthesis, and Implementation.

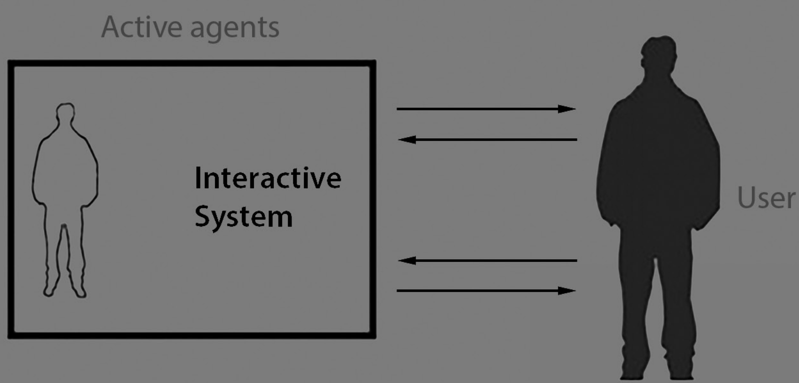
Stage
Analysis
Synthesis
Implementation

Figure2: Basic Stages of Design Method

Facilitating the connection between analysis and synthesis of data and the three concepts behind interactive architecture (place, space, and interaction) is the main goal of the design.[15] For the purpose of this thesis - reacting to environmental changes in real time to change the model behavior during runtime- I need to generate a live stream of data to control the automation and movement of the digital model, simulating the process and providing the required out put data for an Ardouino board (a microcontroller). for this reason, I will use the simulation based approach.

Stage	Attribute	Technique	Concept
Analysis	Cultural Practice	Focus groups Observational techniques	Place
	Morphological	Site analysis: Movement and usage Physical affordances	Space
	Conceptual	Theoretical intent Precedents Model	Interaction
Synthesis	Prototyping	Sketching interaction Storyboarding Description of user scenarios Mapping physical computing Information flow diagrams Computing system diagrams Sections and construction details	Place Space Interaction

Figure4: Attributes and Techniques per Stage



There are four degrees of interactive systems: Passive, Reactive, Autonomous, and Agent systems

CHAPTER 1
SYSTEM DESIGN

System Design

Keywords: Interaction, Interactive System, Environment, Visual Interaction, Passive/Active Input, Data, Data Collection, Degree of Interaction, Agent System.

Interaction

Interaction is the reciprocal influence two or more agents have on each other (in the same interaction) to make a complementary behaviour based on a mutually constructed pattern.[16]

Interactive System

An Interactive system is an intelligent system/robot or Artificial Interactive agent that is often learned through continuous interaction with its environment (its real physical or virtual world, humans and/or other machines).[17] Interactive systems and robots are intelligent systems since they learn through continuous interaction with their environment. They perceive (through their sensors), act (through their effectors) and communicate with their environment, using different machine learning frameworks and techniques, such as Neural Networks, or Genetic Algorithms, for learning.[18]

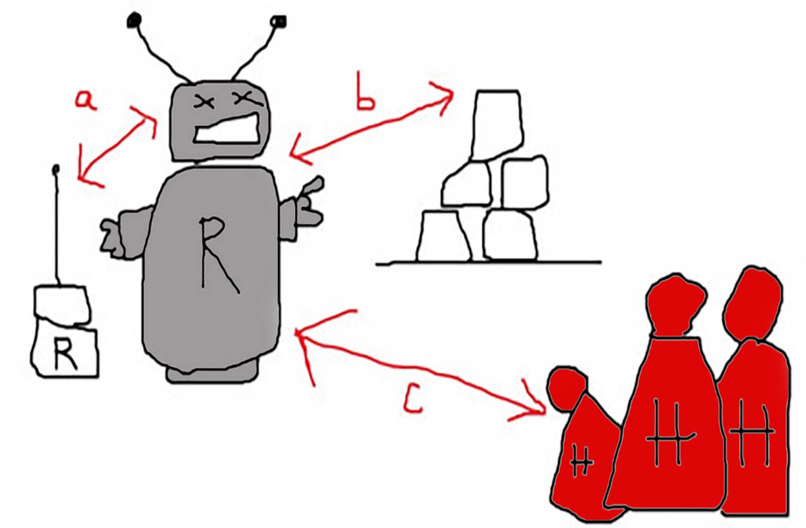


Figure 5: An interactive system, interacting with other robots (a), the world (b), or with humans (c).

It is a dynamic autonomous system with predefined responses to predefined behaviour scenarios, concerning with observing or engaging in interaction.[19]

> "The communication between the architectural components and the users in such system is a dialogue with new messages being related to previous ones." [20]

The system can actively modify its own behavior and respond to changes according to its user's behaviour. Also, it has the ability to learn and/or build upon previous interactions. There are different interaction methods, in particular, this thesis is interested in a visual interaction, an interaction in which a sight is detecting the actions of participants as in the case of their movement and gesture, identifying the actions of interest and labeling its identifications with a probability of correctness.

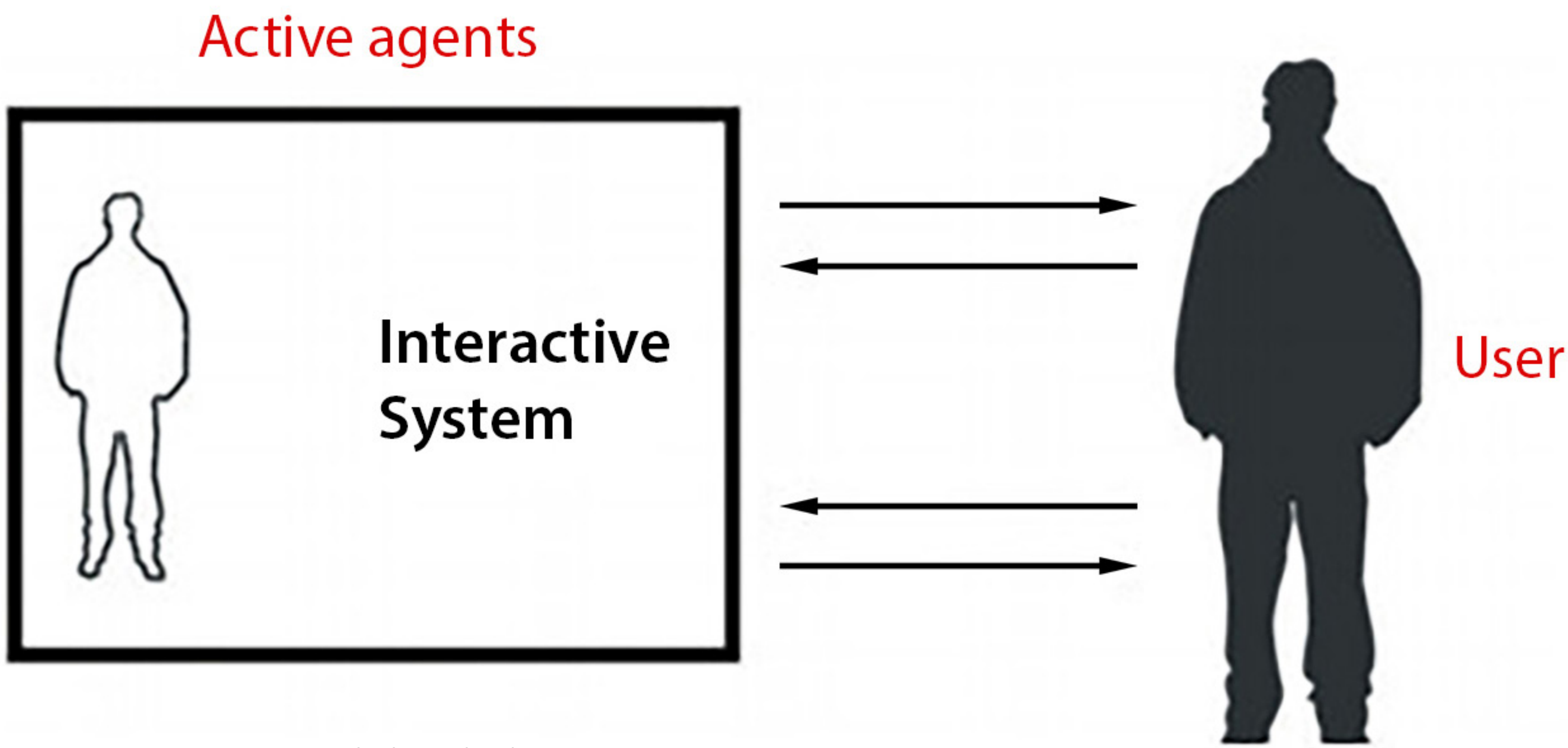


Figure 6: An interactive system is engaged with a user, based on an internal representation of its goals and the user goals.

Passive/Active Input

Two kinds of inputs can define an interactive environment: active and passive. The active users participate in the interaction as a dynamic input, where it is harder to speculate its behaviour in comparison to the behaviour of its environment. While active input occurs when participants directly influence the system (pushing a button, pulling a switch, or something other), passive users indirectly influence the system by some action based on user needs through observation, monitoring, measuring, or other means of detection.[21]

Interactive Systems and Data Collection

When we make an algorithm we need input data for the design to work. The data is not already there, it comes from the interaction in real time. The system senses the participants' behaviour and collects required data through the use of sensors. As a result, the system constantly collects passive input data as the environmental data and to the user the system seems completely environmental.

The active input data then as a part of the system are used as the system evaluation tools to compare the current environment with the desired environment. In relation to data, for the purpose of this thesis, trends, duration and movement are considered the fundamental information in explaining an individual's habit through time.

Degree of Interaction

There are four degrees of interactive systems: passive, reactive, autonomous, and agent systems.[22] In passive and reactive systems the responses of the system to its environment are like a reflex action. Whereas, in autonomous and agent systems, the environmental inputs goes through several constant evaluations against the required outputs to shift the design goal from designing the behavior to designing the process that leads to the behaviour. In this case, the desired output forms part of the input, therefore, the system can evaluate and perform the optimum behaviourin order to reach the required outputs. “An autonomous system is a reactive system with a mind of its own. [23] Agent systems also are autonomous systems capable of communicating directly with other systems and anticipating required changes by acting before an input from the environment is received.[24] Both autonomous systems and agent systems are intelligent, therefore only autonomous systems and agent systems can establish interactivity.25 Since the final design has to be able to communicating directly with other systems and anticipating required changes by acting before an input from the environment is received, the focus of this thesis is on agent systems.

Agent System

Agent systems are responsive systems that influence their environment by evaluating their environmental inputs according to the system goals, reasoning mechanisms, and allowed means. They communicate directly with other systems, and anticipate required changes by acting before receiving an input from the environment.

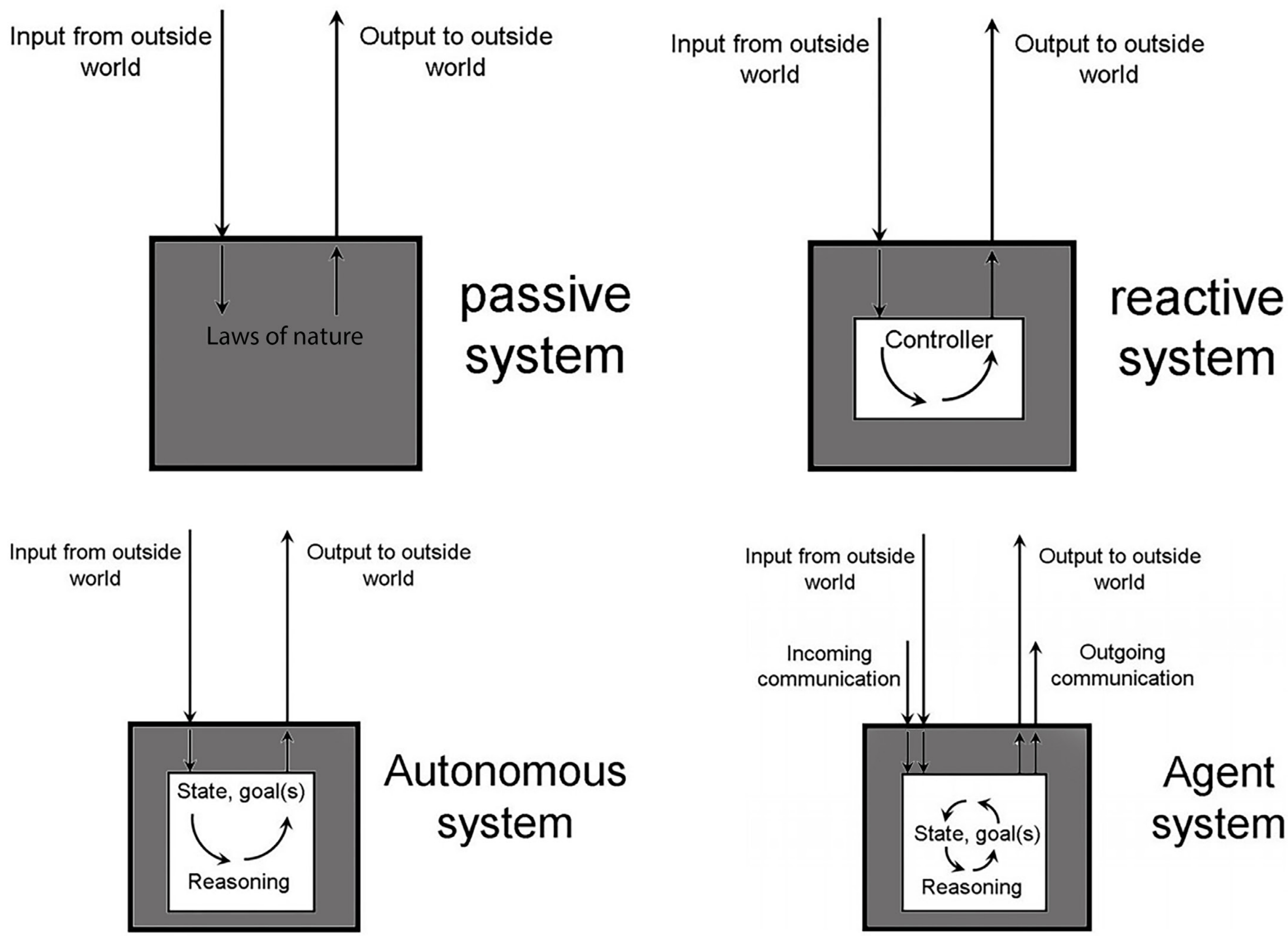


Figure 7: Four Degrees of Interactive Systems

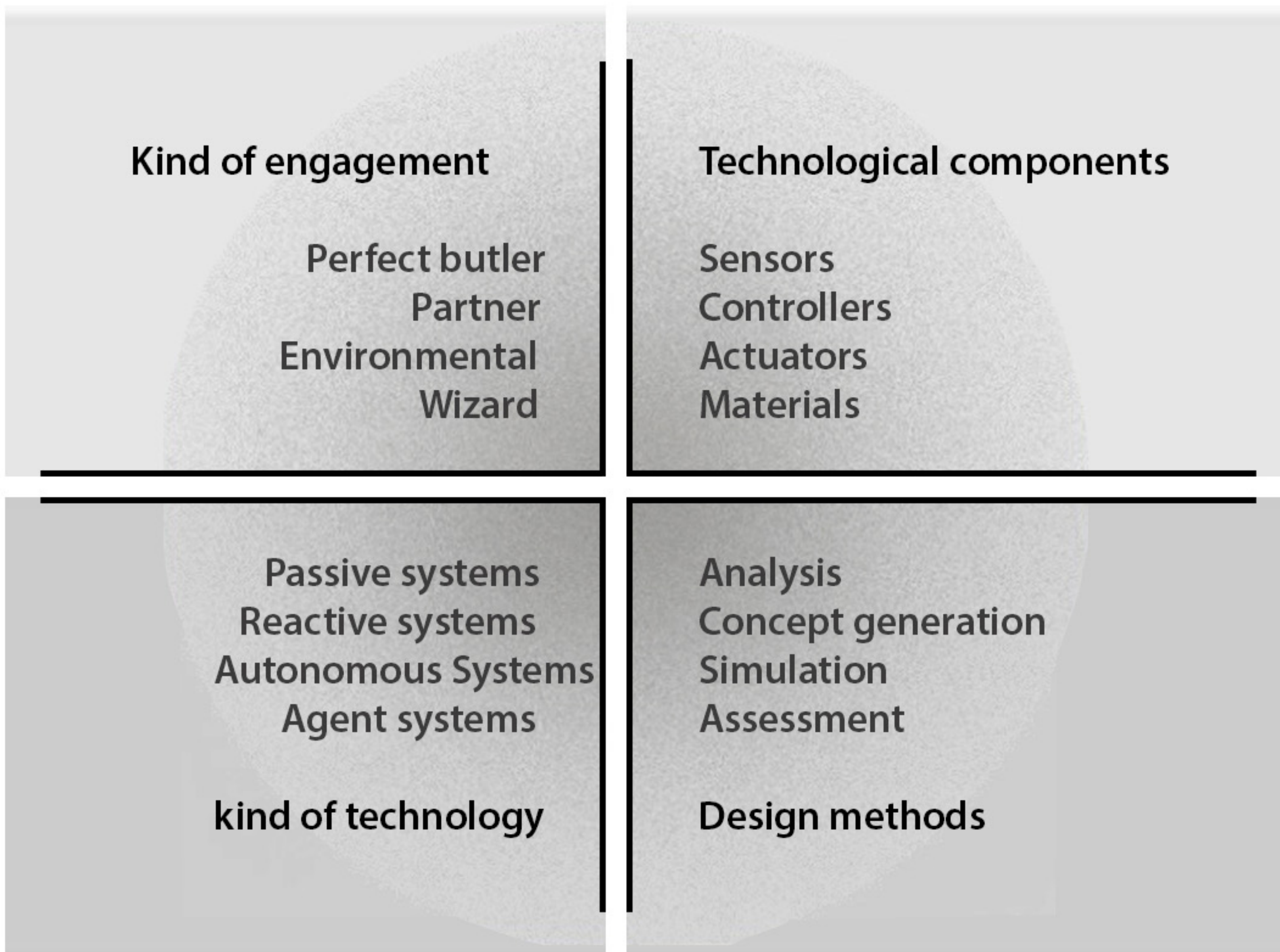
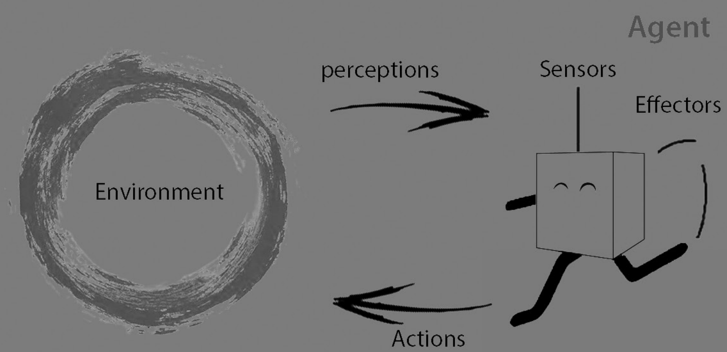


Figure 8: Consideration Areas for the Design of Interactive Architecture



An Interactive system is an intelligent system/Robot or artificial interactive agent that is often learned through continuous interaction with its environment

CHAPTER 2 INTERACTIVE SYSTEM DESIGN

Interactive System Design

Keywords: Human Computer Interaction, Agent, Task, Environment, Artificial Intelligence (AI), Machine Learning, Algorithm, Robotics, Neural Networks, Genetic Algorithms.

Human computer interaction (HCI)

> "HCI is a methodological basis in a quest to advance the locational accuracy of sensing systems. " [26]

HCI stands for Human Computer Interaction. In this interaction the Human is the one using the system, the Computer is the machine or network of machines that run the system, and the Interaction is the interface that connects the human with the computer, facilitating user actions and interactions. In relation to design it is a young discipline. It is about making communication and interaction between a user and a computer system, developing systems that are easy, effective, and enjoyable to use. It refers, in particular, to the use of data as input/output and re-quired devices and supporting software.

It is about design implementation and evaluation of the interface and how people interact with the user interface. In this context, the required database contains information about the interface and HCI research provides a range of methods for evaluating this data.[27]

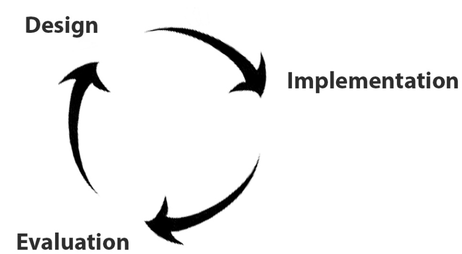


Figure 9: The Process of system design in HCI

HCI is used in all disciplines where there is a possibility for a computer installation, for instance, an application design in computer science for developing interactive products in industrial design. Its use in architecture is recent and suggests a new relation between a space and its users and a user's navigation in the space.

Navigation is an ongoing concern in both Human Computer Interaction and architecture. Using HCI and interaction design to improve navigation suggests new possibilities in redesigning architectural spaces with more navigable landscapes,impactful experiences that reward longer interactions with the space.[28]

Interactive System Design - Artificial Intelligence – Machine Learning- Robotics

Intelligence

> "[the] ability to adapt oneself by responding to a complex stimulating environment in the form of a unified behavior."[29]

Artificial Intelligence

Artificial Intelligence (AI) studies how a machine/ robot can exhibit intelligent behaviour by, expressing and responding intelligently. The application of Artificial Intelligence (AI) technology and its techniques, such as machine learning, robotics, computer vision, and computer perception, are new to architecture and used to develop aspects of intelligence and factors leading to experience intelligent (interactive) architecture,[30] Artificial Intelligence, as an approach, is about how intelligence can be imparted to machines to enable them to perceive their environment and take action to succeed at a particular goal; Giving them the ability to do complex reasoning, logical operations and problem solving. As a result, intelligent machines may perform some human associated functions such as: learning and problem solving, simulating human intelligence by transforming the principles of human intelligence to complete simple tasks such as recognition.[31]For the purpose of this thesis, Artificial intelligence is considered as an approach to design a facade for communicating, perceiving, and acting in a two-way interaction with people. To do so AI suggests algorithm called Genetic Algorithms wich is used in Grasshopper.

Interactive System Design as an Artificial Intelligence system

Designing an interactive intelligent system for Interactive architecture based on Artificial Intelligence: An AI system has two main components: an agent and its environment. The environment is where the agent or agents (may contain other agents) act.[32]

1- Agent= in AI an agent is “anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors.”[33] The agent’s goal is called Task.

2- Architecture = the machinery that an agent executes on.

3- Agent Program = an encoded software agent that decide which, if any, action is appropriate, in other words, the implementation of an agent function.

4- Condition-Action Rule = is a rule (defined by algorithm) that maps a state (condition) to an action. The state is updated based on the information about how the world evolves, and how the world is affected by the agent’s actions.

5- Vision System a system enable to understand, interpret, and comprehend visual inputs on the computer.[34]

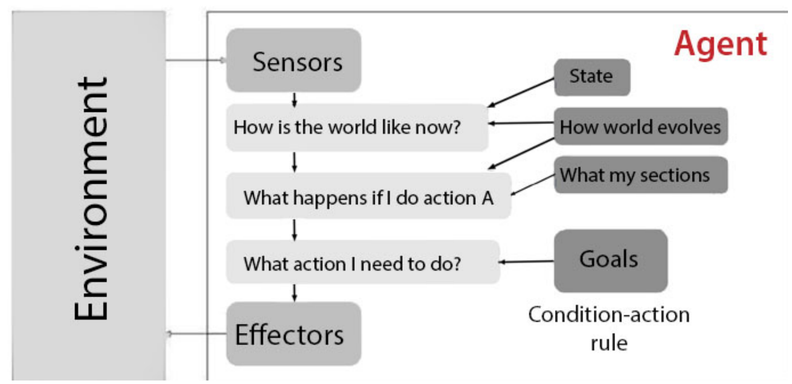


Figure 11: Agent in an Interaction

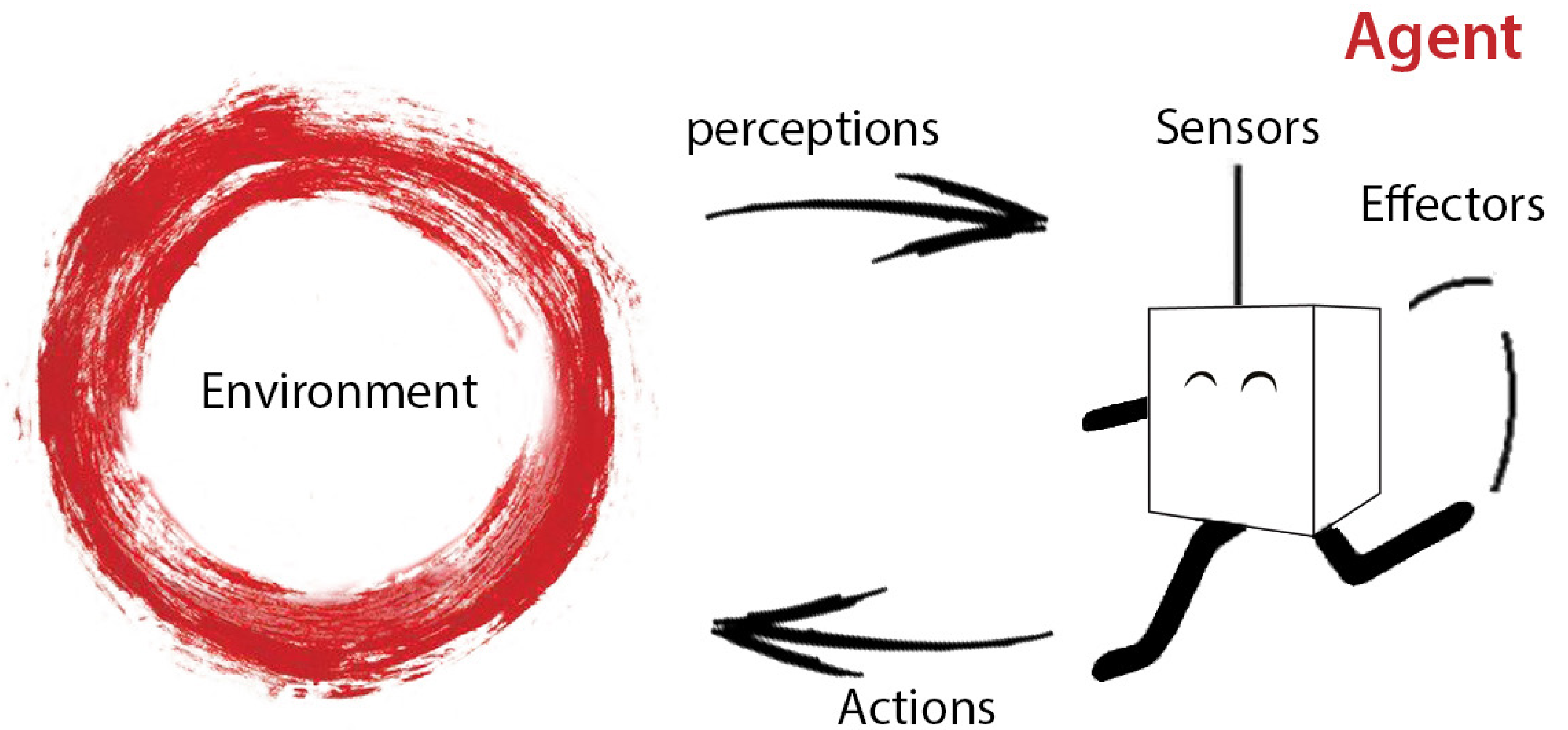


Figure 10: An agent is anything that can perceive its environment through Sensors and act upon that environment through Effectors

Algorithm

A step by step procedure, a sequential list of instructions that solves a particular problem.[35] There are several computational techniques and high-level Artificial Intelligence methods used to understand and solve complex problems. These solutions may include evolutionary algorithms, such as Artificial Neural Network or Genetic Algorithms: the systems that learn from data and are used in machine learning to solve a wide variety of tasks like computer simulation, computer vision, and speech recognition.[36]

Artificial Neural Networks

Artificial Neural Networks (ANN) is a computational technique that "[offers] a promising approach to a number of different recognition and computation tasks….. Networks that can learn to compute a particular classification function by examining a number of examples of the functions." [37]

Genetic Algorithms

A Genetic Algorithm (GA) is different from a Artificial Neural Network. It is a learning system and a search-based optimization technique based on the principles of evolution, natural selection, and genetics. Where ANN is used for recognition, GA is used to find optimize problem solving in machine learning, the process of making something better from a set of inputs, as a set of outputs.[38]

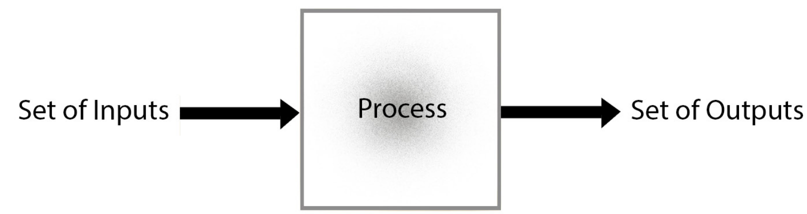


Figure 12

> "Genetic Algorithm is developed primarily for problem-solving and optimization in situations where it is possible to state clearly both the problems and the criteria to be fulfilled for their successful solution."[39]

In the final thesis project Genetic Algorithms where selected over Artificial Neural Network be-cause they use architectural evolutionary systems as an architectural design tool to optimize and evaluate architectural optimizations. One of the scenarios in Genetic Algorithm is the technical interactive selection, which allows users and audience to participate in the evolutionary process of simulation, using genetics as agents to move around the screen and experience the artificial world. In particular in interactive systems it enables the architectural application (such as Grasshopper) to change its behavior during runtime. The following figure shows the single loop of typical Genetic Algorithm, the logic that shows how Genetic Algorithms work (repetitive selection, variation and mutation).

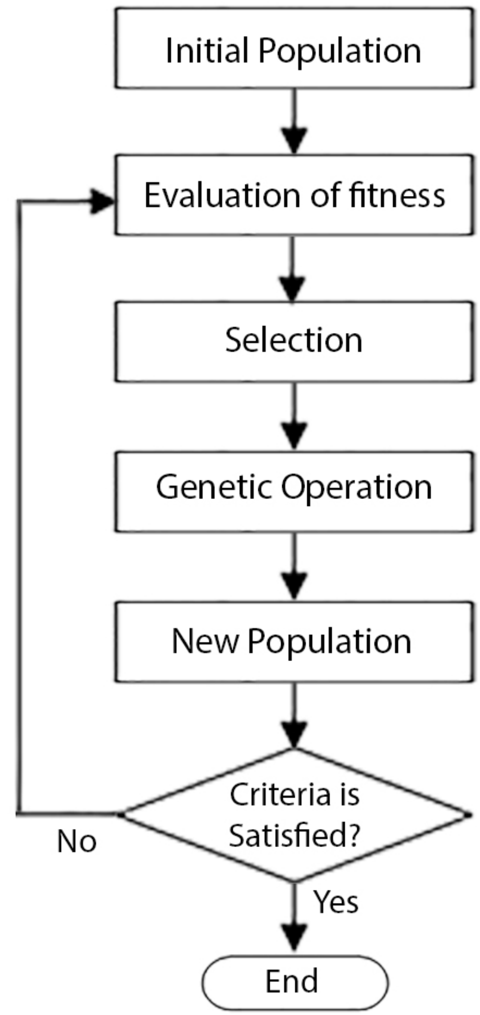


Figure13: the single loop of typical Genetic Algorithm

The loop is a highly parallel, evolutionary, adaptive search procedures. In a single loop of a typical Genetic Algorithm, the system selects and operates the basic elements as the system inputs to produce the system output. If the resulting output does not fit the criteria, the algorithm loop will restart, using the new data. In every loop the data are ranked and generated one by one by the system operator, allowing the system to choose the fittest results and use them as the new database for generating new data to keep the database active. The process makes the architectural evolution and adaptation possible.[40] In the case of interactive design, with this kind of algorithm, numerous adaptive possibilities are searched to find the fitting solution to efficiently behave, defining a Living Design.[41]

> "Living design is not specified or imposed, but emerges from self-created structures that are dynamically responsive to the inhabitants' comfort"[42]

There is a continuous change during the runtime in interactive process, the design responses in real time and adapts its form. If there is a short-time change in the process of adaptation, there is a real-time response in the system, and if the change is a long-term architectural adaptation, it is considered as an evolutionary optimization process.[43] There are several other optimization techniques in Artificial Intelligence. All approaches seek to find a way for optimization and find a fitting result for the process of adaptation of the machine/Robot, but they are different in their inner computing methods. Whatever solution is used, the appropriated approach should be selected based on the nature of the problem, the particular needs of the project, and how efficiently the method can respond to the demands.[44]

Architecture, Inhabitation and User in the Design Proposal

The purpose of my design is to create a double skin facade system, an intelligent facade where multiple users can interact and modify its behaviors. It is a double skin to generate an intelligent response on both interior and exterior. Here the design is capable of acting as a context-sensitive user interface to respond to gestures and developing a design method for building a relationship between architectural form and user behaviour. Putting all the information together, the development of the final design would be based on the applications of Human Computer Interaction. It will use basic levels of Artificial Intelligence, Robotic, and Machine Learning algorithms, through the development of dynamic visualizations, to allow for the recording, control, and simulation of the environment. The design will imply developments of interactive robotic components from conceptualization to use. Physically the project will build a prototype of a robotic environment with robotically driven processes. The end result aims to respond to bodily (hand) gesture and movement.

Electrical Requirements

I need the following electrical components and in my final design:

- Arduino Board: Is a microcontroller board, a small circuit with a microcontroller (a small computer on a designed for embedded applications, working as the brain of the design.) on it. The board digitally controls non digital electronic devices such as sensors and actuators, to control the proces sof interaction. They are different versions of Arduino board. For my design I will use Arduino Mega 2560 R3.

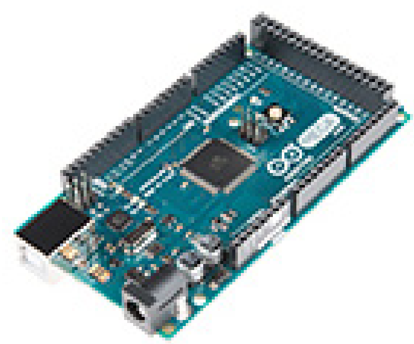


Figure14: Arduino Board

The Arduino Board will be programed, using C++ (the object-Oriented software agent (the computer program) will be encoded as the Arduino program to decide which, if any, action is appropriate and tell the board what to do by sending a set of instructions to the microcontroller on the board. A tool called IDE (Integrated Development Environment) will be used to wright the program (sketch), using C++, and upload it to the Arduino Board.

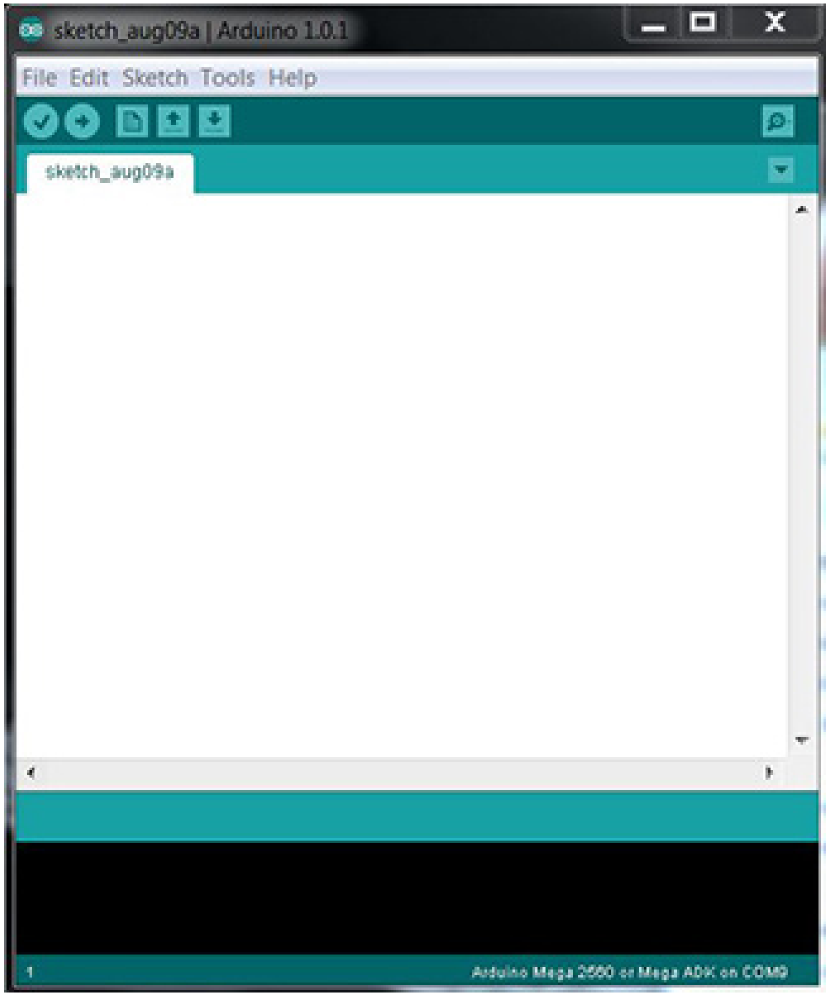


Figure15: IDE, (Integrated Development Environment)

- A sensor – to provide knowledge of real time information on the task environment. Sensors act as eyes, ears, and other human's organs to gather required data from their environment for the simulation. In my design a motion sensor (gesture sensor) called "ZX Distance".

Figure16: motion sensor

It is a touchless sensor that is capable of looking for simple gestures, detecting the motion of the person who has direct interaction with it, will be used.

- Actuators – various motors can be used for the effectors (effectors act as hands, legs, mouth, and other body parts.) Actuators convert energy into required movement. An actuator is a servo or servo motor moving the mechanical design parts (rotary actuator or linear actuator) based on output data provided by the microcontroller, Arduino Board. Different actuators provide different powers. Its type depends on my final mechanical design. I will use a servo as my actuator. A smart control servo - FT-SCServo Smart Servo Motor - With sensory feed back and control speed (velocity) to set the required velocity.

Figure17: Smart control servo

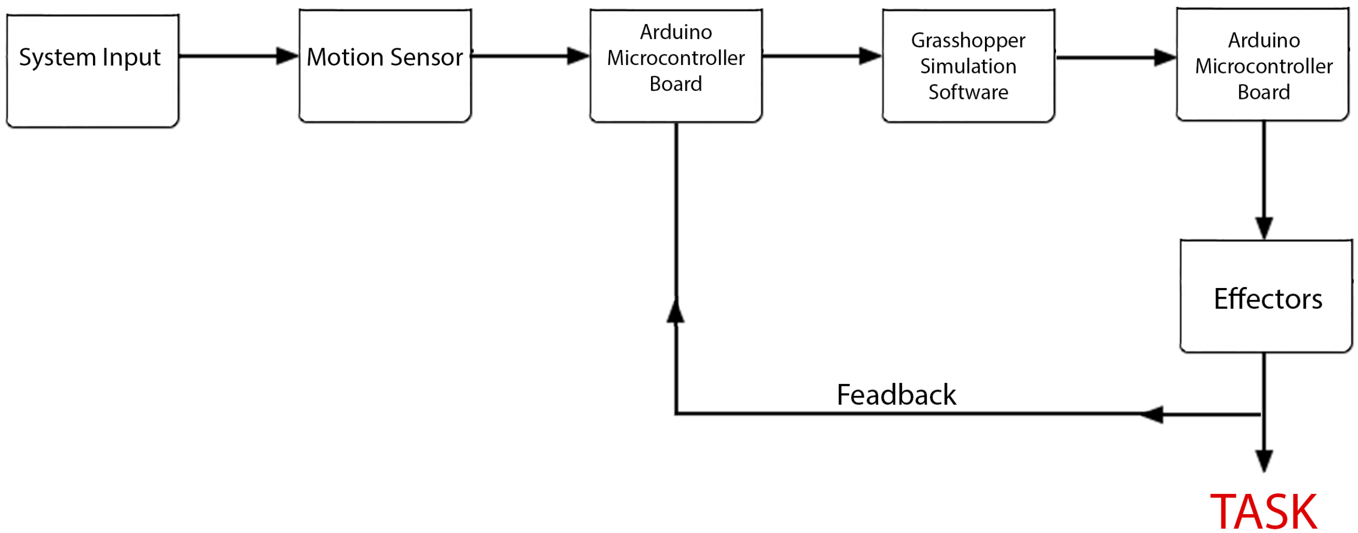
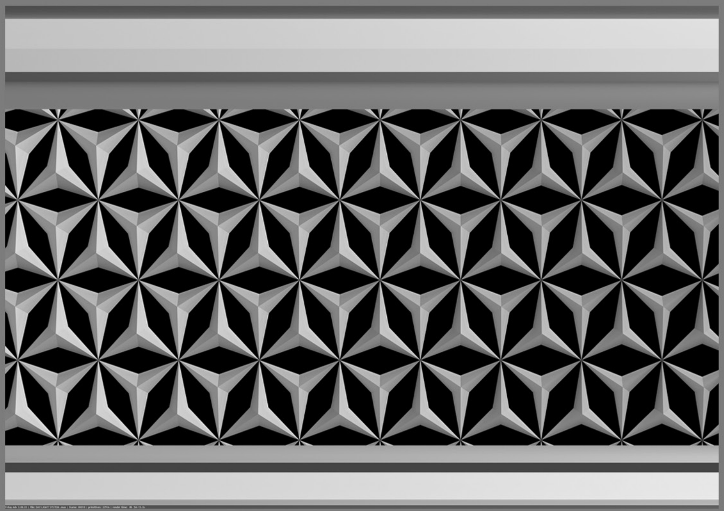


Figure 19- Block diagram representation
of the overal controler



A double skin facade with a digital, mechanical adaptive solution system, which can react and appear to be intelligent

CHAPTER 3
INTERACTIVE FACADE
DESIGN PROCESS

Interactive Facade

Keywords: Behavior

The project is an interrelationship between architecture, computer science, mathematics, mechanical, and industrial design, using multi-functionality of architecture, information of algorithmic design from mathematics, intelligent automation design from computation, and system design from mechanical and industrial design. The result is a double skin facade with a digital, mechanical adaptive solution system, which can react (based on its program) and appear to be intelligent. What I means by intelligent is that the system can move according to the provided program. What makes it intelligent is that it can come up with this movement based on its interaction with the world. The concept is that the interior side of the facade interacts with its inhabitant, perceiving their body movement and adjust its form accordingly. The exterior side interacts with the movement of the sun, working as a shade, reacting to environmental changes. The designed geometric pattern on the outside, complies with a different interior pattern, to provide shade. Each side of the facade has its own behaviour.

Behavior in this case is the reaction to the stimulus which depends on a certain set of parameters.[45]

stimulus → **BEHAVIOR** → response

In order to accomplish the project I have focused on software such as Rhinoceros, Grasshopper, and Grasshopper's plugins (such as Firefly and Kangaroo).

Physical Implementation Proposal

The proposal is to design both sides of the facade but for proof of the concept the project will only physically construct a working prototype of the interior skin.

Data Flow

Initial experimentations were conducted in Grasshopper to understand both the individual movement of each unit and entire system . (In response to one person and multiple people). Subsequently, a digital model was made in Rhino, using Grasshopper Plugin. The developed algorithm in Grass hopper is controlling the automation and movement of the digital model in Rhino. Since the final design, I only designed the module in Rhino as there is no need to simulate in Grasshopper. In the final design stage the Arduino microcontroller will produce the required output to digitally control devices (non-digital electronic system) and complete the movements. The movement is simulated in Rhino using Grasshopper to activate the digital model and pro-duce the required data to show how bodily movement can transform the model. Simulating the interaction based virtual model, using a slider in Grasshopper to control the movement of the geometries along with an attractor point. The system moves according to the environmental inputs, the control is intuitive and corresponds exactly with the virtual modelling space. The result a sensitive system, demonstrating digital model activated based on a single attractor point.

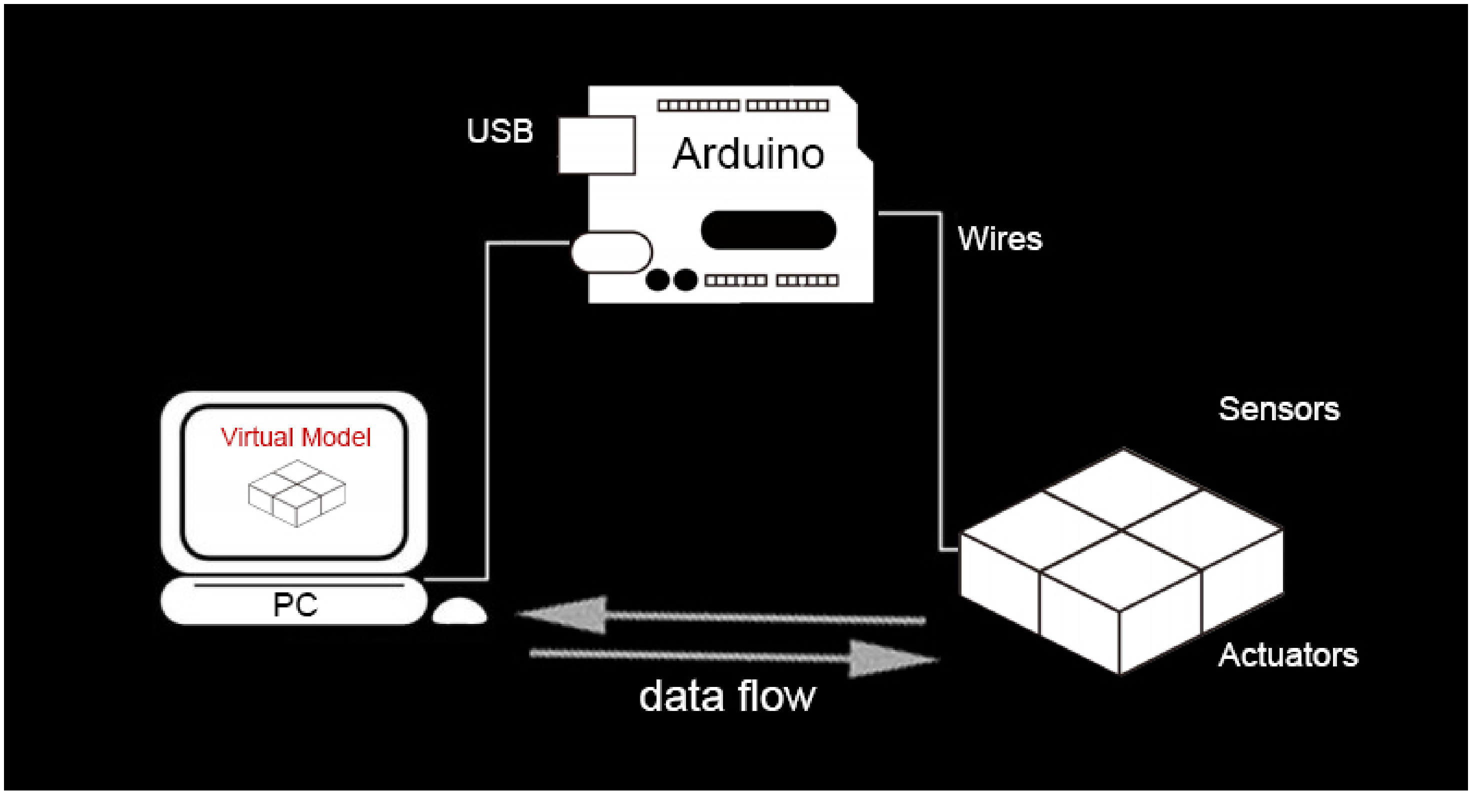


Figure17: Data Flow through the system

The data flow through the system provides automatic calculations for continuous self-modelling to enable the system to response to human input in real time.

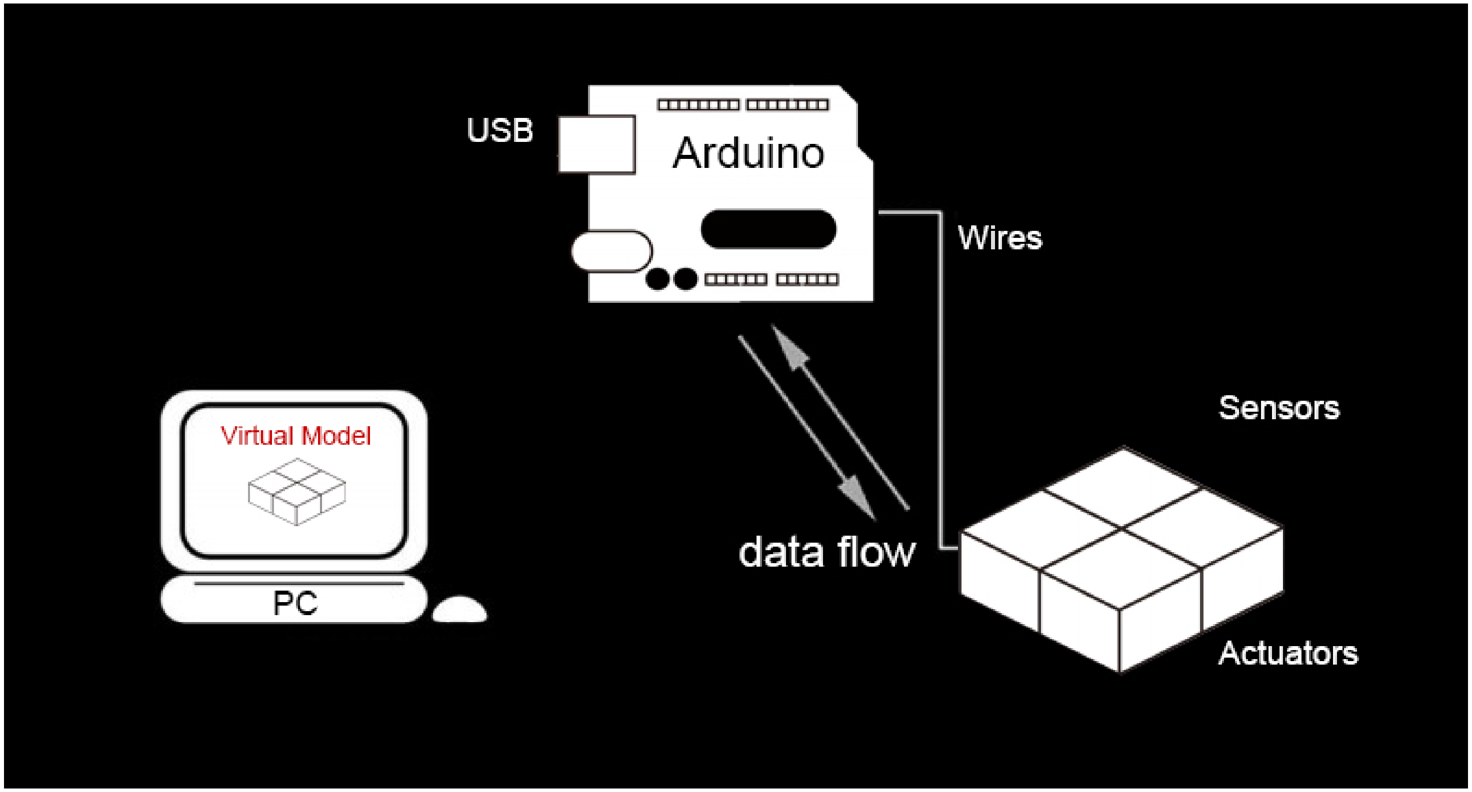


Figure18: Data Flow through the system
for one module (my design)

Initial experimentations were conducted in Grasshopper to understand both the individual movement of each unit and the entire system in response to one person and multiple people. Subsequently, a more detailed digital model was made in Rhino, using the Grasshopper Plugin. The algorithm in Grasshopper controls the automation and movement of the digital model in Rhino.

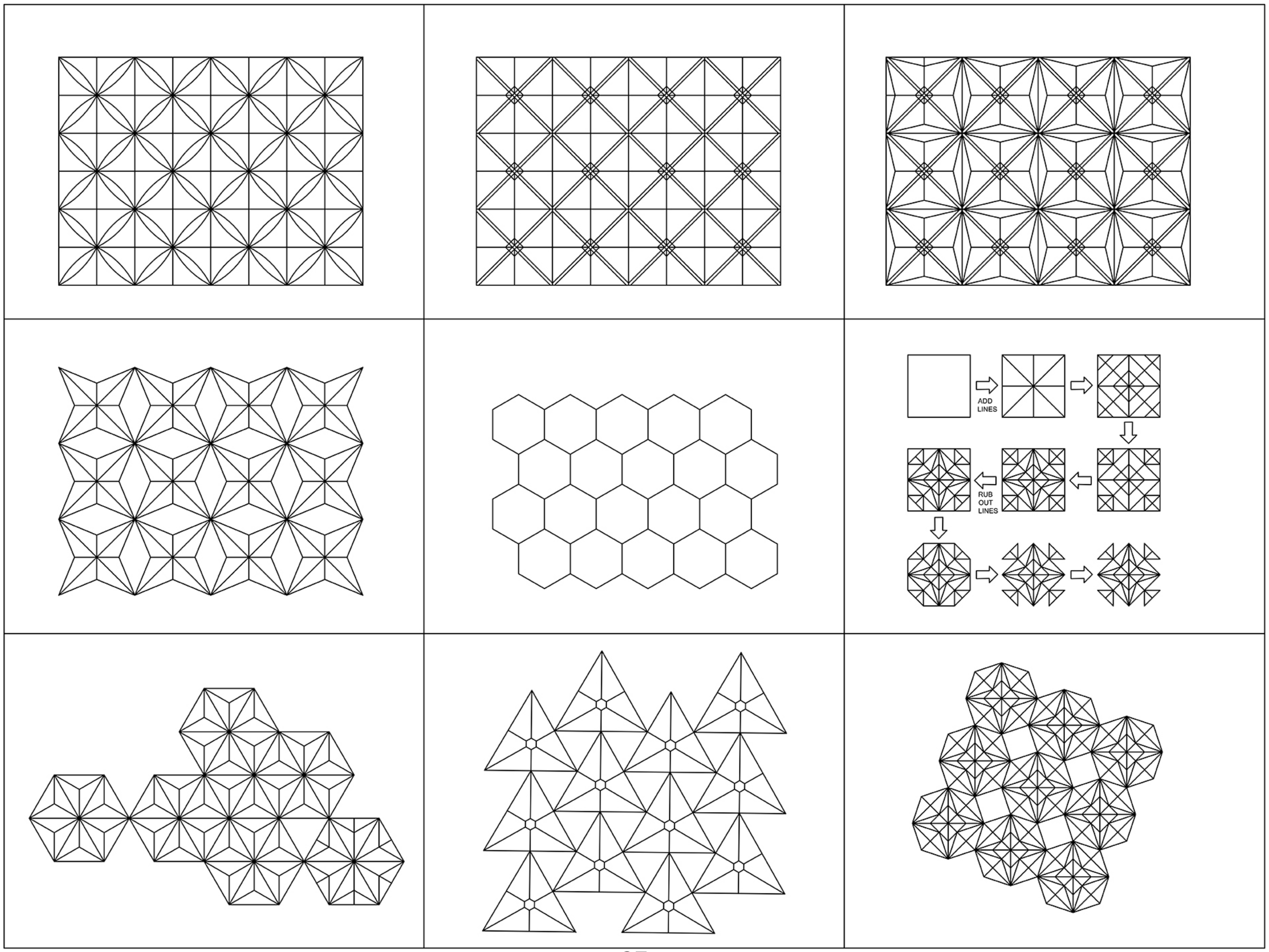
In the final design, I will design a module, not entire system. There is no longer a need to simulate the movement in Grasshopper. In the final design, the Arduino microcontroller will produce the required output to the digitally control devices (non-digital electronic parts of the system- the sensors, actuators, and effectors) to complete the movement.

Design Process

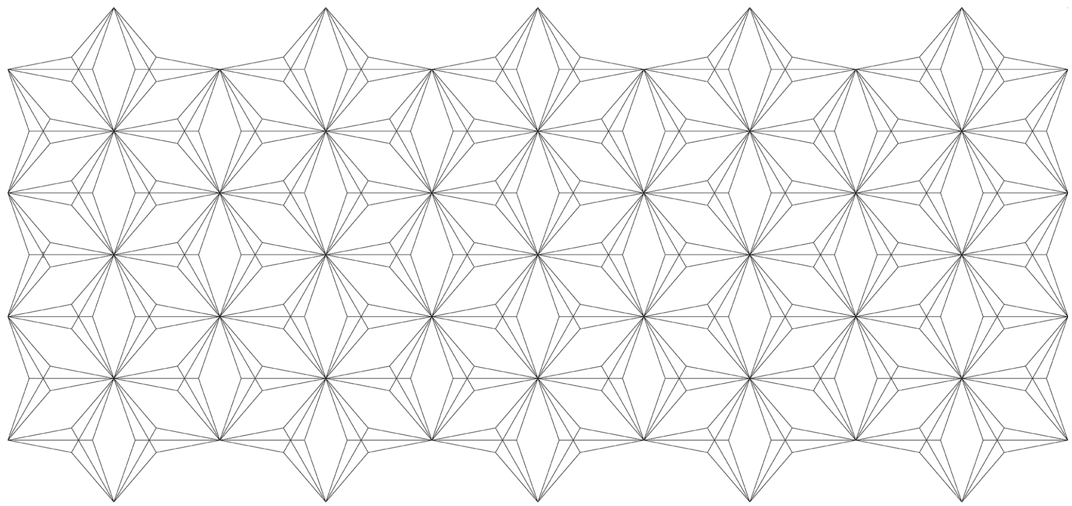
Brainstorm and Development

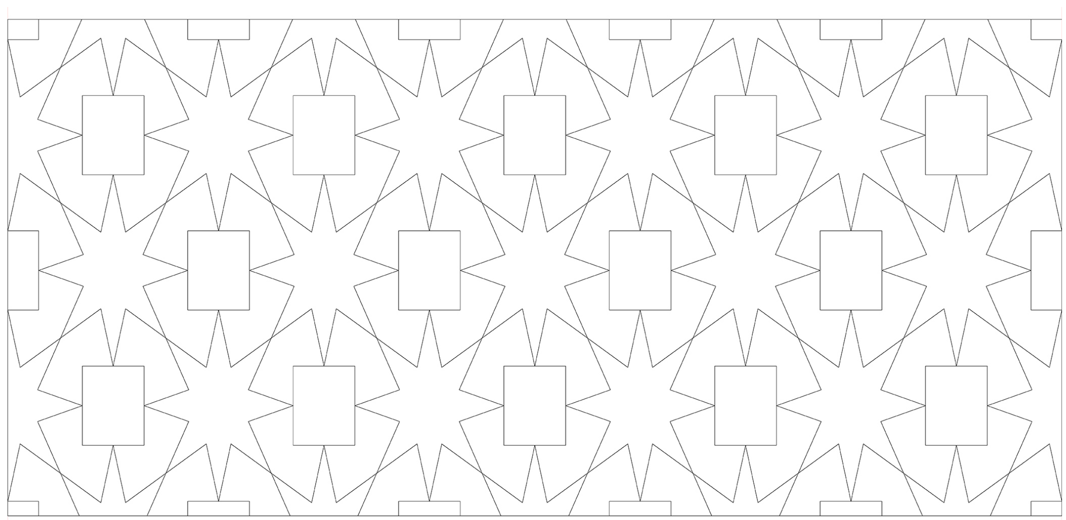
Developed mechanism of the unit's movement

A basic set of geometries come together to make the system. I conducted a series of geometric studies, regular hexagonal, quadrangular, and triangular tessellations were tested to develop the geometric pattern and movement of the facade. During the exploration I noticed that if I replace faces and edges in a uniform tiling by individual reciprocal frames and their connections, respectively, I can obtain the large structure.

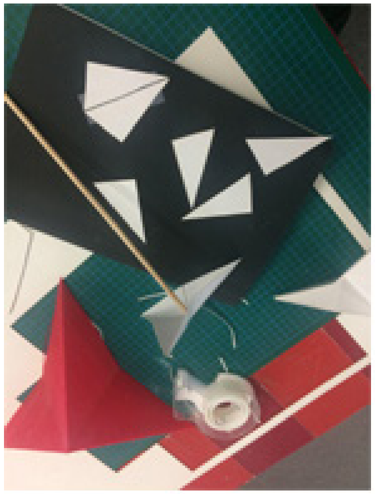
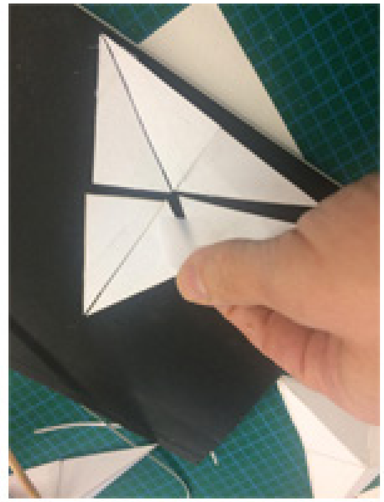
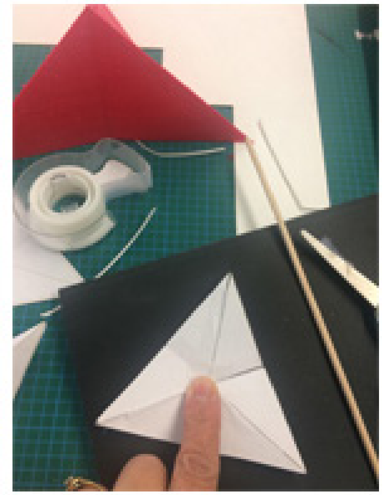
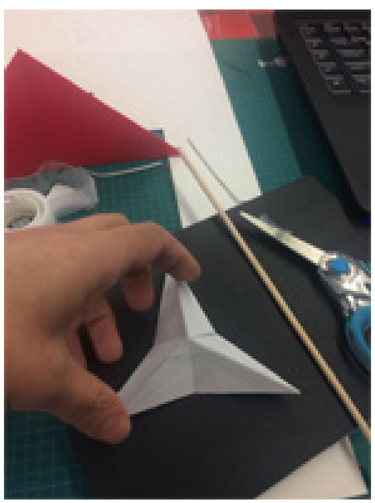
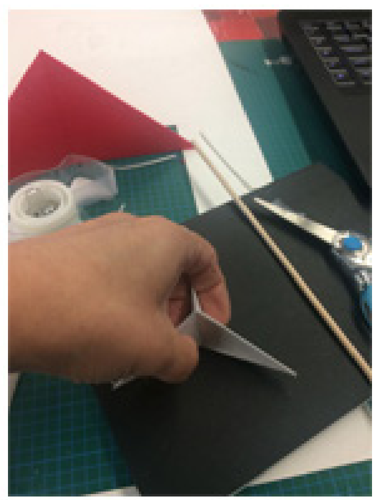
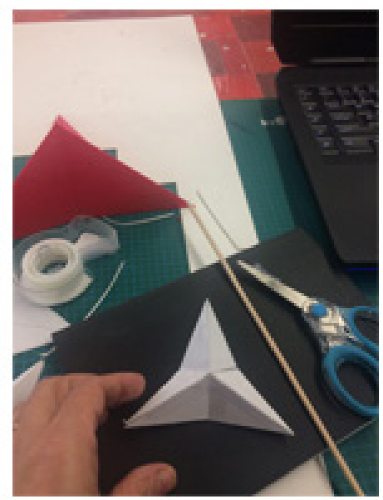

ADD LINES
RUB OUT LINES

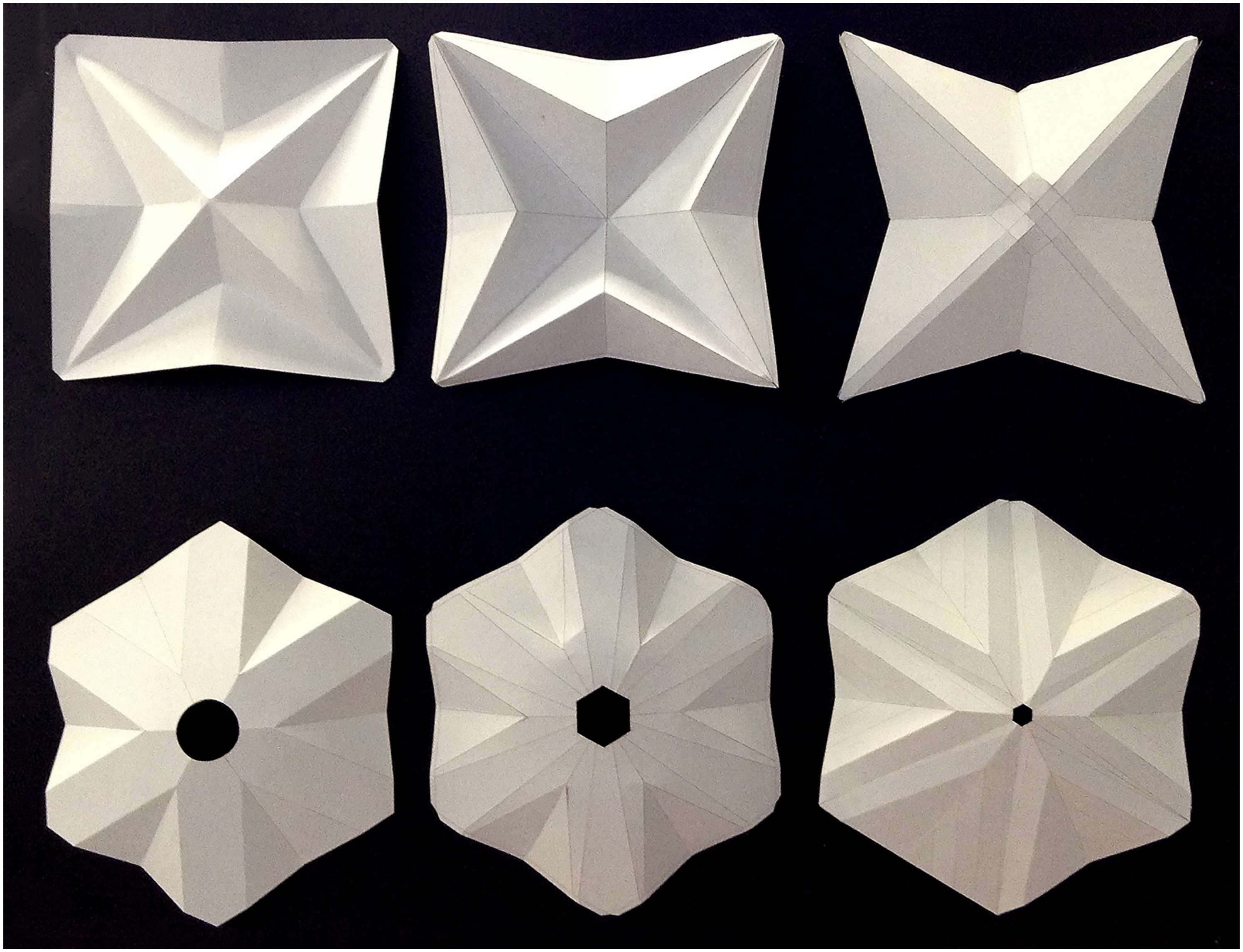
Finding two patterns for both interior and exterior side

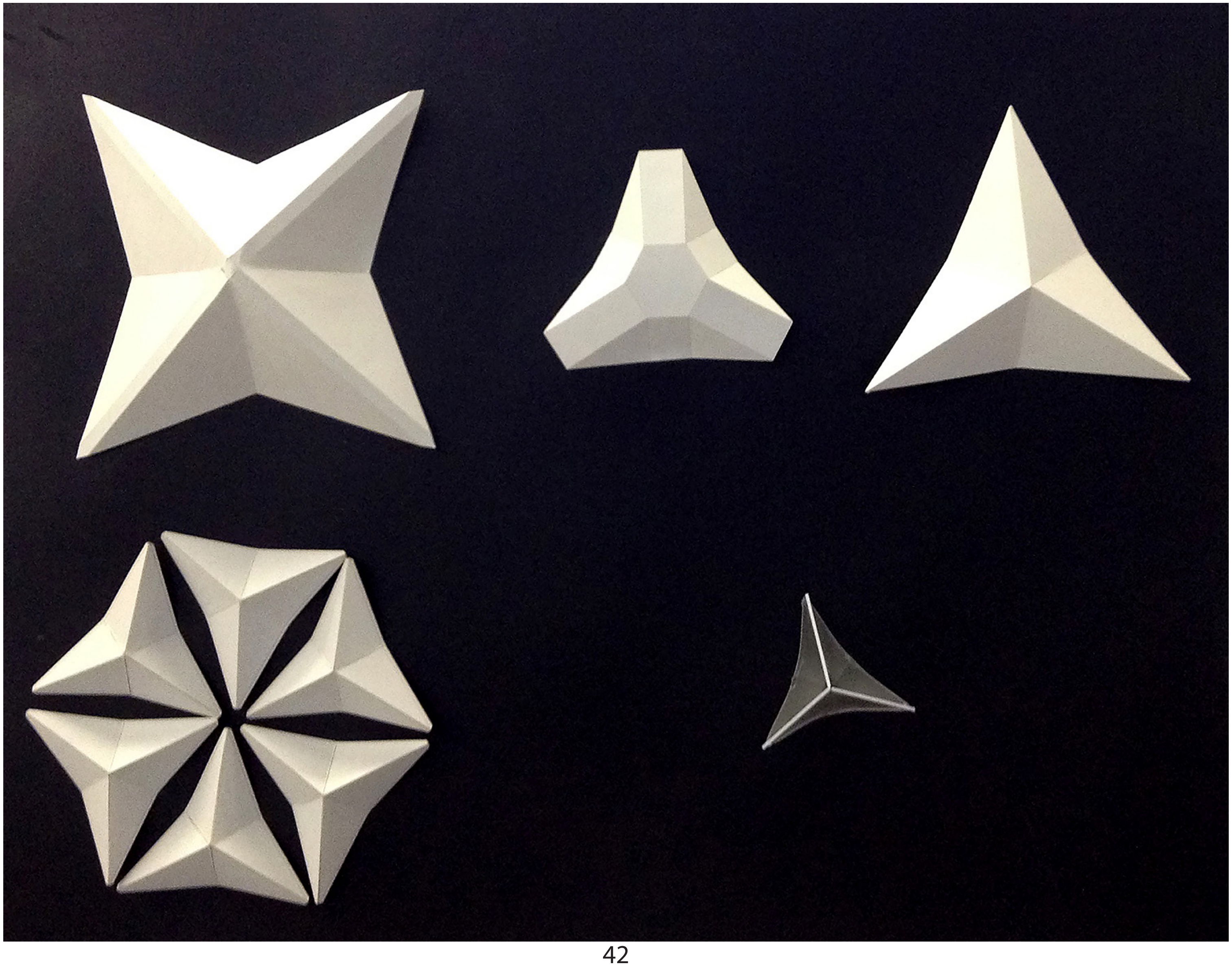




These units were first constructed from paper to develop the final composition. I further completed a number of Origami studies to find a radial symmetry that could easily repeat across the plane of the facade. In this case there is a common point allowing any of the three corners to work as a rotation point to generate new shapes around each pattern.



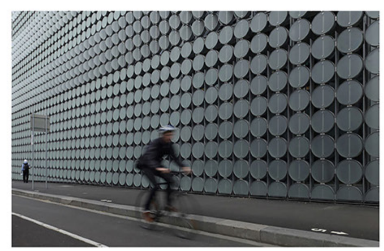


Design Process

looking at Adaptive, Kinetic, and Responsive architecture, 3 case studies have been chosen to investigate their design.

Al bahr Tower Abu Dhabi

Jean Nouvel Arab Institute



Rmit's Design School Melbourne

Panelization

a fully periodic tiling was achieved, In a following step the panels were planarized, a hexagonal pattern serves as the base for the facade, spliting the surfaces on both side of the facade into individual hexagonal panels.

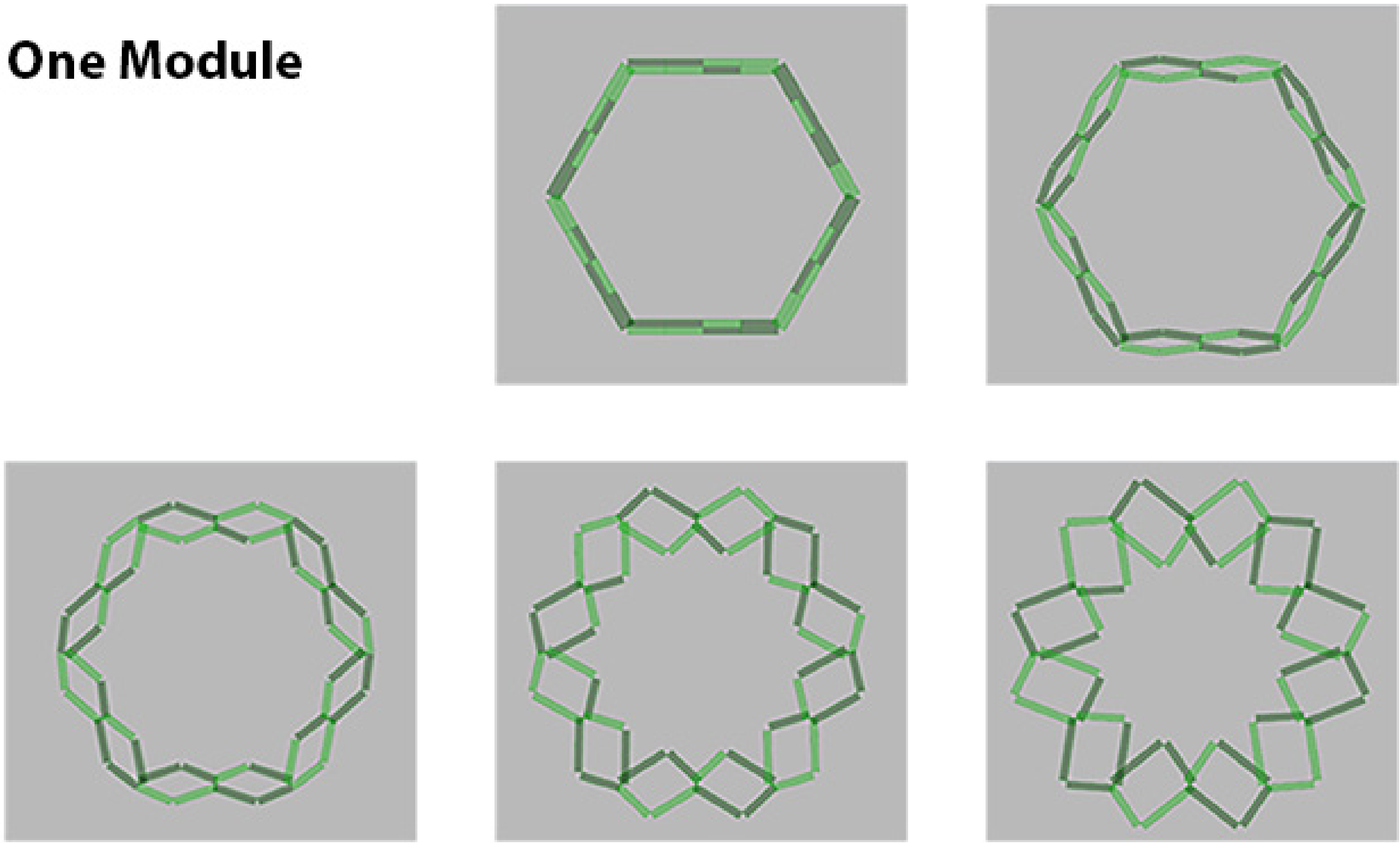
Design Process

Simulation

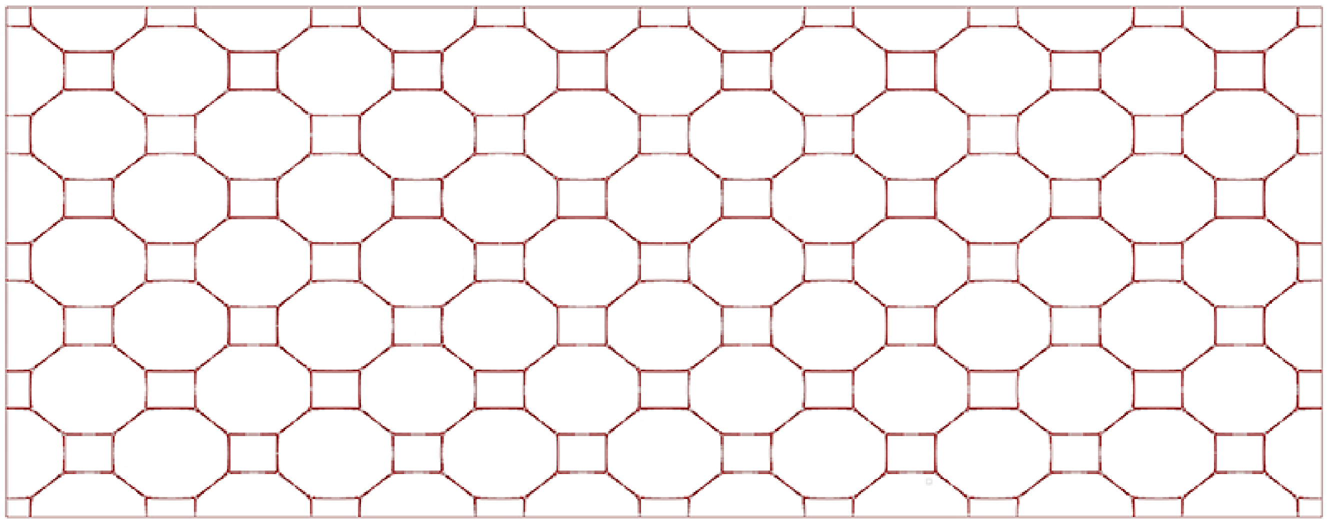
The interactive computational tool (Grasshopper) considers two aspects: design and assembly. The generated interior and exterior of the facade systetem are parametric developed in Grasshopper. I focused on tiling surfaces with a large distortion on both sides of the facade. The method developed in Grasshopper required a certain level of expertise in developing algorithm in Grasshopper which I have done a lot of research about and I will attached them to the thesis in my defence.

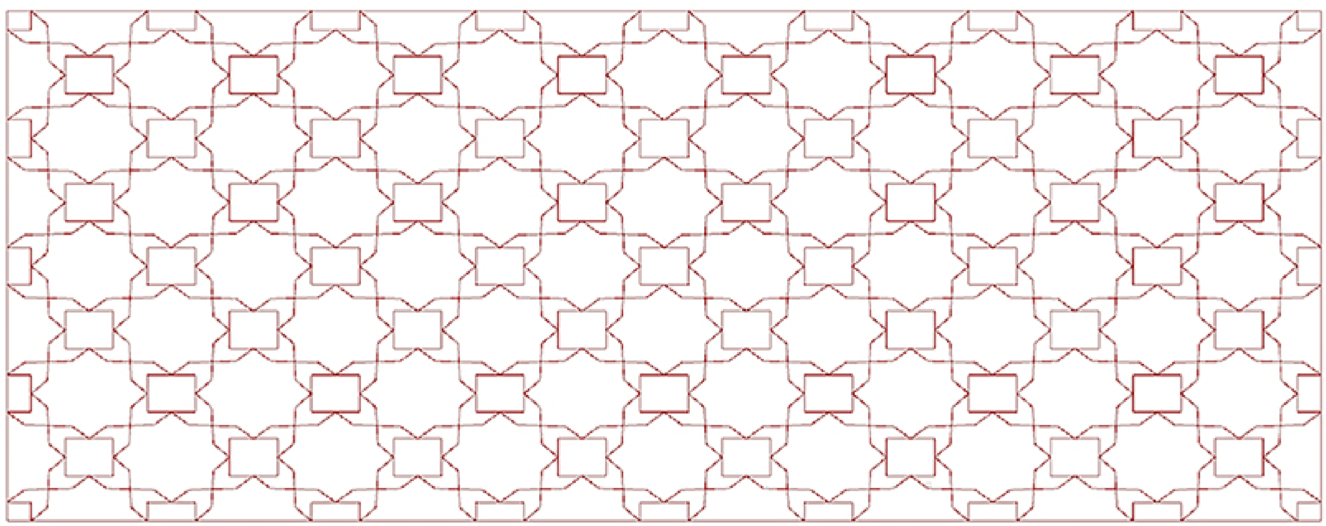
Exterior Movement

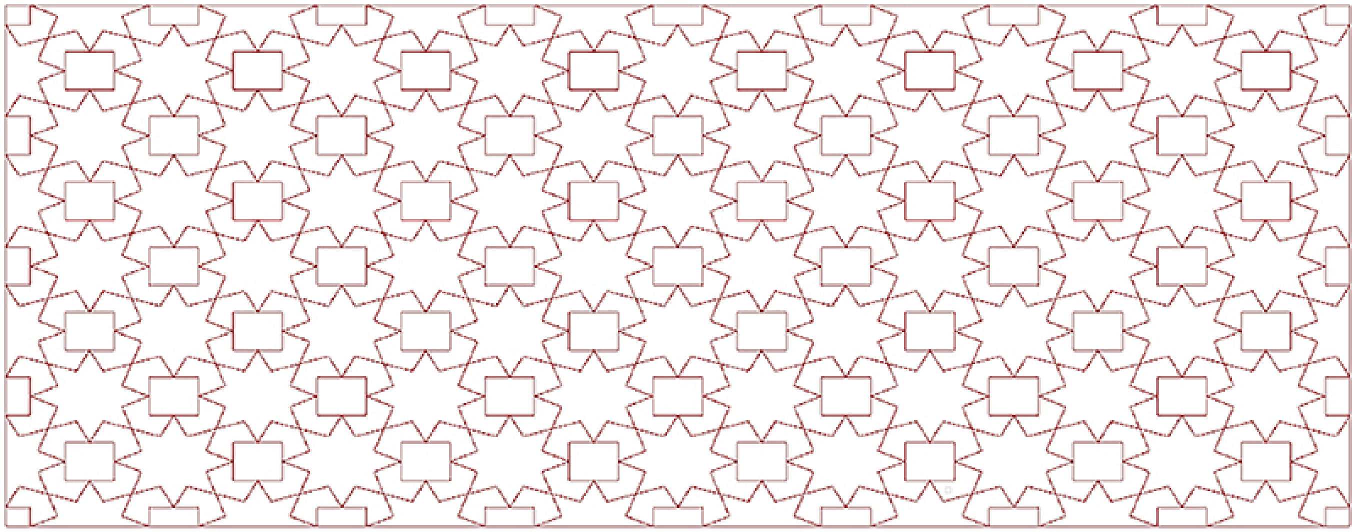
One Module

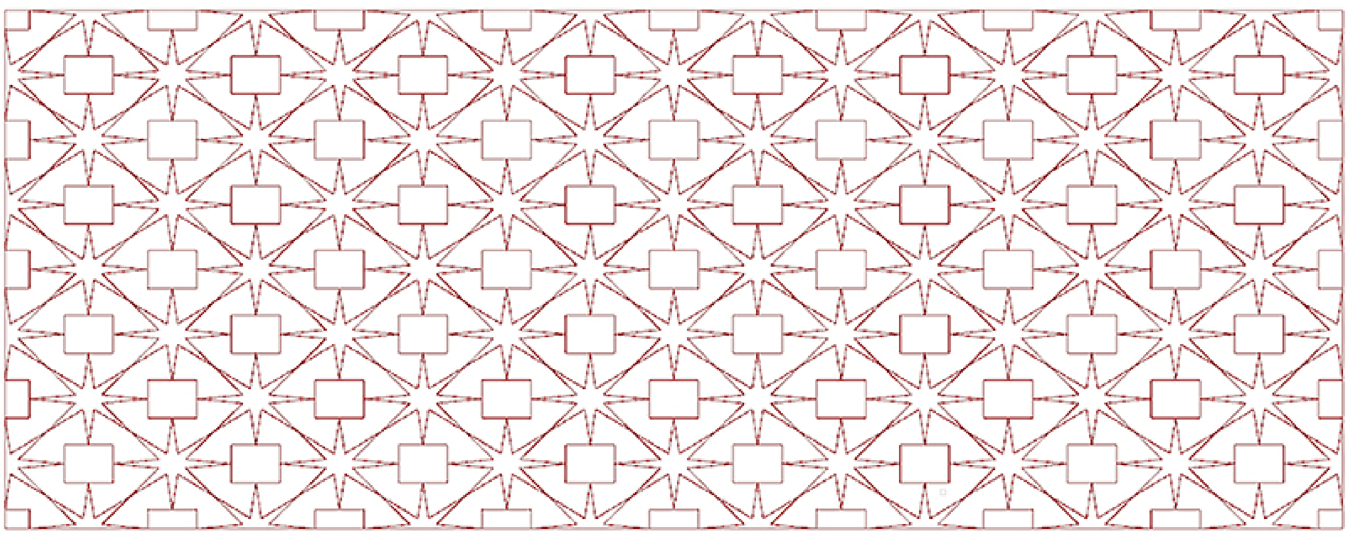


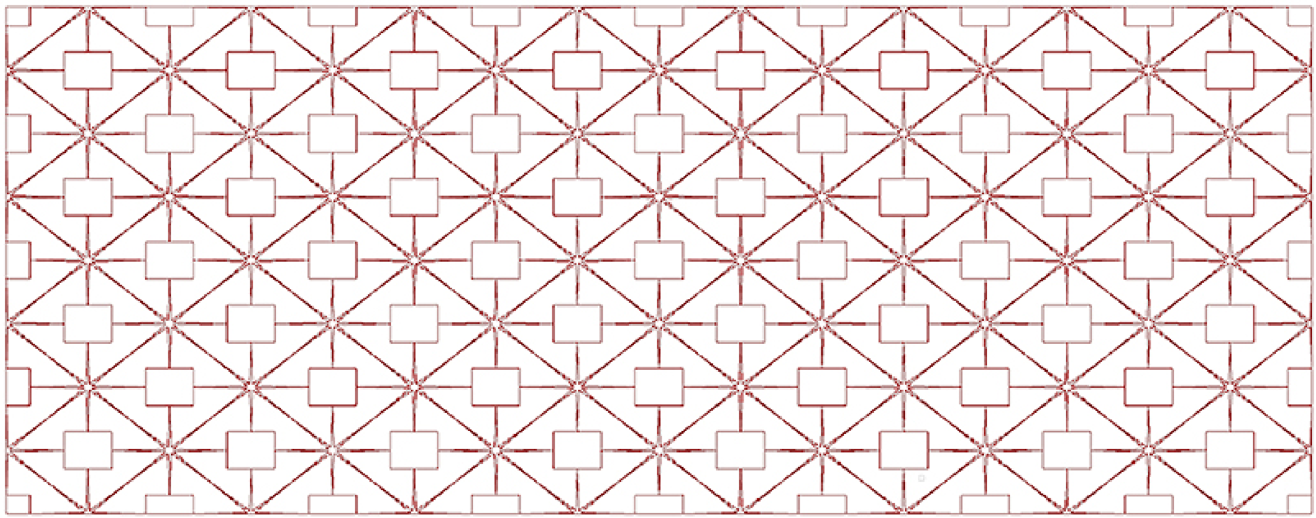
Entire System- Exterior

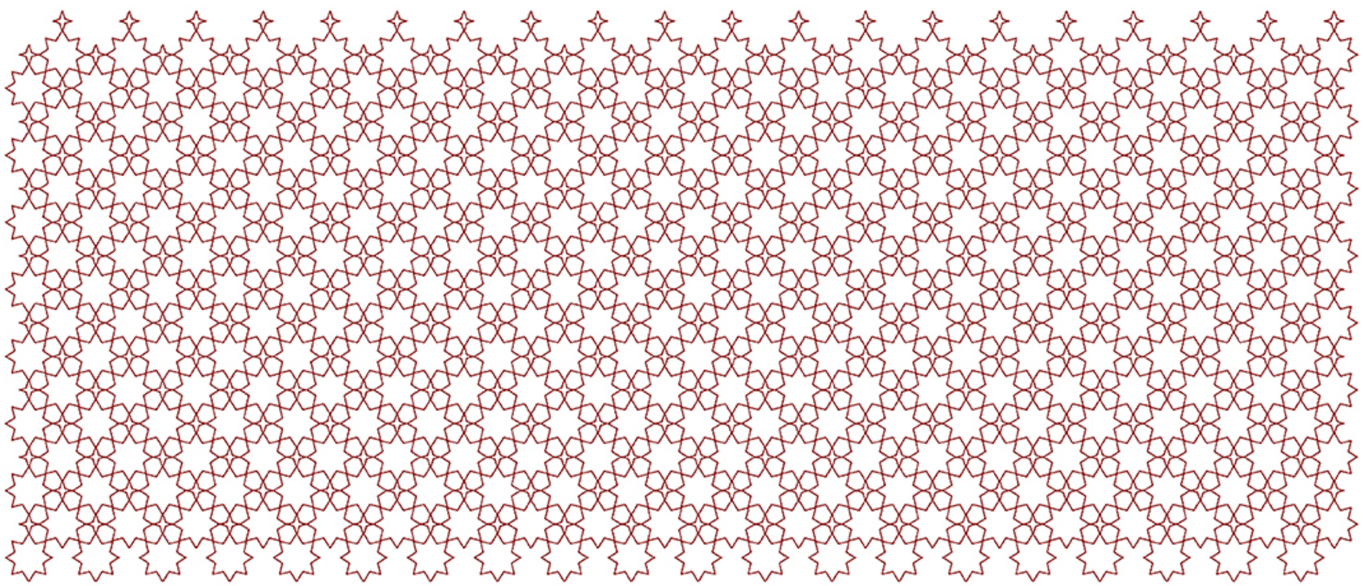




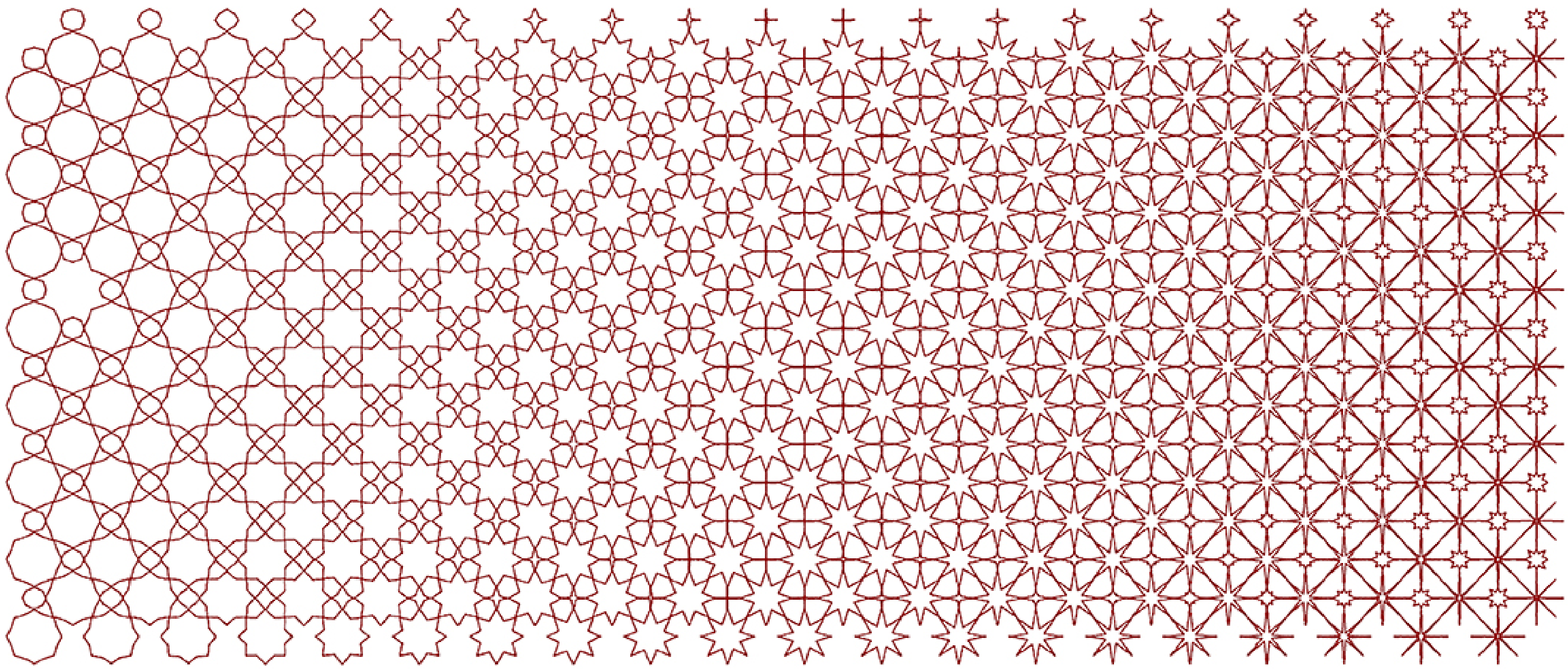




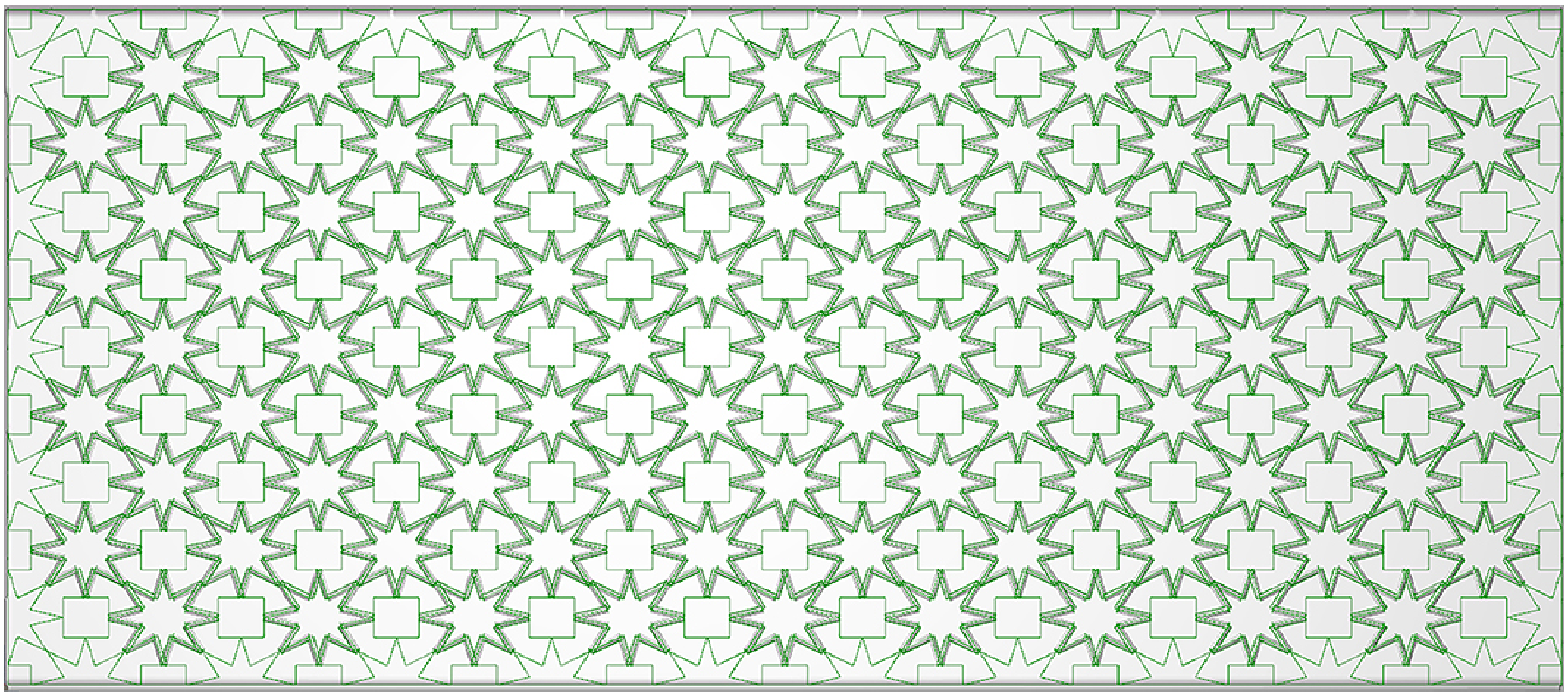




Entire system movement based on its interaction with the movement of the sun

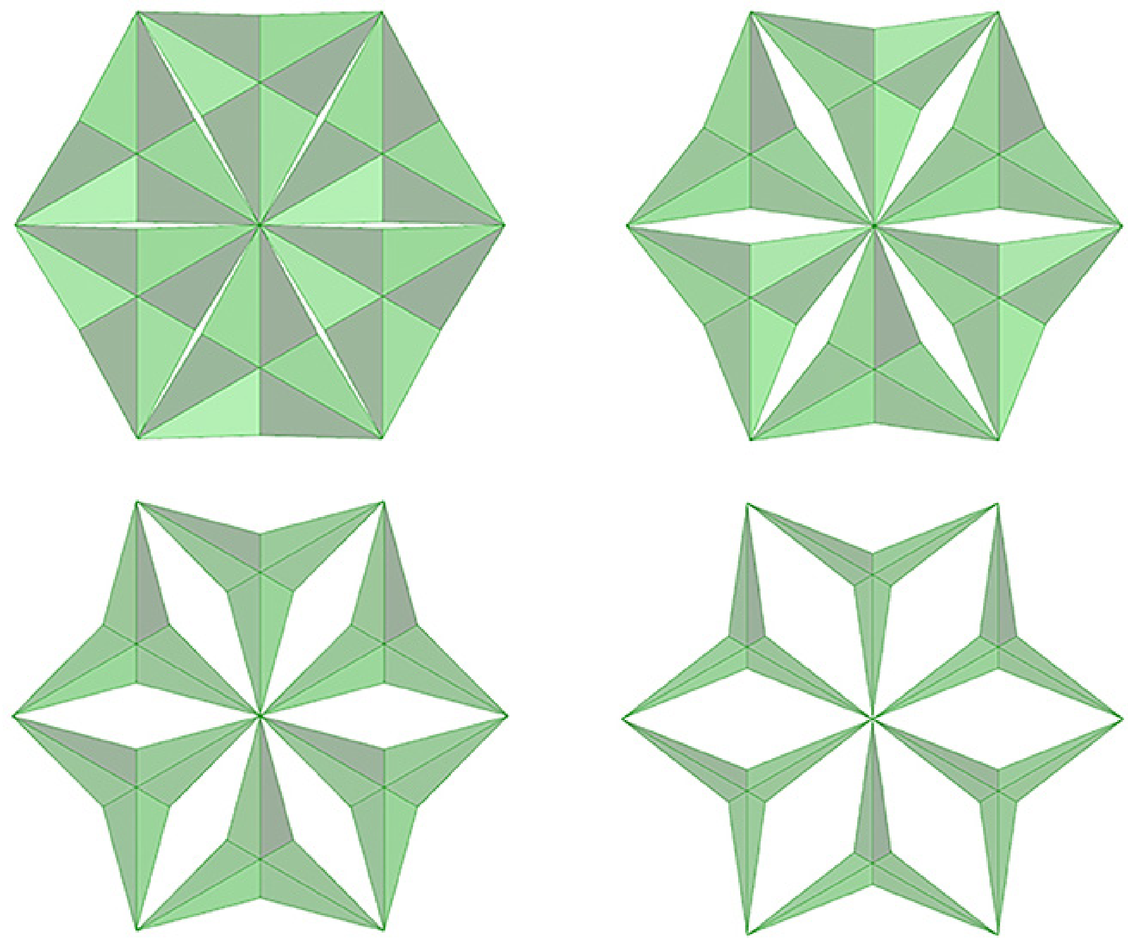


The exterior side is shaded to either close and block the sun or open and let the sunlight enter into the building.

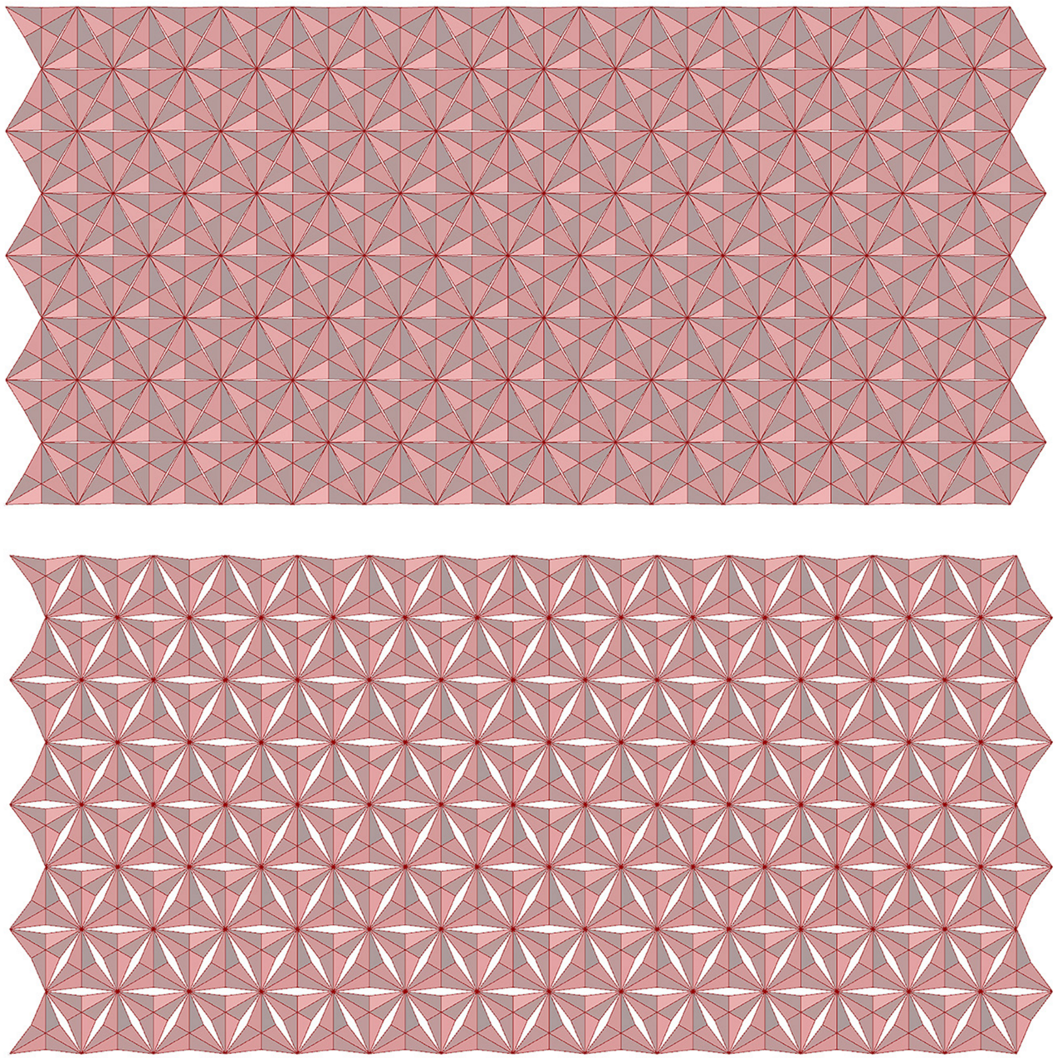


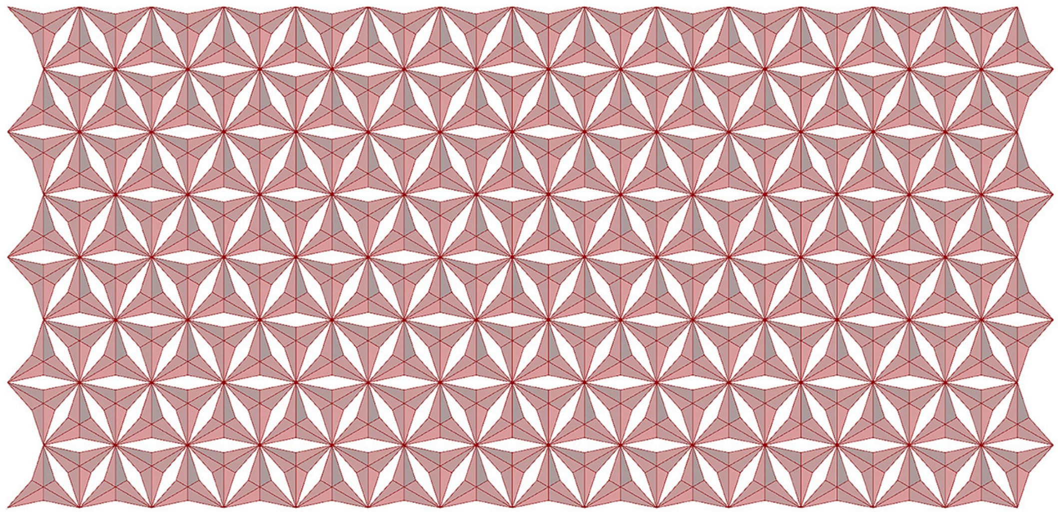
Interior Movement

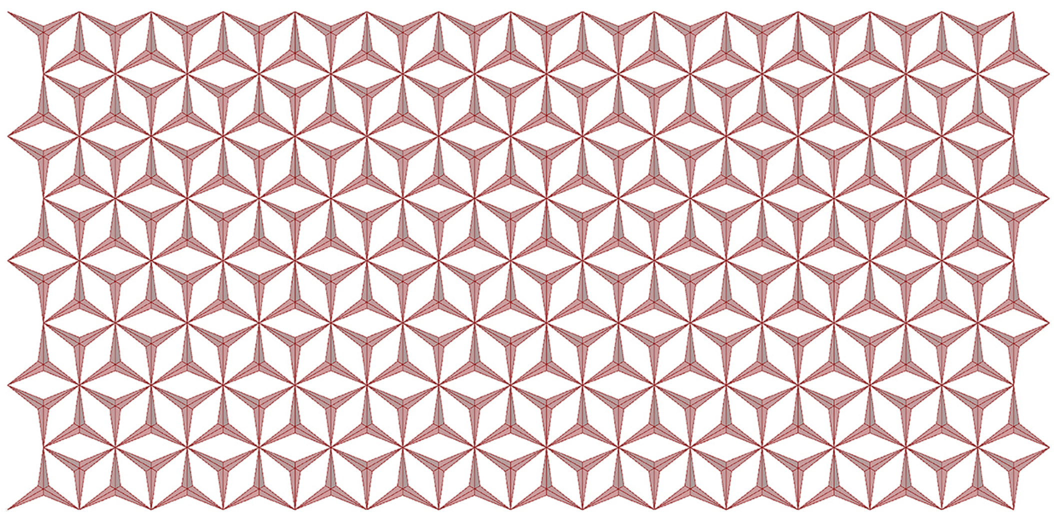
One Module



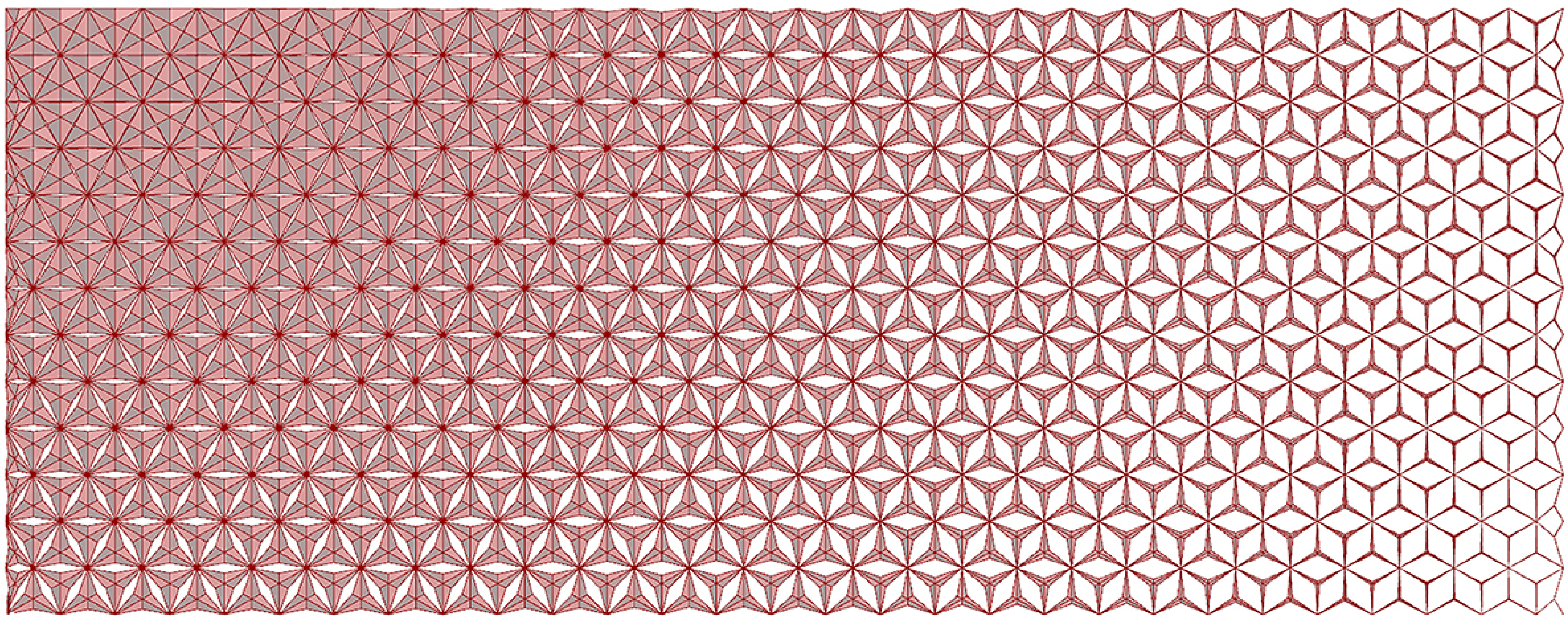
Entire System- Interior







Entire System movement based on its interaction with people

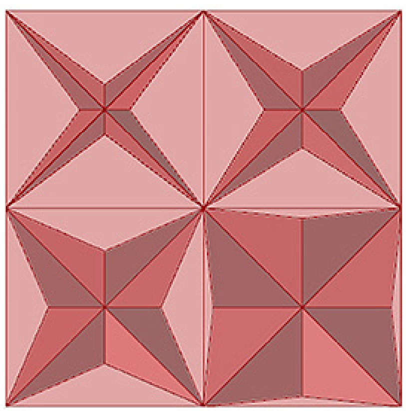


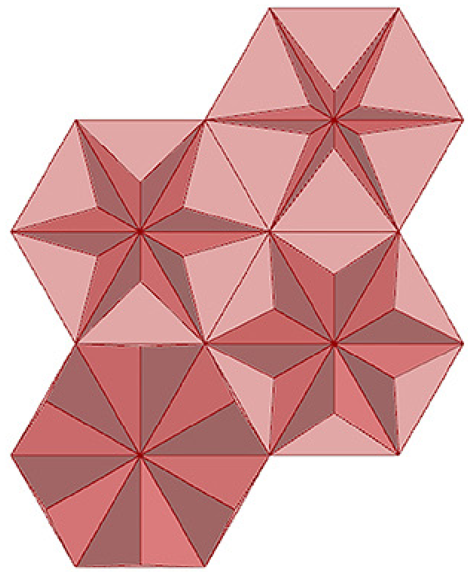
Design Process

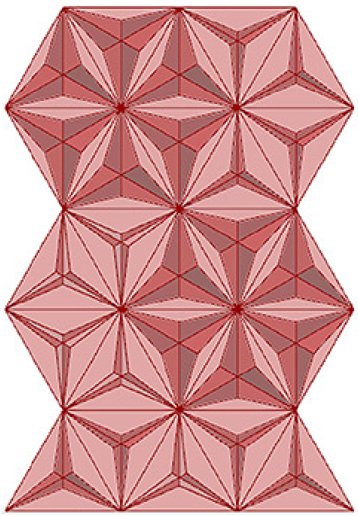
Development of the design algorithm in Grasshopper

Parametric modeling

The computational design in this case can help to figure out the rules, to go from rules to in-stances, using parametric generative computational design in Grasshopper to design the system's movement, writing algorithm that can set the particular movement based on interaction. I took different 3 dimensional models in Grasshopper and change them constantly over and over again, looking at several possibilities in design to find the one that is fundamental to design. I finally came to the final algorithm that gave me the form and its required movements. Using polygon as the base to give one form and generate others as much as are needed in the facade; Generating smart geometry that have all the information and data needed which can be applied to other models to create the entire system. To do so, I must came with a very simple set of rules in a parametric model and generate a lot of complexity from them. Finally there are different representation in each unit based on their interaction with people, each one has its own character having difference from with other ones. The result developing required algorithm in Grasshopper as an integrated universal design solution to adjust the movement of different units to the movement of the people passing by in which each unit can self-adjust to different cases.

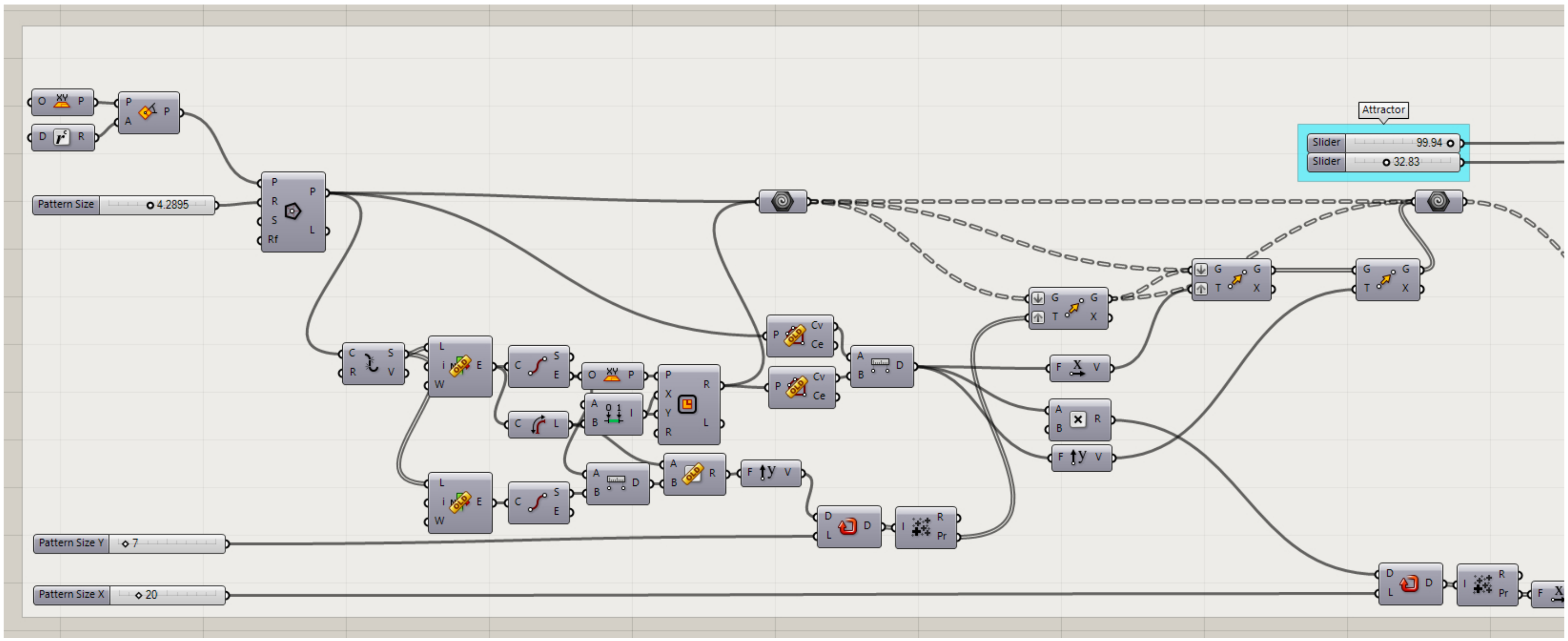


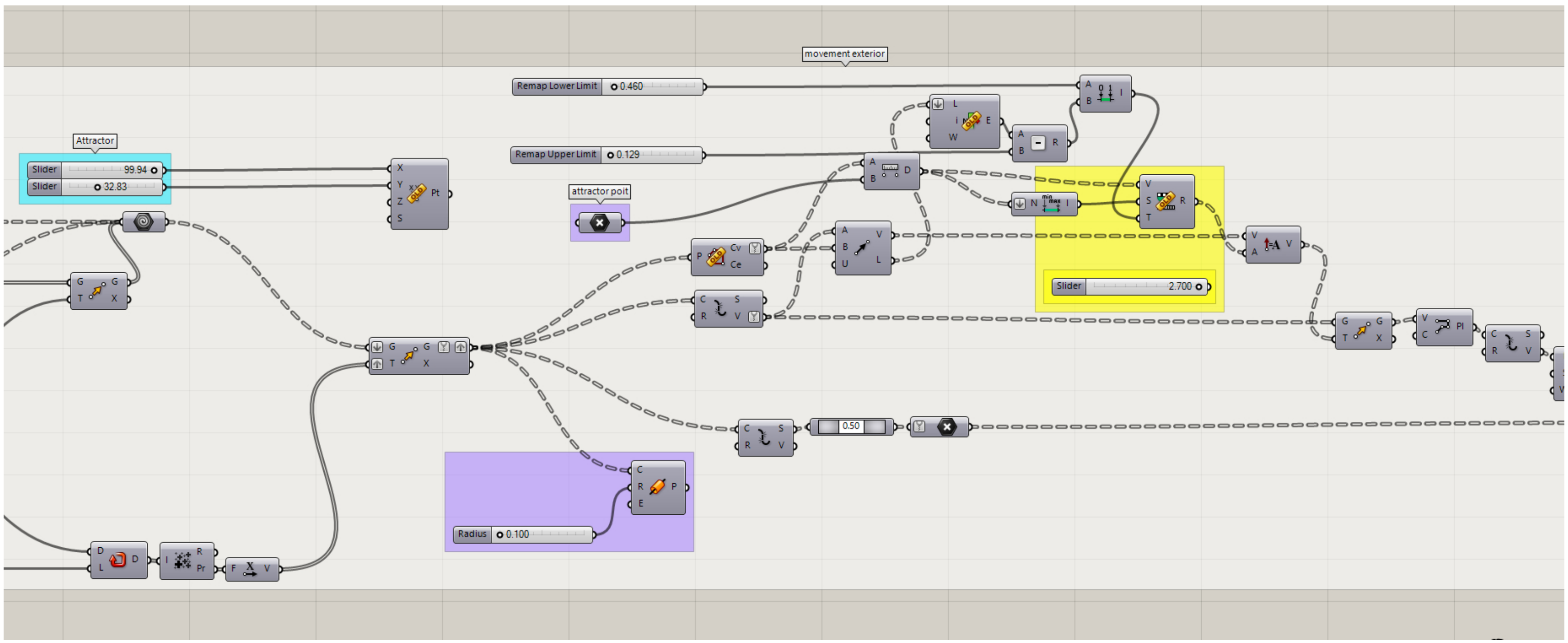


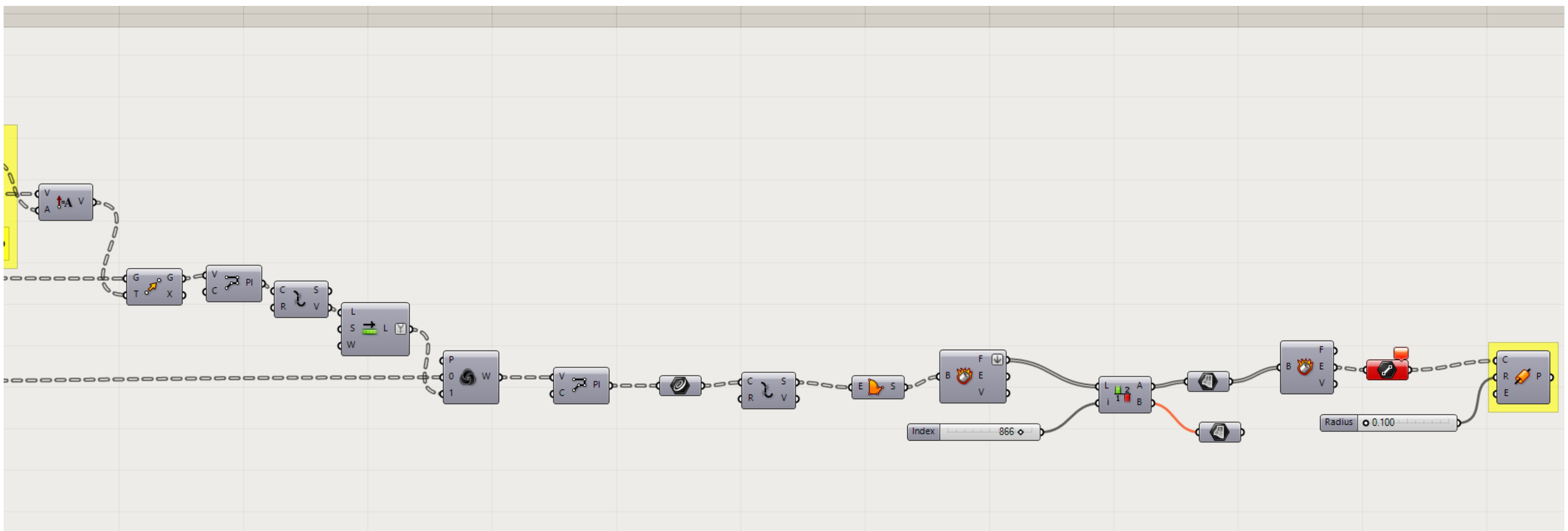


Exploring different geometry and the way they can come together 3 dimensionally in Grasshopper

Grasshopper Algorithms
1- Exterior Side Movement

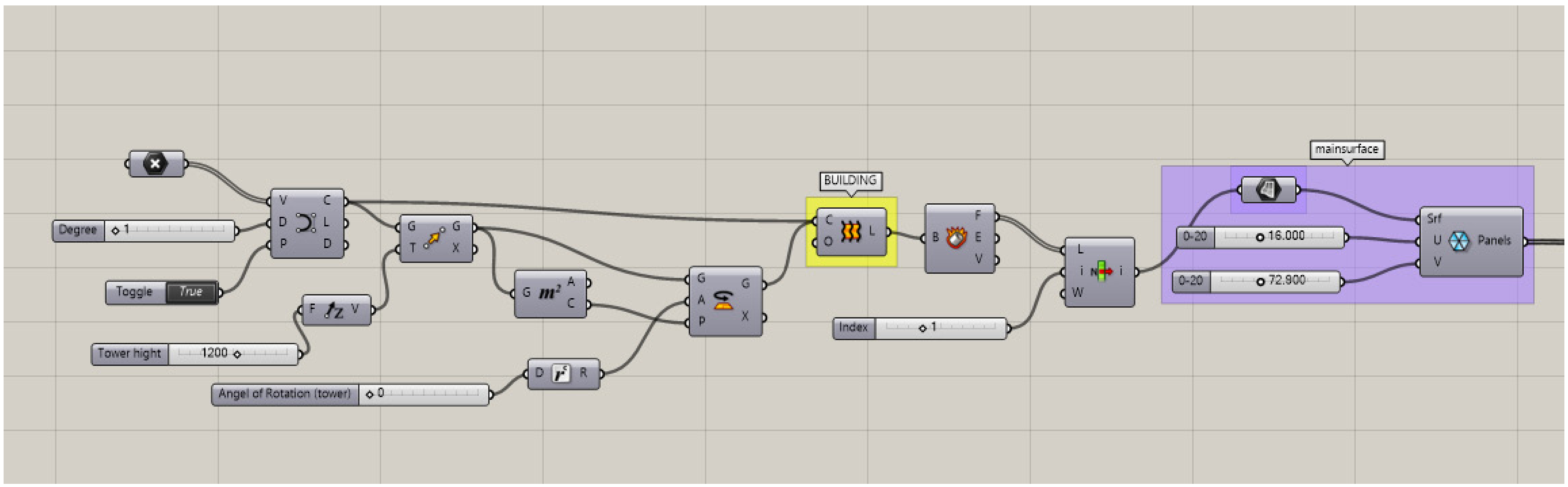


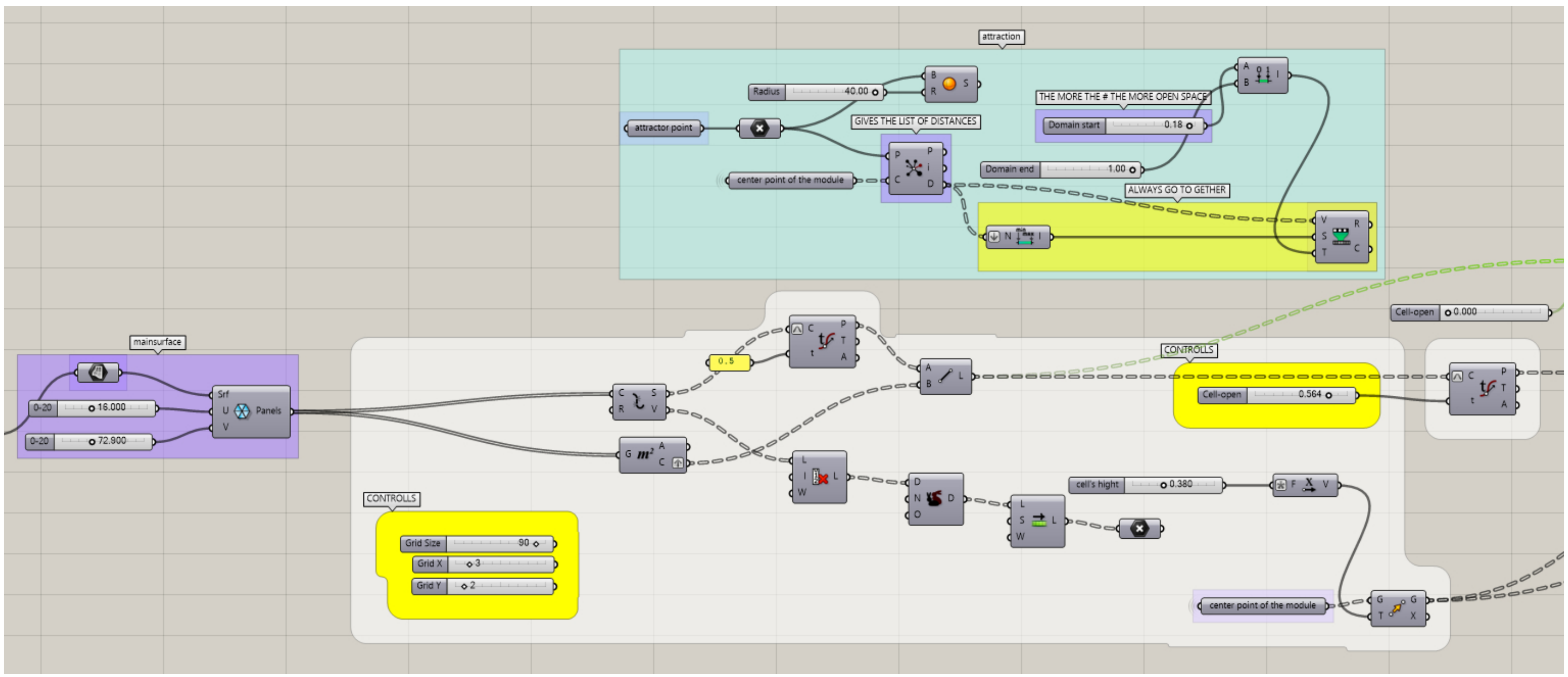

movement exterior
Remap Lower Limit
0.460
Remap Upper Limit
0.129
Attractor
Slider
99.94
Slider
32.83
attractor poit
Slider
2.700
0.50
Radius
0.100

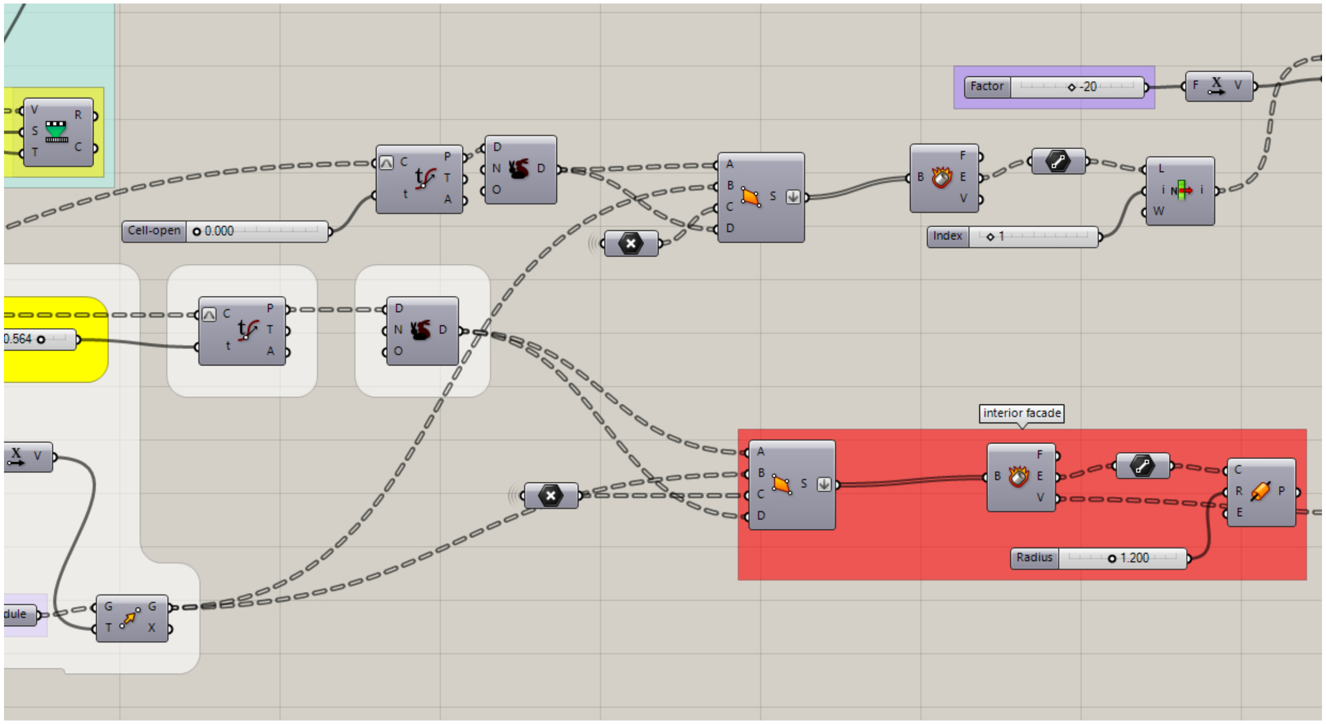


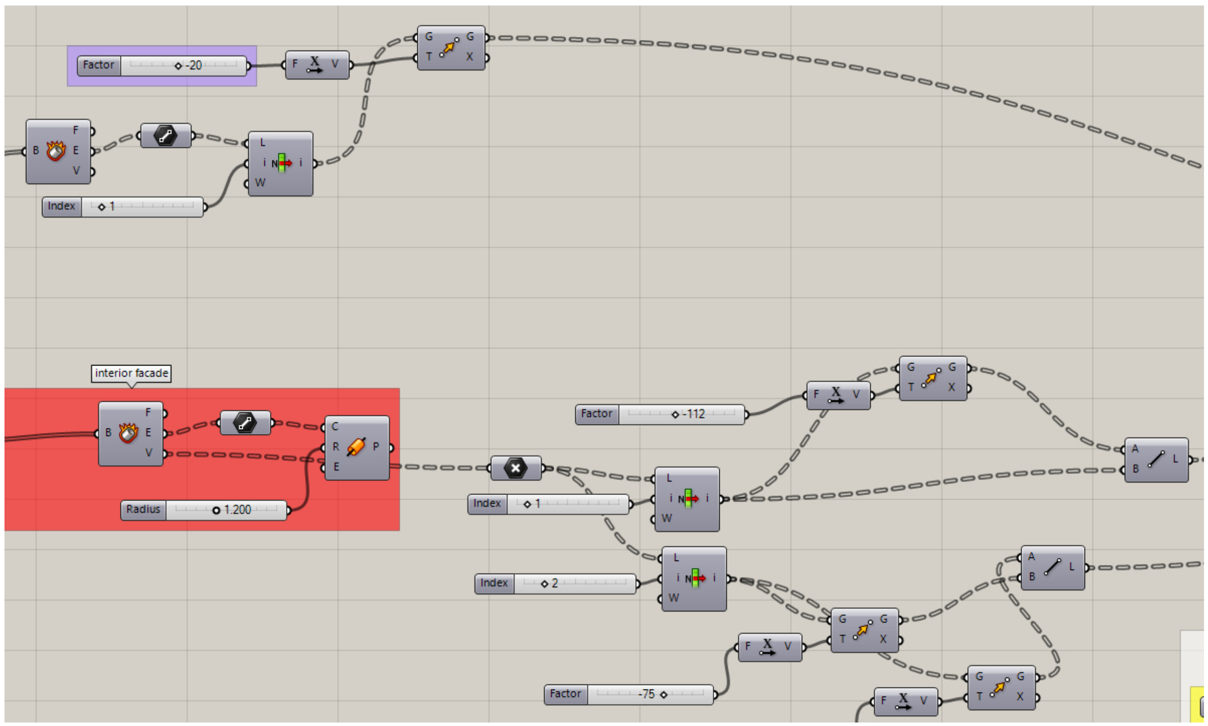
Index
866
Radius
0.100

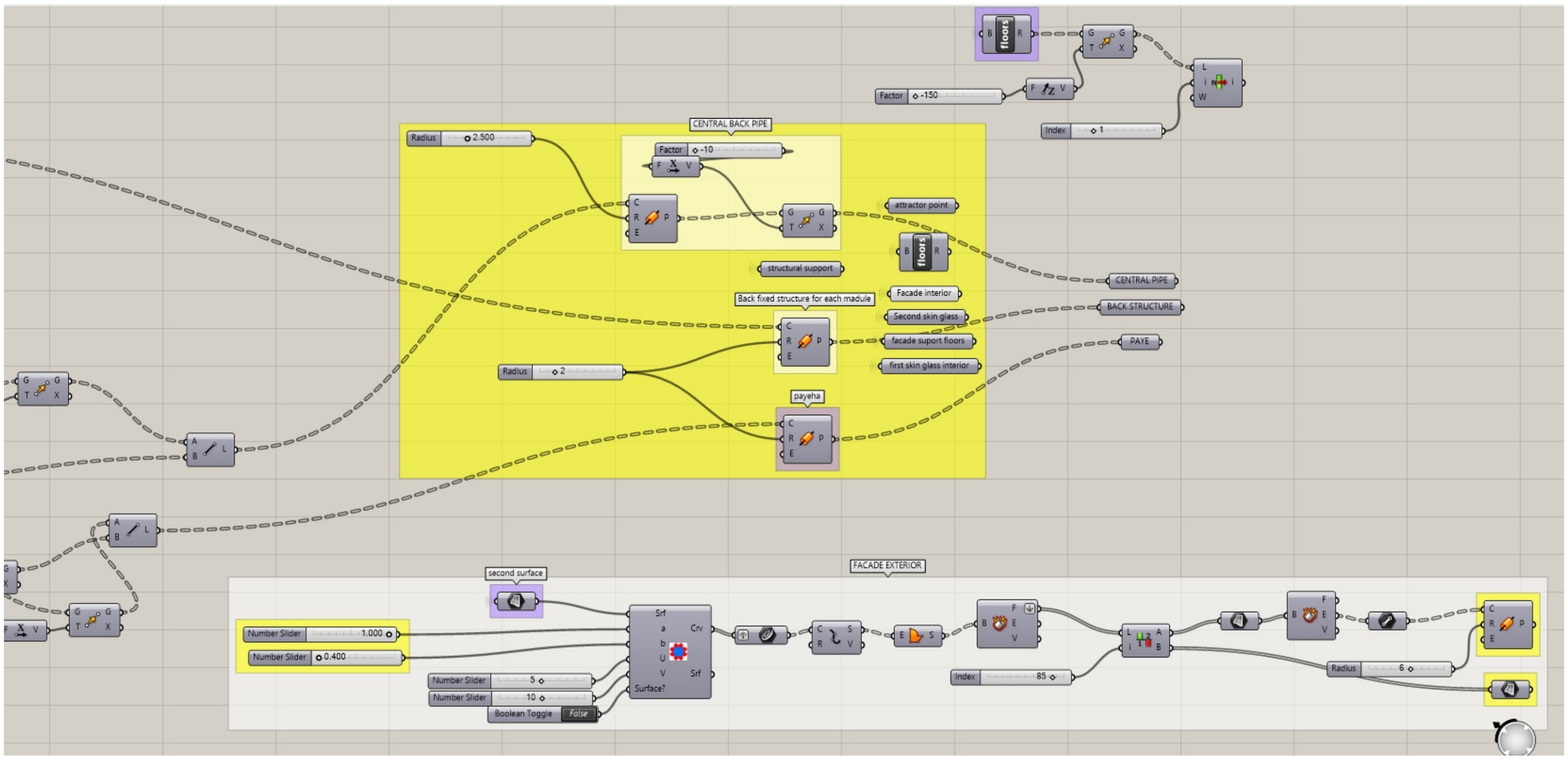
Grasshopper Algorithms
1- Interior Side Movement and its Structure




attraction
Radius
40.00
attractor point
GIVES THE LIST OF DISTANCES
center point of the module
THE MORE THE # THE MORE OPEN SPACE
Domain start
0.18
Domain end
1.00
ALWAYS GO TO GETHER
mainsurface
0-20
16.000
0-20
72.900
Panels
0.5
CONTROLLS
Cell-open
0.564
Cell-open
0.000
cell's hight
0.380
center point of the module
CONTROLLS
Grid Size
90
Grid X
3
Grid Y
2


Factor
-20
Cell-open
0.000
Index
1
0.564
interior facade
Radius
1.200


Factor
-20
Index
1
interior facade
Radius
1.200
Factor
-112
Index
1
Index
2
Factor
-75



floors
Factor
-150
Index
1
CENTRAL BACK PIPE
Radius
2.500
Factor
-10
attractor point
structural support
Facade interior
Back fixed structure for each madule
Second skin glass
facade suport floors
first skin glass interior
Radius
2
payeha
CENTRAL PIPE
BACK STRUCTURE
PAYE
second surface
FACADE EXTERIOR
Number Slider
1.000
Number Slider
0.400
Number Slider
5
Number Slider
10
Boolean Toggle
False
Index
85
Radius
6

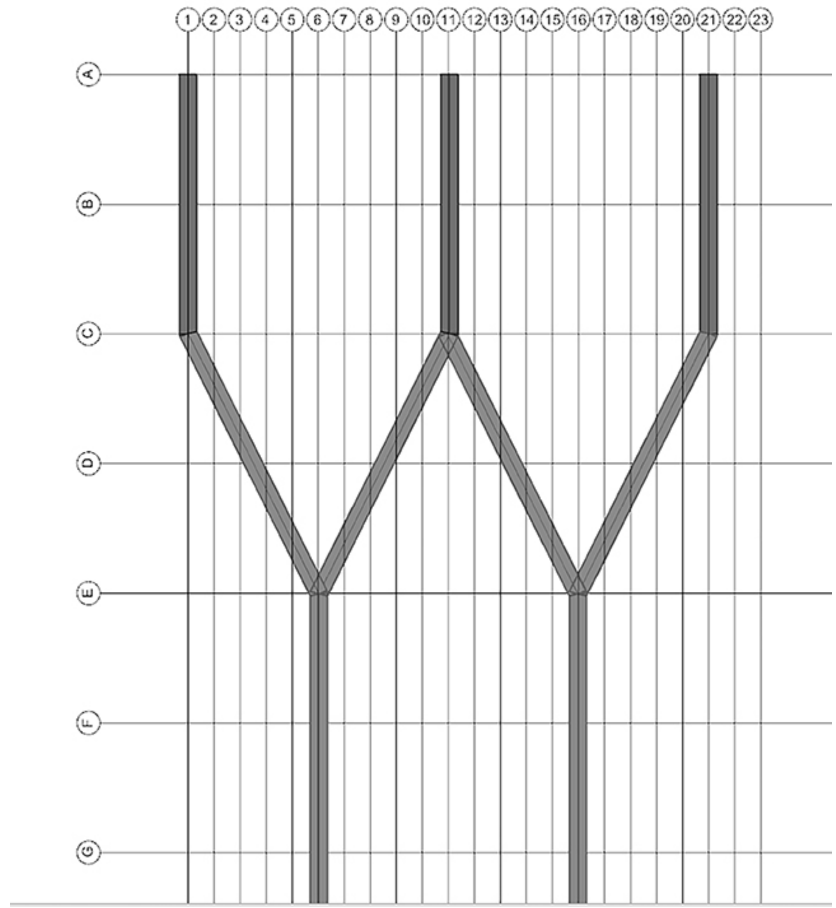
All the components follow the mathematical models provided in Grasshopper.

Design Process

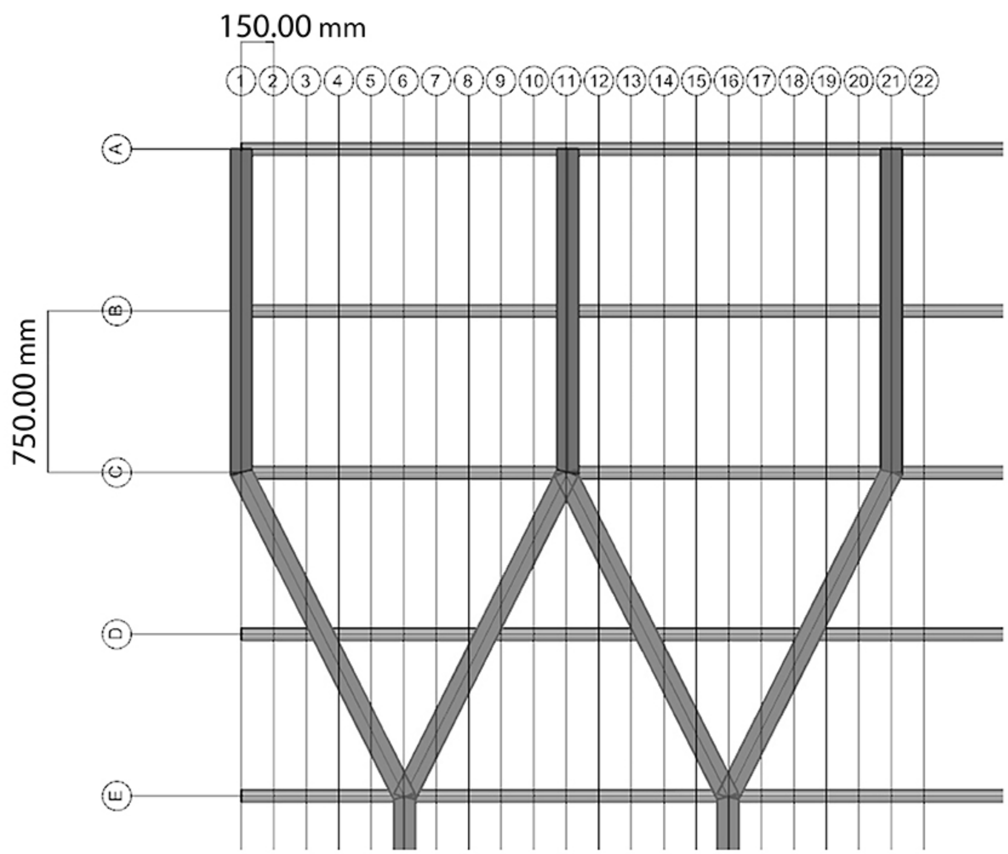
Development and design of the structure of the facade

The honeycomb structure of the facade system, supporting the units.

Main structure of the facade

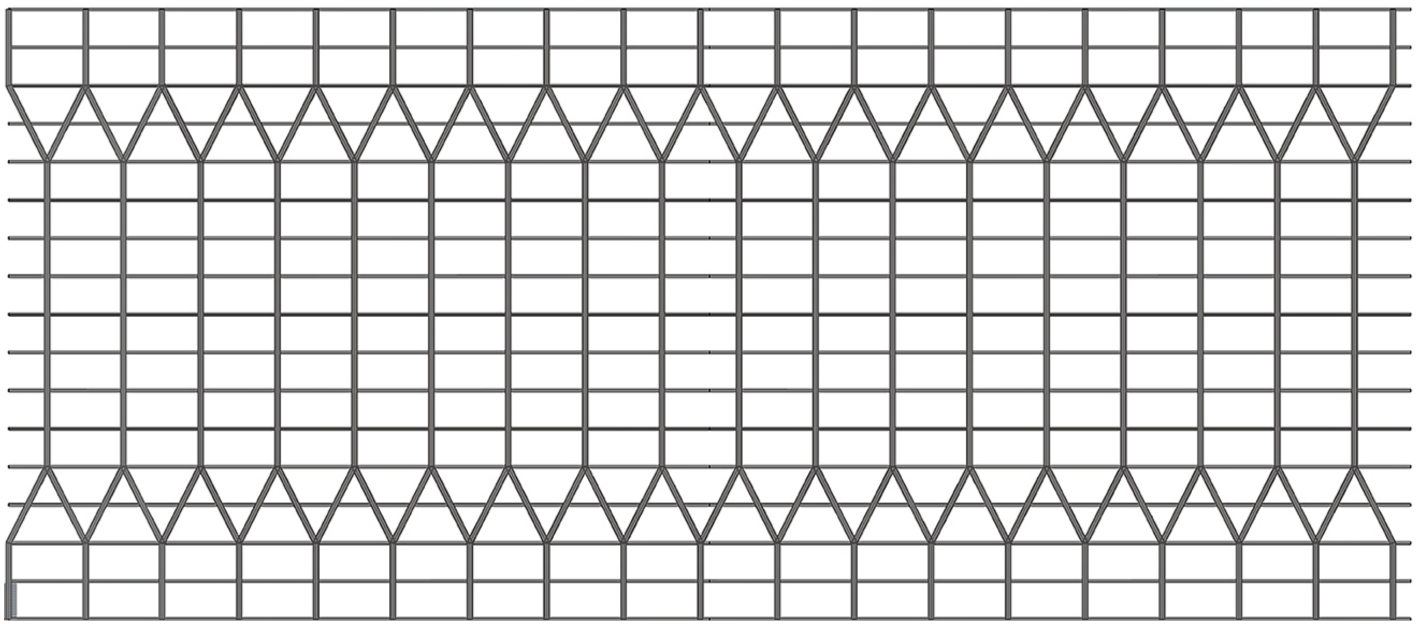


connecting vertical structures with horizontal structures

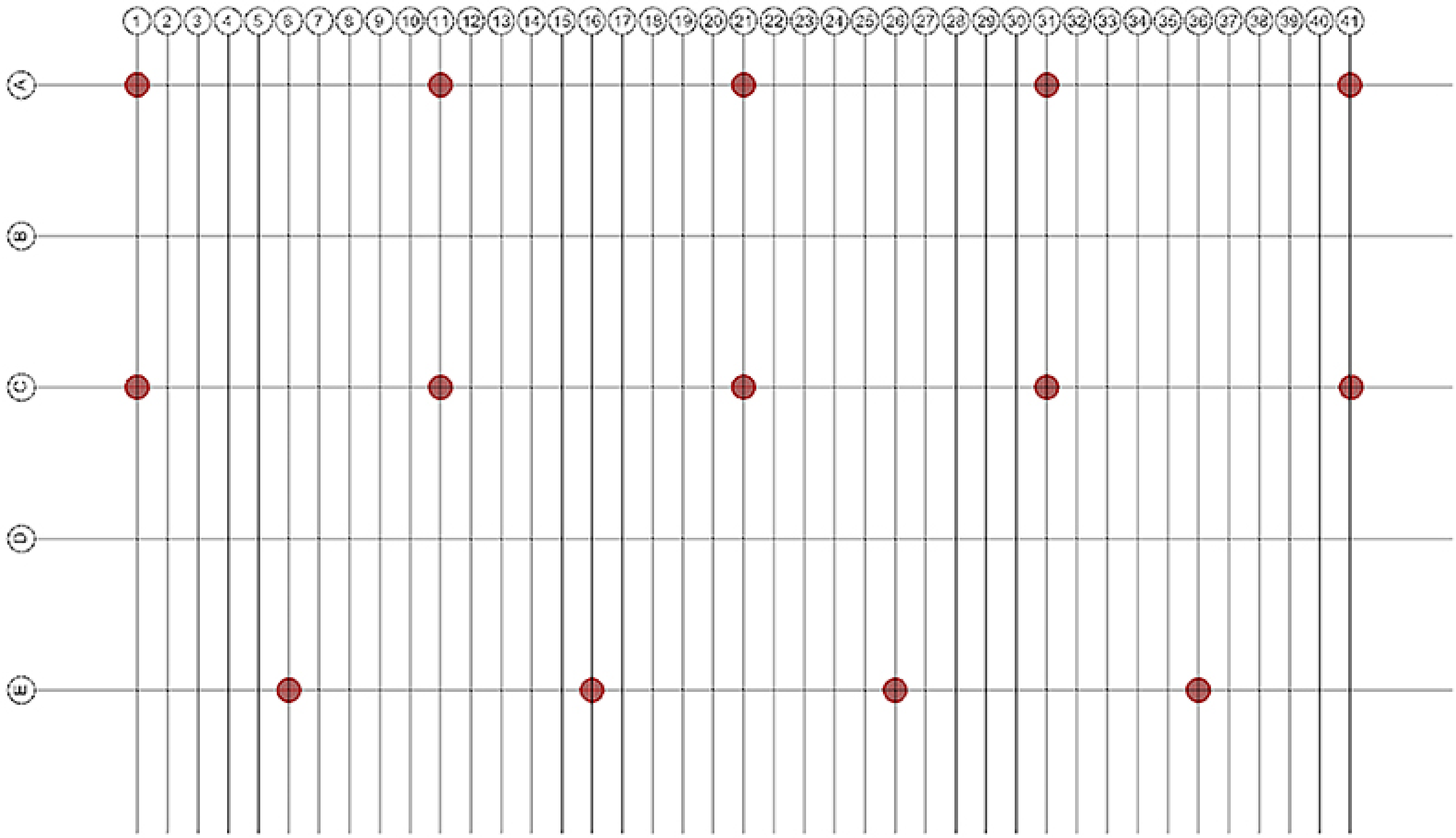


Structure

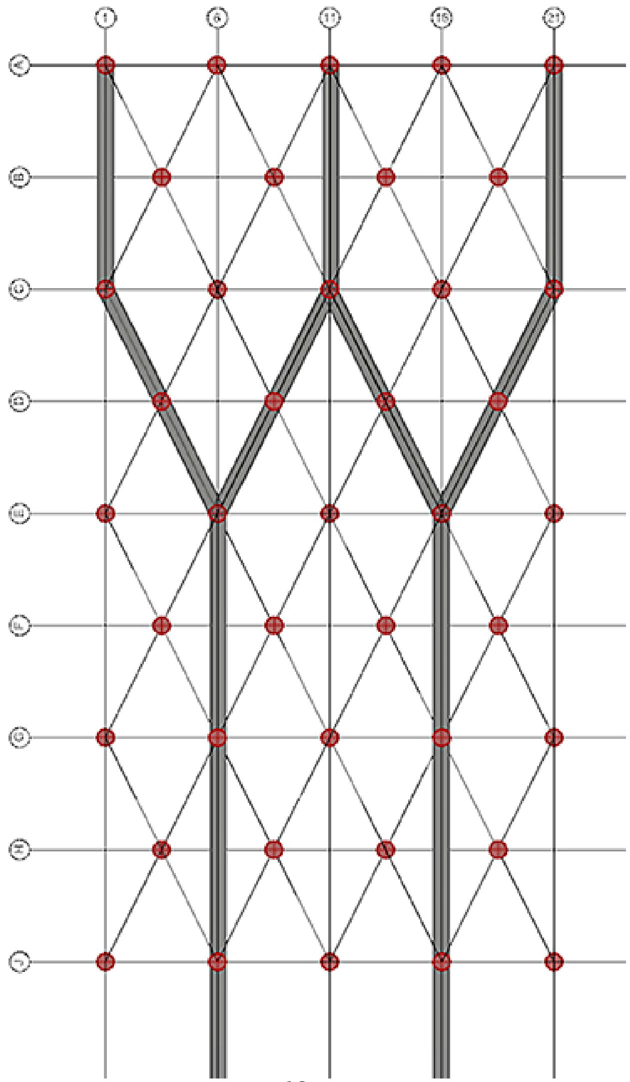
The horizontal and vertical structure of the facade.



Developing support points

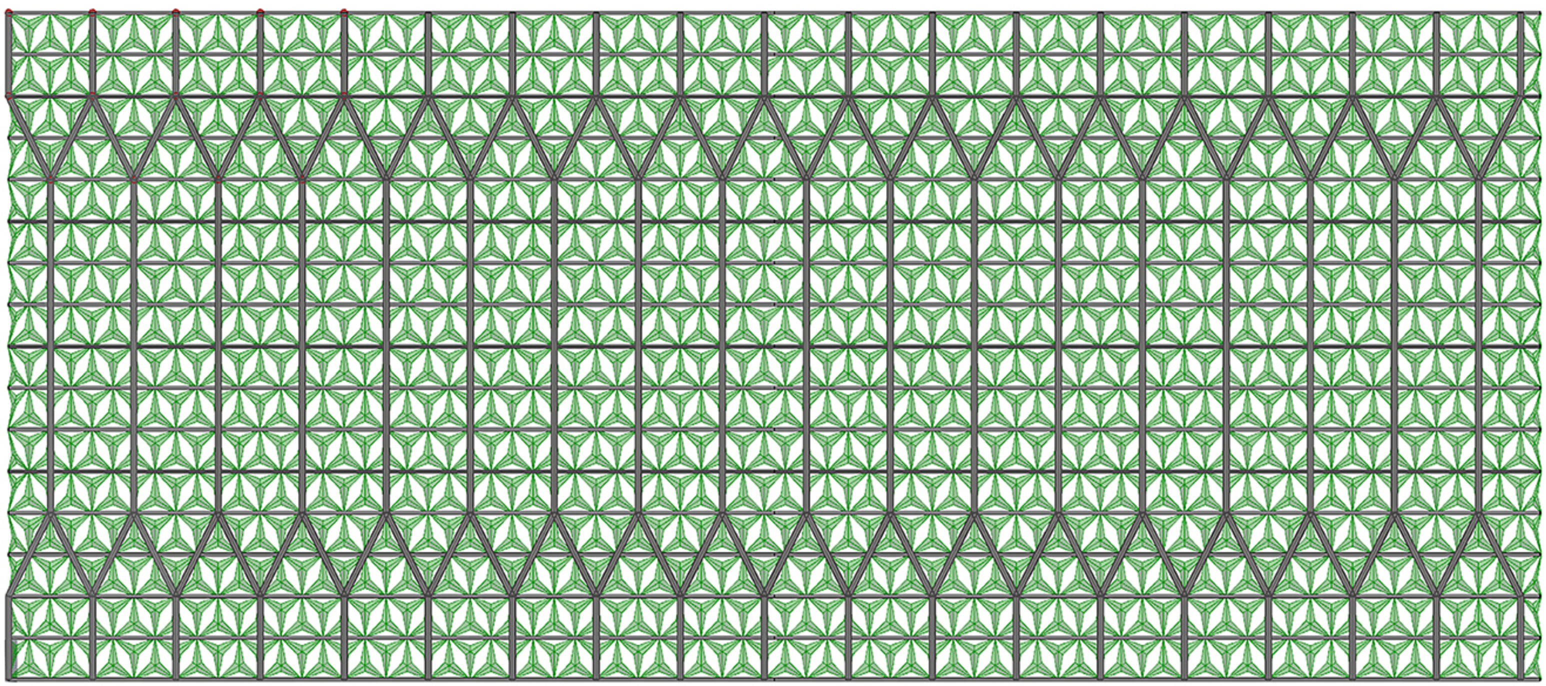


Support points



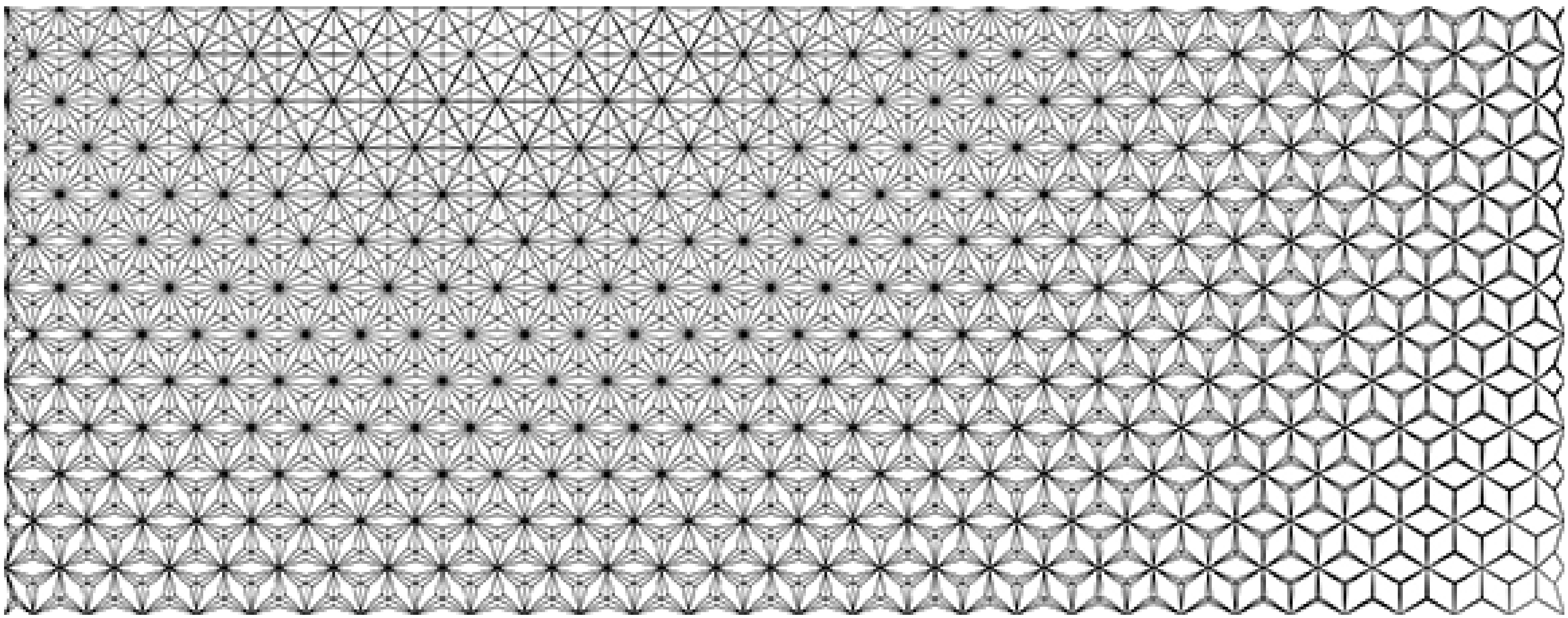
Structure

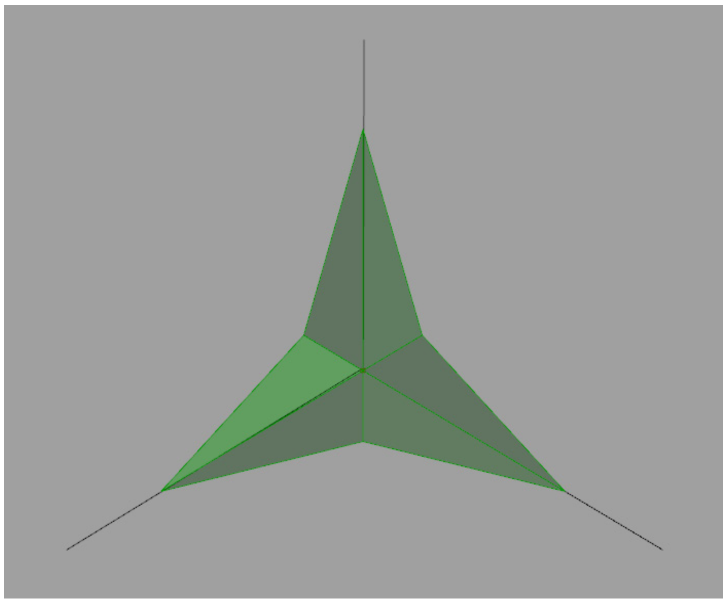
The structure is supporting the units.

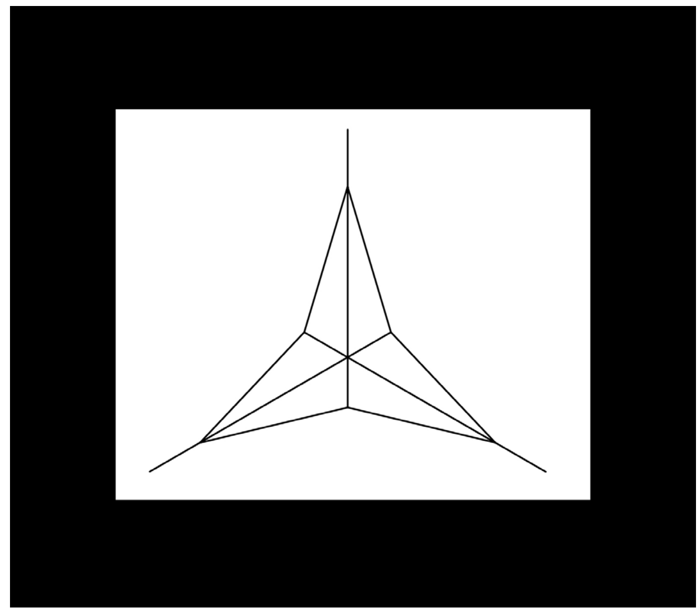


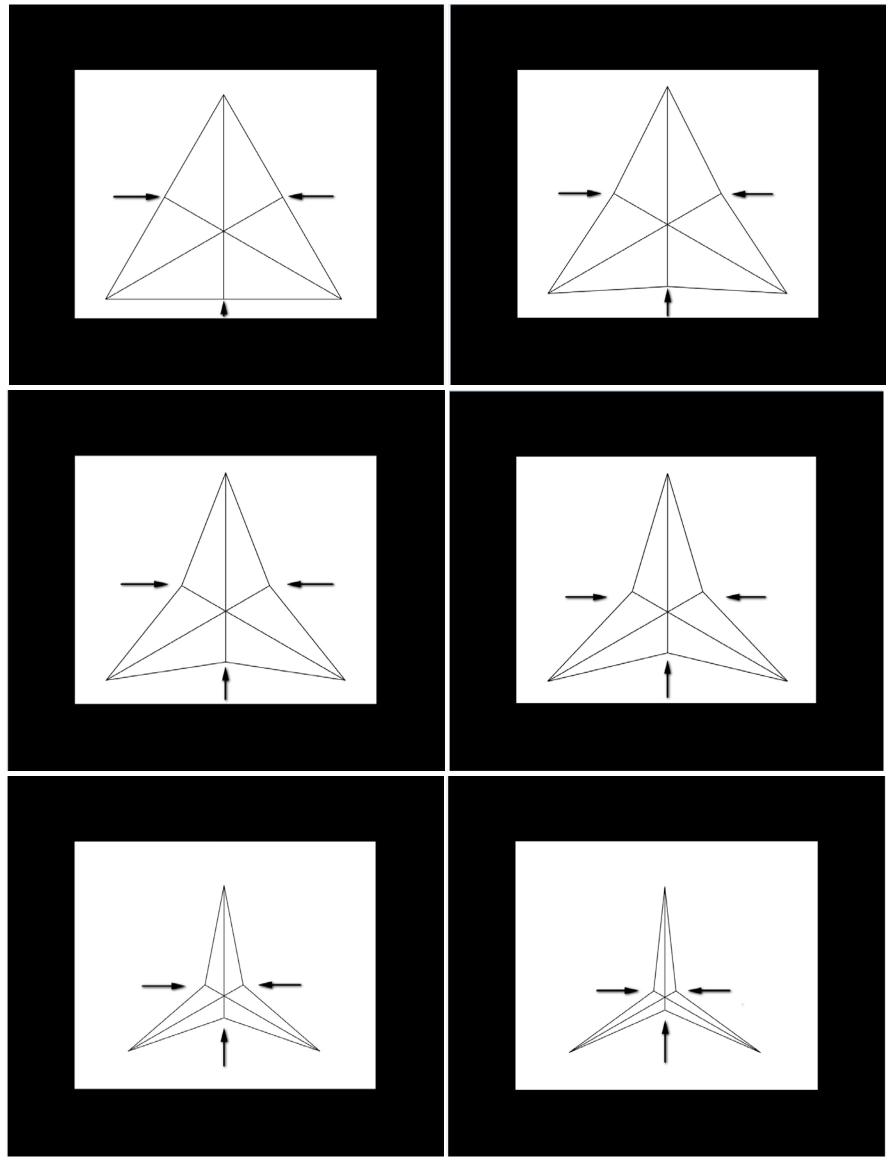
Design Process

Developing one unit from inside of the facade as proof of concept









Computation process in Grasshopper

Following the cosine relationship imposed by the spine and rotational angle of the wing, further trigonometrical relationship are also found based on this movement. Thus, rather than writing in fixed measurements, parametric equations were written as formula which slots in to satisfy those relations, making all lengths become scalable elements.

Lh = h * (1-NCP)

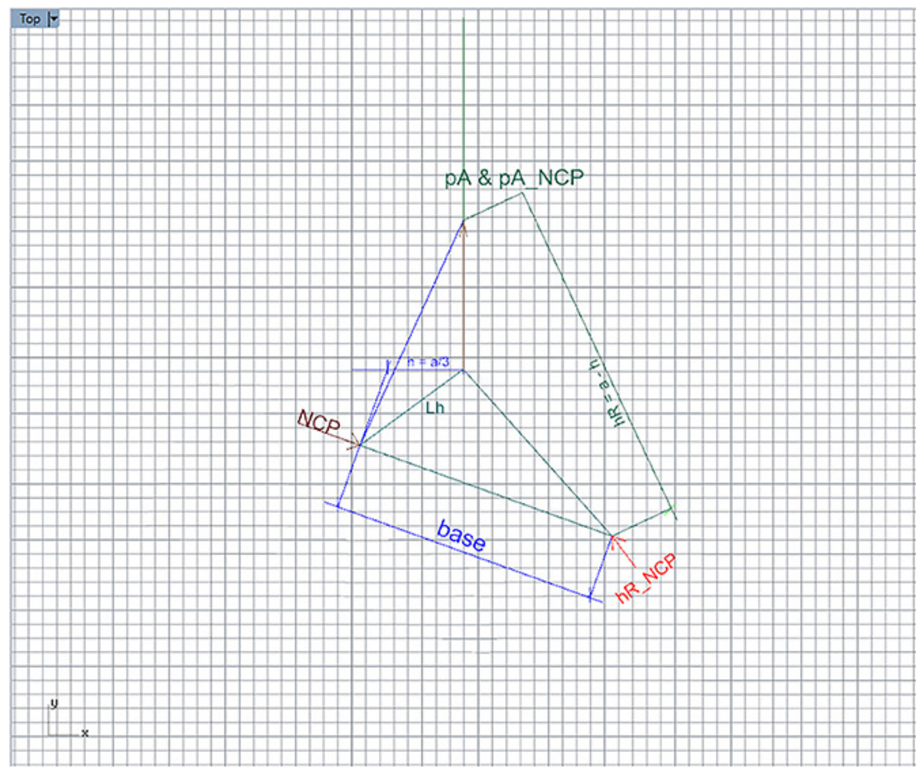
pA= sqrt ((h^2) - (Lh^2))

pA_NCP = PA/a

hR_base = sqrt ((hR^2) - (PA^2))

hR_NCP = hR_base / hR

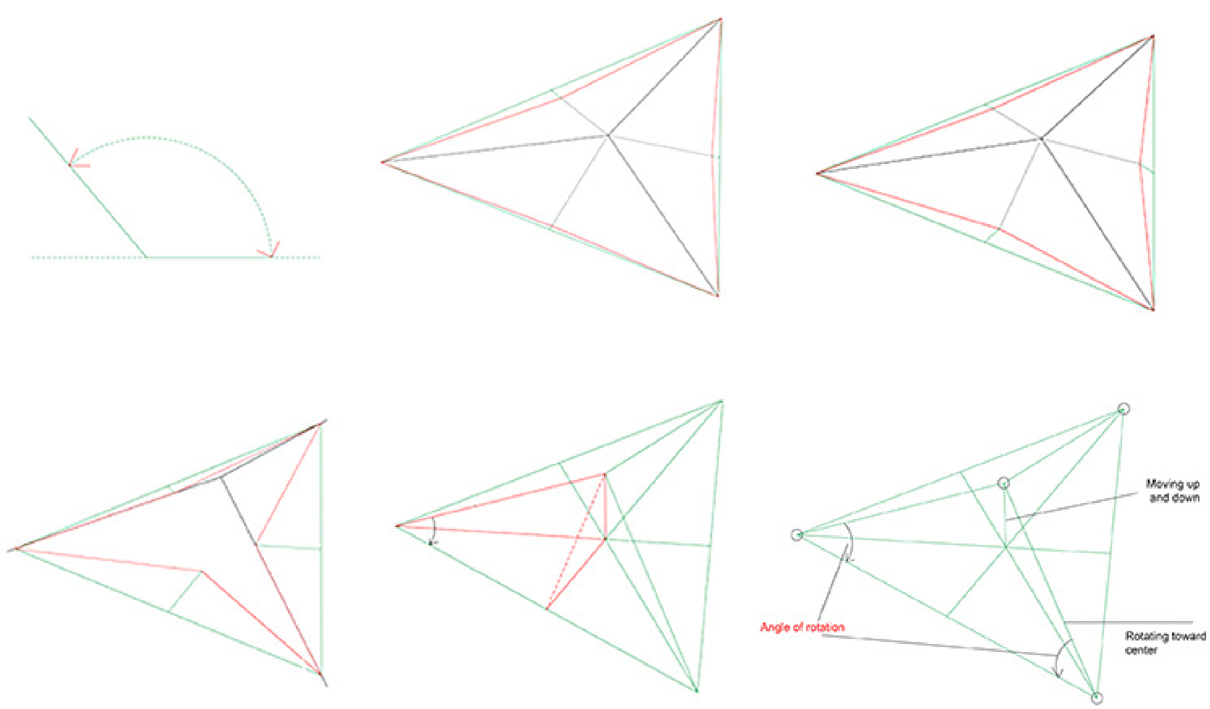
NCP = The variable controlled by the user or eventually Grasshopper
a= user defined or (sqrt(3)*base)
base= user defined or a/sqrt(3)/2

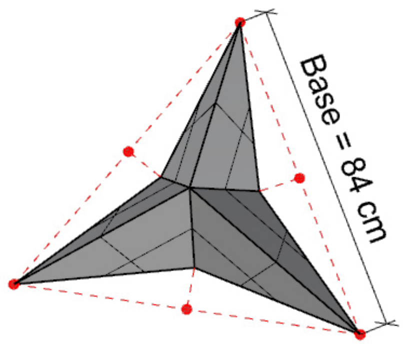


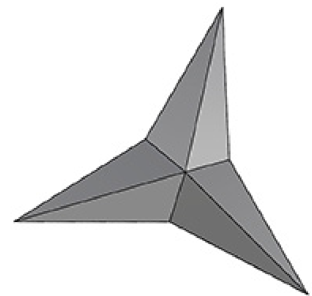
The Mathematical Relationship between Joint Elements:

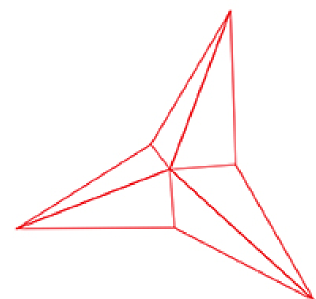
$a2 = b2 + c2 - 2bc \cos A$

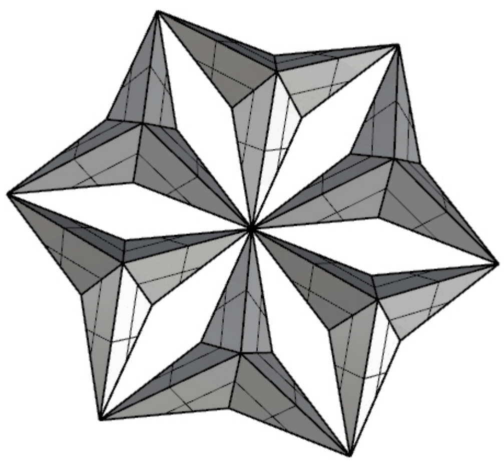
These components are then nested into the family of main panel, where its spine movement is driven by the rotation angle of the wings.

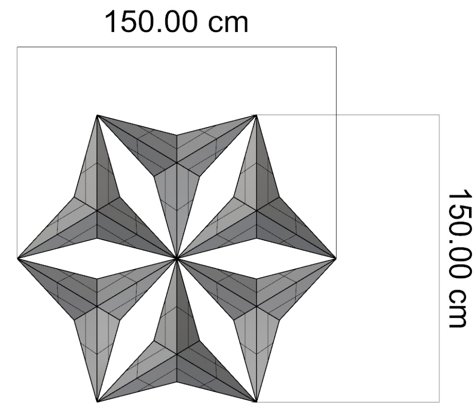


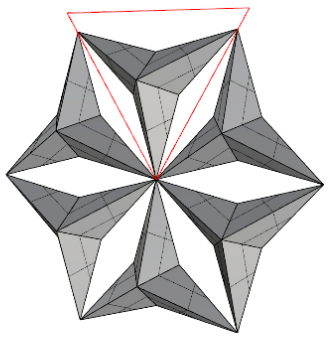

Base = 84 cm

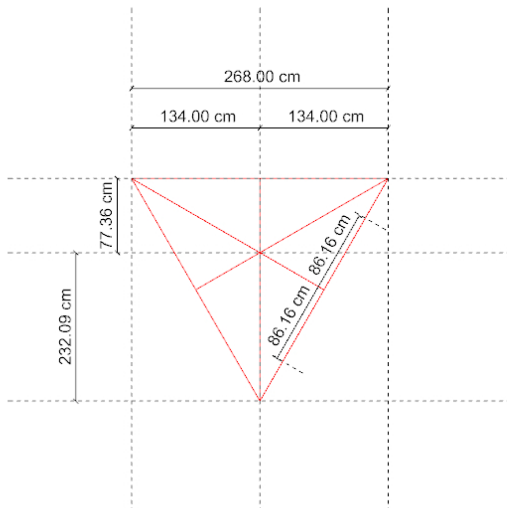


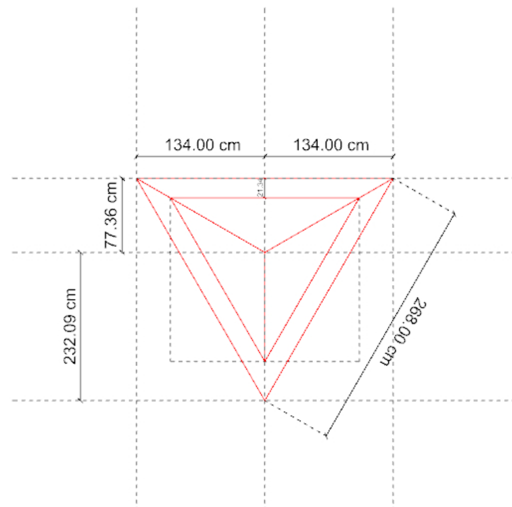





150.00 cm
150.00 cm

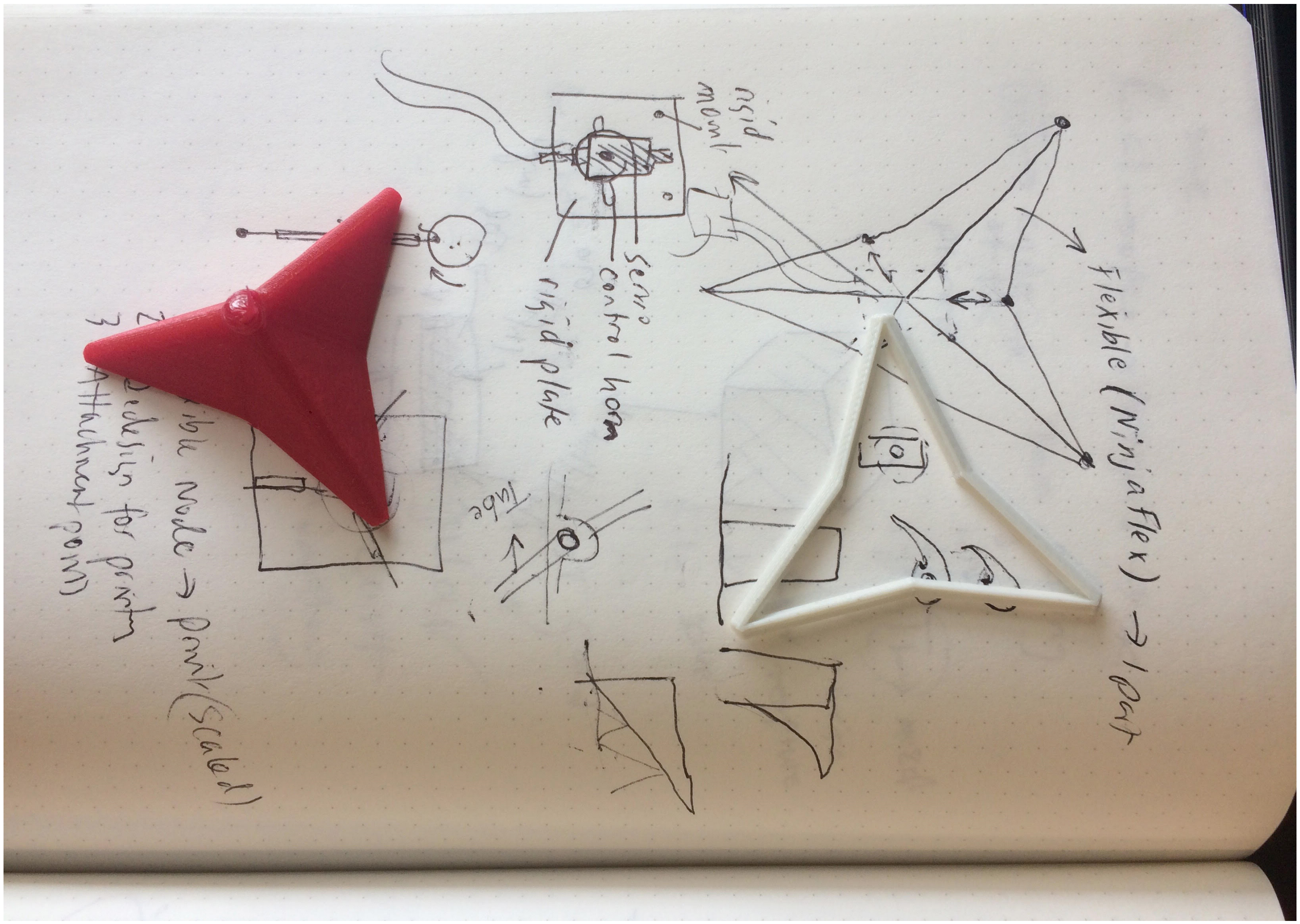


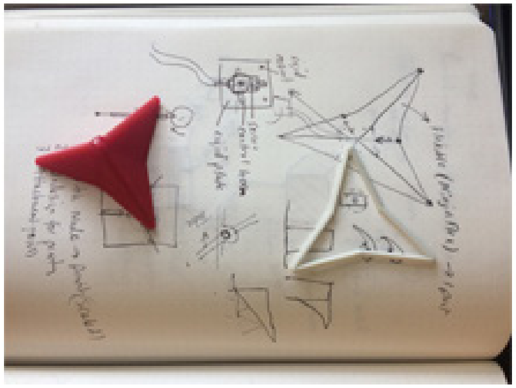
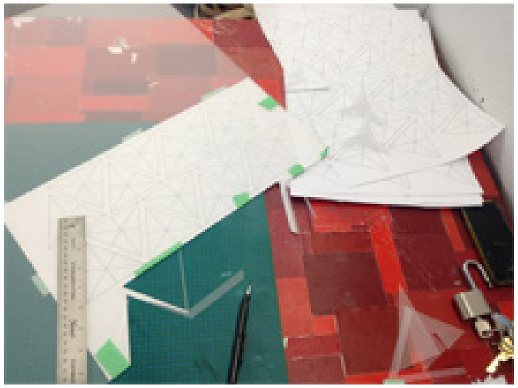
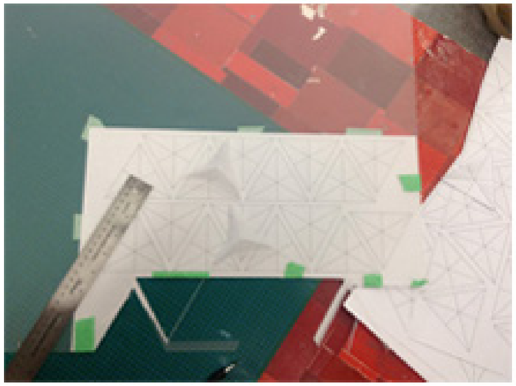
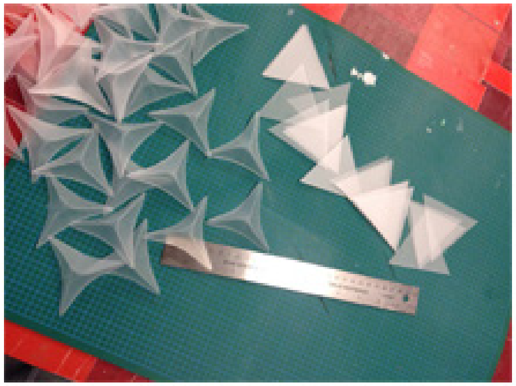


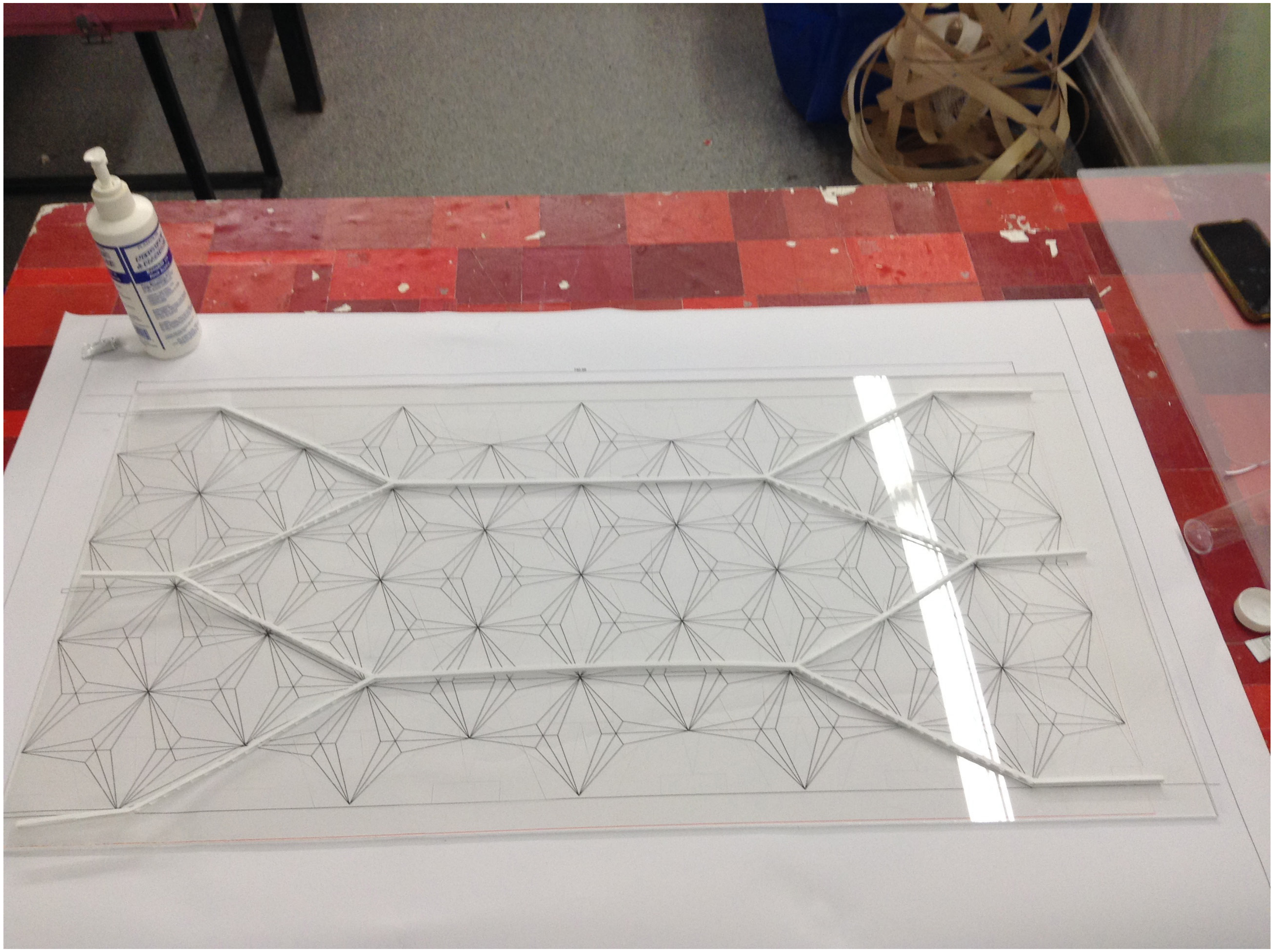


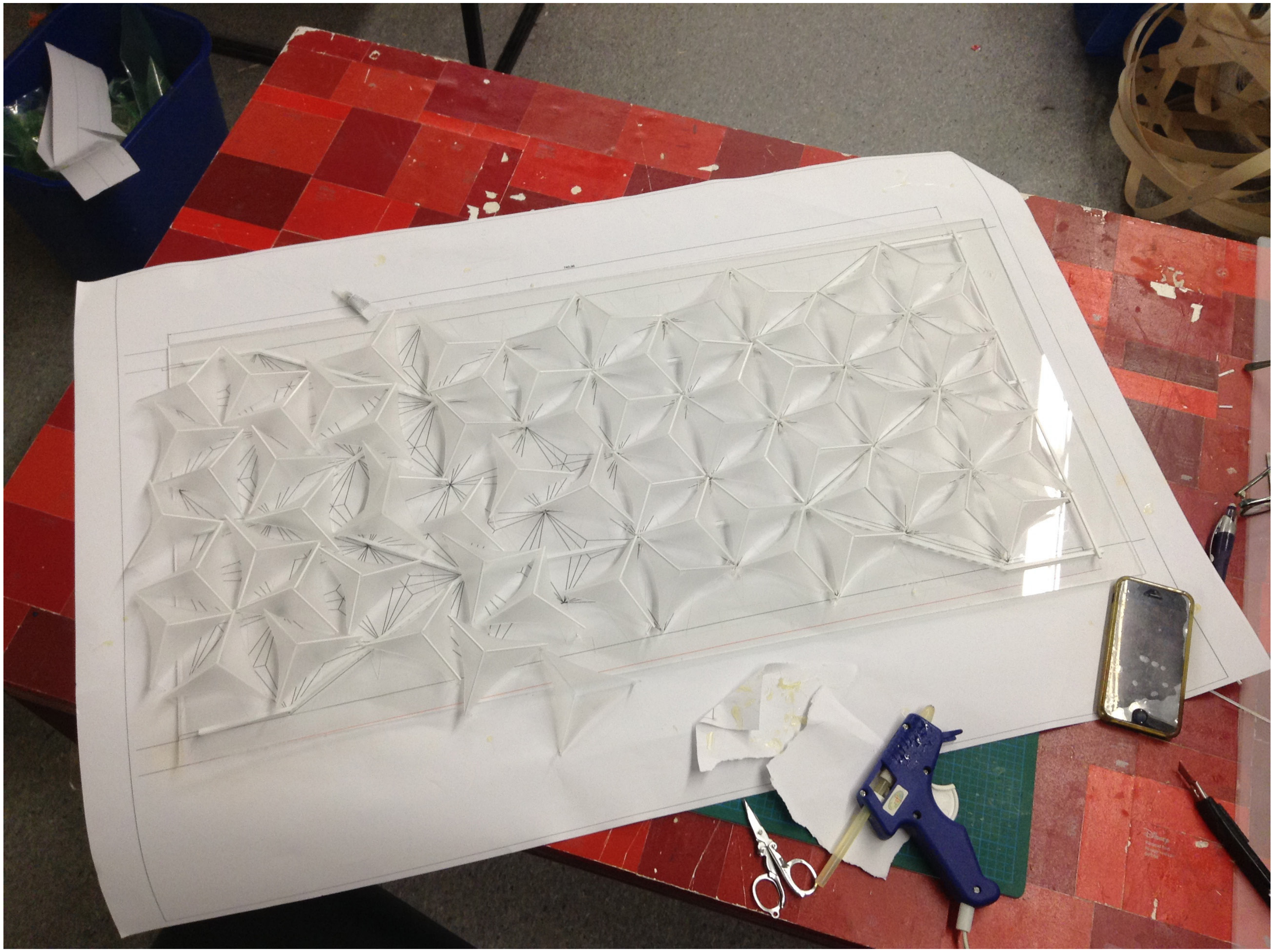
The smaller triangle is a scaled down version of the main triangle, forming the base of the panel.

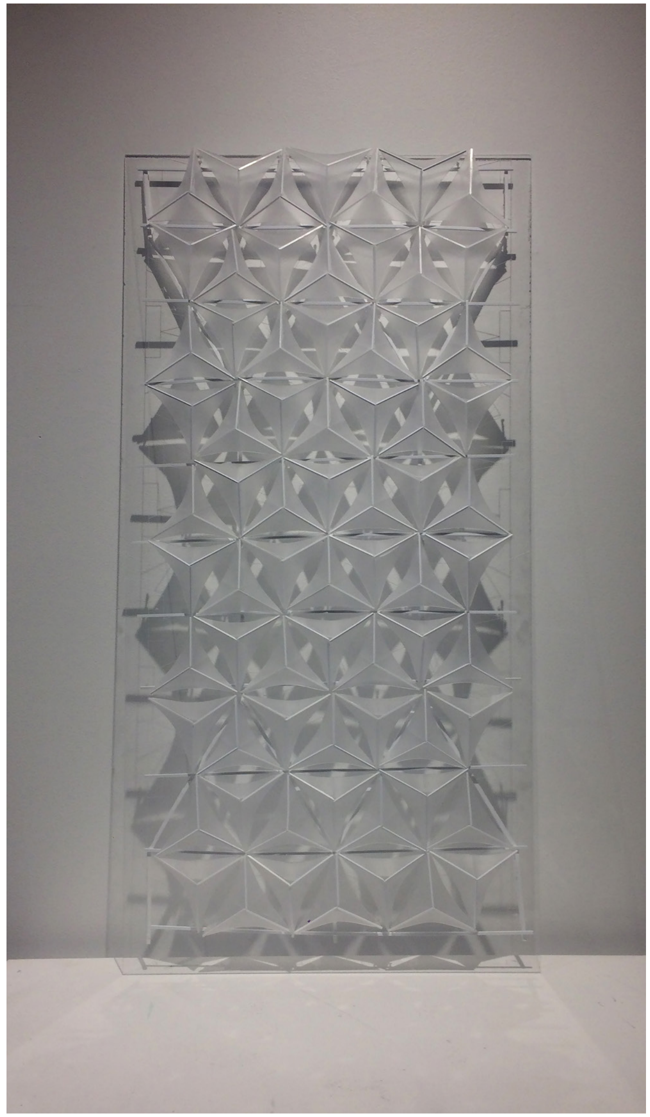
Making the First Model





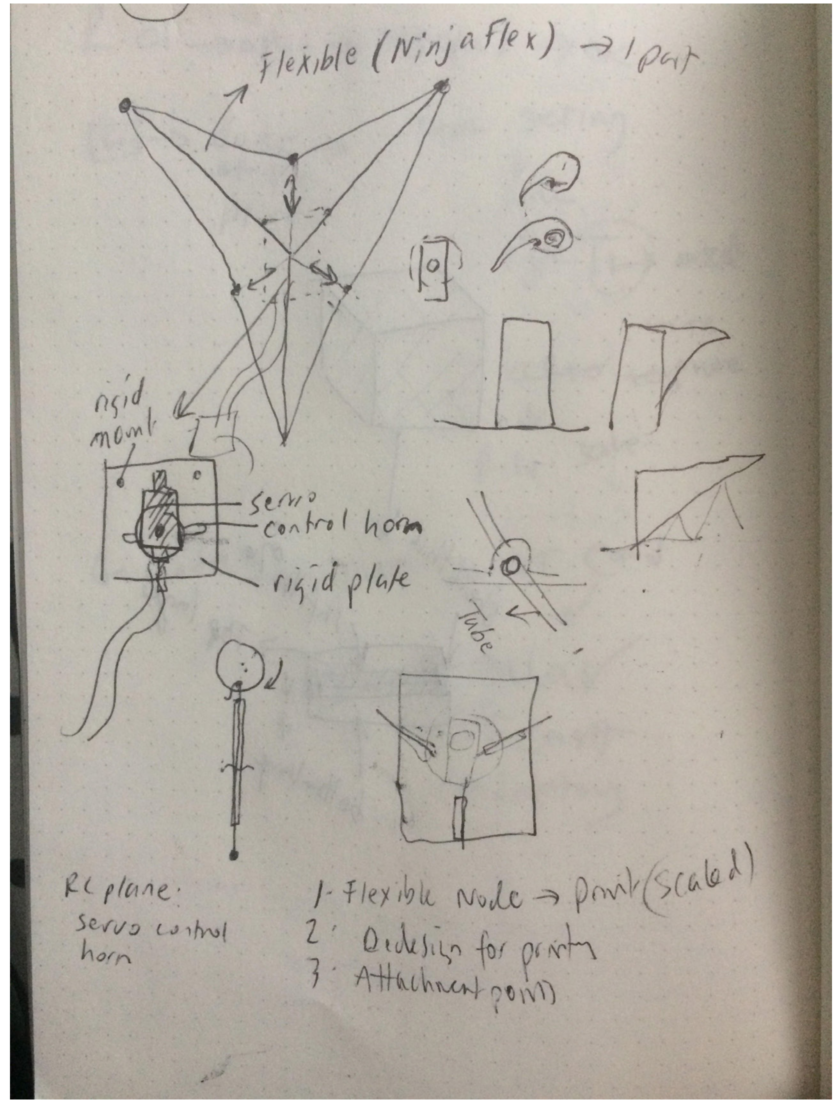


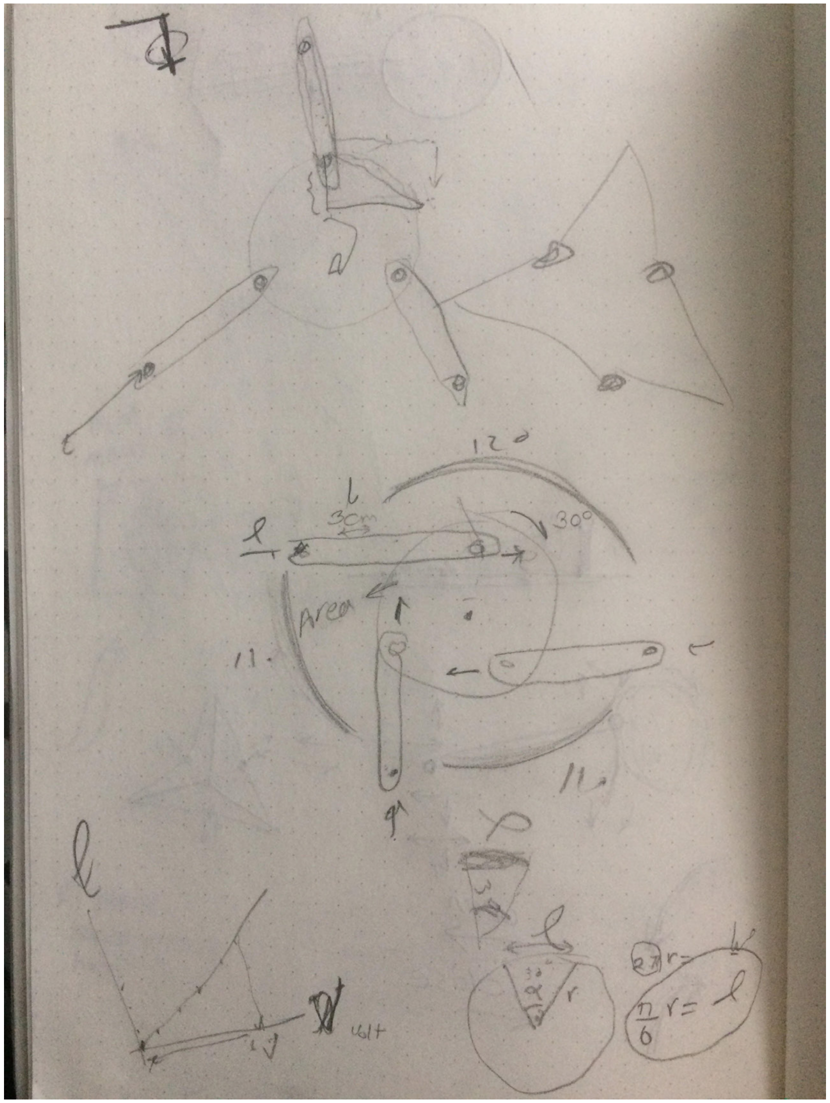


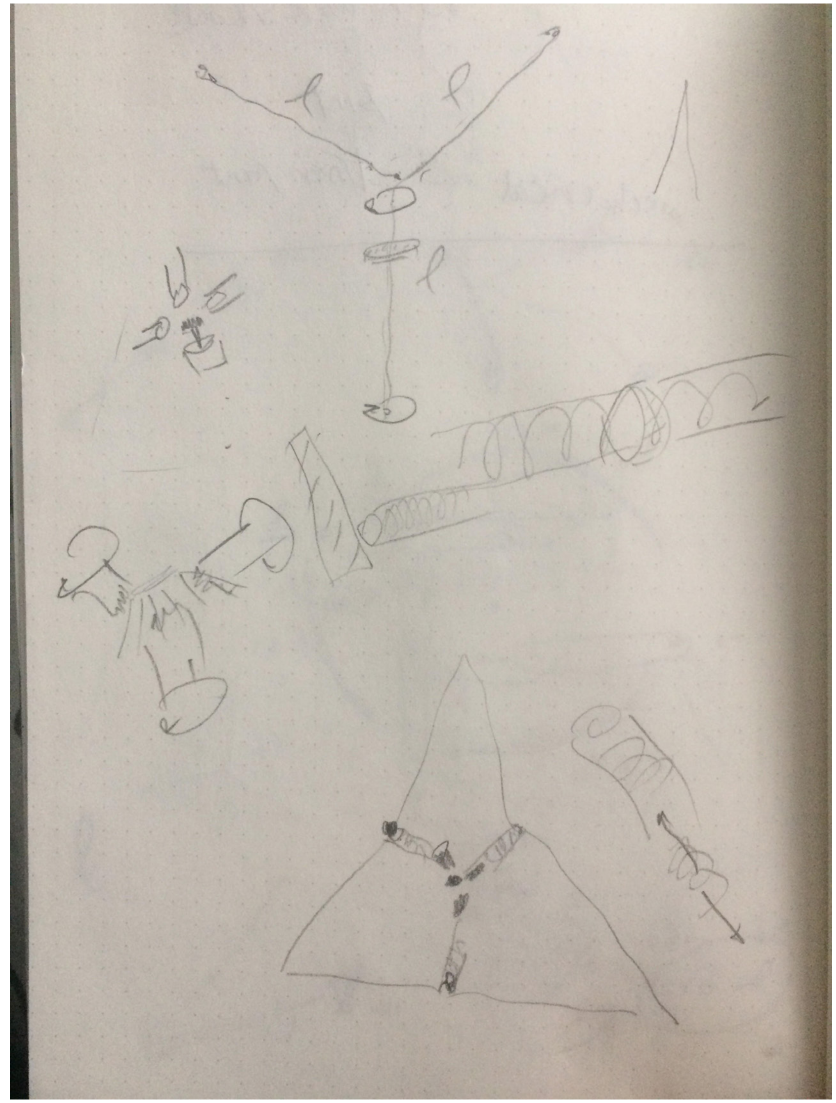
Design Process
Prototyping

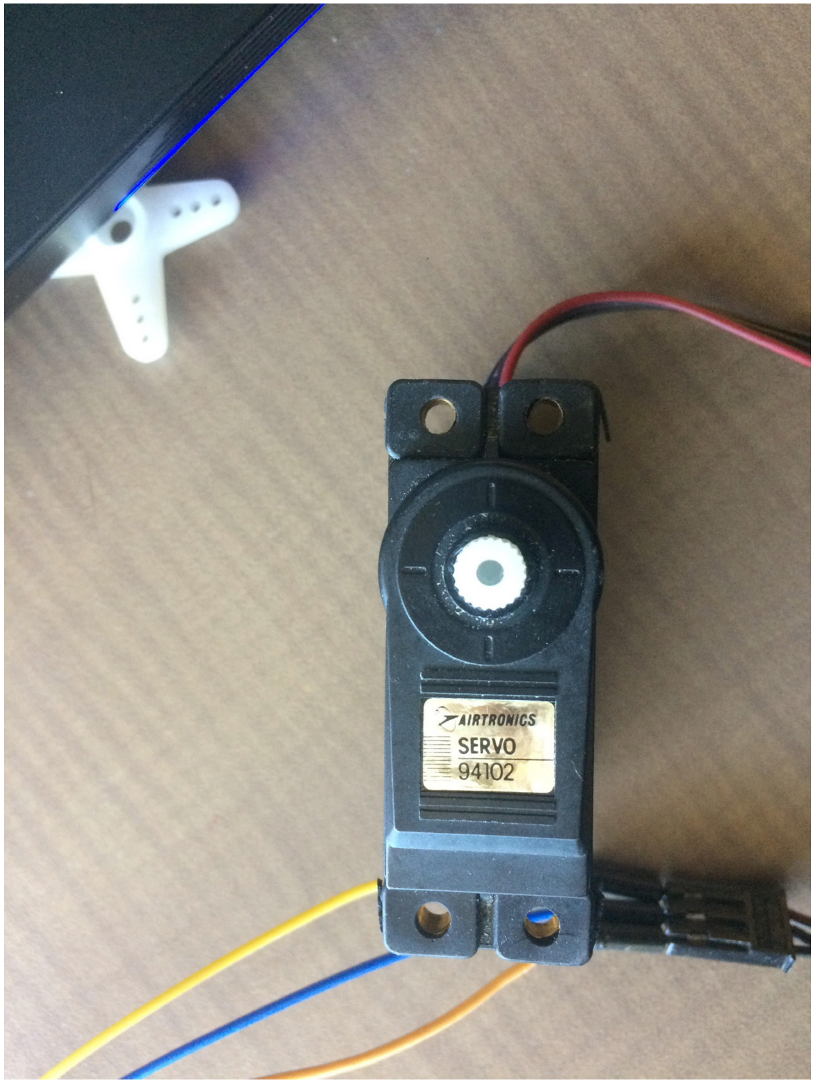
The design of the prototype

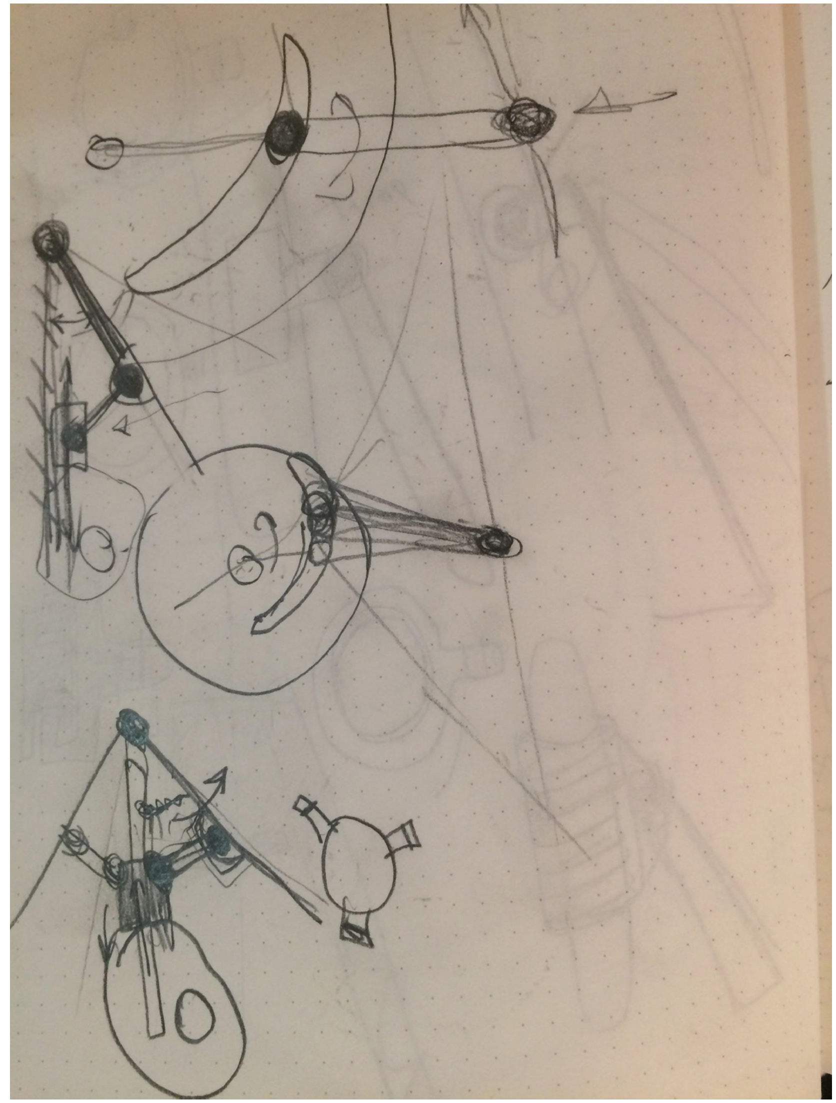
Doing sketches to find out the relation between combined components.

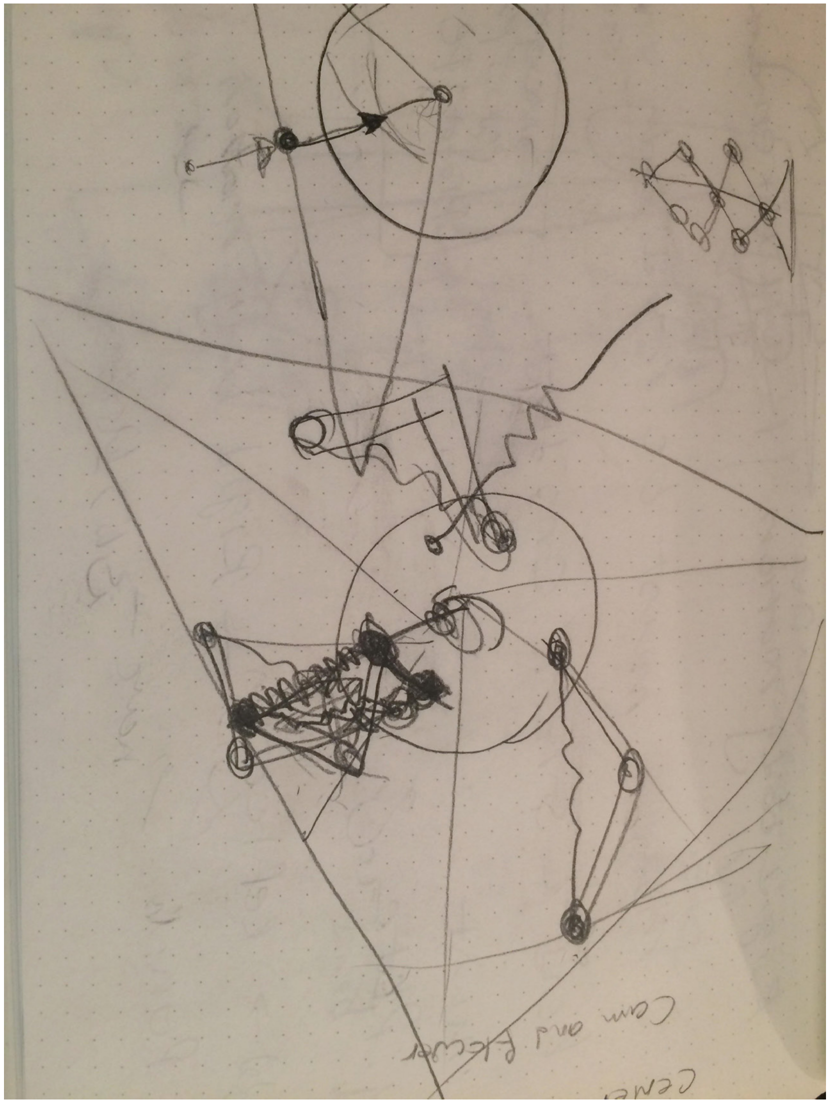


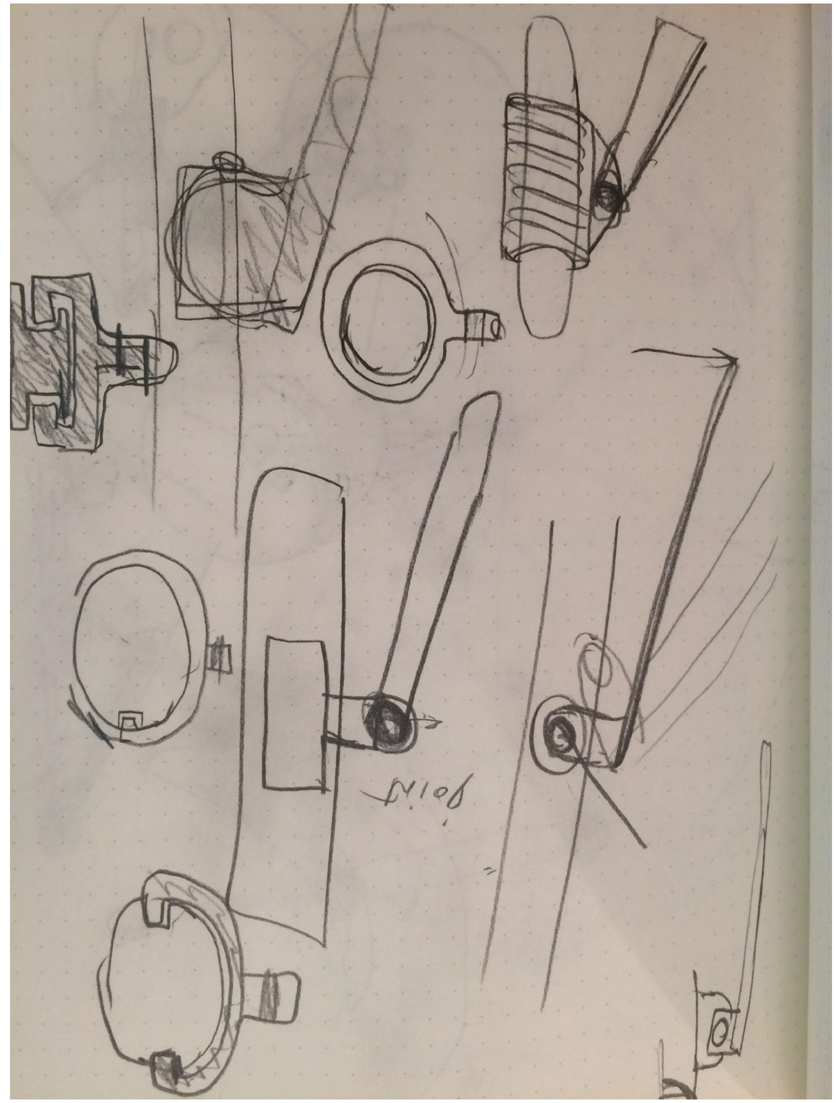

120
l
30cm
30°
Area

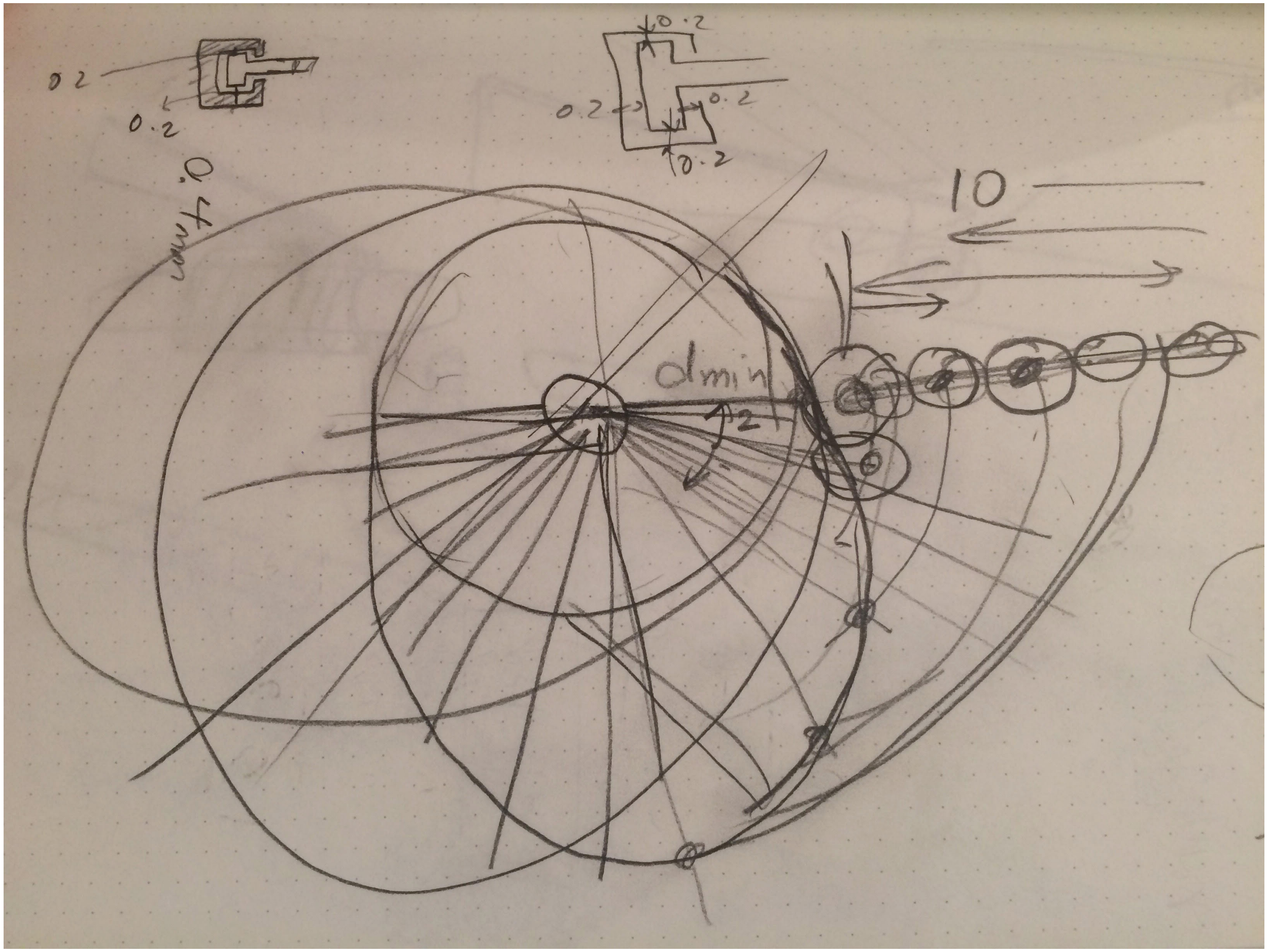


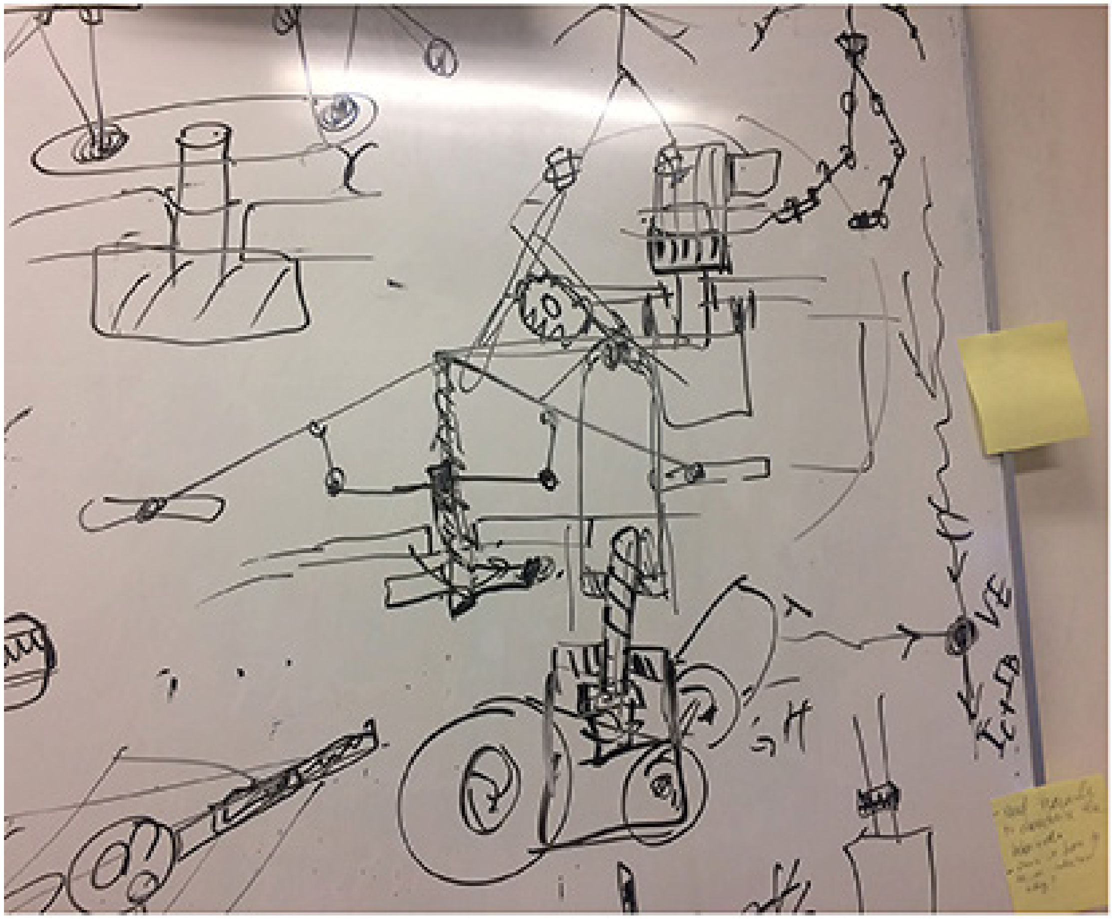

AIRTRONICS
SERVO
94102

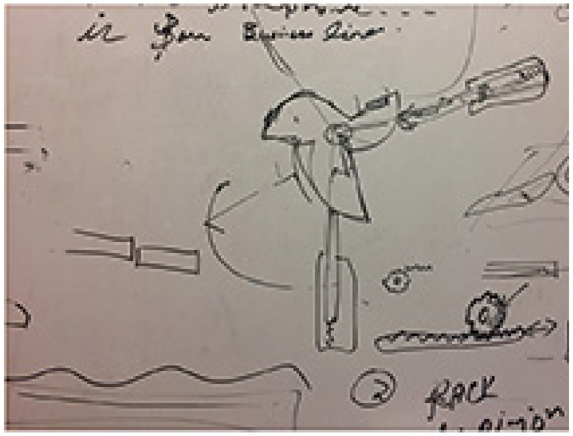


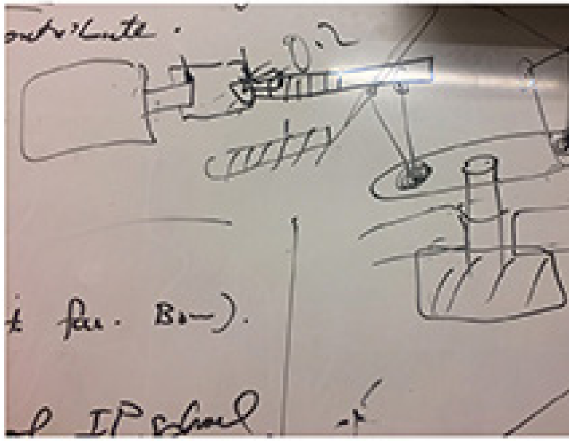




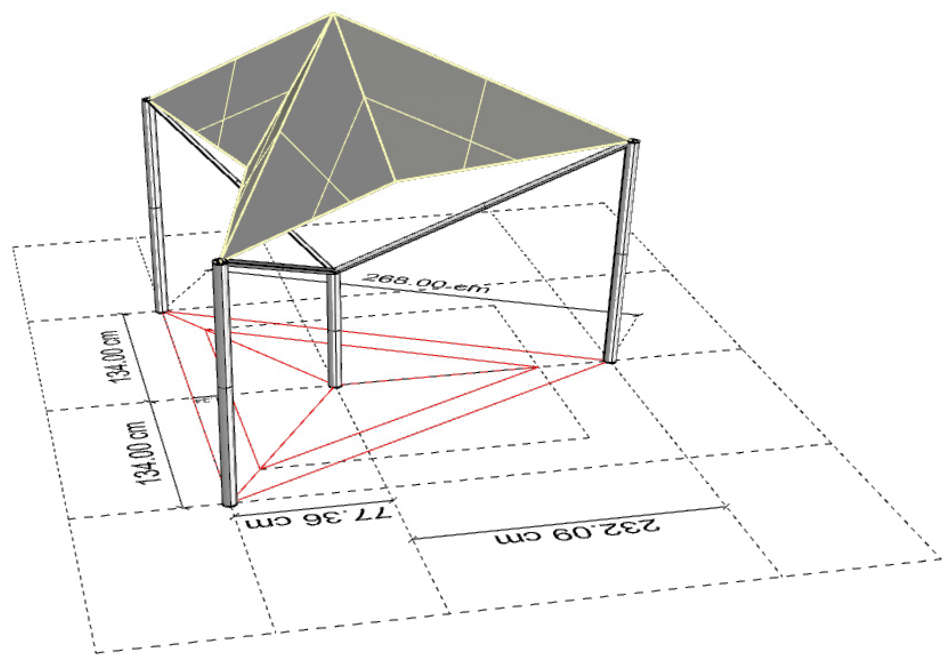

0.2
0.2
0.4mm
0.2
0.2
0.2
0.2
10
dmin
2







In the final model there will be an extension panel which the main panel is nested into. Their parameters coexisted within one another and forms inter-relating chain effects.

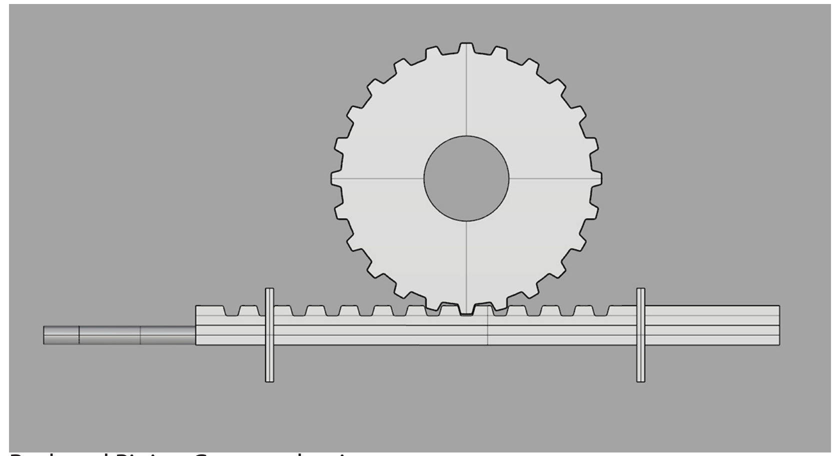


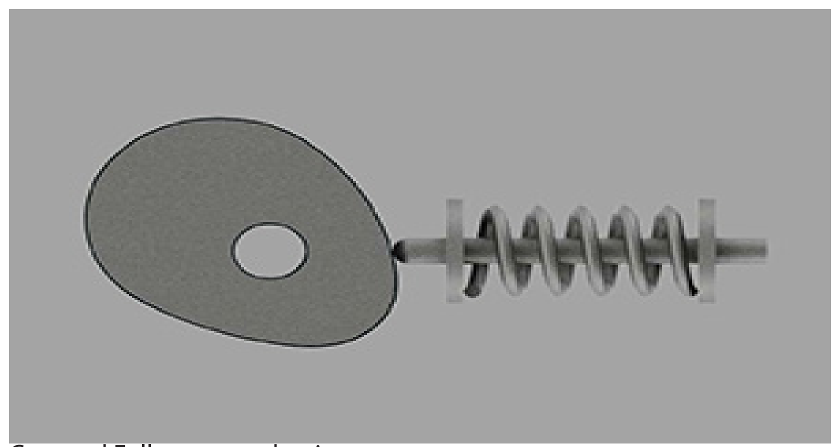
Prototyping design alternatives

Mechatronic integration of the system

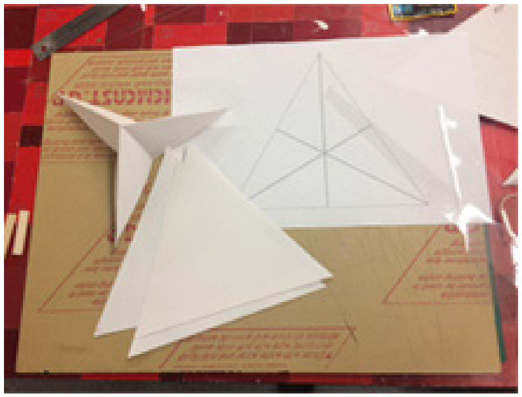
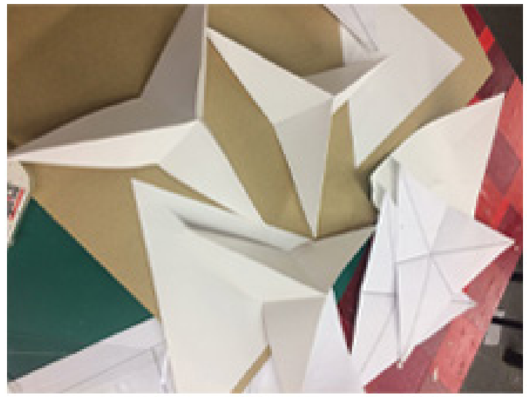
Building the mechatronic integration of the design. Two motion transformation mechanisms are investigated, Rack an Piion Gear mechanism, and Cam and Followers mechanism, to transform the rotary motion of the server into linear motion required for the movement. Both mechanisms are designed to compel the mo-tion of one part to the motion of the other by changing the type, direction, or speed of movement or the amount of torque or force available to do design.46 The mechanism are shown in the following figures provided by me in Grasshopper. The motion must be a rigid body motion (the motion that involves not only translation but rotation as well) whereas the bend and stretch trough the movement must be considered. For the pro-pose of this thesis I am going to use the Cam and Followers mechanism since first, the motion in Rack and Pinion Gear mechanism is a reflection motion and it is nota rigid body motion.

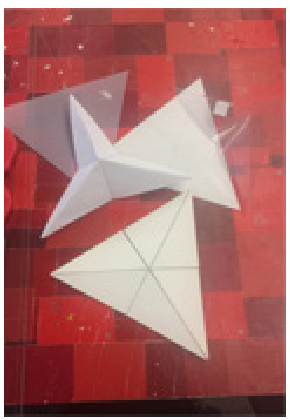
Second, I can manage the shape of the Cam to get the linear motion I need from 3 different directions.

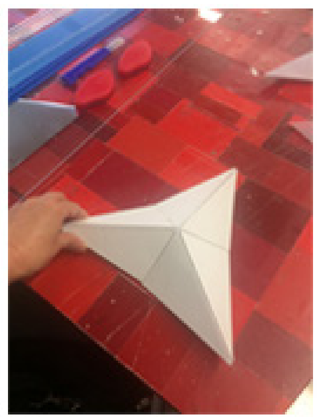
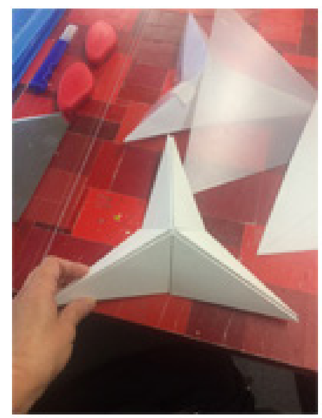

Rack and Pinion Gear mechanism

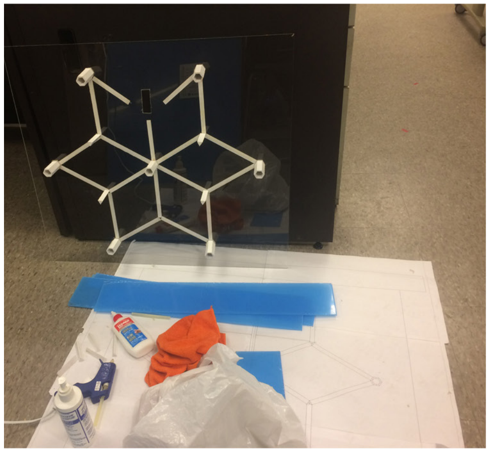
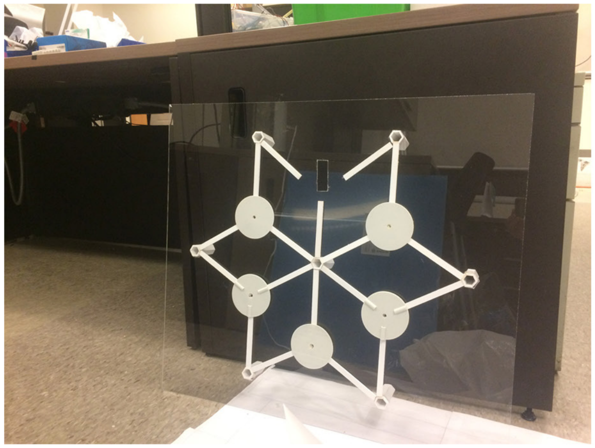

Cam and Followers mechanism

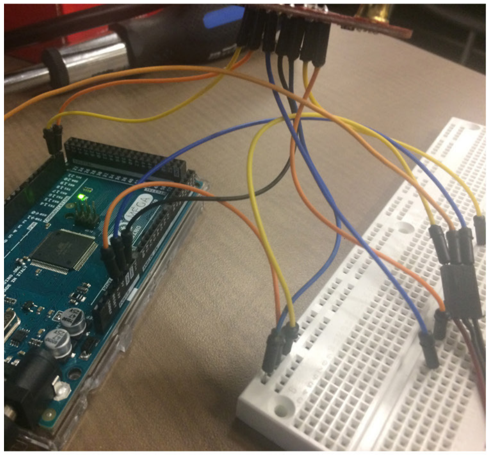
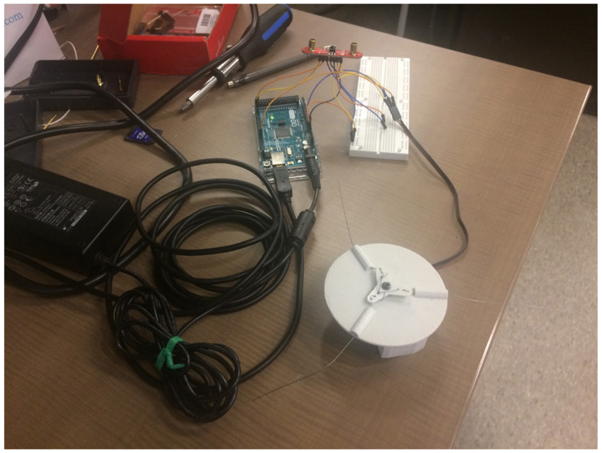
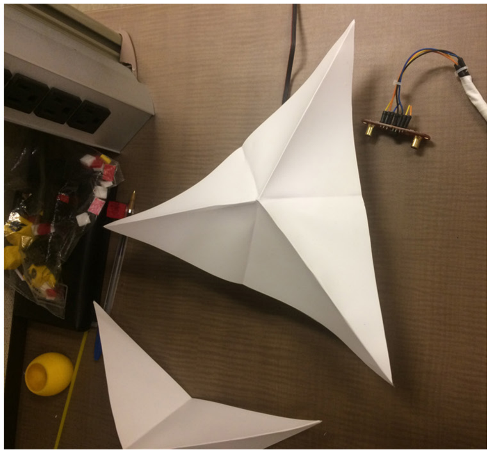
Mechanical and Computer System

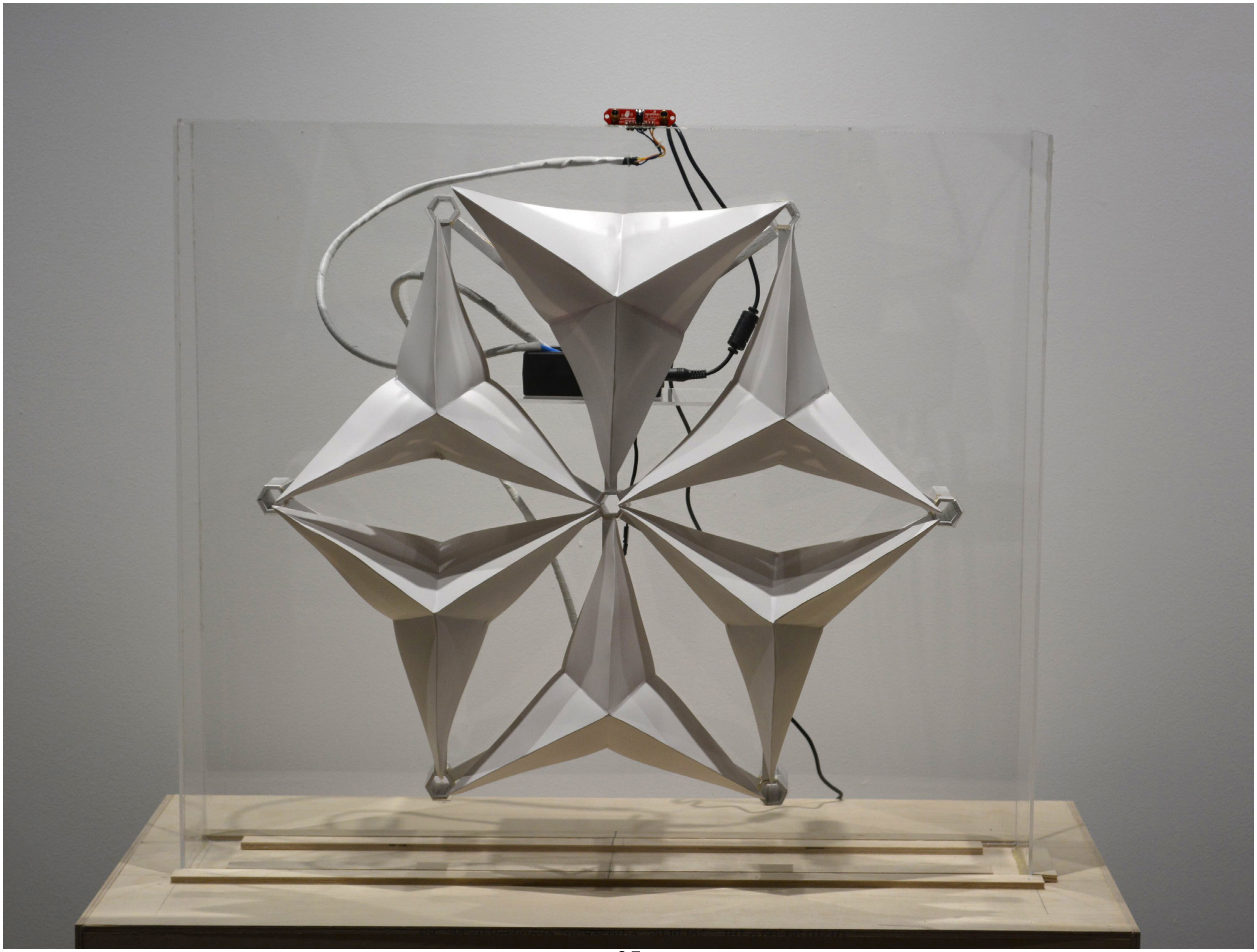







Arduino Sketches

Final.ino ReadMe.adoc

```
// --------------------------------------------------------------------------
// Title:         Tella's Final Thesis Prototype Code
// Description:   Arduino measures hand distance from sensor using I2C
//                protocol. This measurement is mapped into servo angle to
//                move, in order to animate facade cell prototype.
// Notes:
//
// Author:        Tella
// Date:          April 19, 2017 @ 7:05 PM
// --------------------------------------------------------------------------

// --------------------------------------------------------------------------
// DEPENDENCIES
// --------------------------------------------------------------------------
#include "Wire.h"          // For I2C communication with gesture sensor.
#include "Servo.h"         // For generation of PWM timing signal for motor.
#include "ZX_Sensor.h"     // Gesture sensor library.
// --------------------------------------------------------------------------

// --------------------------------------------------------------------------
// DEFINITIONS
// --------------------------------------------------------------------------
// Constants
const int MAX_SERVO_ANGLE = 80;    // Maximum servo should travel.
const int MIN_SERVO_ANGLE = 30;     // Minimum servo should travel.

const int MAX_HAND_POS = 240;      // Right hand position output.
const int MIN_HAND_POS = 0;        // Left hand position output.

const int MIN_HAND_DISTANCE = 0;   // Min distance for hand to be from sensor.
const int MAX_HAND_DISTANCE = 240; // Max distance for hand to be from sensor.

const int SCALE = (MAX_HAND_POS - MIN_HAND_POS) / (MAX_SERVO_ANGLE - MIN_SERVO_ANGLE

const int ZX_ADDR = 0x10;          // Gesture Sensor I2C Address
const int SERVO_PIN = 9;           // Servo Pin.
const int LED_PIN = 13;           // LED Pin.

// Objects
ZX_Sensor zx_sensor = ZX_Sensor(ZX_ADDR);   // Gesture
```

Final.ino | ReadMe.adoc

```
const int LED_PIN = 13;           // LED Pin.

// Objects
ZX_Sensor zx_sensor = ZX_Sensor(ZX_ADDR);   // Gesture
Servo myservo;                              // Servo

// Hand Measurements
uint8_t x_pos;
uint8_t z_pos;

// Servo Output
int servo_angle = MIN_SERVO_ANGLE;   // Servo angle to move each measurement step.
// ---------------------------------------------------------------------------

// ---------------------------------------------------------------------------
// INITIALIZATION
// ---------------------------------------------------------------------------
void setup()
{
  // This function initializes all sensors and Arduino.

  // Hardware-Related
  pinMode(LED_PIN, OUTPUT);
  digitalWrite(LED_PIN, HIGH);

  // Serial-Related
  Serial.begin(115200);
  Serial.println("Tella Thesis Prototype");

  // Initialize Gesture Sensor
  if (zx_sensor.init())
  {
      Serial.println("Sensor Okay!");
  } else {
      Serial.println("Sensor Error :(!");
  }

  // Get the model and register map of sensor (for diagnostics).
  uint8_t ver = zx_sensor.getModelVersion();
  ver = zx_sensor.getRegMapVersion();

```

Final.ino ReadMe.adoc

```
  // Get the model and register map of sensor (for diagnostics).
  uint8_t ver = zx_sensor.getModelVersion();
  ver = zx_sensor.getRegMapVersion();

  // Initialize Servo
  myservo.attach(SERVO_PIN);
  myservo.write(servo_angle);

  // Debugging LED information.
  //delay(1000);
  //digitalWrite(LED_PIN, LOW);
  //delay(1000);
}
// ---------------------------------------------------------------------------

// ---------------------------------------------------------------------------
// MAIN LOOP
// ---------------------------------------------------------------------------
void loop()
{
  // Main program loop.

  // Acquire sensor measurement.
  if (zx_sensor.positionAvailable())
  {
    // Get the X and Z measurements.
    x_pos = zx_sensor.readX();
    z_pos = zx_sensor.readZ();

    // Check that the measurements were okay.
    if ((x_pos != ZX_ERROR) && (z_pos != ZX_ERROR))
    {

      // Print the measurements for debugging.
      Serial.print("X\t");
      Serial.print(x_pos);
      Serial.print("\t\tZ:\t");
      Serial.print(z_pos);

      // We only want to have this move if hand is within range.
      if ((z_pos >= MIN_HAND_DISTANCE) && (z_pos <= MAX_HAND_DISTANCE))
```

Final.ino | ReadMe.adoc

```
      // Print the measurements for debugging.
      Serial.print("X\t");
      Serial.print(x_pos);
      Serial.print("\t\tZ:\t");
      Serial.print(z_pos);

      // We only want to have this move if hand is within range.
      if ((z_pos >= MIN_HAND_DISTANCE) && (z_pos <= MAX_HAND_DISTANCE))
      {
          // Set visual indicator.
          digitalWrite(LED_PIN, HIGH);

          // Calculate the servo angle to set.
          servo_angle = (z_pos / SCALE) + MIN_SERVO_ANGLE;

          Serial.print("\t\tAngle:\t");
          Serial.print(servo_angle);

          // Write new servo angle to servo.
          myservo.write(servo_angle);
      } else {
        // Set LED off.
          digitalWrite(LED_PIN, LOW);
      }

      // Finish printing.
      Serial.println("");
    } else {
      // Set LED off.
      digitalWrite(LED_PIN, LOW);
    }
  }
}
// ------------------------------------------------------------------------

// ------------------------------------------------------------------------
// END OF FILE
// ------------------------------------------------------------------------
// ------------------------------------------------------------------------

```

Movement.ino ReadMe.adoc

```
#include <Wire.h>
#include <ZX_Sensor.h>
#include <Servo.h>

// Constants
const int ZX_ADDR = 0x10;  // ZX Sensor I2C address

// Global Variables
ZX_Sensor zx_sensor = ZX_Sensor(ZX_ADDR);
uint8_t x_pos;
uint8_t z_pos;

// Servo
Servo myservo;  // create servo object to control a servo
                // a maximum of eight servo objects can be created

int pos = 0;    // variable to store the servo position

void setup() {

  uint8_t ver;

  // Initialize Serial port
  Serial.begin(9600);

  Serial.println("Tella Thesis Prototype");

  // Initialize ZX Sensor (configure I2C and read model ID)
  if ( zx_sensor.init() ) {
    Serial.println("SENSOR OK");
  } else {
    Serial.println("SENSOR ERROR");
  }

  // Read the model version number and ensure the library will work
  ver = zx_sensor.getModelVersion();
  if ( ver == ZX_ERROR ) {
    Serial.println("Error reading model version number");
  } else {
    Serial.print("Model version: ");
```

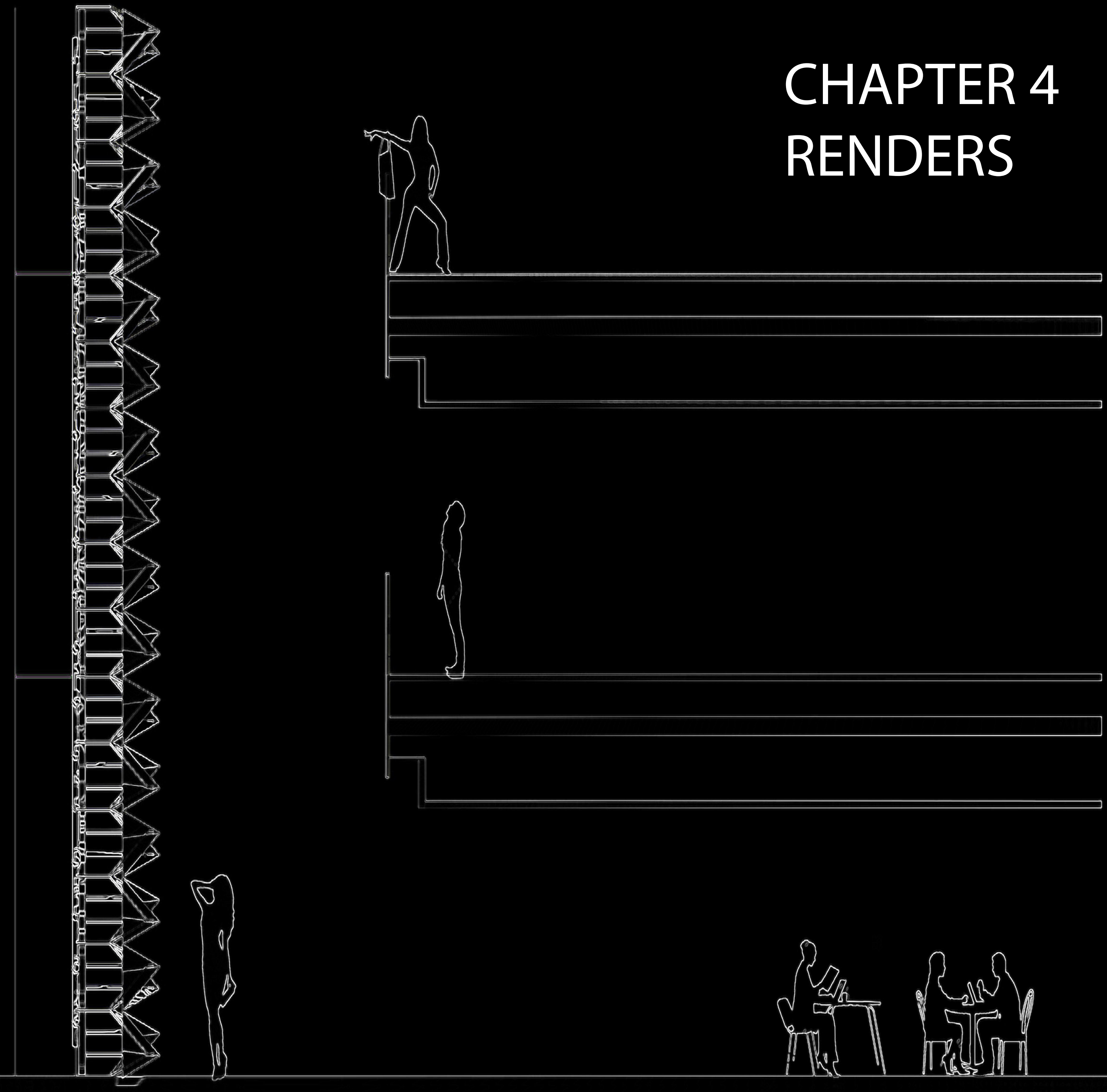
Movement.ino ReadMe.adoc

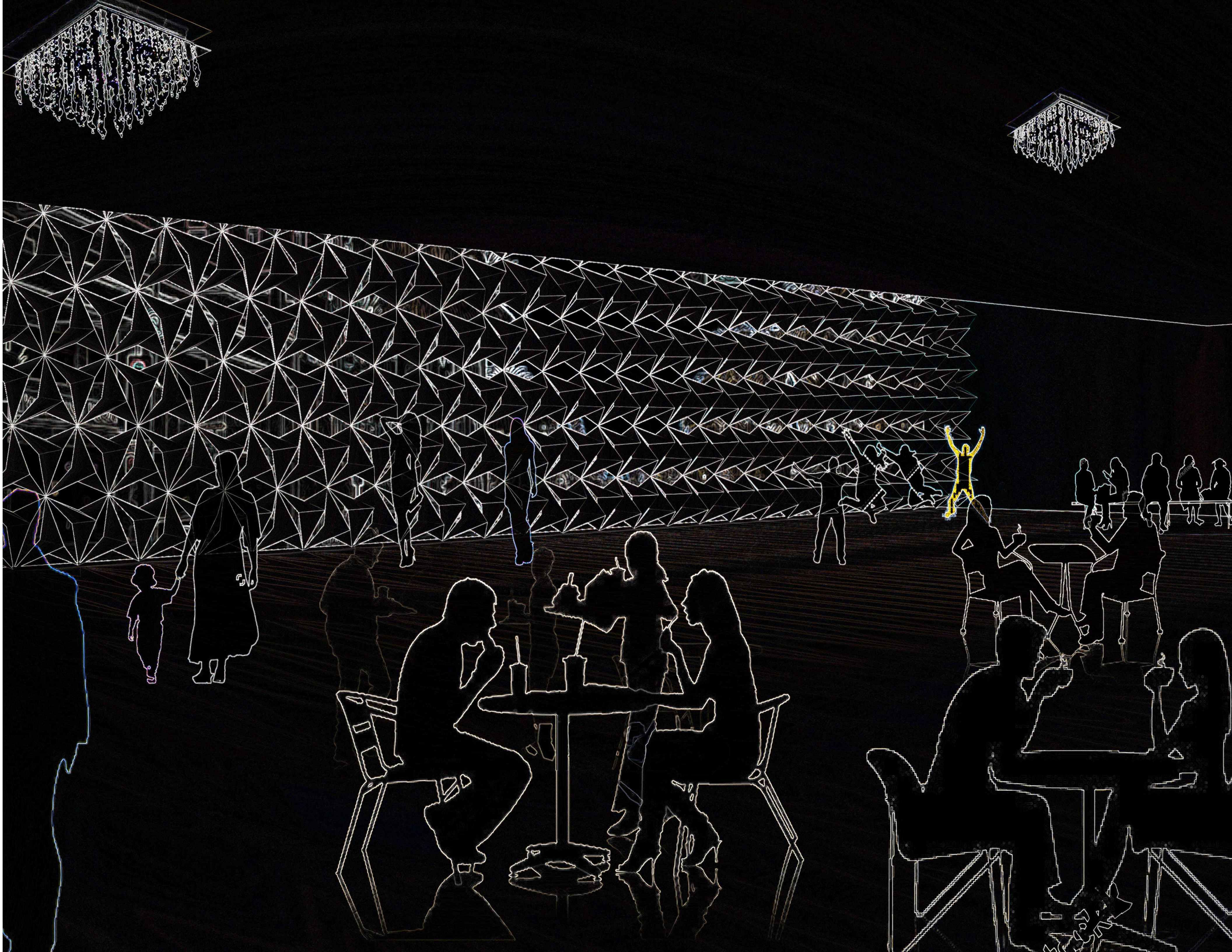
```
  if ( ver == ZX_ERROR ) {
    Serial.println("Error reading model version number");
  } else {
    Serial.print("Model version: ");
    Serial.println(ver);
  }
  if ( ver != ZX_MODEL_VER ) {
    Serial.print("Model version needs to be ");
    Serial.print(ZX_MODEL_VER);
    Serial.print(" to work with this library. Stopping.");
    while(1);
  }

  // Read the register map version and ensure the library will work
  ver = zx_sensor.getRegMapVersion();
  if ( ver == ZX_ERROR ) {
    Serial.println("Error reading register map version number");
  } else {
    Serial.print("Register Map Version: ");
    Serial.println(ver);
  }
  if ( ver != ZX_REG_MAP_VER ) {
    Serial.print("Register map version needs to be ");
    Serial.print(ZX_REG_MAP_VER);
    Serial.print(" to work with this library. Stopping.");
    while(1);
  }

  // Attach Servo
  myservo.attach(9);
}

void loop() {

  // If there is position data available, read and print it
  if ( zx_sensor.positionAvailable() ) {
    x_pos = zx_sensor.readX();
    if ( x_pos != ZX_ERROR ) {
      Serial.print("X: ");
      Serial.print(x_pos);
    }
```

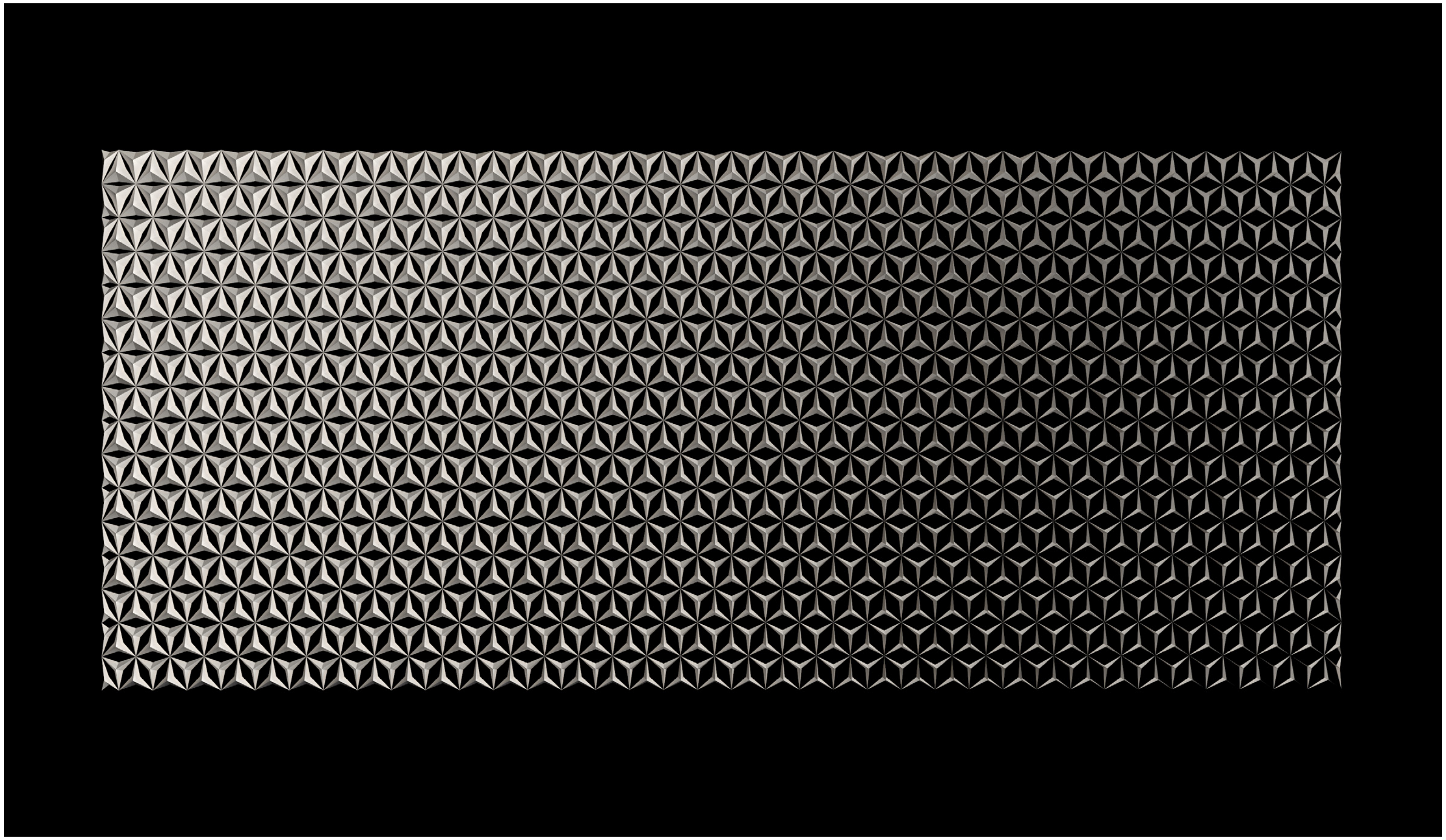
Movement.ino ReadMe.adoc

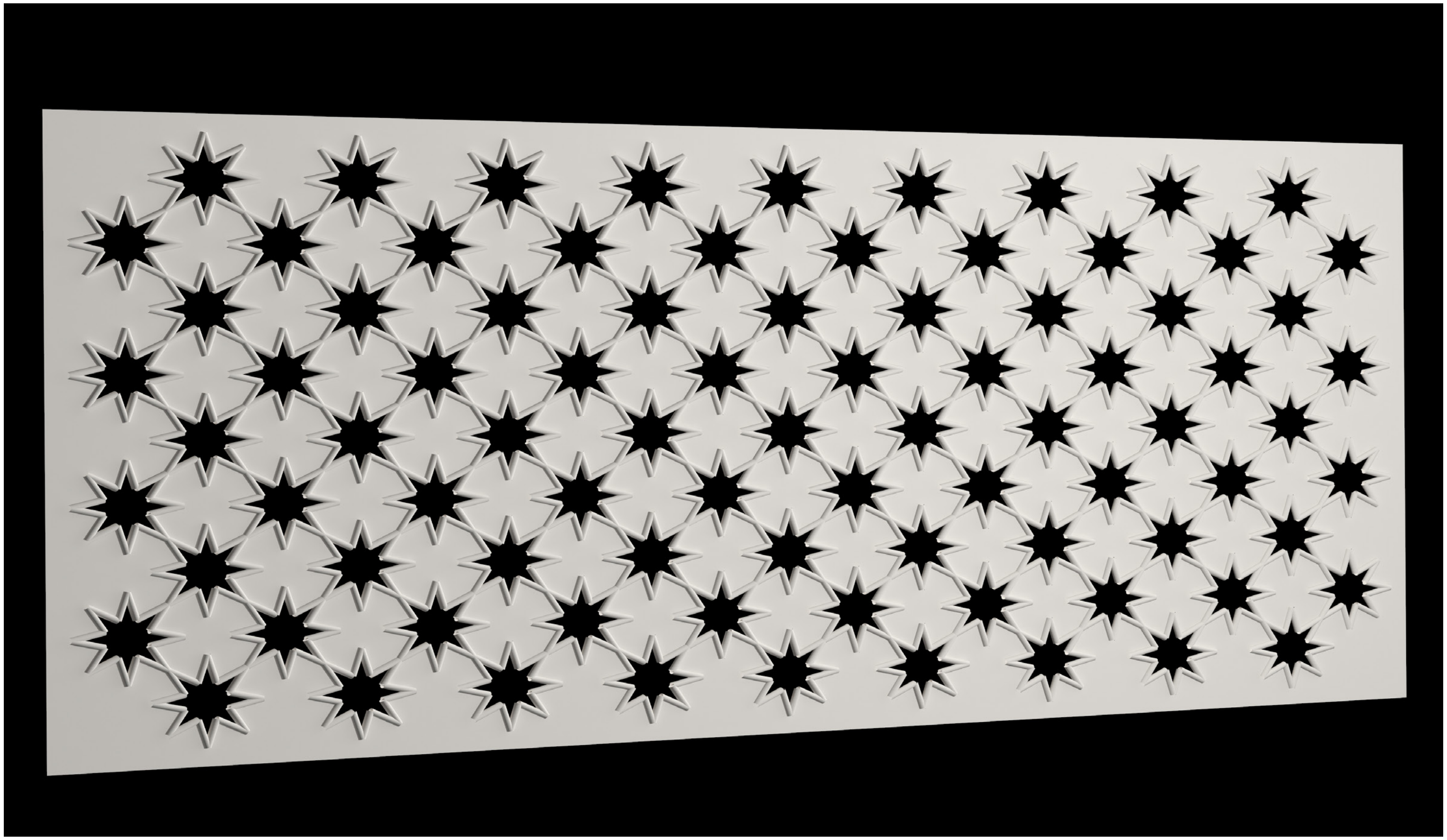
```
    while(1);
  }

  // Read the register map version and ensure the library will work
  ver = zx_sensor.getRegMapVersion();
  if ( ver == ZX_ERROR ) {
    Serial.println("Error reading register map version number");
  } else {
    Serial.print("Register Map Version: ");
    Serial.println(ver);
  }
  if ( ver != ZX_REG_MAP_VER ) {
    Serial.print("Register map version needs to be ");
    Serial.print(ZX_REG_MAP_VER);
    Serial.print(" to work with this library. Stopping.");
    while(1);
  }

  // Attach Servo
  myservo.attach(9);
}

void loop() {

  // If there is position data available, read and print it
  if ( zx_sensor.positionAvailable() ) {
    x_pos = zx_sensor.readX();
    if ( x_pos != ZX_ERROR ) {
      Serial.print("X: ");
      Serial.print(x_pos);
    }
    z_pos = zx_sensor.readZ();
    if ( z_pos != ZX_ERROR ) {
      Serial.print(" Z: ");
      Serial.println(z_pos);
    }
    myservo.write(x_pos);
  }
}
```

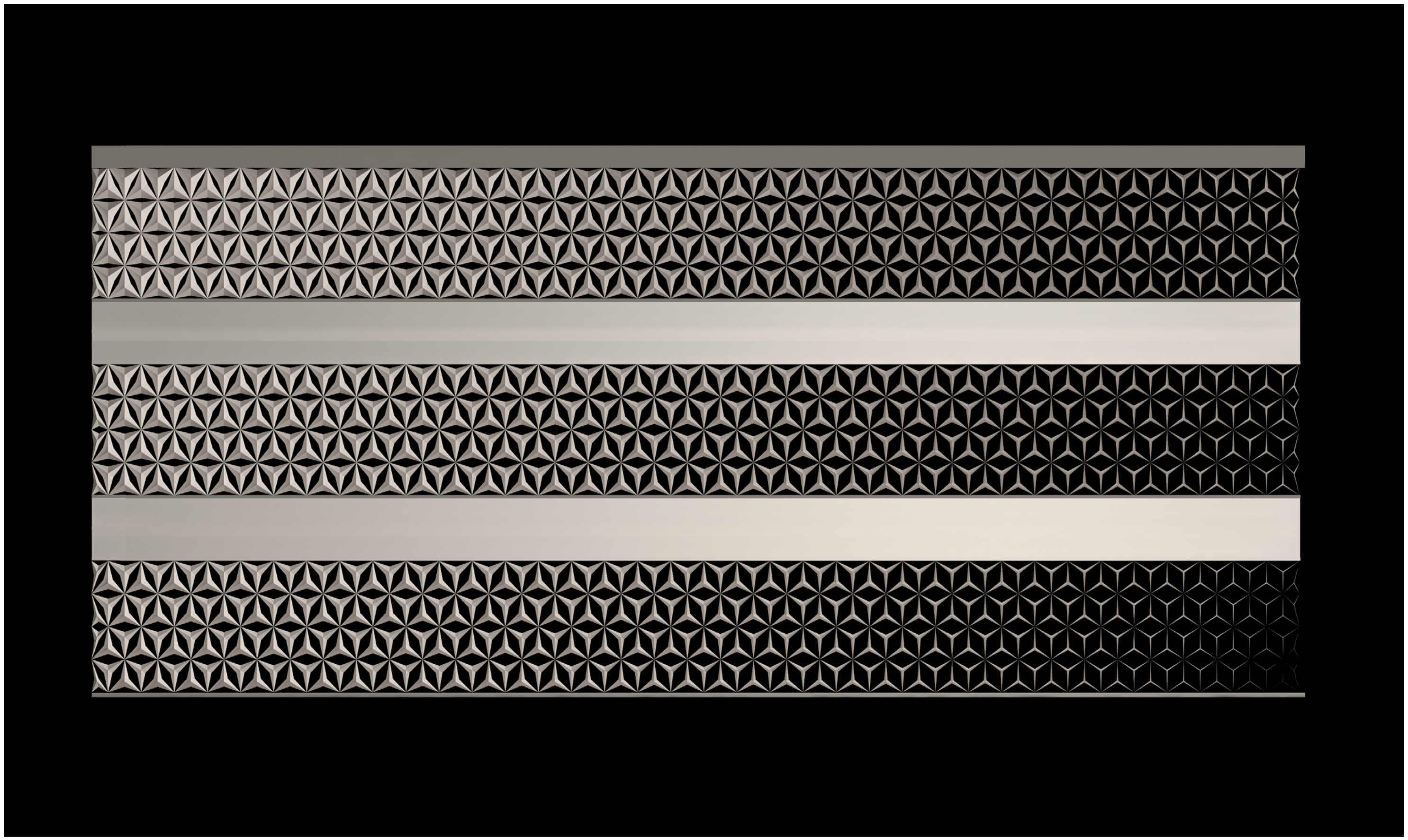
CHAPTER 4
RENDERS

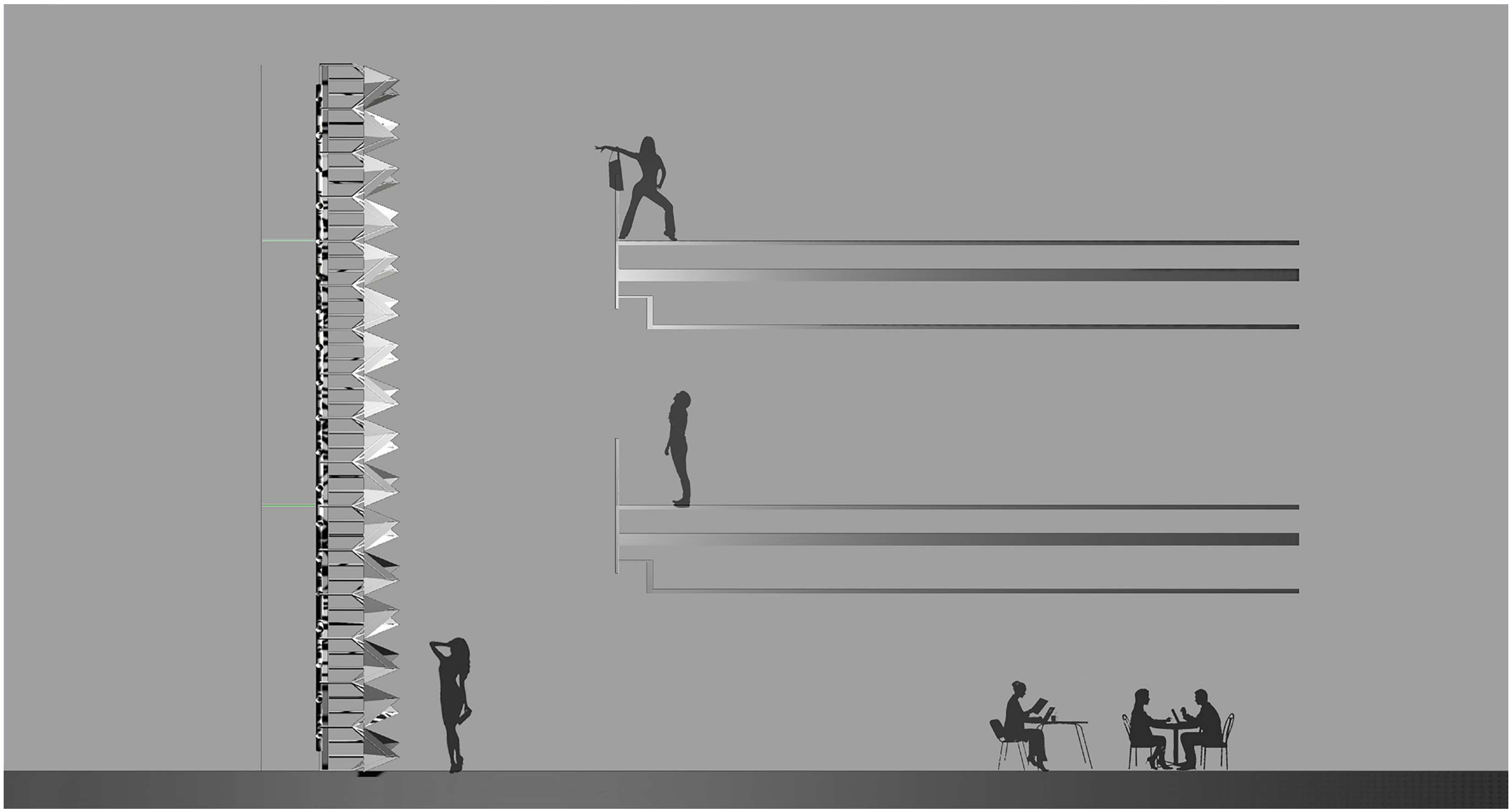


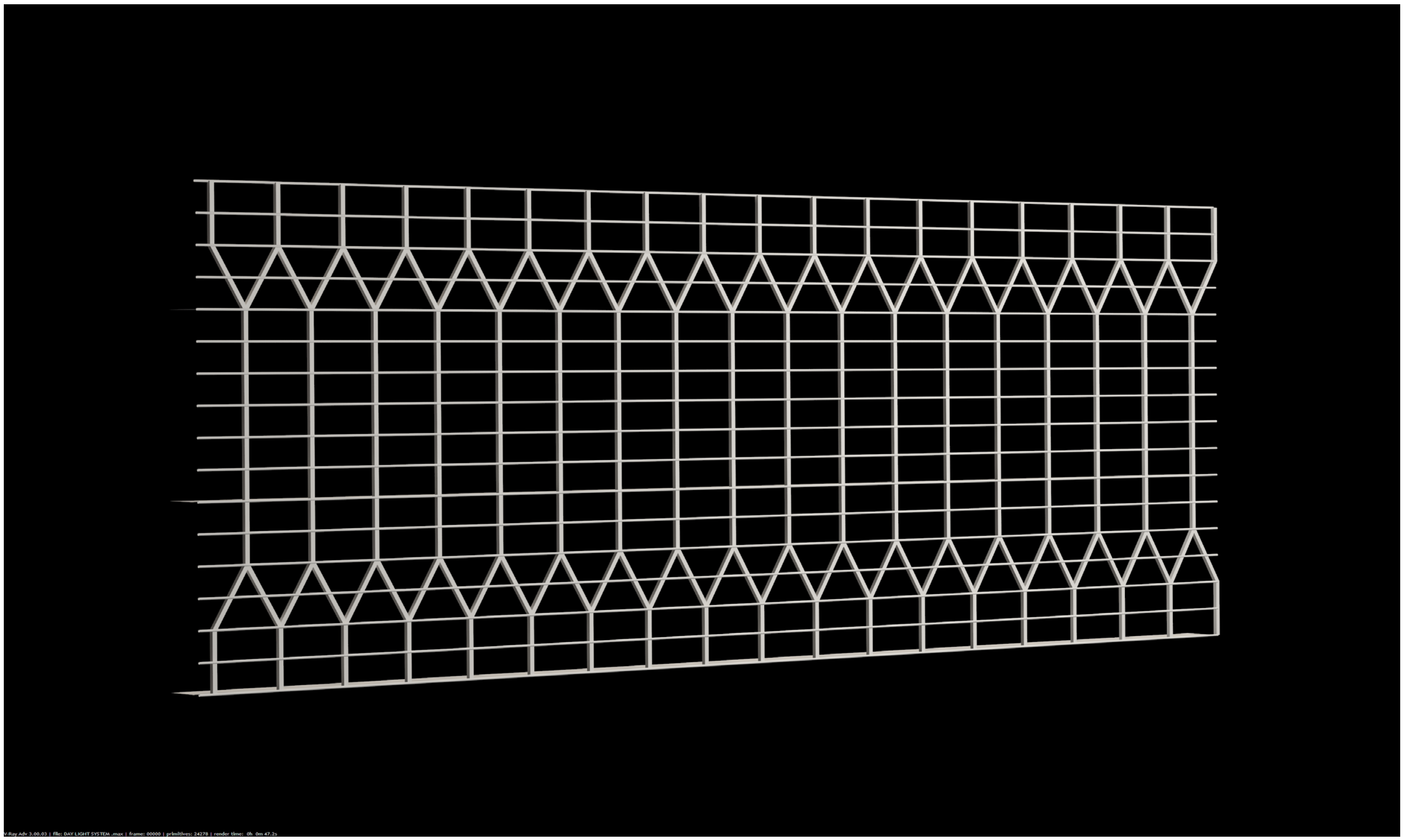


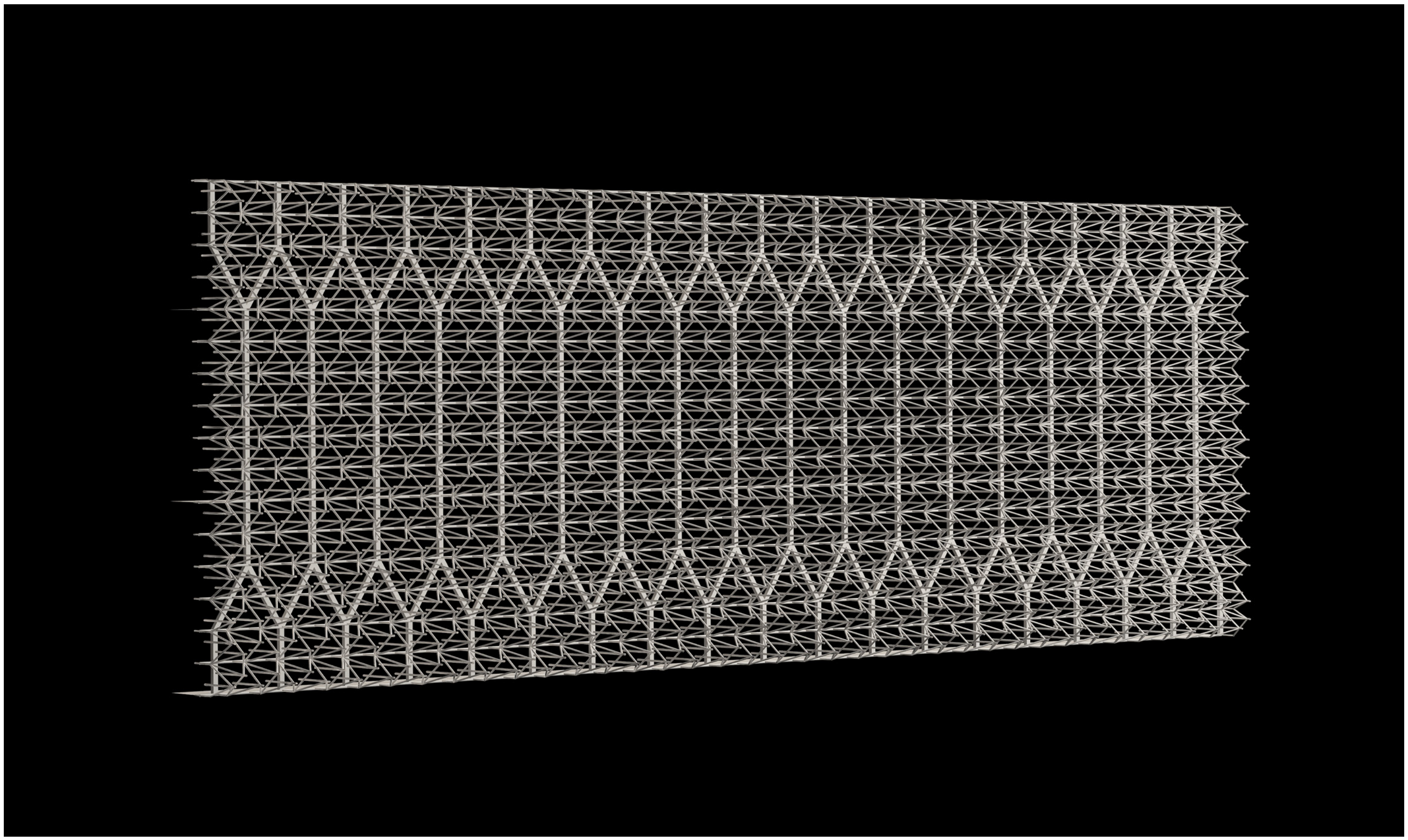


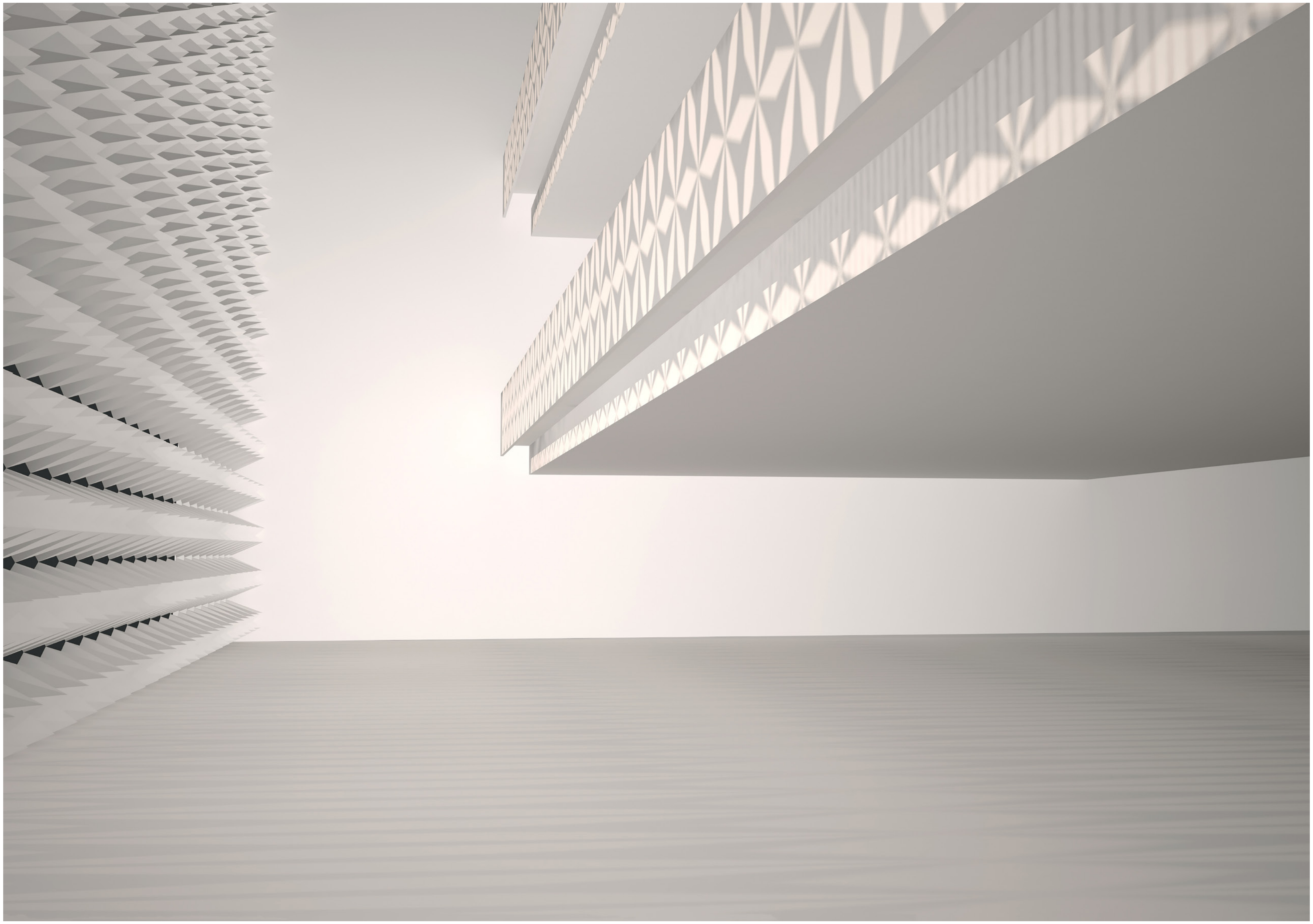


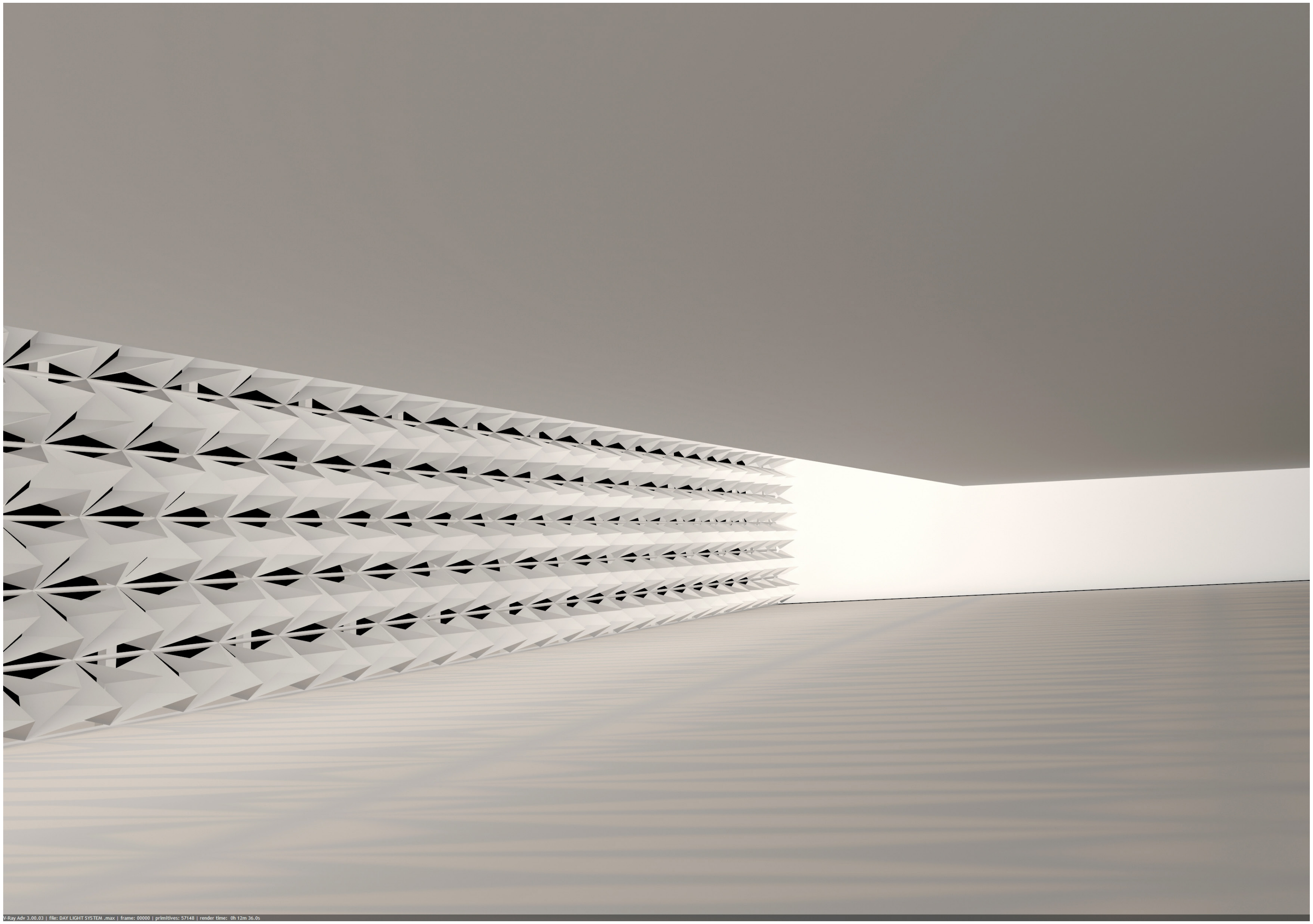


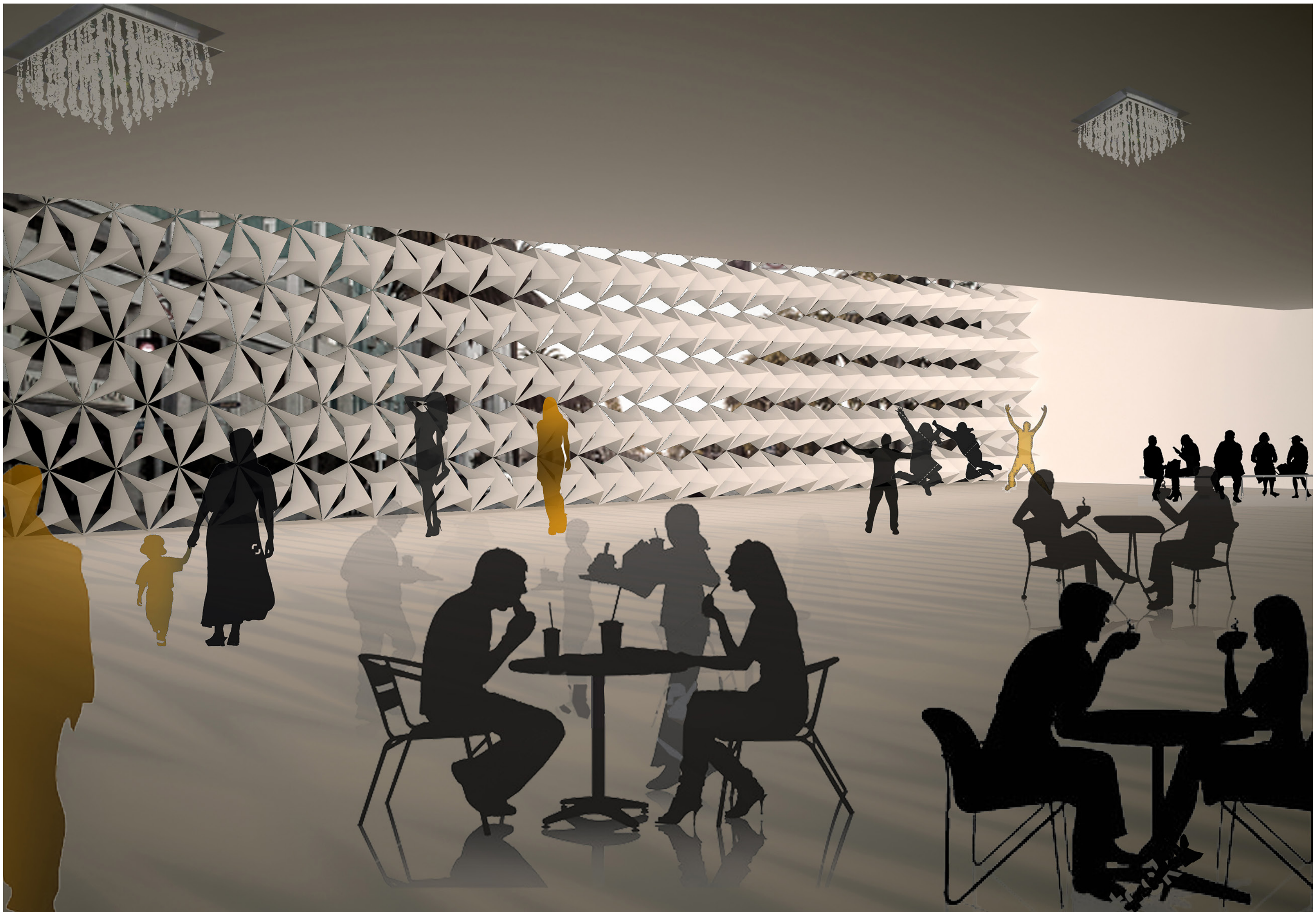


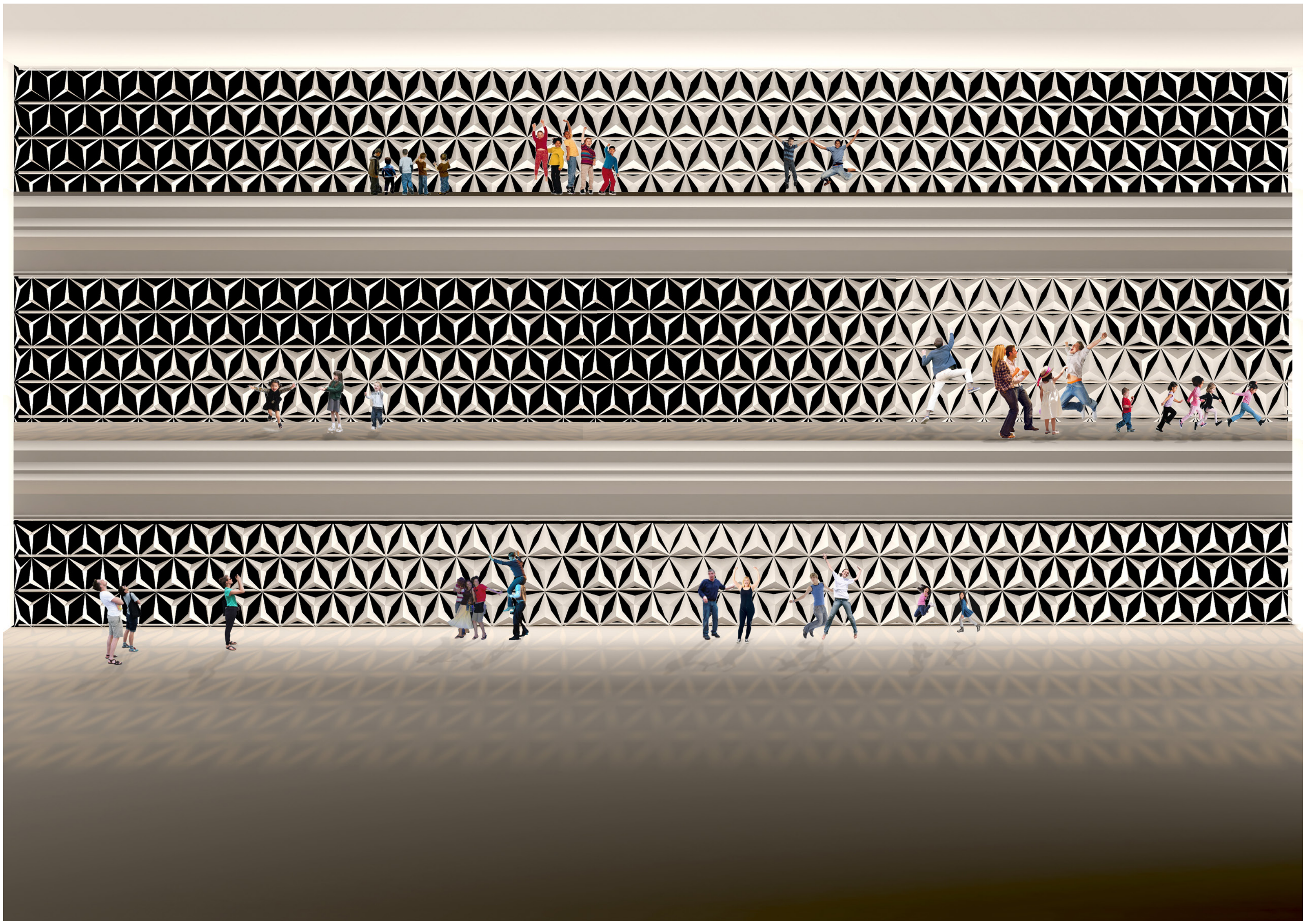







V-Ray Adv 3.00.03 | file: DAY LIGHT SYSTEM .max | frame: 00000 | primitives: 57148 | render time: 0h 12m 36.0s





The project is about experiencing a new way of making direct engagement between space and its users, making a playful relationship with architecture.

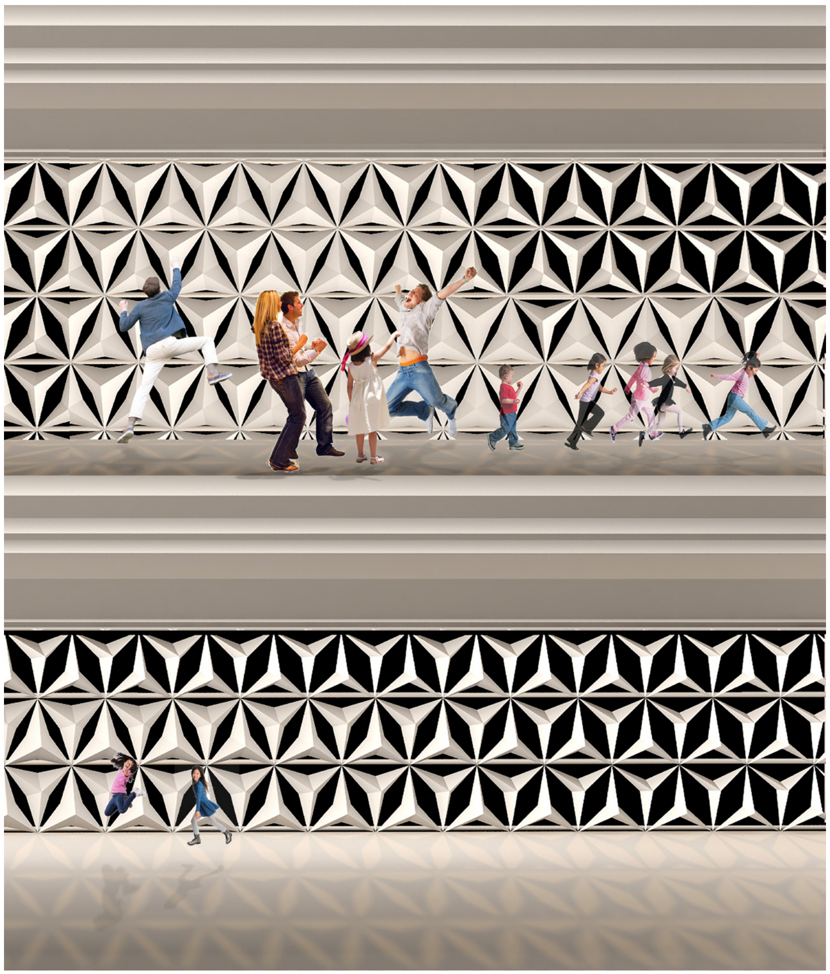
One of the ways that spaces and places can provide social glue is play, using the nature of a public spaces with the possibility of engaging the public into play, a way through which people involve with strangers and specific types of activities, leading to different social relations and interactions to engage people to redefine spaces and find something that have in common with others. Creating a space, where according to Henri Lefebvre is a process of production, in which ongoing interactions of social relations are happening.[47] Play is an important aspect of people's experience of space. Quentin Stevens in his book "The Ludic City" states that:

> "One of the fundamental functions of public space is as a setting for informal, non-instrumental social interaction, or play." [48]

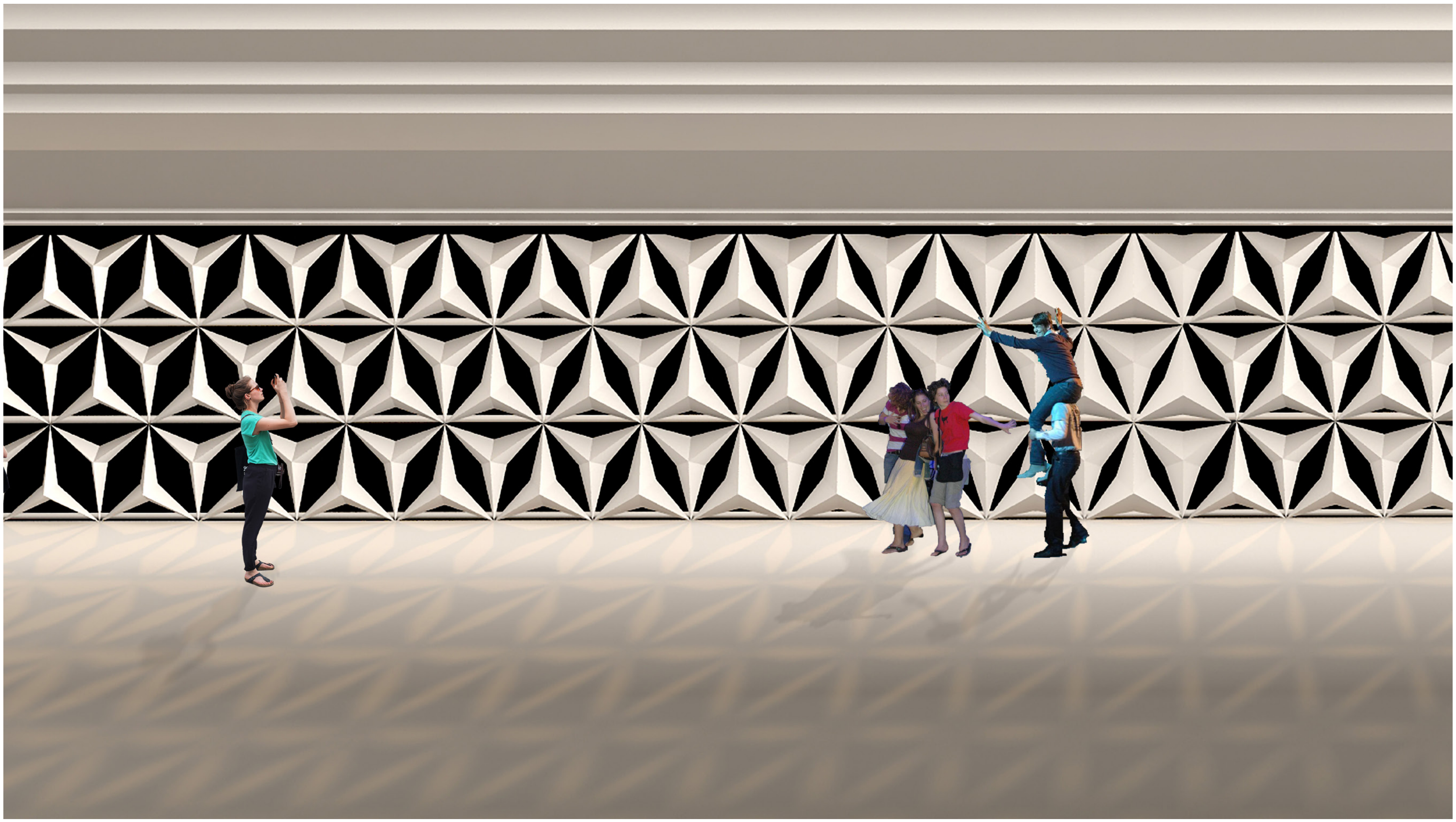
In addition, for Lefebvre the need to play is a social need as necessary as the need for creative activity, information, symbolism, and imaginary. An essential need of the city and urban life, for bringing people together and seeing architecture as a social form. [49]

> "The human being has the need to accumulate energies and to spend them, even waste them in play." [50]

Understanding play as a social activity and thinking about space as experiential and social way, allowed me for creating a communal environment where can give individuals the power to see others, relate themselves to the space, and experience freedom. The concept that for Hannah Arendt, one of the seminal political thinkers of the twentieth century, is the most important principle of political life, something that is experienced in action and speech in which people enjoy the freedoms of a space. [51]



The objective of this project is to create a space that provided a potential to play, where engage people in a social interaction through space, enabling them to experience freedom through play. A playful environment, where every person has a personal experience about the space, architecture, and people who might have never known them before. Experiencing a playful interaction with others, inconstant dynamic process of interactions among people where make the space active. Activating the space through play to make social spaces, spaces for public life. The space is activated trough physical means, encouraging its audience to walk through, engaging them in play with architecture and allowing them to participate in a direct interaction with space and each other to experience an activated public space. The space as an activated public space has the power to make people happy, get them moving and have a fun close by either individually or collectively.



Conclusion

In this paper I have aimed to investigate: how with the coloration between architect and robotic can develop a space where has an understanding of its users (through their body gestures) and respond accordingly? I have aimed to give a framework to what interactive architecture is. Generating architecture through behaviour-based AI approaches, using HCI, intelligent models of interaction for a dynamic interaction between buildings and their users. There are two important aspects for designing an interactive systems: 1- the ways they may interact and 2- types of systems and technology to be used. Understanding the process of creating interactive spaces, using methods which deal with computation, data generation, and simulation, helped me to think about different possibilities in design.

Future Work:

There are more possibilities open to this design to improve the design and make it more intelligent. For a later stage of the project: First, the system can be optimized to intelligently react to people modes. Using AI, Machine learning, Neural Network, and picture recognition algorithm, to rec-ognize, caption, and classify the details of people mode, giving them unique experience in which the system is reacting to them. Second, the system can be optimized to intelligently distribute light through dif-ferent area of the building, doing regional custom lighting where as people move from one area to another giving them unique experience with lighting. In addition, in terms of sustainability, the research can be expended to additional exte-rior intelligent system to control the indoor lighting and save energy consumption. The exterior system in this case can react to the sun m path and direction over time and provide sufficient indoor luminance and heat.

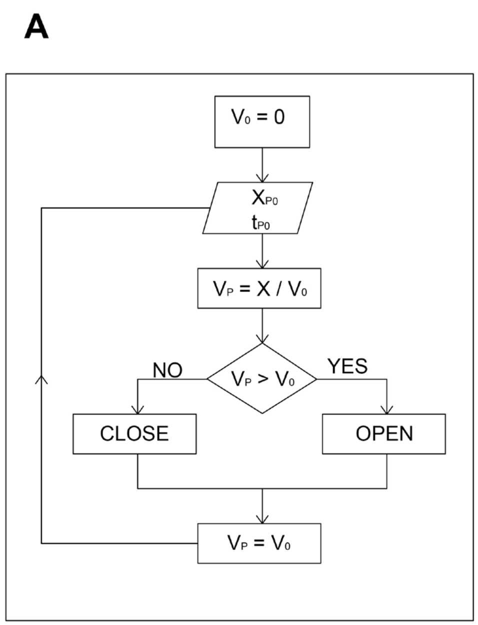
Finally, one also can add and develop its general algorithm to control the dynamic movement of the system of the interior side of the facade (the units open and close) based on a persons velocity (walking speed) and distance to the units. In this case the final algorithm can be developed as:

V0 = Initial velocity of the person

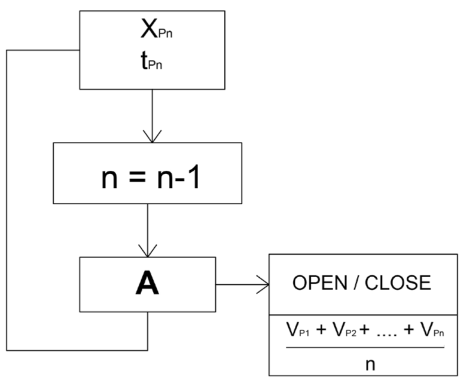
t = the time the person pass the unit by

x = distance between the person and the unit

The Flowchart:
A -> Algorithm for one person



The Flowchart:
Aorithm for n person:



References:

1- Ning Gu, and Xiangyu Wan. Computational Design Methods and Technologies: Applications in CAD, CAM, and CAE eEucation (USA: Information Science Reference, 2012), 257.

2- Joo-Ho Lee, and Hideki Hashimoto. “Intelligent Space —Concept and Contents,” Advanced Robotics 16, no. 3 (2002): 265, doi:10.1163/156855302760121936.

3- Ibid,. 266.

4- ohn Frazer. An Evolutionary Architecture (USA: Architectural Association Publications, January, 1995), 10.

5- Nicholas S Dalton, Holger Schnädelbach, Mikael Wiberg. Human–Computer Interaction Series: Architecture and Inter action: Human Computer Interaction in Space and Place (Switzerland: Springer, 2016), 256-257.

6-Joo-Ho Lee, and Hideki Hashimoto. “Intelligent Space —Concept and Contents,” Advanced Robotics 16, no. 3 (2002): 267, doi: 10.1163/156855302760121936.

7- Michael Fox, Interactive Architecture: Adaptive World (New York: Princeton Architectural Press, 2016), 13.

8- Michael Fox, Interactive Architecture: Adaptive World (New York: Princeton Architectural Press, 2016), 9.

9- Claus-Peter Rückemann. Integrated Information and Computing Systems for Natural, Spatial, and Social Sciences (USA: IGI Global, 2012), 48.

10- Gabriela Celani, David Moreno Sperling, and Juarez Moara Santos Franco. Communications in Computer and Infor mation Science- Computer-Aided Architectural Design Futures. The Next City - New Technologies and the Future of the Built Environment (Berlin Heidelberg: Springer, 2015), 233.

References:

11-Joo-Ho Lee, and Hideki Hashimoto. “Intelligent Space —Concept and Contents,” Advanced Robotics 16, no. 3 (2002): 268, doi: 10.1163/156855302760121936.

12- Ipke Wachsmuth, Timo Sowa, Gesture and sign languages in human-computer interaction: internation al Gesture Workshop, GW 2001, London, UK, And April 18-20, 2001: revised papers (New York, Berlin: Springer, 2002), 255.

13- Ning Gu, and Xiangyu Wan. Computational Design Methods and Technologies: Applications in CAD, CAM, and CAE Education (USA: Information Science Reference, 2012), 280.

14- Ibid,. 280.

15- Ibid,. 281.

16- Ipke Wachsmuth, Timo Sowa, Gesture and Sign Languages in Human-Computer Interaction: Inter national Gesture Workshop, GW 2001, London, UK, And April 18-20, 2001: revised papers (New York, Ber lin: Springer, 2002), 254.

17-Heriberto Cuayáhuitl, Martijn van Otterlo, Nina Dethlefs, and More... “Machine learning for in teractive systems and robots: a brief introduction” MLIS ‘13 Proceedings of the 2nd Workshop on Machine Learning for Interactive Systems: Bridging the Gap Between Perception, Action and Communication (08/2013): 19-28. doi:10.1145/2493525.2493530, 19.

18- Ibid,. 21

19- Ibid,. 254.

References:

20- Gabriela Celani, David Moreno Sperling, and Juarez Moara Santos Franco. Communications in Computer and Information Science- Computer-Aided Architectural Design Futures. The Next City - New Technologies and the Future of the Built Environ ment (Berlin Heidelberg: Springer, 2015), 95.

21- Henri Achten, “Degrees of Interaction: Towards a Classification.” (In Respecting Fragile Places: 29th eCAADe Con ference Proceedings, 565-572. eCAADe: Conferences. Ljubljana, Slovenia: University of Ljubljana. Faculty of Architec ture, 2011), 568-569.

22- Ibid,. 569.

23- Ibid,. 571.

24- Ibid,. 572.

25- Ibid,. 269-572.

26- Nicholas S Dalton, Holger Schnädelbach, Mikael Wiberg. Human–Computer Interaction Series: Architec-ture and Interaction: Human Computer Interaction in Space and Place (Switzerland: Springer, 2016), 141-142.

27- Ibid,. 1-3.

28- Ibid,. 3.

29- Rolf Pfeifer, and Christian Scheier, Understanding Intelligence (USA: Cambridge, Mass: MIT Press, 1999), 7.

References:

30- Liu Hsiang-Chuan, Sung Wen-Pei, and Yao Wenli, Computer, Intelligent Computing and Education Technology (USA: CRC Press, 2014), 579.

31- Stuart J. Russell, and Peter Norvig, Artificial Intelligence a Modern Approach (USA: Alan Apt, 1995), 3-5.

32- Hannu Jaakkola, Hannu Kangassalo, and Eiji Kawaguchi. Information Modelling and Knowledge Bases XII. (The Nether-land: IOS Press, 2001), 2.

33- Ibid,. 2.

34- Stuart J. Russell, and Peter Norvig, Artificial Intelligence a Modern Approach (USA: Alan Apt, 1995), 31-33.

35- Stuart J. Russell, and Peter Norvig, Artificial Intelligence a Modern Approach (USA: Alan Apt, 1995), 11.

36- Artificial Intelligence - MIT Open Course Ware- lecture 1 and 2 https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-034-artificial-intelligence-fall-2010/

37- K. Mäkisara, O. Simula, J. Kangas, and T. Kohonen. Artificial Neural Networks (Netherland: North Holland, 1991), 1033.

38- Eleftheria Fasoulaki. Genetic Algorithms in Architecture: a Necessity or a Trend. Department of Archi-tecture, Mas sachusetts Institute of Technology, 20012, 2.

39- John Frazer. An Evolutionary Architecture (USA: Architectural Association Publications, January, 1995), 2.

References:

40- Ibid,. 13.

41- Eleftheria Fasoulaki. Genetic Algorithms in Architecture: a Necessity or a Trend. Department of Architecture, Mas-sachusetts Institute of Technology, 20012, 3.

42- J Scott Turner. “Evolutionary Architecture.” Architectural Design 82. no. 2 (March 2012): 33. doi: 10.1002/ad.1376

43- John Frazer. An Evolutionary Architecture (USA: Architectural Association Publications, January, 1995),13.

44-Eleftheria Fasoulaki. Genetic Algorithms in Architecture: a Necessity or a Trend. Department of Archi-tecture, Mas sachusetts Institute of Technology, 20012,7.

45-Cristian Secchi, Cesare Fantuzzi, and Stefano Stramigioli. “Control of interactive robotic interfaces: a port-Hamiltonian approach” (Berlin ; London : Springer, 2007), 170.

46-Vinogradov, Oleg. Fundamentals of kinematics and dynamics of machines and mechanisms. The USA: CRC Press, 2000.

47-Henri Lefebvre, The Production of Space (Cambridge, Mass., USA, Oxford, OX.UK: Blackwell, 1991),33.

48-Stevens Quentin, The Ludic City: Exploring the Potential of Public Space (New York: Routledge, 2007), 3.

49-Henri Lefebvre, Writings on Cities (USA, UK: Blackwell, 1996), 147-148.

50-Ibid,. 147.

51-Stevens Quentin, The Ludic City: Exploring the Potential of Public Space (New York: Routledge, 2007), 4.

Bibliography:

Achten, Henri. "Degrees of Interaction: Towards a Classification." In Respecting Fragile Places: 29th eCAADe Conference Proceedings, 565-572. eCAADe: Conferences. Ljubljana, Slovenia: University of Ljubljana. Faculty of Architecture, 2011.

Artificial Intelligence - MIT Open Course Ware https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-034-artificial-intelligence-fall-2010/

Briegel, Hans J., and Cuevas, Gemma De las. "Projective Simulation for Artificial Intelligence." Scientific Reports 2 (15 May 2012): 400- 416. doi:10.1038/srep00400.

Celani, Gabriela, Sperling, David Moreno, and Santos Franco, Juarez Moara. Communications in Computer and Information Science- Computer-Aided Architectural Design Futures. The Next City - New Technologies and the Future of the Built Environment. Berlin Heidelberg: Springer, 2015.

Cuayáhuitl, Heriberto, van Otterlo, Martijn, Dethlefs, Nina, and More... "Machine learning for nteractive systems and robots: a brief introduction" MLIS '13 Proceedings of the 2nd Workshop on Machine Learning for Interactive Systems: Bridging the Gap Between Perception, Action and Communication (08/2013): 19-28. doi:10.1145/2493525.2493530

Dalton, Nicholas S, Schnädelbach, Holger, and Wiberg, Mikael. "Human–Computer Interaction Series: Architecture and Interaction: Human Computer Interaction in Space and Place." Switzerland: Springer, 2016.

Fasoulaki, Eleftheria. Genetic Algorithms in Architecture: a Necessity or a Trend. Department of Architecture, Massachusetts Institute of Technology, 20012.

Bibliography:

Fox, Michael. Interactive Architecture: Adaptive World. New York: Princeton Architectural Press, 2016.

Frazer, John. An Evolutionary Architecture. USA: Architectural Association Publications, January, 1995.

Gu, Ning, and Wan, Xiangyu. Computational Design Methods and Technologies: Applications in CAD, CAM, and CAE Education. USA: Information Science Reference, 2012.

Hsiang-Chuan Liu, Wen-Pei Sung, and Wenli Yao. Computer, Intelligent Computing and Education Technology. USA: CRC Press, 2014.

Jaakkola, Hannu, Kangassalo, Hannu, and Kawaguchi, Eiji. Information Modelling and Knowledge Bases XII. The Netherland: IOS Press, 2001.

Jaskiewicz, Tomasz Jan. "Towards a Methodology for Complex Adaptive Interactive Architecture" PhD thesis, Department Architectural Engineering + Technology Delft University of Technology, 2013, doi:10.4233/uuid:a81827c5-7d65-4cc7-9fab-20fab3a14c30.

K., Mäkisara, O., Simula, J., Kangas, and T. Kohonen. Artificial Neural Networks. Netherland: North Holland, 1991.

Lee, Joo-Ho, and Hashimoto, Hideki. "Intelligent Space — Concept and Contents." Advanced Robotics 16, no. 3 (2002): 265 – 280. doi: 10.1163/156855302760121936.

Lefebvre, Henri. Writings on Cities. USA, UK: Blackwell, 1996.

Bibliography:

Pfeifer, Pfeifer, and Scheier, Christian. Understanding Intelligence. USA: Cambridge, Mass. : MIT Press, 1999.
Rückemann, Claus-Peter. Integrated Information and Computing Systems for Natural, Spatial, and Social Sciences. USA: IGI Global, 2012.

Rückemann, Claus-Peter. Integrated Information and Computing Systems for Natural, Spatial, and Social Sciences. USA: IGI Global, 2012.

Russell, Stuart J., and Norvig, Peter. Artificial Intelligence a Modern Approach. USA: Alan Apt, 1995.

Secchi, Cristian, Fantuzzi, Cesare, and Stramigioli, Stefano. "Control of interactive robotic interfaces: a port-Hamiltonian approach" Berlin ; London : Springer, 2007.

Stevens, Quentin. The Ludic City: Exploring the Potential of Public Space. New York: Routledge, 2007.

Turner, J Scott. "Evolutionary Architecture." Architectural Design 82. no. 2 (March 2012): 28-33. doi: 10.1002/ad.1376

Wachsmuth, Ipke, Sowa, Timo. Gesture and Sign Languages in Human-Computer Interaction: International Gesture Workshop, GW 2001, London, UK, And April 18-20, 2001: revised papers. New York, Berlin: Springer, 2002.

Warwick, Kevin. Artificial Intelligence: The Basics. New York: Routledge, 2013.

Vinogradov, Oleg. Fundamentals of kinematics and dynamics of machines and mechanisms. The USA: CRC Press, 2000.