COMPUTATIONAL COMMUNITY:

AN AGGREGATION BASED MODULAR HOUSING DEVELOPMENT

James William Thomson III

COMPUTATIONAL COMMUNITY:

AN AGGREGATION BASED MODULAR HOUSING DEVELOPMENT

A Design Thesis Submitted to the
Department of Architecture
North Dakota State University

By:
James William Thomson III

In Partial Fulfillment of the Requirements for the Degree of
Master of Architecture

May 2022

TABLE OF CONTENTS

FIGURES

THESIS ABSTRACT

How can computation and aggregation-based design tools be utilized in the creation of a modular housing development? Using existing computer modelling and computational programming tools, such as Rhinoceros and its plugin Grasshopper, a set of architectural objects can be created. Geometric data will be derived from these architectural objects and a set of spatial relationship rules will be developed. By using aggregation based and genetic solver Grasshopper plugins such as WASP and GALAPAGOS the arrangement of these architectural objects will organize themselves, based on set spatial relationship rules as well as a controlled set of testing constraints. The development of these spatial rules and testing constraints will be catered to facilitating the creation of a modular housing development, specifically located on a later specified site on the Southern part of the City of St. Paul Minnesota.

THESIS NARRATIVE

We live in an age of data and technology. It is up to us to use that data and technology responsibly and in ways that give back to the human community. There have been major technological advances in Robotics, SMART Devices, Artificial Intelligence Models, and even Autonomous Vehicles, all of which have made our lives more efficient and brought us closer and more aware of those even on the other side of the planet, but what about Architecture? Energy modeling and structural integrity simulations have certainly made the buildings that we design more efficient and stronger than ever. But how can Architects utilize generative processes to output iterative design solutions?

Inspired by projects such as Habitat 67 by Moshe Safdie, Nakagin Capsule Tower, and Alexandra Road Estate; The Computational Community project aims to explore how Modular Housing can be generated through existing computational programming tools.

This Thesis aims to focus on the process of designing an Aggregation Based Modular Housing Development facilitated by the use of Discrete Modeling tools to optimize for various points of data. The primary data points that were chosen for the simulation in this thesis include maximizing sun exposure to windows of the various unit types and maximizing views from said units.

These is a lot of criteria to juggle but with the power of computation and genetic algorithms meeting these needs can be simplified, provided the problem is laid out very carefully.

PROJECT TYPOLOGY AND PRECEDENT

The final project typology will be a Modular Housing Development utilizing a set of architectural objects that will virtually arrange themselves based on a set of criteria for optimization. A similar precedent can be seen through housing developments such as Habitat 67 or Nakagin Capsule Tower and their modular approach to housing. The main difference between the goal of this project and Habitat 67 is that this project will not use subjective criteria for the arrangement of its units but will utilize and optimize for quantifiable data that will act as the typological constraints.

MAJOR PROJECT ELEMENTS

- Housing Development
- Modular Housing
- Prefabricated Building Components
- Computational Building Constraints
- Rhino-Grasshopper
- Simulation Modeling
- Discrete Modeling Design Tools
- Maximizing Output Data

USER/CLIENT/AUDIENCE DESCRIPTION

The Primary Audience for the project is those who live or want to live in the St. Paul Area specifically in the Dayton's Bluff neighborhood although the site is in St. Paul proper and is disconnected from the traditional neighborhood layout. The Computational Community Projects aims to create livable units that meet three different living constraints. One unit comprising of amenities for a single to two occupants. Another unit comprising of amenities for two to four occupants. And a final unit comprising of amenities for two to four occupants with walkable roof access. The project will aim to maximize the occupancy of this development within the constraints of the testable data.

SITE/CONTEXT

Union Depot Rail View Picnic Area Parking Lot (394 E Kellogg Blvd, St Paul, MN 55101)

Figure 1.0

Buildable Area -- *Figure 1.1a*

Contestable Buildable Area ------------------------

THE PROJECT EMPHASIS

The project emphasis will be a comparative analysis of multiple generated outputs from the virtual program. These volumes will have a series of output data that match the optimization constraints that can be compared and analyzed. This analysis will reveal positives and negatives of each generated volume with unit aggregations inside of them. The comparison is there to show how a new system of computationally driven modularity can be used to improve or progress the previously explored Metabolist movement. Additionally, due to the site's proximity to railways a significant redevelopment of the site will be necessary for safety purposes and will be cogent when generating the model for simulation.

GOALS OF THE THESIS PROJECT

- Comprehensive site analysis
- Data Collection regarding Habitat 67 + Other Case Studies (and to be used as Final Constraints for St. Paul Site)
    - Maximized Sun Received from Windows
    - Maximized Isovist Success Rate (Views from Units)
    - Maximized Circulation Unit Score
- The creation of a modular discrete design conceptual housing generator based on the criteria I have laid out using Rhinoceros – Grasshopper – WASP – Galapagos
    - An optimization of Unit layout based on the set training goals.

PLAN FOR PROCEEDING

a. Design Methodology

The Design Methodology will be as follows. First, a series of 3 Case Studies which will be analyzed for Typological Design considerations. Second, the creation of a Rhinoceros Grasshopper Code that will aid in facilitating and generating my Unit Organizations for my buildable area. Third, Multiple unit organizations will be generated based on which optimization factor I am prioritizing to show the differences until multiple well-rounded masses can be generated based on the various optimization factors.

b. Plan for Documentation

Once all of the volumes have been generated from the Algorithmic Code their respective data will be collected via an excel spreadsheet which will show all the associated information.

c. Project Schedule (See Figure 2.0)

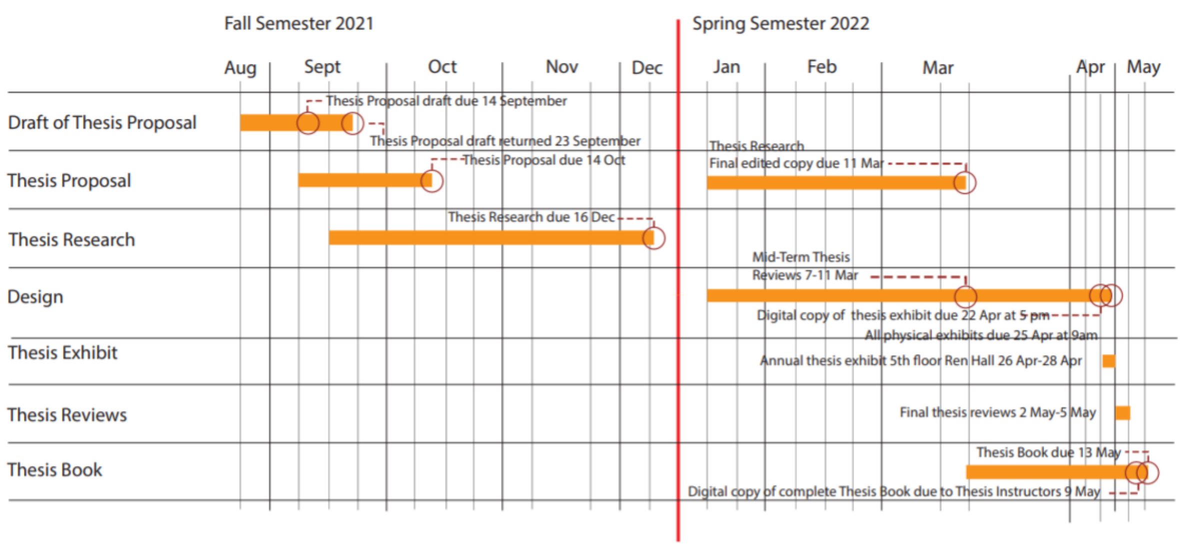
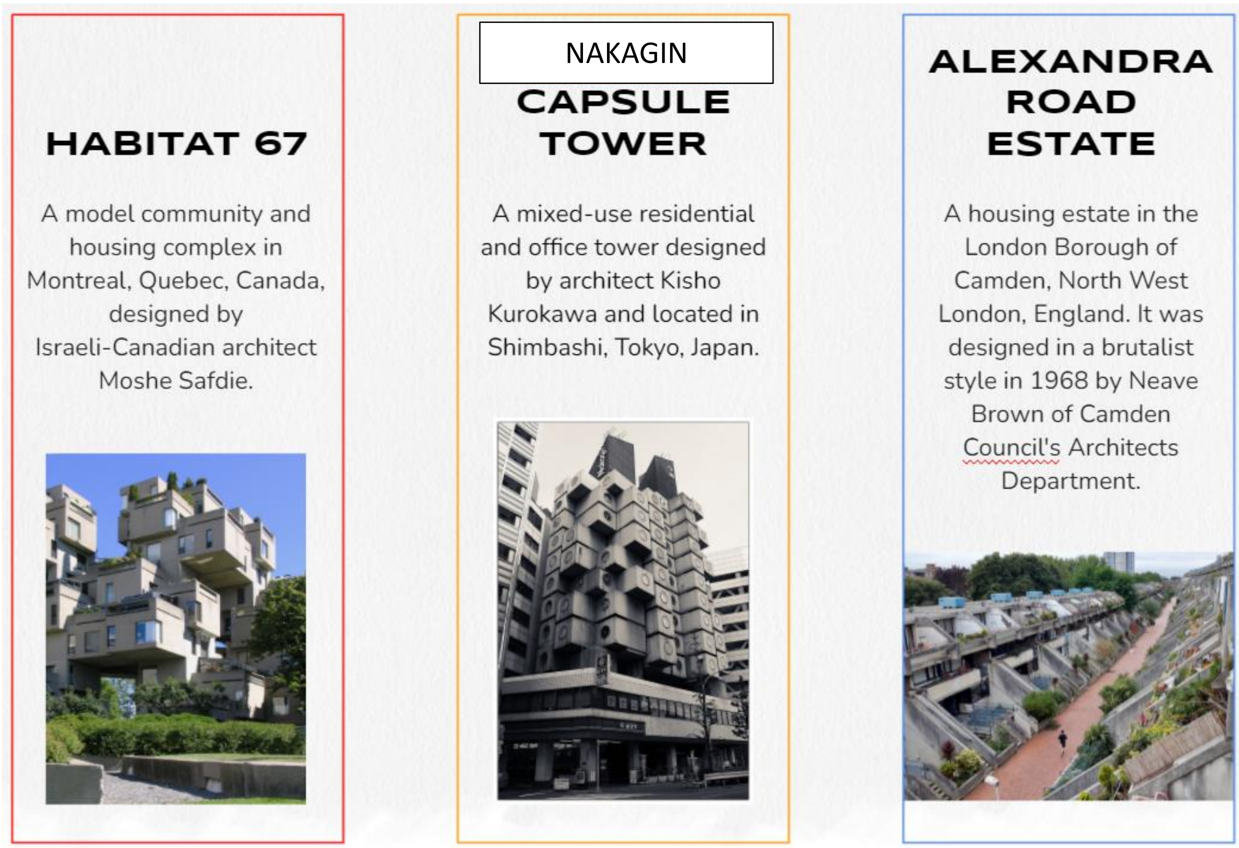


Figure 2.0

THESIS RESEARCH

PRECEDENT RESEARCH (CASE STUDIES)



The most important piece of information derived from the Habitat 67 Community housing complex by Moshe Safdie in relation to the thesis described in this document is the Modular-like aspect of the units. There are many variations in unit size and massing but when stacked it creates a complex form. I believe that through the use of Computational Design this scattered form can be brought to a type of order if the units for the new site are optimized according to various environmental constraints. On the next page is Figure 3.0 which shows a section of one part of the Habitat 67 Complex.

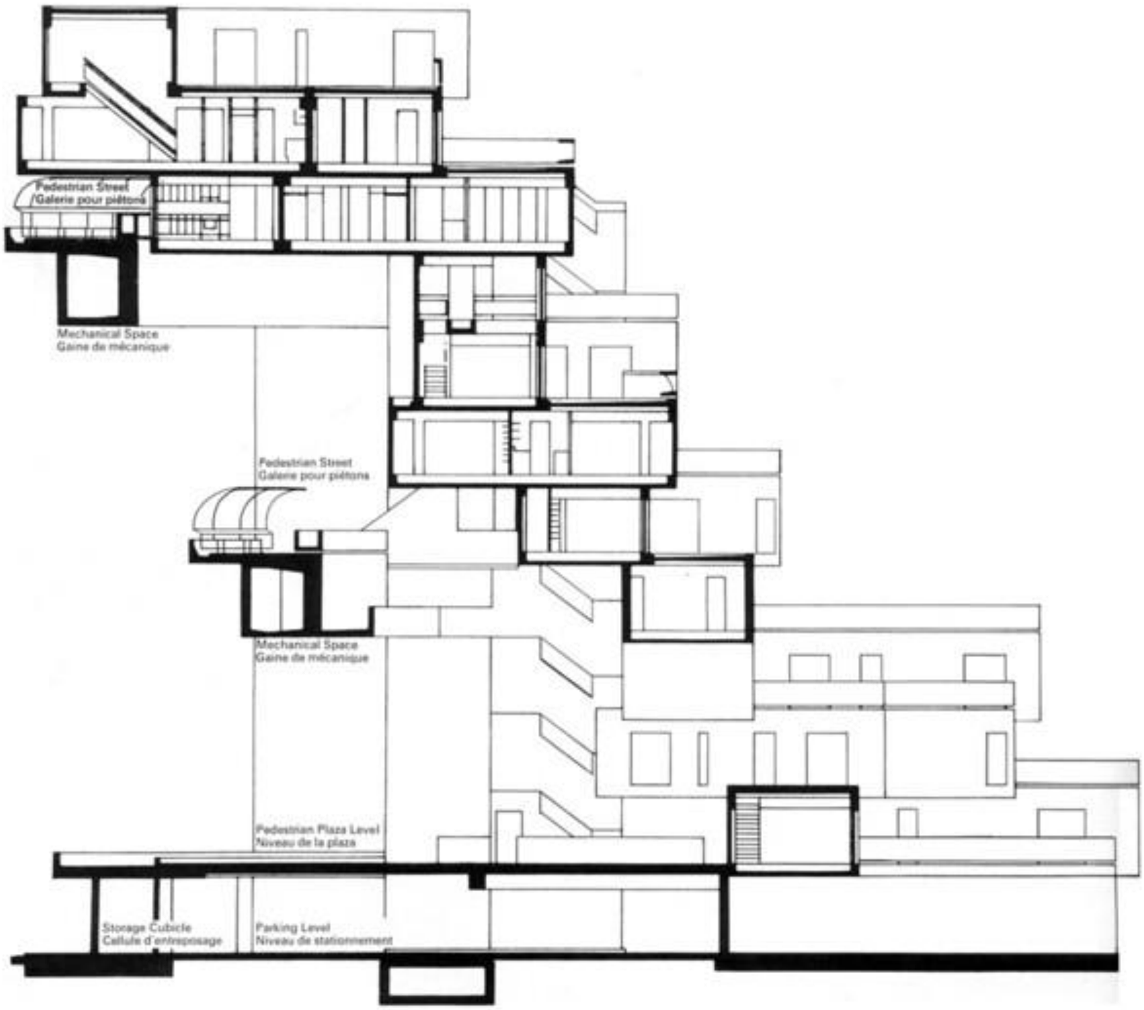

Pedestrian Street
Galerie pour piétons
Mechanical Space
Gaine de mécanique
Pedestrian Street
Galerie pour piétons
Mechanical Space
Gaine de mécanique
Pedestrian Plaza Level
Niveau de la plaza
Storage Cubicle
Cellule d'entreposage
Parking Level
Niveau de stationnement

Figure 3.0

In regards to the Nakagin Capsule Tower of Tokyo, although I do not personally see it as ideal living, it is however a fantastic study into modularly designed concrete units which are extremely easy to manufacture especially in comparison to the complexly arranged Habitat 67. But in terms of its efficiency, the Nakagin tower looks quite promising. Below is Figure 3.1 which shows a section of the Nakagin Capsule Tower.

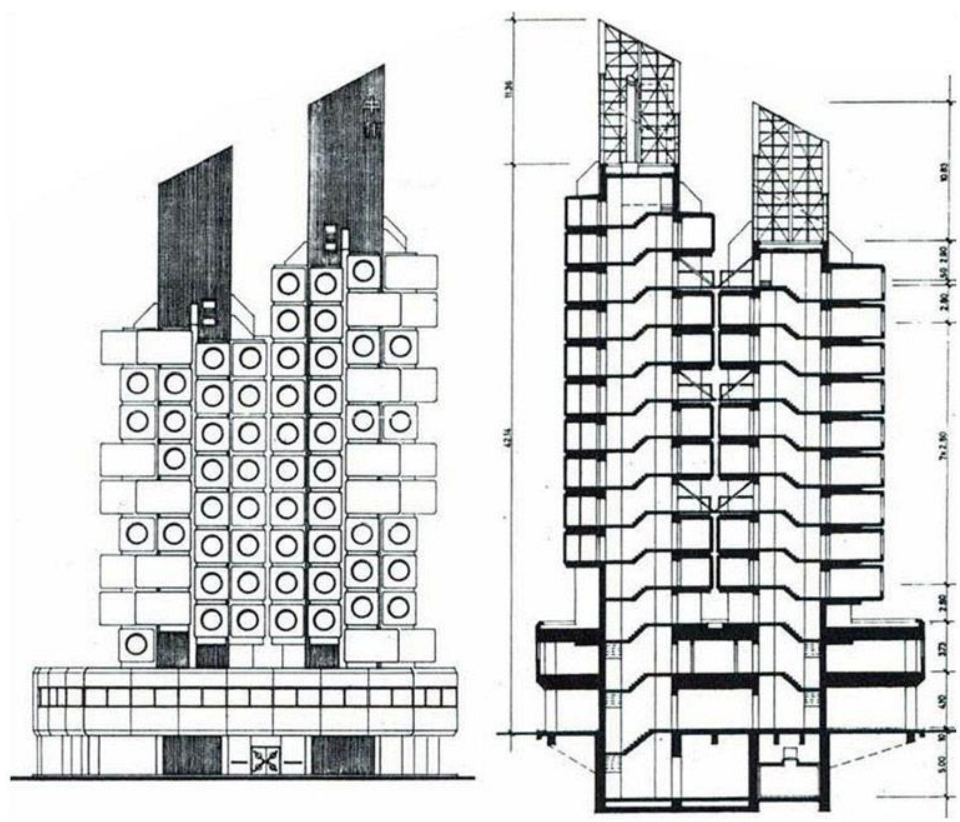


Figure 3.1

The Alexandra Road Estate Flats in London England is much more desirable in terms of its unit size. Each unit being able to hold at least a small family. The development does act as a wall an either side with units on both sides and a walkway in the middle. The walkway in the middle is fantastic for pathing and prevents roadway traffic as all the parking is located below the units. (as with the previous two case studies.) However the downside to having units stacked on either side so tall is that it actively diminishes views to the surrounding landscape but an upside is that the units act as an enclosed tight knit community because every unit can see every other unit. Below is Figure 3.2 which shows a section of the Alexandra Road Estate Flats.

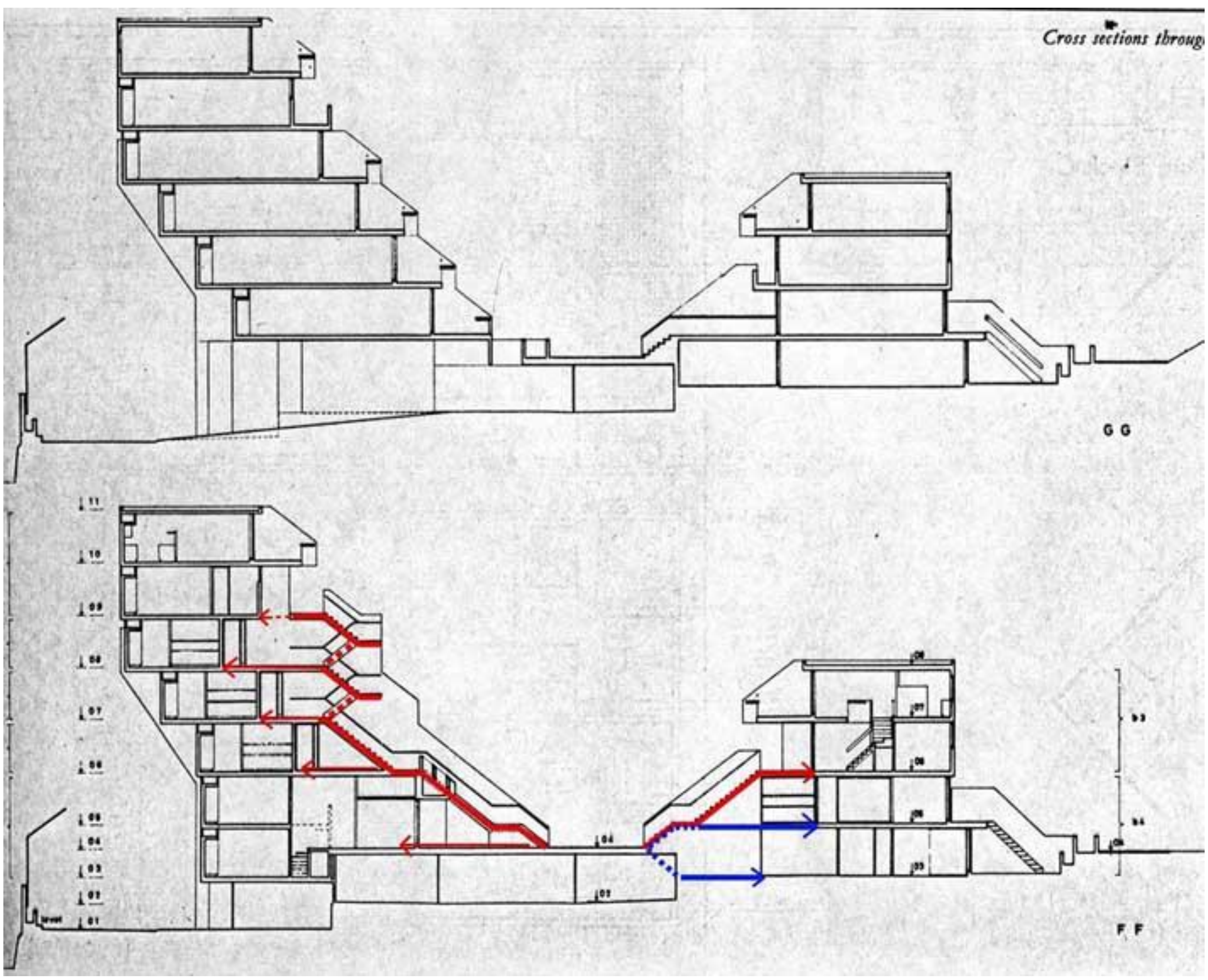


Figure 3.2

PROJECT JUSTIFICATION

The following is a list of sources that consist of already existing documented research and papers that were deemed to be relevant or potentially relevant to the thesis presented in this document. This section will consist of a review of said article or published paper describing its premise and contents. Then, noteworthy in formation will be selected and elaborated on. Finally, its usefulness to the overall thesis presented in this document will be evaluated and probed for various potential avenues and conduct for the thesis presented in this document.

- *Between Form and Information: Early Philosophies of Computer-Aided Design by Daniel Cardoso Llach from the Nexus Network Journal*
- *Supporting product architecture design using computational design synthesis with network structure constraints by David F. Wyatt, David C. Wynn, Jerome P. Jarrett, and P. John Clarkson through the Research and Engineering Design Journal*

The first piece of literary information to be evaluated is a research paper titled *Between Form and Information: Early Philosophies of Computer-Aided Design by Daniel Cardoso Llach from the Nexus Network Journal*. An excerpt from its abstract is as follows; "This article draws from primary historical sources to examine the origin of, and tensions between, two postwar era modeling techniques that shaped the early history of computer-aided design: the plex, developed by Douglas T. Ross, and the Coons patch, developed by Steven A. Coons. The article shows how each of these two techniques can be seen as emblematic of a fundamentally different understanding of design—one centered on information, and another on form—crucially foreshadowing present-day practices of computational design in architecture and other creative fields." In short, the paper is comparing the "Plex"; a technique developed by Douglas T. Ross who was a mathematician at the Servomechanisms Laboratory at MIT, and a technique called the

“Coons Patch”; which was developed by Mathematician Steven A. Coons who was an employee at Chance Vaught Aircraft. Each of these techniques were developed in the 1950’s and 1940’s respectively. The Plex was originally developed as a series of subroutines that made the process of directing a 3 Axis Milling Machine, using punched paper tape, much easier. But Ross had greater ambitions for the Plex and developed a theory that could have the Plex store data for more complex 3D objects than just the motions of a milling machine. The three elements of his theory were” *data, structure,* and *algorithm. “*“The data are “units or indivisible entities in terms of which the ‘thing’s’ properties are described or measured;” structure refers to the relationships between the data; and the algorithm is “the capstone that allows the data in the structure to be interpreted, manipulated, and filled with meaning” (Ross 1968: 14). The algorithm relates to the behavior and the interpretation of the whole: a sort of logical rule set for operation and assembly.”” (Llach) “Plex” then became a word to describe a line and its sub entities such as points and their coordinates. “The plex delineates an approach to modeling that eschews the description of an artifact’s physical or geometric attributes as its chief purpose. It emphasizes instead the architecture of the relationships between the different elements—physical or not—that comprise an artifact. In doing so, the plex elicits an understanding of computational design premised on the structured nature of digitally encoded information. This understanding foreshadows and continues to underpin ongoing sensibilities in architectural modeling—in particular, those linked to building information modeling, where form is only one of many descriptive layers in a complex arrangement of information.” (Llach) In regards to the “Coons Patch”: this mathematical technique was primarily used for interpolating the position of points on a three-dimensional curved surface defined by a set of parametrically-defined curves. Coons work became highly regarded and his work went on to inspire the Cambridge CAD Group to develop a graphics library called GINO and "B-Reps" or what would now today be called "Breps" which is a way to digitally describe a "Boundary Definition" in terms of its specific 3Dimensional coordinates. Llach went on to describe his own thoughts and comparisons in conclusion.

This Research Paper was extremely helpful in describing the very early processes and concepts that define modern day Computational Design and Computer Graphics. In regards to the thesis presented in this document which involves the use of 3D Modeling such as Rhinoceros and its associated plugin Grasshopper, the concept of a "Plex", "Coons Patch", and "BREP" are present in their software display and make up how the user interacts with its programming. Rhinoceros and Grasshopper allow the user to identify and modify geometries in groups that are like a "Plex", surfaces are defined by a form of "Coons Patch" and their boundaries are defined by "BREPs". 3D Modeling and Visualization programs of today would not exist without the work of Ross and Coons.

The second piece of literary information for review is a research paper titled *Supporting product architecture design using computational design synthesis with network structure constraints by David F. Wyatt, David C. Wynn, Jerome P. Jarrett, and P. John Clarkson through the Research and Engineering Design Journal*. In the research papers' abstract it states "This paper explores how computational tools can augment creative methods in product architecture design. Based on an empirical study aiming to understand the context of product architecture design, a new computational method is proposed to support this activity. In the method, product architectures—networks of components linked by connections—can be synthesized using constraints on the structure of the network to define the set of 'realizable' architectures for a product." Now although this paper does not directly deal with the Architecture profession, it does however relate quite a deal to the thesis described in this document. But first an overview of the research paper. The paper first defines a “Product Architecture.” “Product Architecture represents a product’s functions, their mapping to physical components, and the interfaces between the components. Maier and Rechtin (2000) quote a range of definitions for ‘architecture’, focusing on structure in terms of components and relationships, but also point out that ‘architecture is what architects produce, and […] what architects do is help clients make decisions about building systems'." (Wyatt) The paper then proposes a model by which product architecture is designed as pictured in Figure 4.0 below.

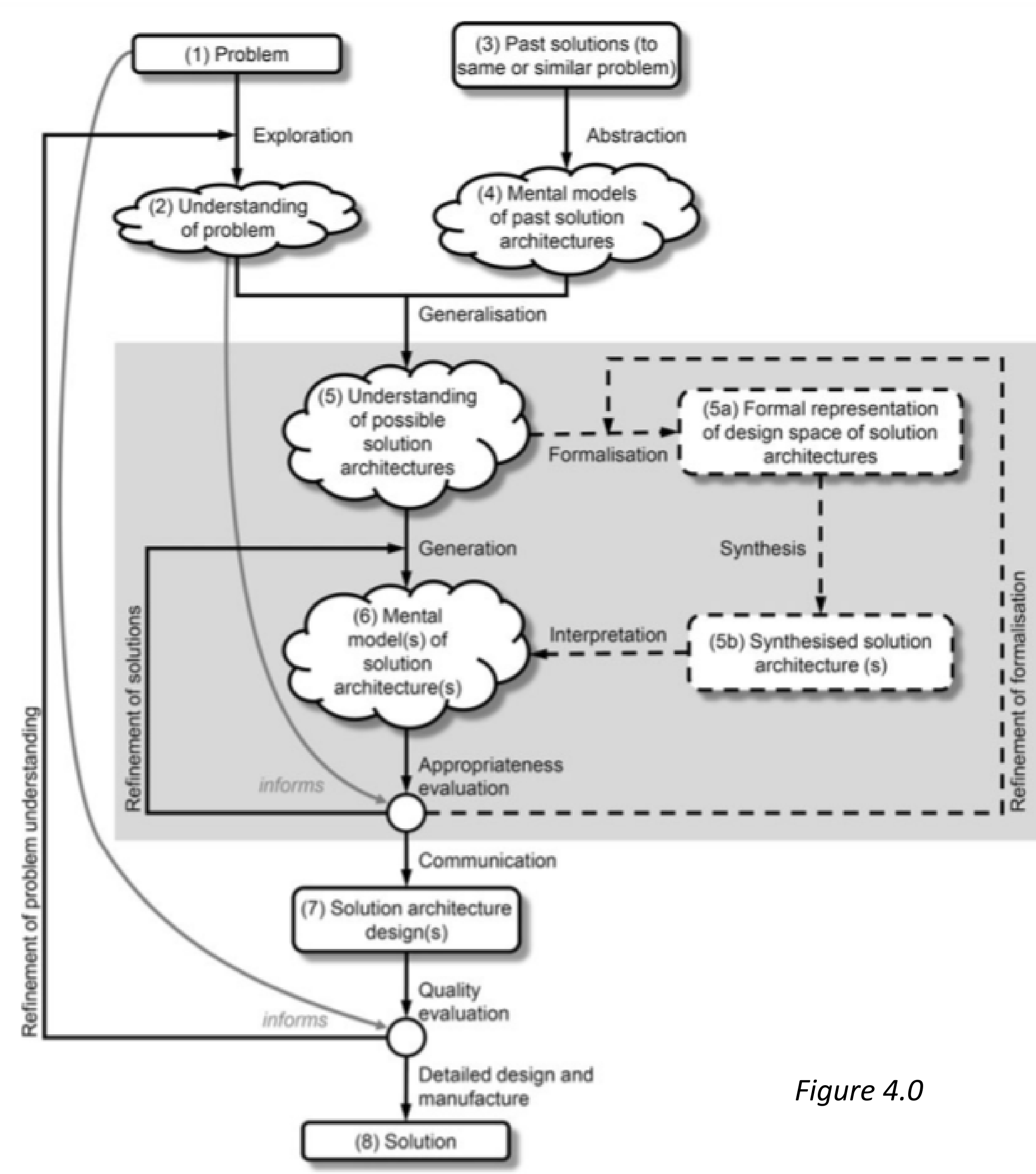


Figure 4.0

Which involves identifying a problem, understanding a problem, which then moves on to a generalization and formalization of design generation and synthesis until a solution can be developed. The paper continues by showing how to model product architectures by defining the space of the architectures. This involves breaking down the product into its possible configurations or rather what components are involved in defining its possible configurations.

These components are defined as the “Network Structure Constraints”. In this research paper the authors used this example to describe NSC’s, “One NSC may apply to all component types within a higher-level grouping (‘AND’), for instance ‘Every car [an abstract component type with child component types of hatchback, saloon, estate and sports car] must comprise four wheels’. Alternatively, an abstract component type may express the possibility for alternatives (‘OR’), for instance ‘Every sports car must comprise one roof [an abstract component type with child component types of solid roof and folding roof]’.” The paper then describes the use of a “Schema” or a plan for various components of the product, in this case a vacuum cleaner gear system. Then a “Synthesis Algorithm” is described where “based on a schema, sets of solution architectures may be generated using computation design synthesis,” (Wyatt) as seen in Figure 4.1 on the next page.

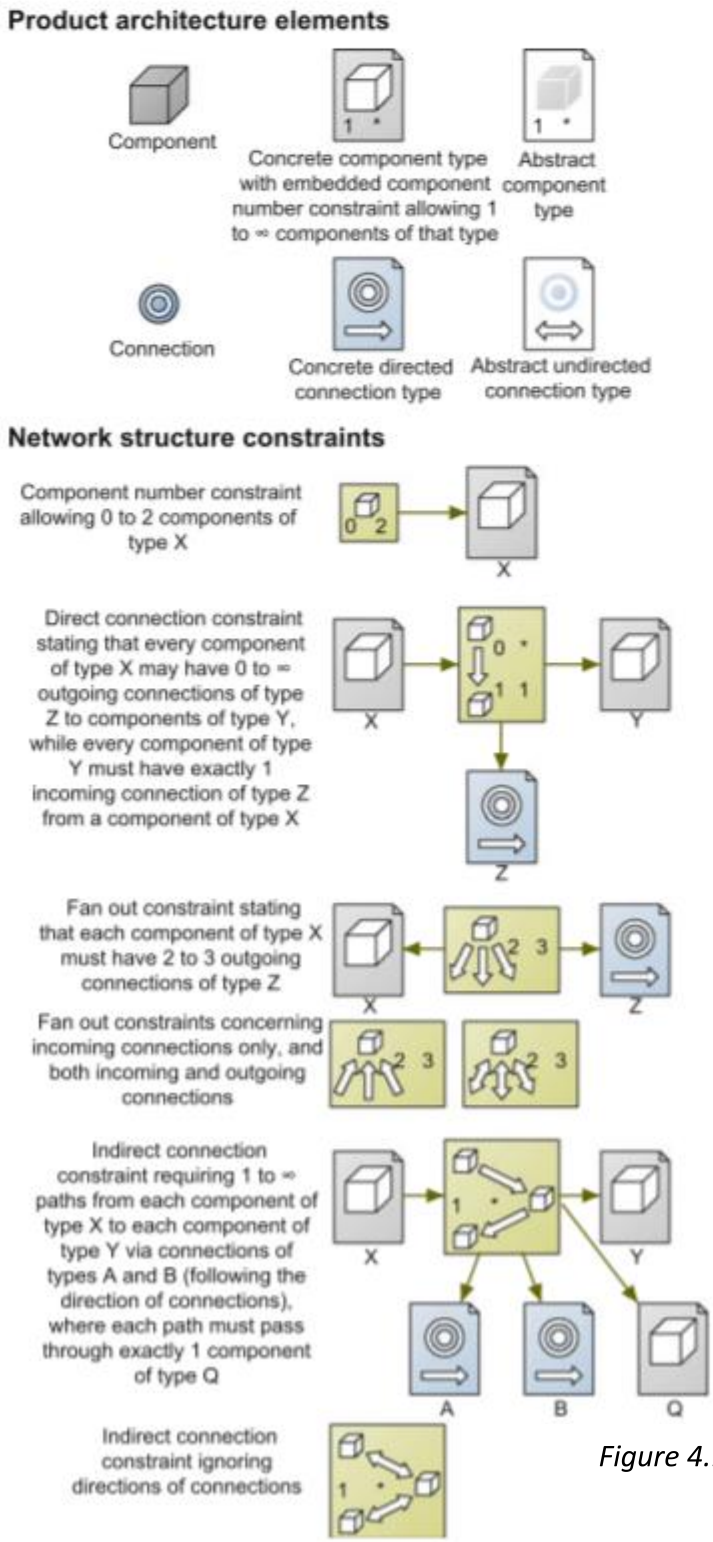

Product architecture elements
Component
1 *
Concrete component type with embedded component number constraint allowing 1 to ∞ components of that type
1 *
Abstract component type
Connection
Concrete directed connection type
Abstract undirected connection type
Network structure constraints
Component number constraint allowing 0 to 2 components of type X
0 2
X
Direct connection constraint stating that every component of type X may have 0 to ∞ outgoing connections of type Z to components of type Y, while every component of type Y must have exactly 1 incoming connection of type Z from a component of type X
0 *
1 1
X
Y
Z
Fan out constraint stating that each component of type X must have 2 to 3 outgoing connections of type Z
2 3
X
Z
Fan out constraints concerning incoming connections only, and both incoming and outgoing connections
2 3
2 3
Indirect connection constraint requiring 1 to ∞ paths from each component of type X to each component of type Y via connections of types A and B (following the direction of connections), where each path must pass through exactly 1 component of type Q
1 *
X
Y
A
B
Q
Indirect connection constraint ignoring directions of connections
1 *

Figure 4.1

And using the proposed method of computational design synthesis the came up with this structure for solution finding. The paper then concludes with a graphical and qualitative analysis of their findings.

This research paper, although not strictly related to architecture, is extremely helpful to the thesis presented in this document. This method of mapping product architecture although specific to physical products like vacuums can be applied to Buildings as well. Buildings are merely products that also require and integrated system of parts that can be refined and even solved for through this particular methodology. There are also several parallels between the structure of this process and that of the Rhinoceros Plugin Grasshopper which relies on utilizing components to link and modify data in a similar structure to that presented in this research paper. The papers conclusion of digitizing structures of architecture as a means to save time and resources is perfectly adequate in reinforcing the type of problem solving being utilized in the thesis presented in this document. Lastly, on the next page is an image from the research paper (Figure 3.2) that shows their formalization and modeling technique as applied to various physical components of the vacuum cleaner. This shows that a digitized and formalized approach to something like a Modular Housing Development is plausible and potentially feasible.

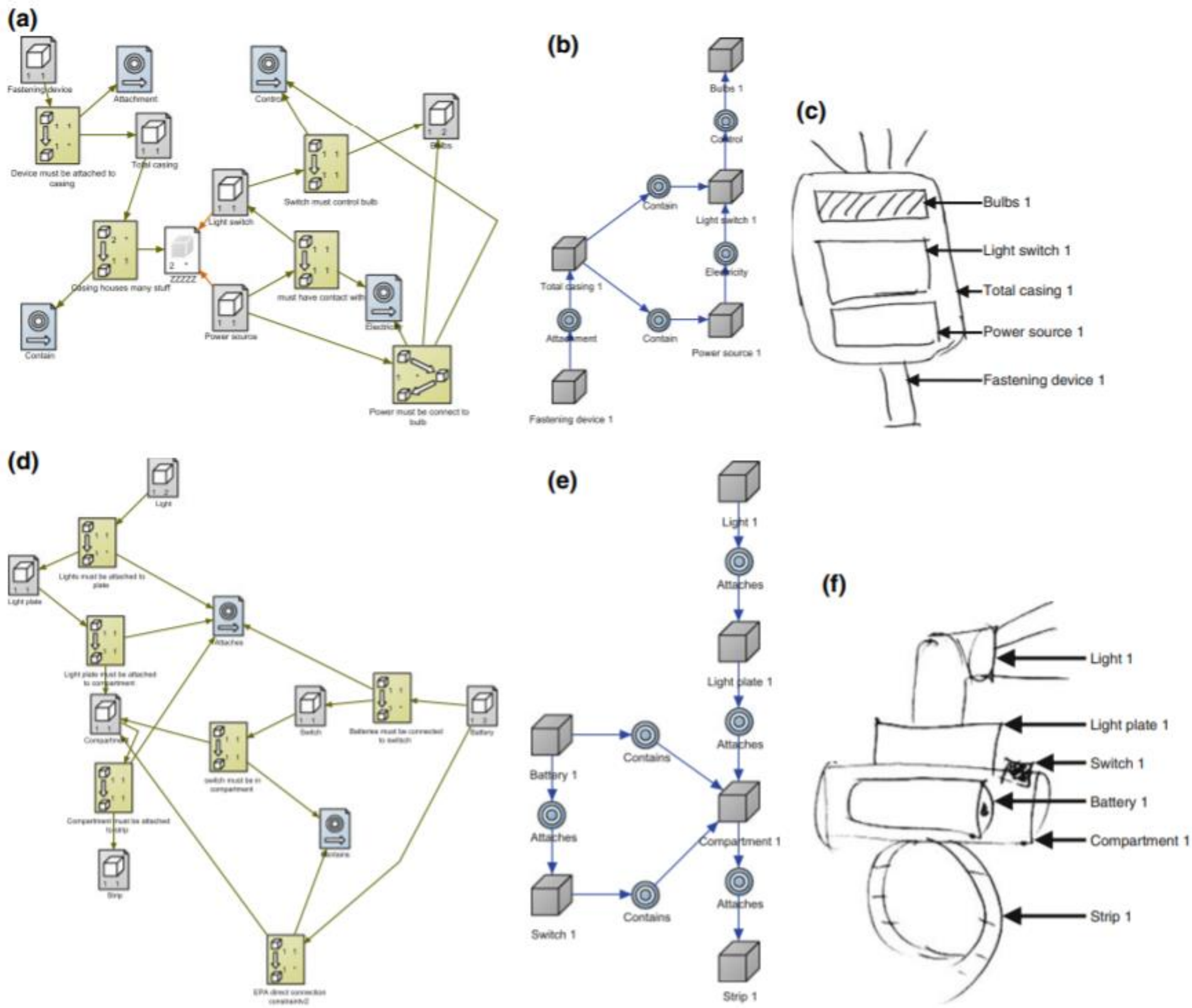

(a)
(b)
Bulbs 1
Control
Contain
Light switch 1
Electricity
Total casing 1
Attachment
Contain
Power source 1
Fastening device 1
(c)
Bulbs 1
Light switch 1
Total casing 1
Power source 1
Fastening device 1
(d)
(e)
Light 1
Attaches
Light plate 1
Attaches
Battery 1
Contains
Attaches
Compartment 1
Contains
Attaches
Switch 1
Strip 1
(f)
Light 1
Light plate 1
Switch 1
Battery 1
Compartment 1
Strip 1

Figure 4.2

SITE/CONTEXT ANALYSIS

Union Depot Rail View Picnic Area Parking Lot (394 E Kellogg Blvd, St Paul, MN 55101)

The St. Paul Union Depot has a large parking lot that is usually only sparsely populated by employees, but the space could be redesigned to accommodate a moderately sized modular housing development.

The Site is unique in its locale that it has a challenging road configuration, but the goal is that through our testing constraints we will be able to come up with an optimized unit configuration that accounts for these factors.

Figure 4.3

Figure 5.0

The Union Depot Rail View Picnic Area Parking Lot is a great place to add additional housing units due to the amenities located nearby.

1. The site is located near other housing of varying types meaning that a new alternative or housing type is healthy for its available market.

2. It is nearby to several parks.

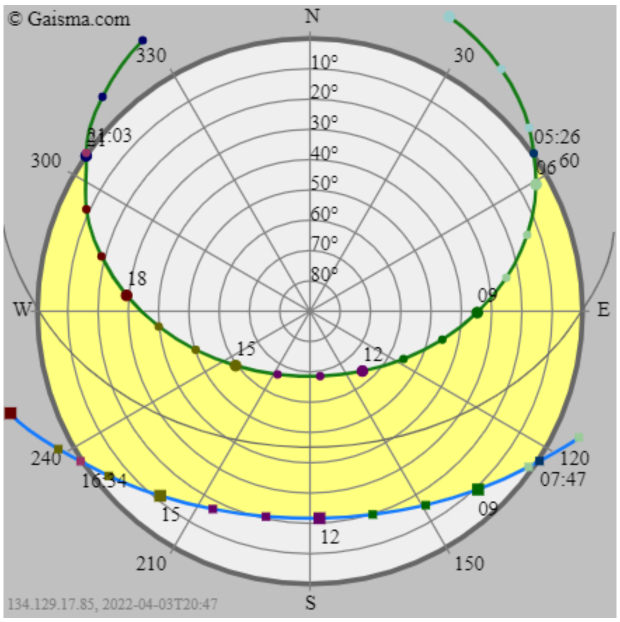
3. It is near the downtown of St. Paul which provides a place nearby for residents to find jobs.

4. The Hospital is close for health-related minutia.

5. Cultural Centers such as Museums and Education Center like the University of Minnesota and CHS Baseball Field are close.

The only downside being the railway, but this is going to become a positive, especially when taking into consideration the modular nature of this development. The railway can eventually be used to transport completed housing units to the site. All of this can be seen in Figure 5.0 Above.

SITE CLIMATE ANALYSIS



All Year Climate & Weather Averages in St. Paul

High Temp: **83 °F**

Low Temp: **9 °F**

Mean Temp: **47 °F**

Precipitation: **1.13"**

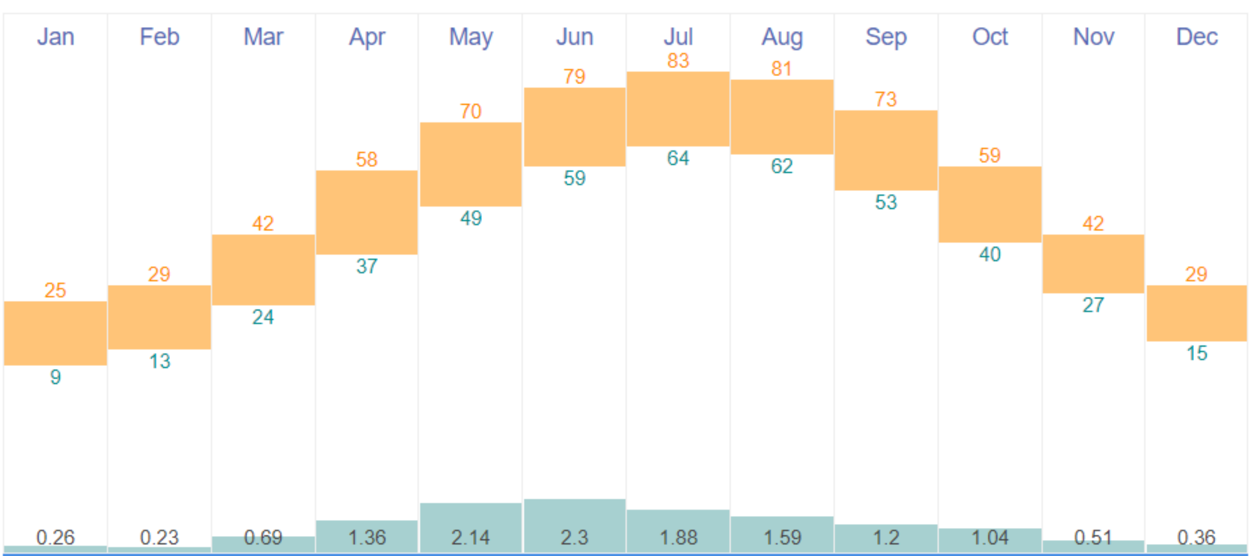
Humidity: **68%**

Dew Point: **36 °F**

Wind: **7 mph**

Pressure: **30.03 "Hg**

Visibility: **10 mi**



https://www.timeanddate.com/weather/usa/st-paul/climate

PERFORMANCE CRITERIA FOR THE PROJECT

The performance criteria for the final design outcome of the project, which is a layout of stacked units, that is being tested through program that optimizes for a series of selected Testing Constraints of my own choosing that I felt were most appropriate for the subject at hand. The many solutions that will be generated based on the program developed through Rhinoceros and Grasshopper and one will be chosen based on a score derived from the testing constraints. That score will be determined as follows in Figure 6.0 below.

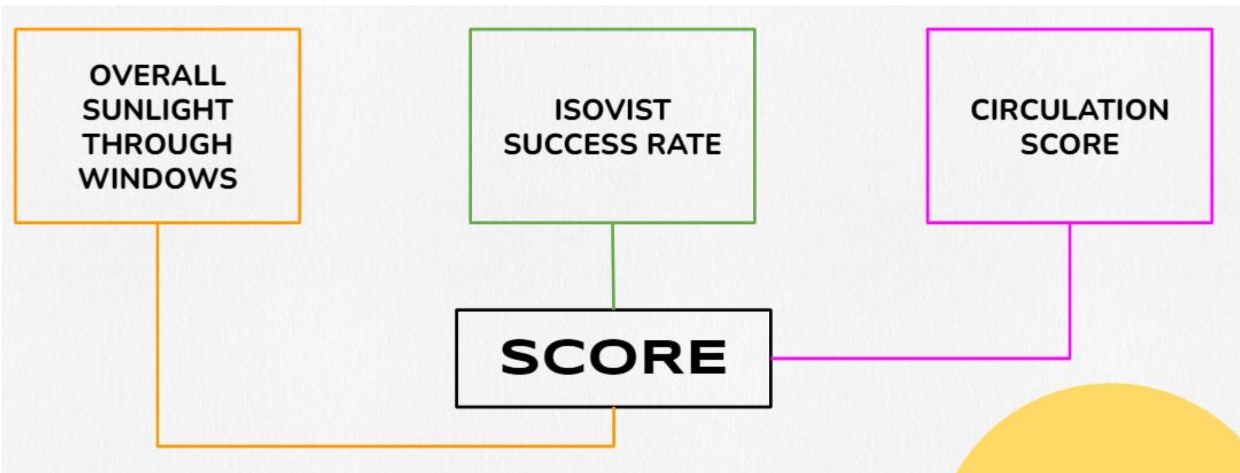


Figure 6.0

DESIGN SOLUTION

PROCESS DOCUMENTATION

Figure 7.0

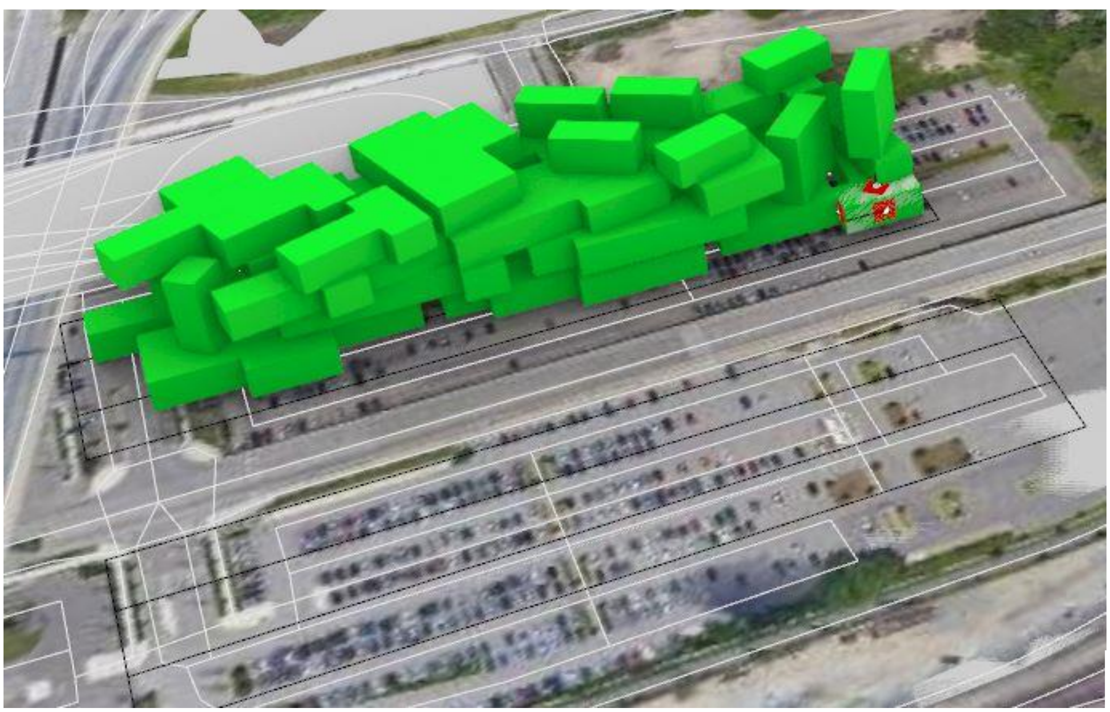


Figure 7.1

The process begins with a testing of the WASP plugin. Connection rules are generated for all connections on the currently only 1 geometry type and can rotate 360 degrees in all directions of their connections.

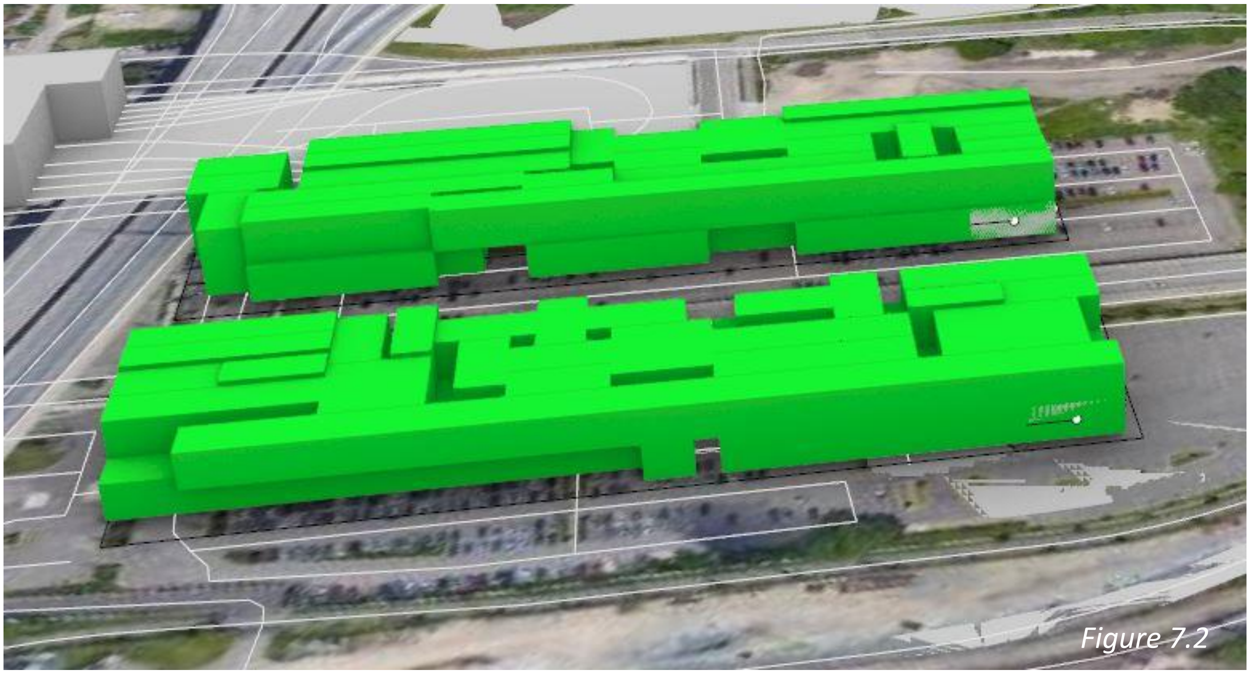


Figure 7.2

Figure 7.3

A more uniform series of connections were developed, and the scale of the site was corrected. In terms of processing if I were to proceed with this level of generation on a volume this large the generation time of the completed program would be too long with the processing power that my Laptop is capable of to get meaningful information out of it quickly.

Figure 7.4

Figure 7.5

In order to save on processing power and make sure that the program is able to generate multiple iterations of volumes that will eventually respond to each other, the volumes were shrunk and the size of a basic unit is conceptualized.

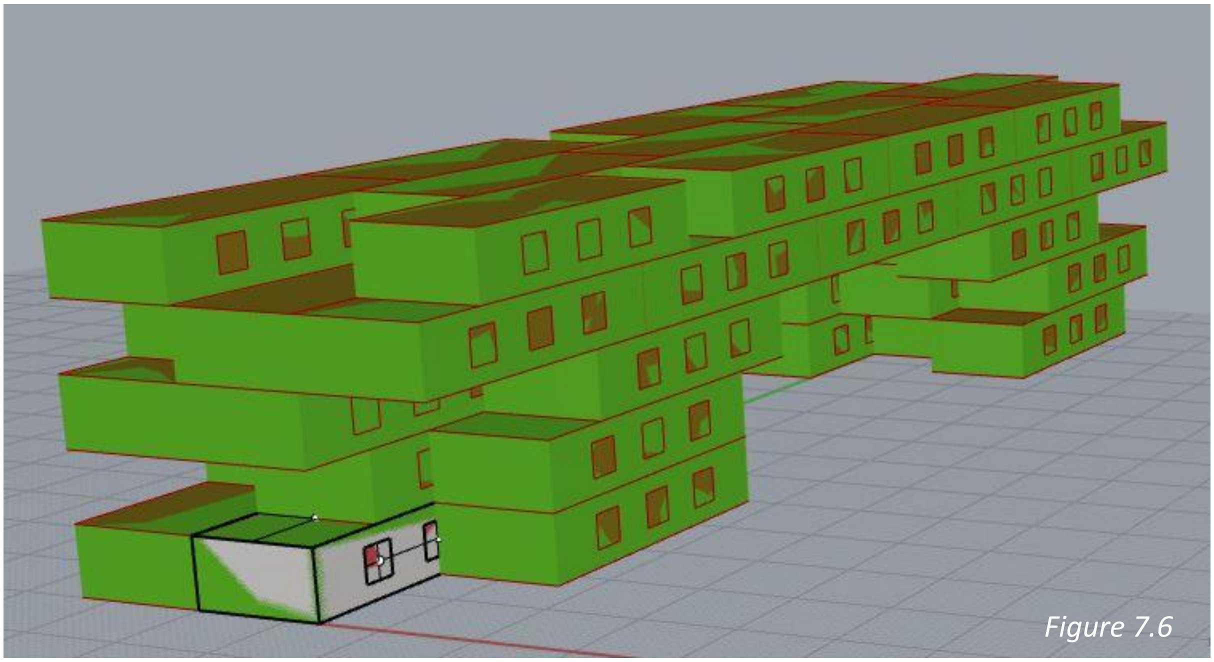


Figure 7.6

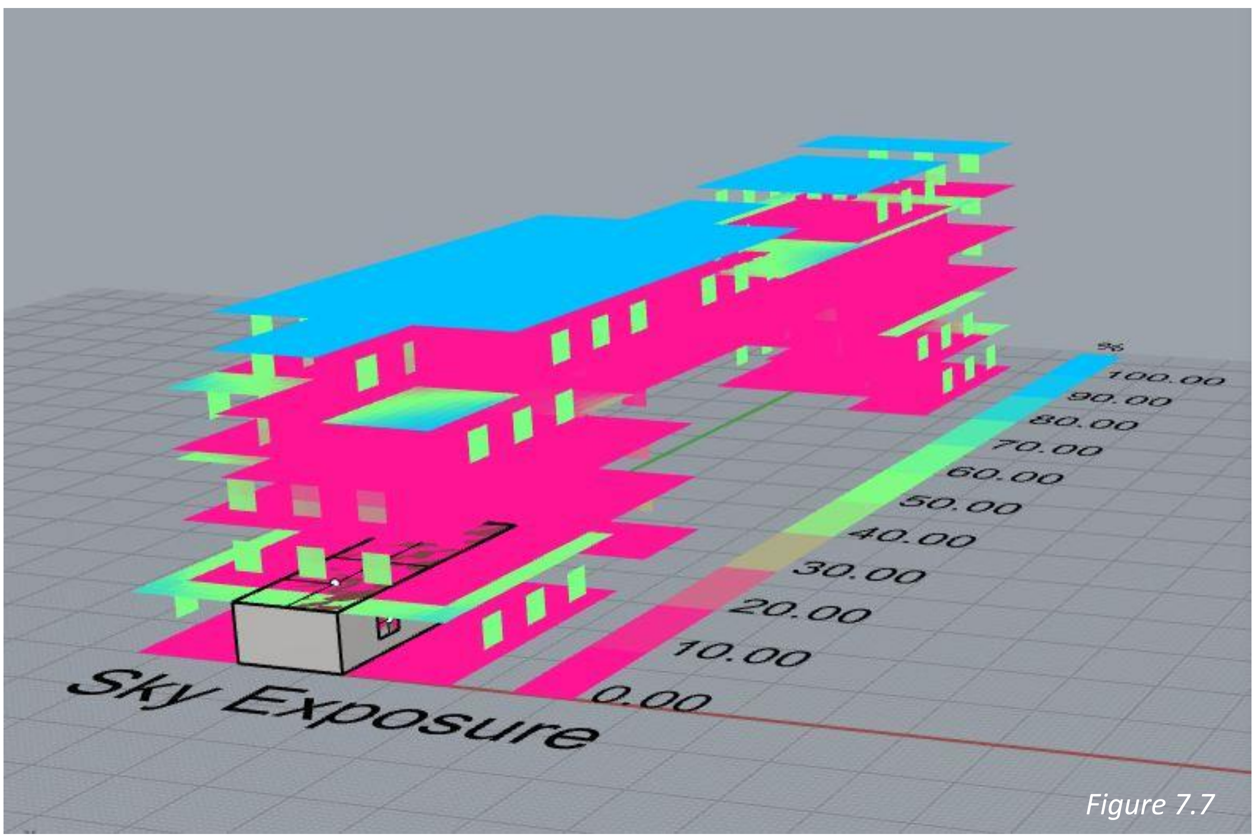


Figure 7.7

A test is run on applying Testing Surfaces to Geometries to see if the information input to one geometry could be output and collected via several geometries of the same time.

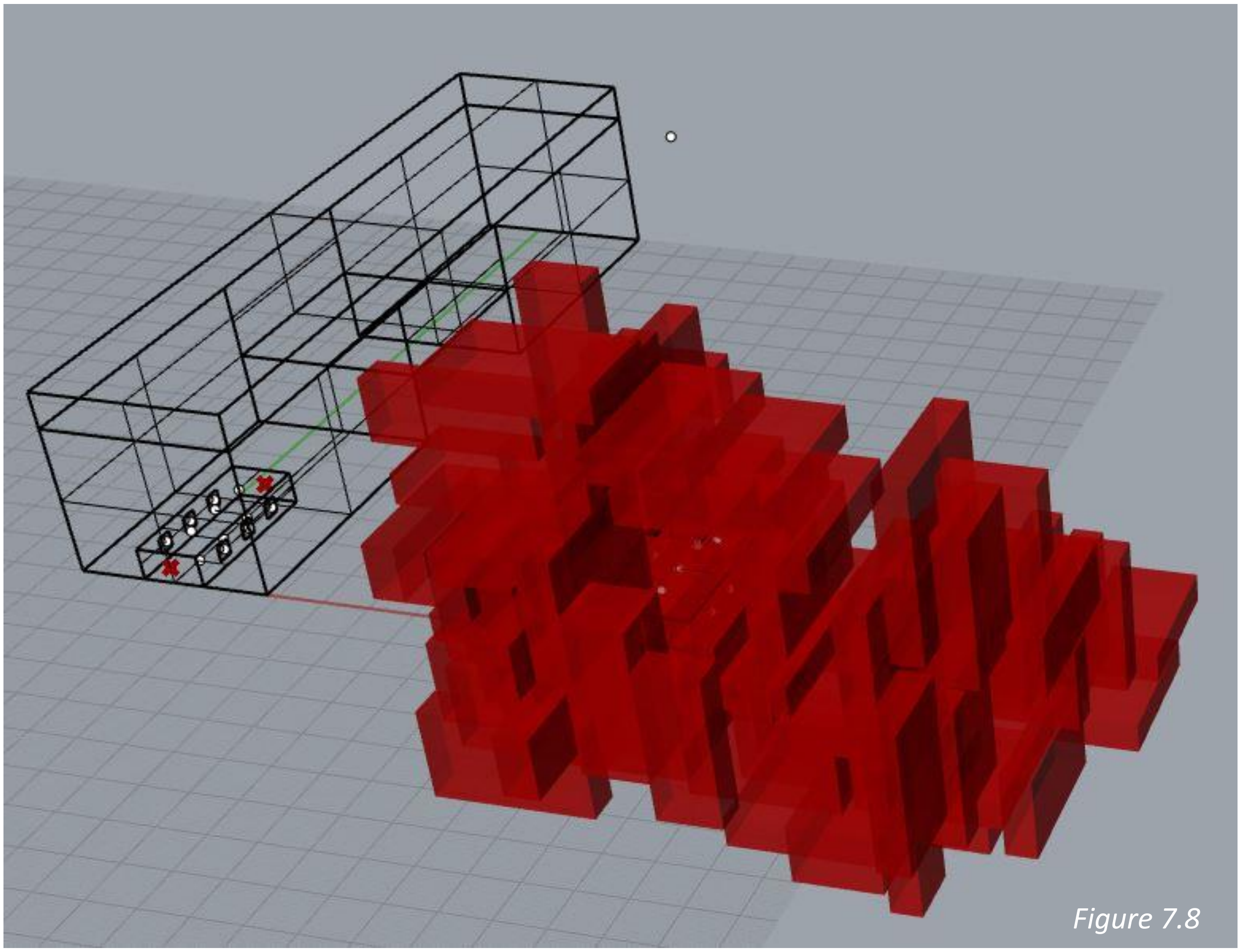


Figure 7.8

More population tests were run but an error was stumbled upon when not referencing the host volume. This was easily corrected with the addition of a Reset Button to the Grasshopper Code.

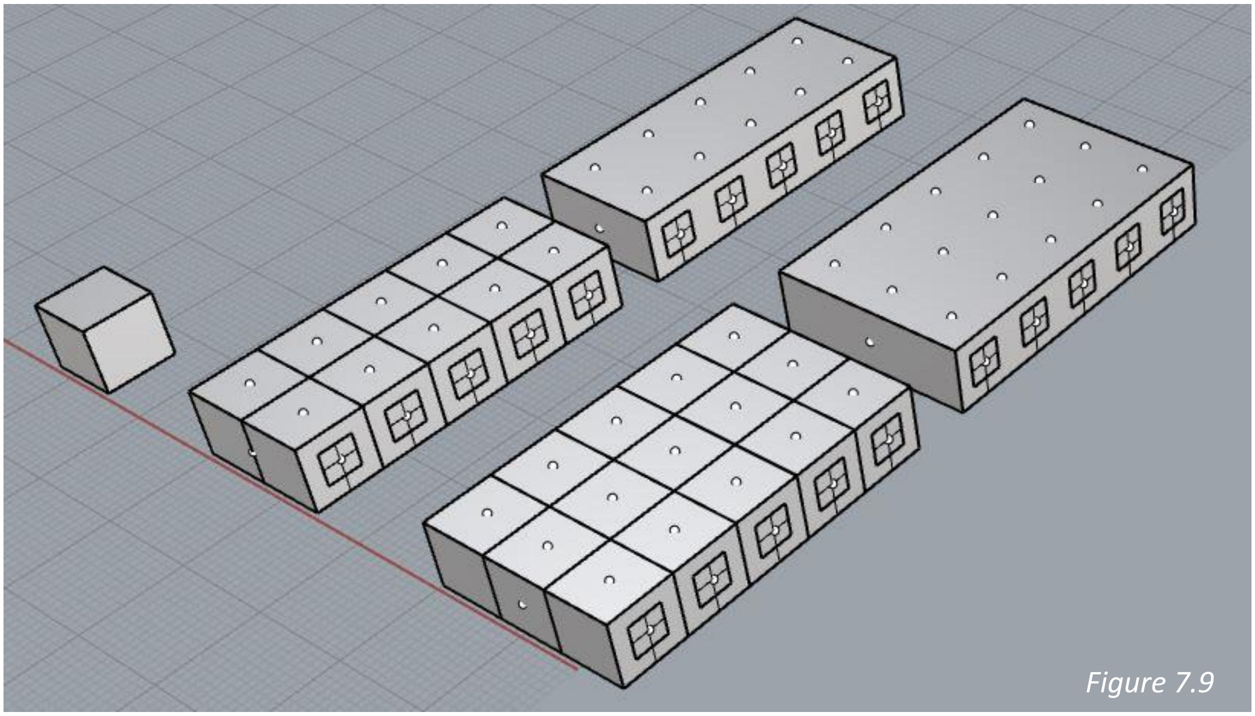


Figure 7.9

Early Concepts for the Unit Geometries that would be used in the Final Aggregations.

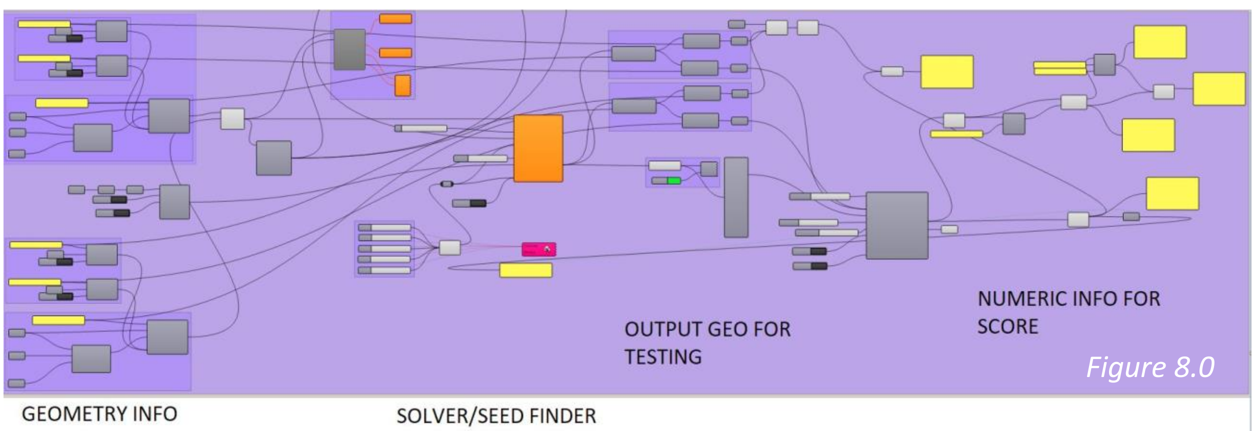


Figure 8.0

Early Code Development and organized for ease of change in the future.

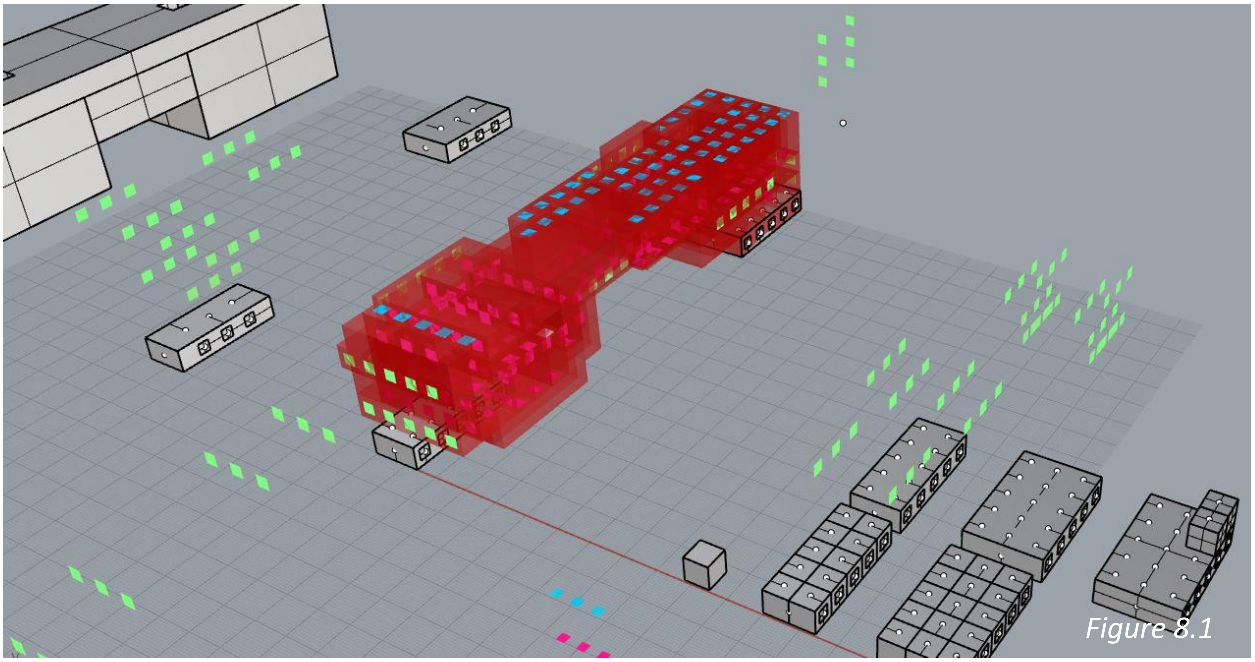


Figure 8.1

Another bug where the referenced testing surface was too far away from the associated geometry.

Figure 8.2

Volume size and spread throughout the site was edited to be more consistent and to allow for roads in between the volumes. This would be perfected later.

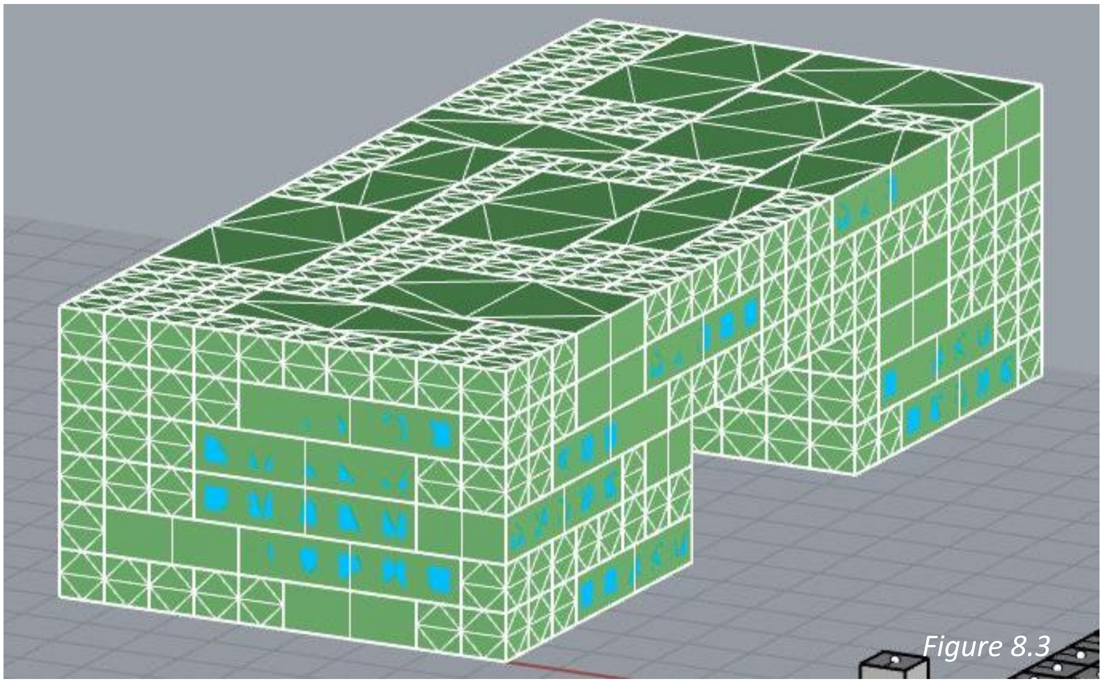


Figure 8.3

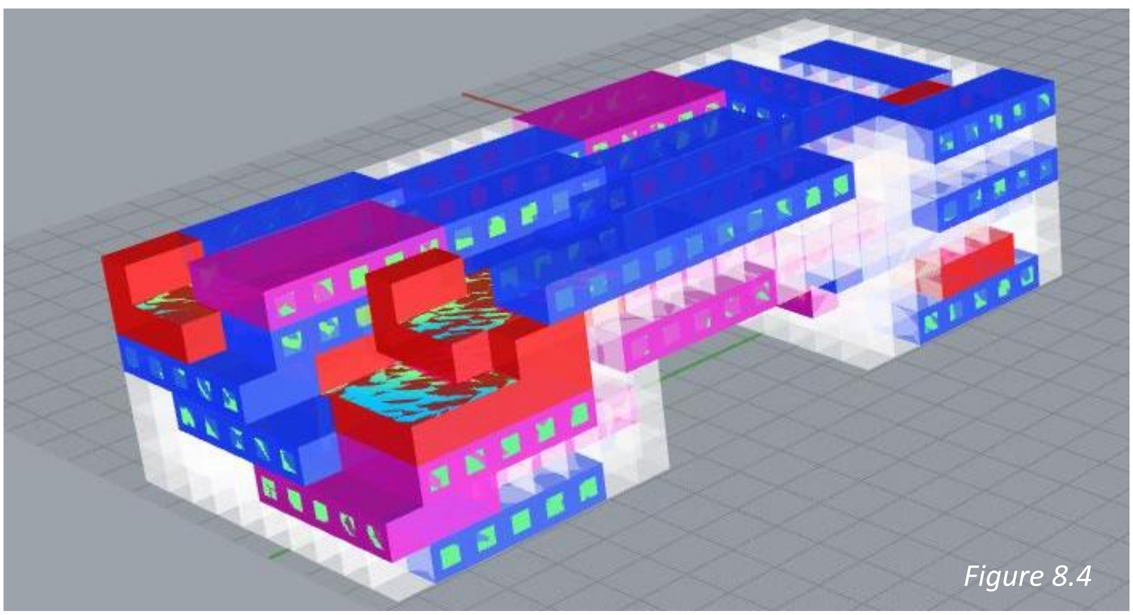


Figure 8.4

This was a Eureka moment in development as the Units were propagating throughout the geometry consistently and the testing conditions were modified in an attempt to get out desirable configurations consistently.

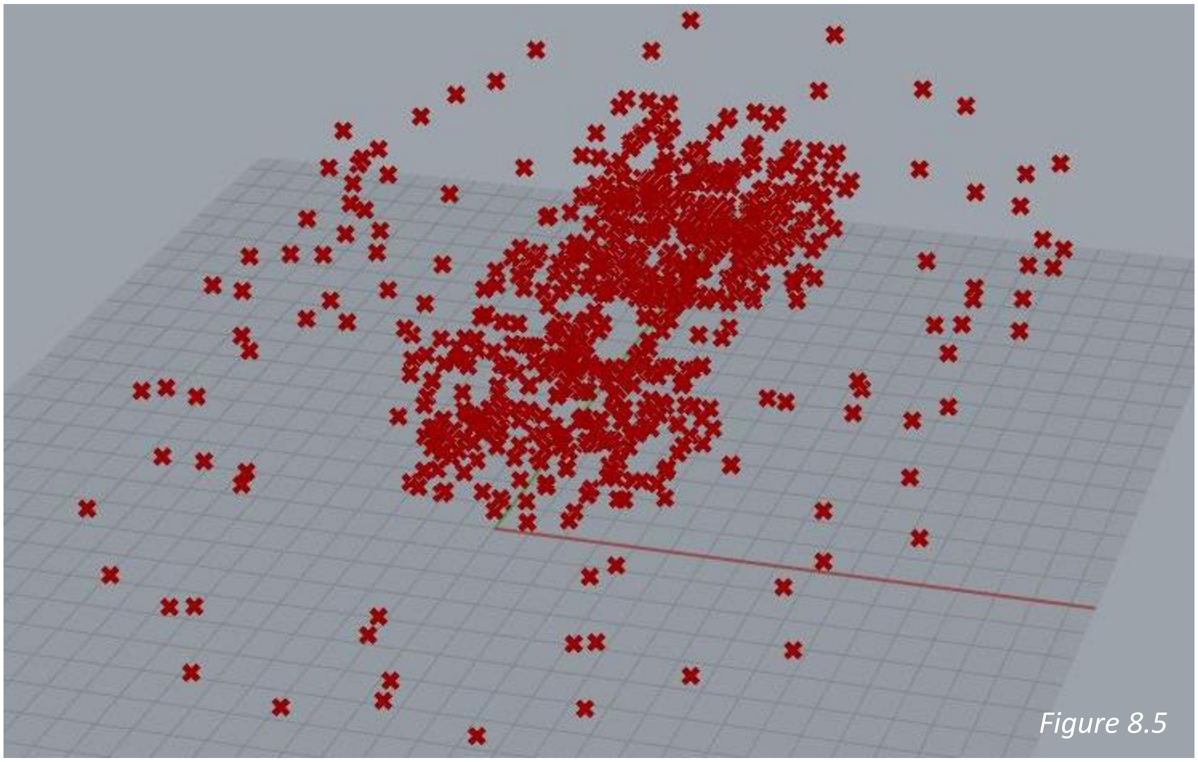


Figure 8.5

This is an example of how the Code sees Isovist information. The Success is determined by how many of the X's reach their maximum distance from their origin.

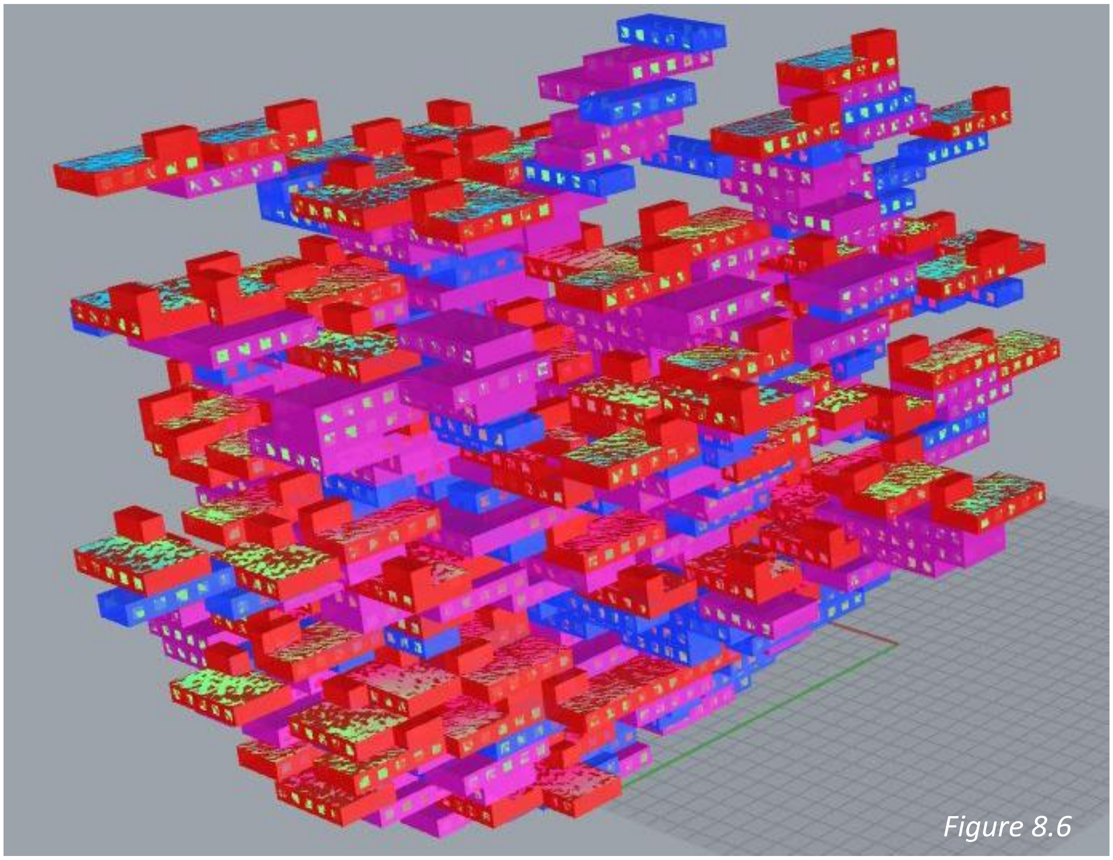


Figure 8.6

This is another example of a bug where the units were not propagating inside the host volume. But this was extremely insightful as the program here was optimizing for the sun and so the solution that it deemed the best from the seeds that the solver looked through was one that resembles a tree. This actually means that the program is functioning properly.

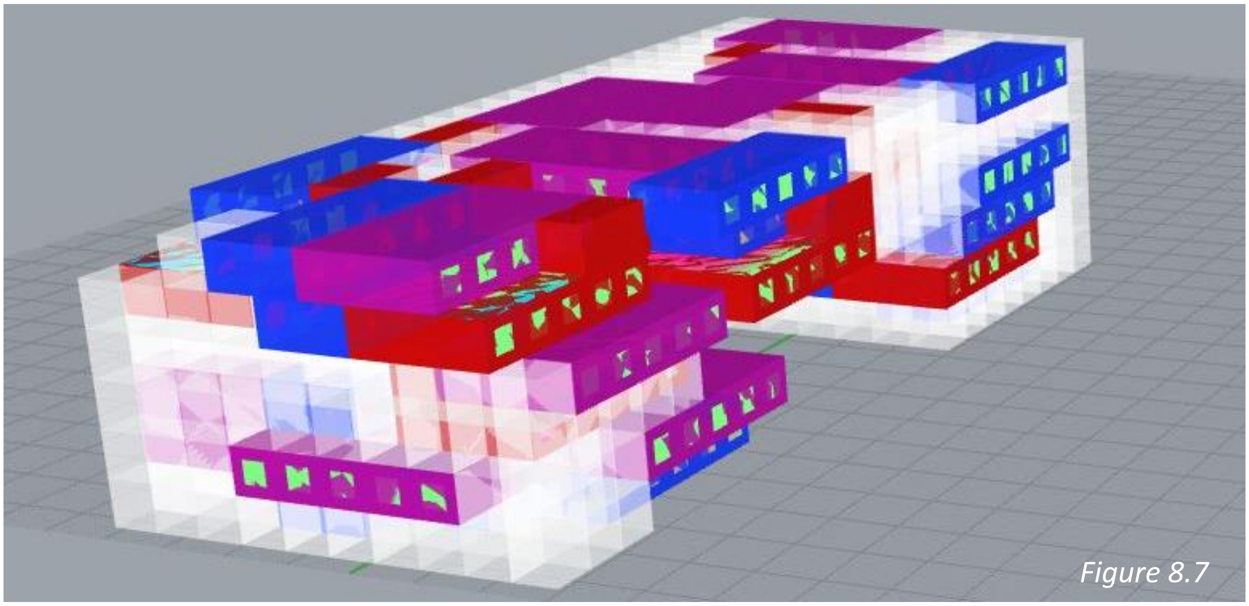


Figure 8.7

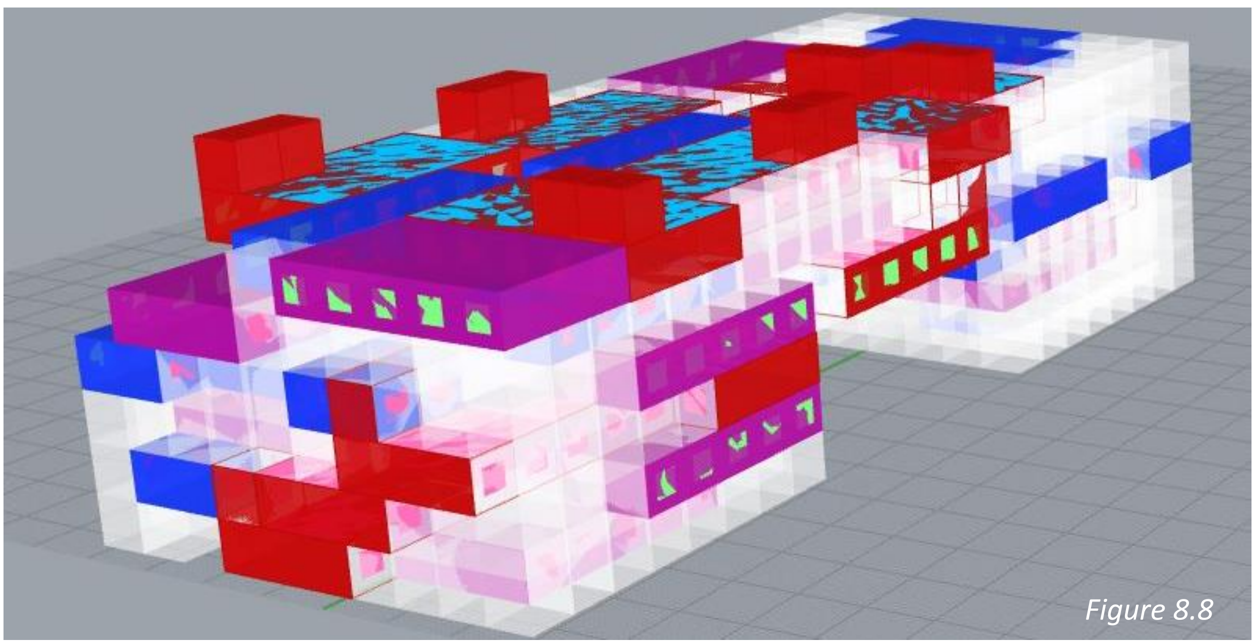


Figure 8.8

Just some minor adjustments to how the Code reads the Red Unit's volumetric information. This would ultimately be changed but was necessary to understand how unit volumes can relate to each other and what information needed to be added so that our testing surfaces could be read unimpeded.

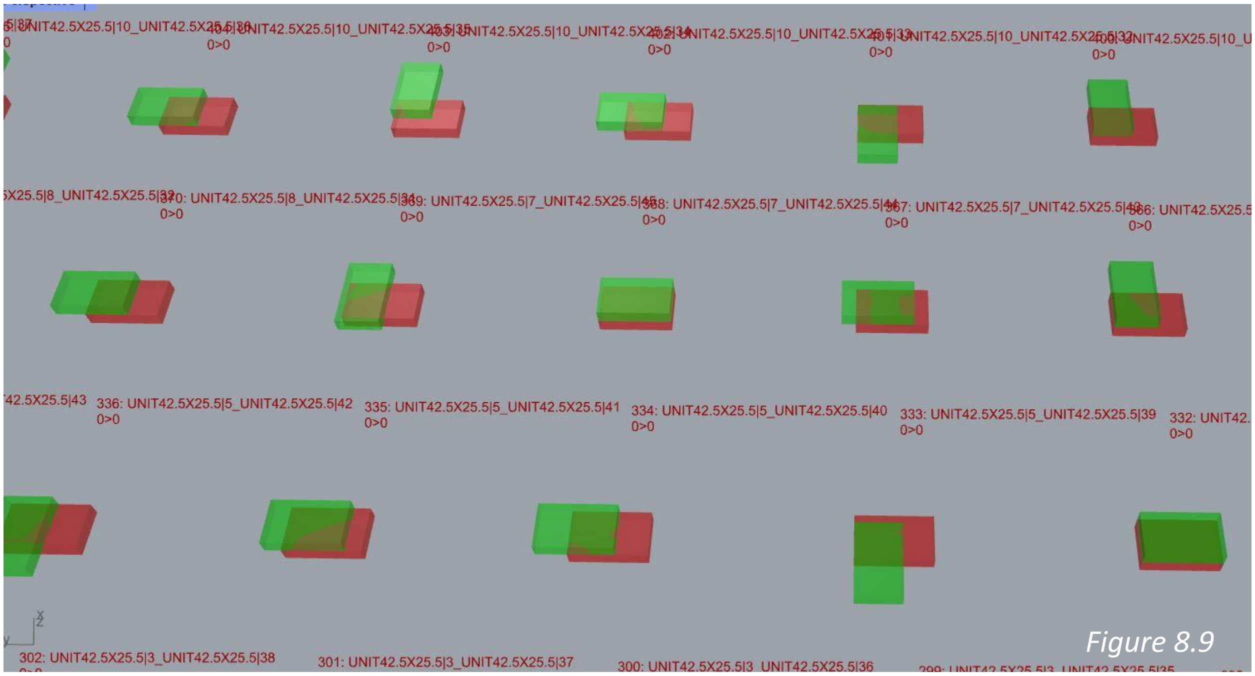


Figure 8.9

Figure 9.0

This is an example of the visualized version of the connection rules of the unit geometries. There were 1010 Rules that made were allowed to be used but only about half of them appear in the final geometries. This can be made as a note for simplification of rules and connections in the future.

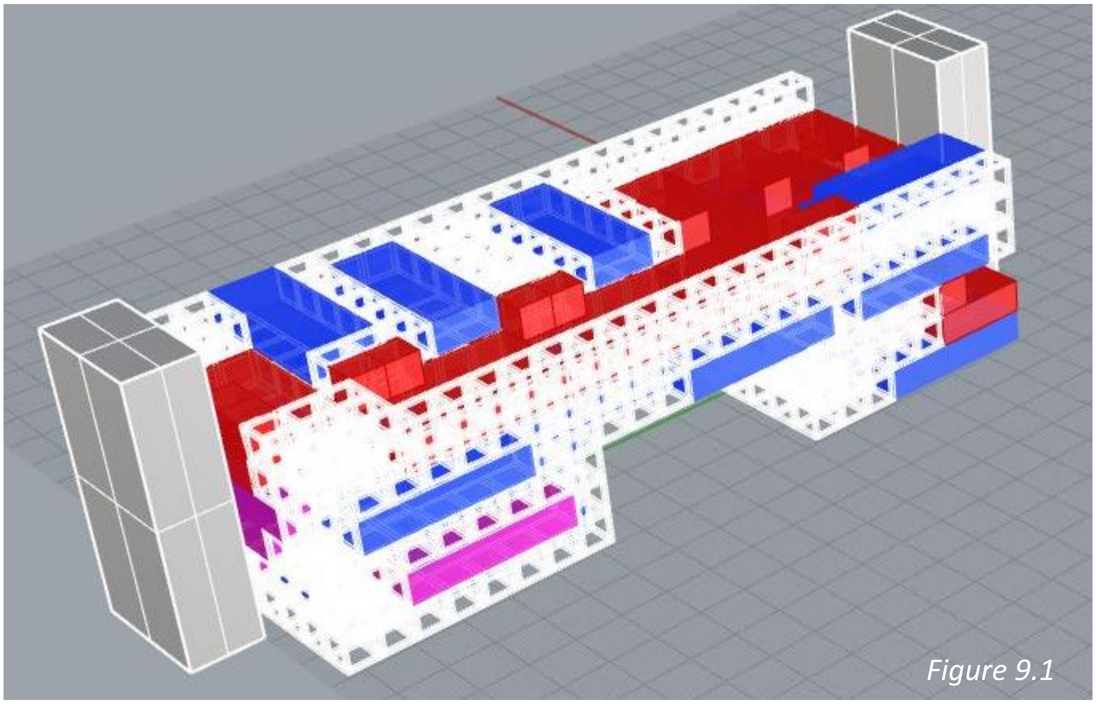


Figure 9.1

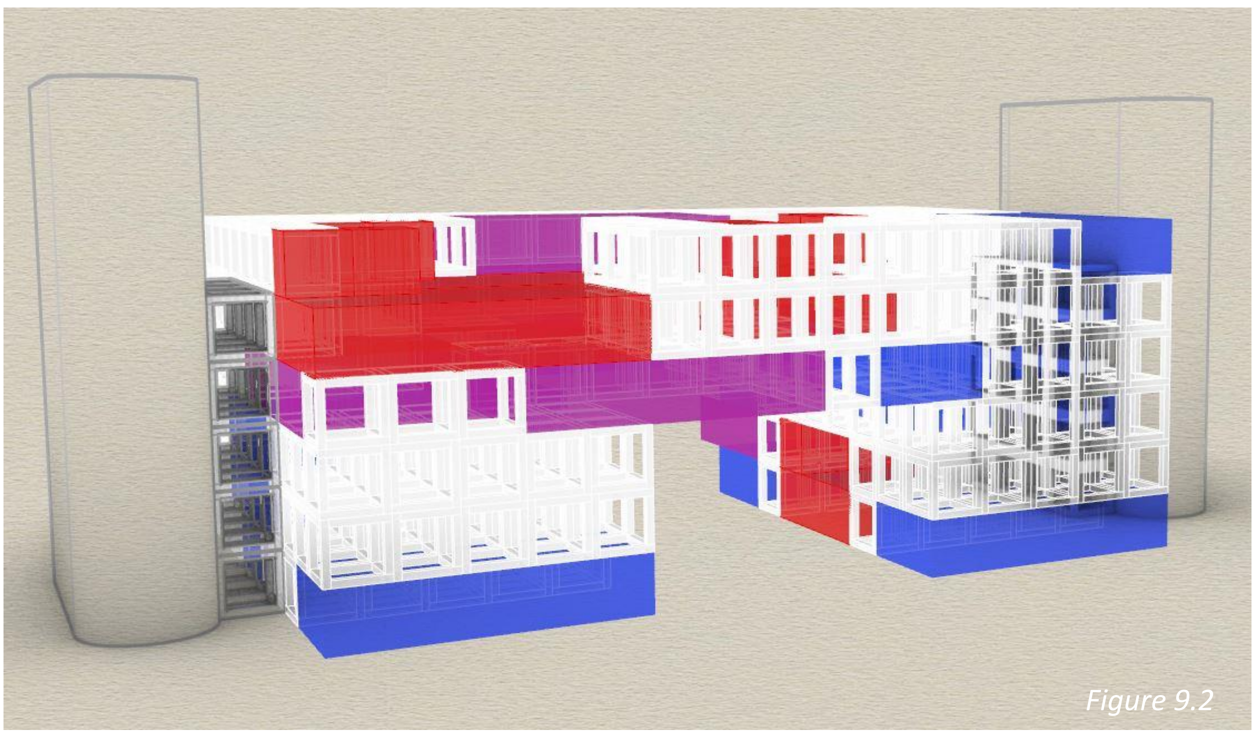


Figure 9.2

Here we have the completed program execution with added circulation towers. The final volume was shrunk significantly to fit into the site properly.

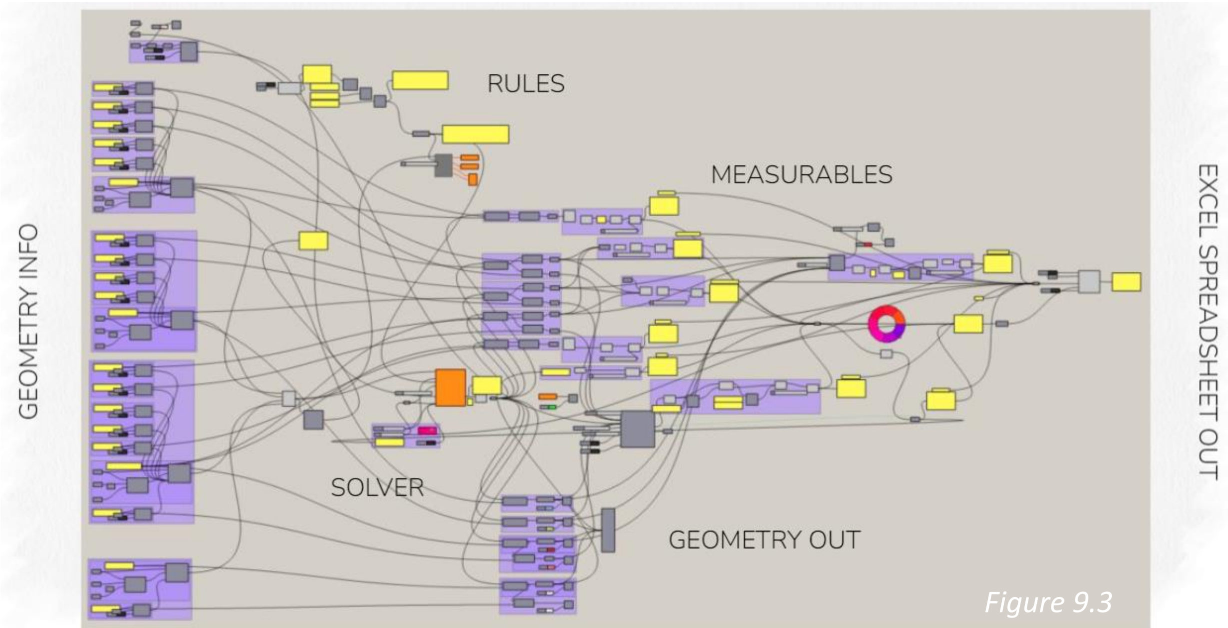


Figure 9.3

This is the final grasshopper program. It has the ability to accept geometry information of various sizes and shapes along with associated information specific to those geometries. The rules determine how the geometries connect together with an extreme level of inclusion or exclusion. The program can test for certain properties that are associated with the output of the geometries inside (or outside) a volume. The solver will search for the solution that outputs the best score from the measurables. Additionally, the score information can be exported to an excel file.

PERFORMANCE ANALYSIS: RESPONSE TO PRECEDENT RESEARCH

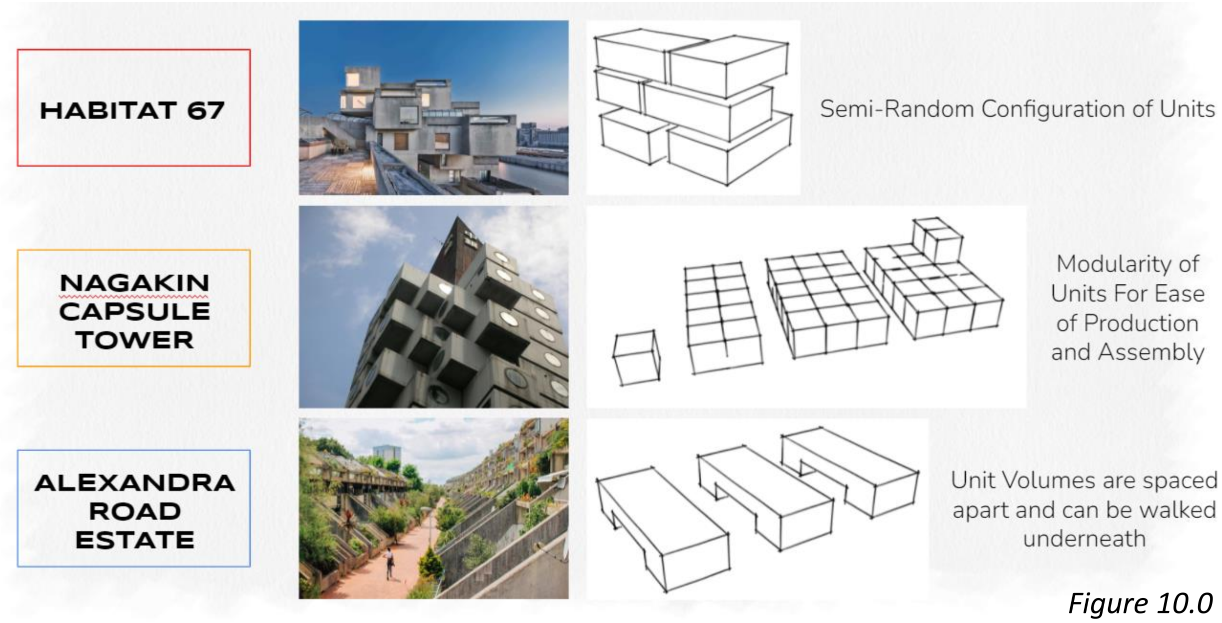


Figure 10.0

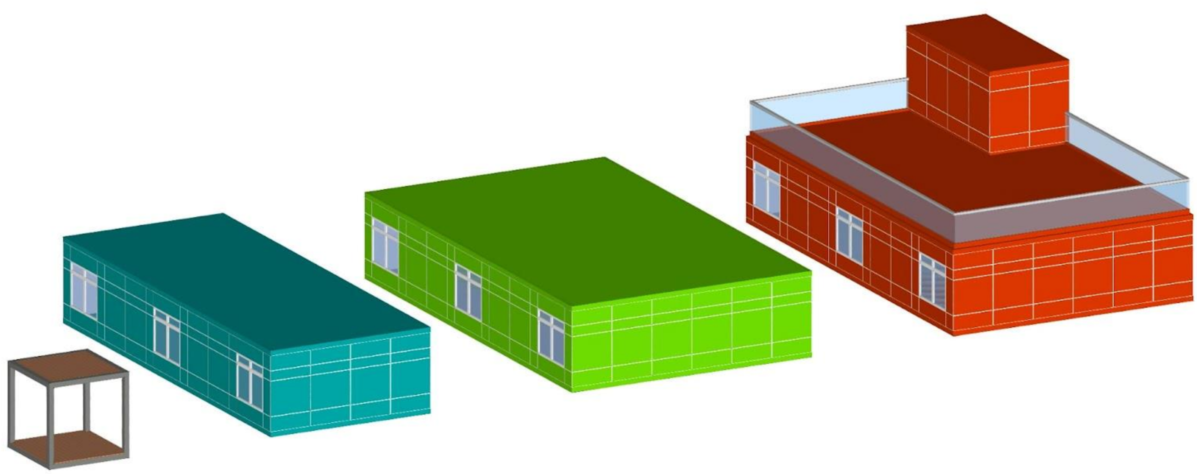
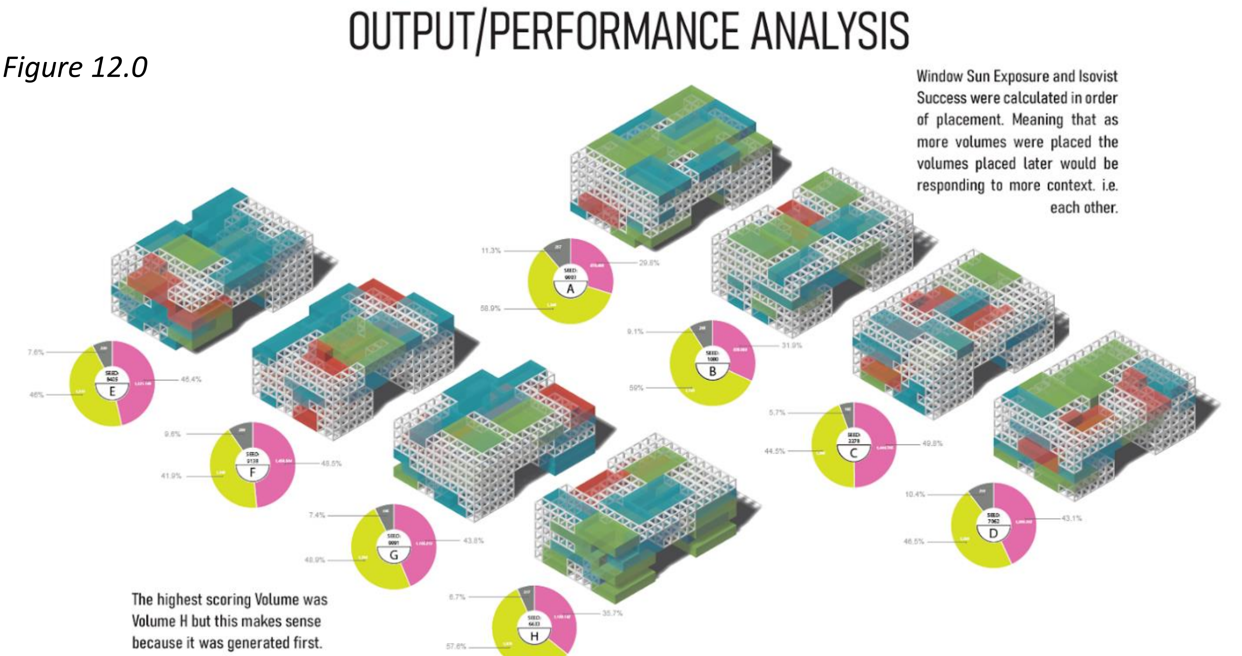


Figure 11.0

The goal was to apply these determined principles gathered from the precedent case studies. I think this was done successfully in terms of form with the exception of the negative space seen in Habitat 67 but the constraints of circulation in my programming meant that the volume would be as full as possible.

Figure 12.0



VOLUME	SEED	SUN EXPOSURE	ISOVIST SUCCESS	CIRCULATION	ENTRANCE DISTANCE	BLUE HEIGHT	RED HEIGHT	DISTANCE FROM CENTER	TOTAL SCORE
A	9993	676.405888	1340	257	1729.046012	76.015558	25.512085	414.836424	3842.410079
B	1080	938.999494	1740	268	1605.288808	78.567371	68.02417	395.250917	5094.13076
C	2278	1444.704705	1290	165	1151.25182	95.142371	170.04834	326.175552	4642.322788
D	7062	1280.302152	1380	310	1212.593751	100.249622	136.036255	341.159887	4760.341667
E	8405	1221.16859	1210	200	2154.614919	50.501056	93.52417	457.014922	5386.823657
F	9138	1458.503952	1260	289	1202.558016	61.986932	229.57251	346.765509	4848.386919
G	9991	1155.213012	1290	195	1459.197697	81.113745	178.54834	369.225846	4728.29864
H	6633	1159.142374	1870	217	1851.149662	74.738745	93.52417	421.724282	5687.279233

As each volume was generated the next responded to the "locked-in" Geometry. This means that any long shadows or views obstructed were accounted for in the subsequent volume generated. This is a very acute way of responding to the site as the site evolves as more information is added to it.

PERFORMANCE ANALYSIS: RESPONSE TO GOAL + PROJECT EMPHASIS

The Goal was to create a series of volumes that aggregate with units of 3 types that are modularly based and respond to testing data.

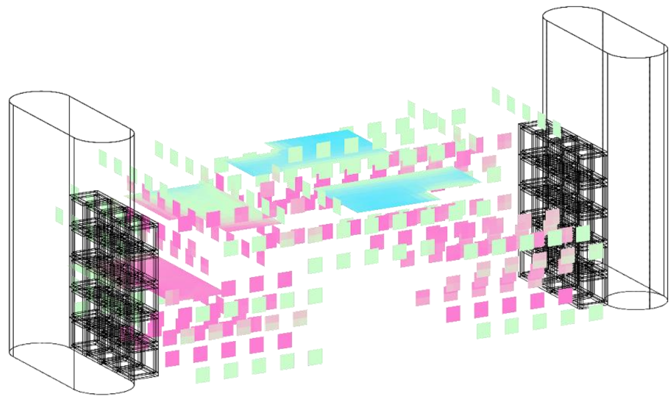


Figure 13.0

Sunlight Exposure Data

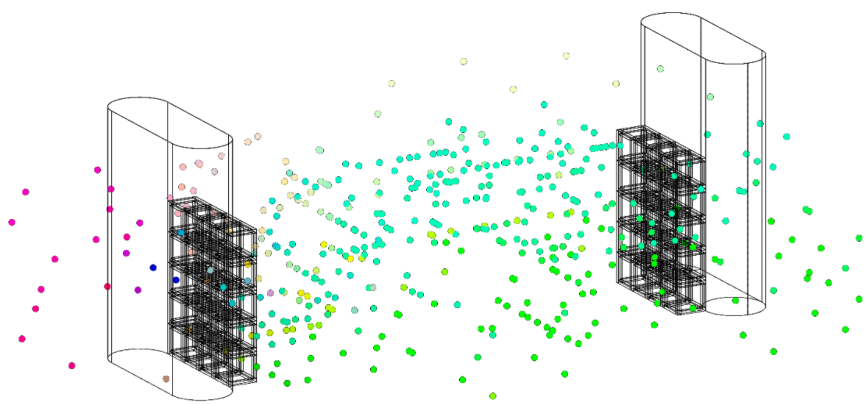


Figure 14.0

Isovist Testing Data

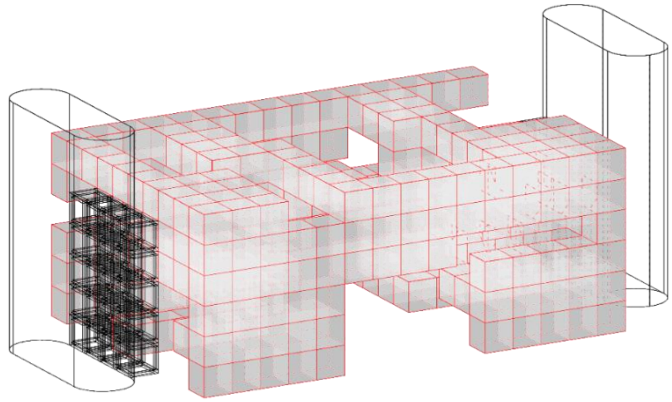


Figure 15.0

Circulation Data

I think that the thesis goal was met adequately. There were other points of data that the volumes responded to and they can be seen in Figure 12. There is of course always room for refinement and improvement.

CRITIQUE OF APPLIED RESEARCH METHODS

I think the way that the design decisions were derived from the Typological Studies were effective especially in relation to the Metabolist movement of the 1960's. The design established in the final iterations is a modern version of modular architecture with a basis in computational processes. Researching computational methodologies and research papers was integral to developing an effective programmatic process for generating these units and volumes. The majority of problems came from inadequately preparing for circulation parameters of the generated units. If I were to continue developing this process I would ensure that the units are only allowed to connect to the circulation geometry on their sides and no other geometry would be allowed to connect in that way. This would generate a solution where every unit always has a buffer of circulation around it and would prevent the overcrowding of units.

DIGITAL PRESENTATION

THESIS STATEMENT

We live in an age of data and technology. It is up to us to use that data and technology responsibly and in ways that give back to the human community. There have been major technological advances in Robotics, SMART Devices, Artificial Intelligence Models, and even Autonomous Vehicles, all of which have made our lives more efficient and brought us closer and more aware of those even on the other side of the planet, but what about Architecture? Energy modeling and structural integrity simulations have certainly made the buildings that we design more efficient and stronger than ever. But how can Architects utilize generative processes to output iterative design solutions?

Inspired by projects such as Habitat 67 by Moshe Safdie, Nagakin Capsule Tower, and Alexandra Road Estate; The Computational Community project aims to explore how Modular Housing can be generated through existing computational programming tools.

This Thesis aims to focus on the process of designing an Aggregation Based Modular Housing Development facilitated by the use of Discrete Modeling tools to optimize for various points of data. The primary data points that were chosen for the simulation in this thesis include maximizing sun exposure to windows of the various unit types and maximizing views from said units.

PERSONAL AREAS OF INTEREST:

- Computational Design
- Generative Design
- Data Driven Design
- Modular Construction Methods
- 3D Printed Construction

EXPLORING THE METABOLIST ARCHITECTURE MOVEMENT

Originating in 1960's Japan.

Metabolism - describes the process of maintaining living cells.

The movement focused on the built environment as something that is not static and is in constant change.

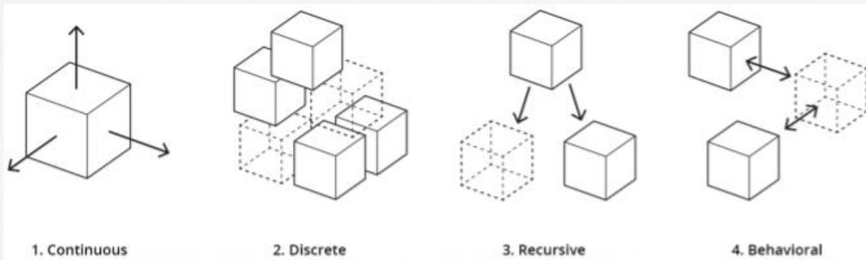
Demolition of NAKAGIN Capsule Tower right now.

NAKAGIN CAPSULE TOWER

A mixed-use residential and office tower designed by architect Kisho Kurokawa and located in Shimbashi, Tokyo, Japan.

HABITAT 67

A model community and housing complex in Montreal, Quebec, Canada, designed by Israeli-Canadian architect Moshe Safdie.



Sequential: 1,2,3...

Aware of State Changes (Not Continuous)

Remembering State Changes

Responding to Internal/External Data

USING COMPUTATIONAL DESIGN STRATEGIES

Definition of *aggregation*

1 : a group, body, or mass composed of many distinct parts or individuals
// A galaxy is an *aggregation* of stars and gas.

How can Computational Programs and **Aggregation-based** design tools be utilized in the creation of a **Modular** Housing Development?

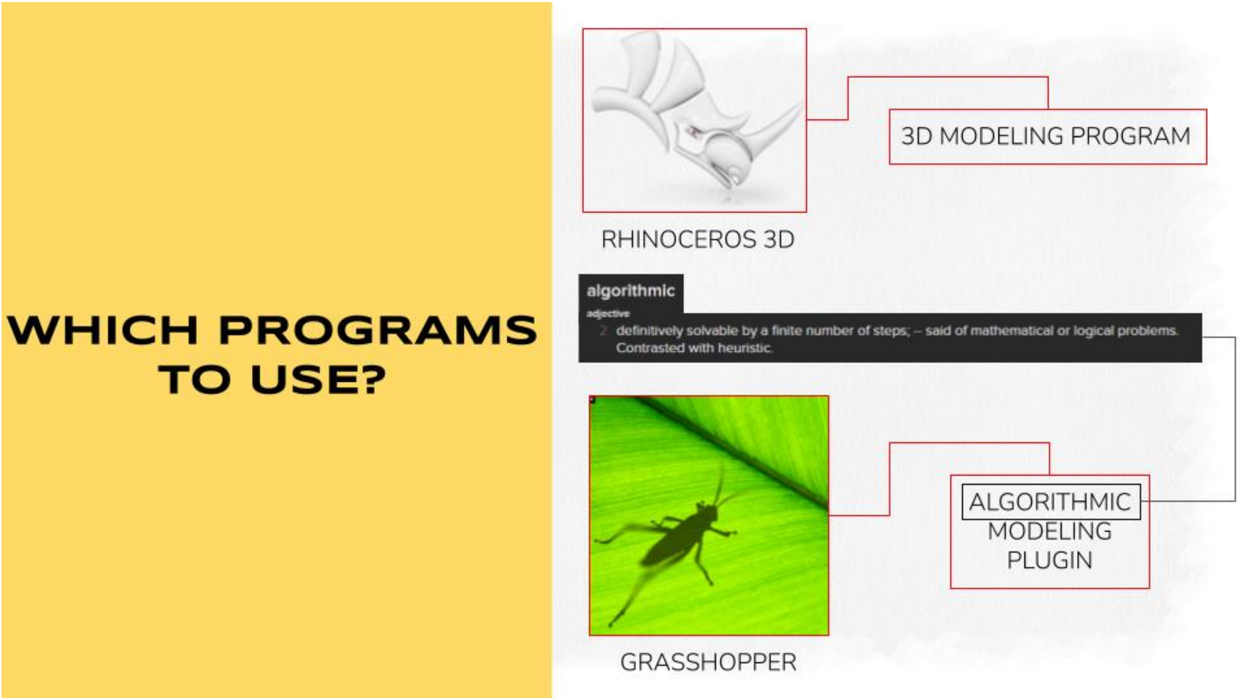
Definition of *modular*

5. Relating to, exhibiting, or being a pattern of growth in which similar morphological units, called modules, are added repeatedly.

THESIS GOAL

Additionally, can the **configuration** of the modular architecture be optimized to meet a series of design constraints through weighted outputs?

https://archinect.com/news/article/150170940/l-a-architects-are-embracing-modular-multi-family-housing


WHICH PROGRAMS TO USE?
3D MODELING PROGRAM
RHINOCEROS 3D
algorithmic
adjective
2. definitively solvable by a finite number of steps; -- said of mathematical or logical problems. Contrasted with heuristic.
ALGORITHMIC MODELING PLUGIN
GRASSHOPPER


discrete
2. Consisting of unconnected distinct parts.
3. Defined for a finite or countable set of values; not continuous.
RHINOCEROS 3D
PLUGINS
WASP PLUGIN
DISCRETE MODELING PLUGIN
GALAPAGOS PLUGIN
GENETIC SOLVER/ANNEALING SOLVER

DESIGN METHODOLOGY OVERVIEW

The new context allows for a blank slate where the desired principles can be applied. Goal: (Modular Housing Development)

Using Rhinoceros 3D + Grasshopper + Wasp + Various other plugins a new program will be developed that can generate a desired Unit Configuration based on the testing constraints.

Generate Outputs from program to determine various SCORES.

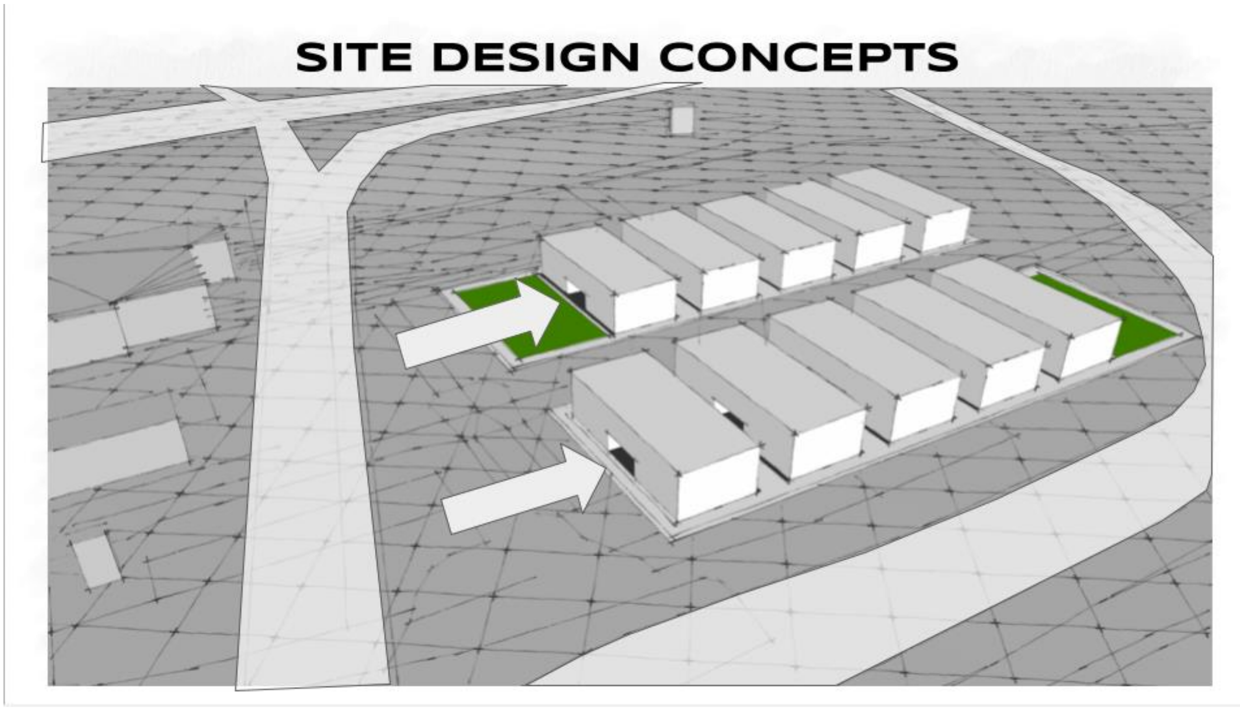
SITE

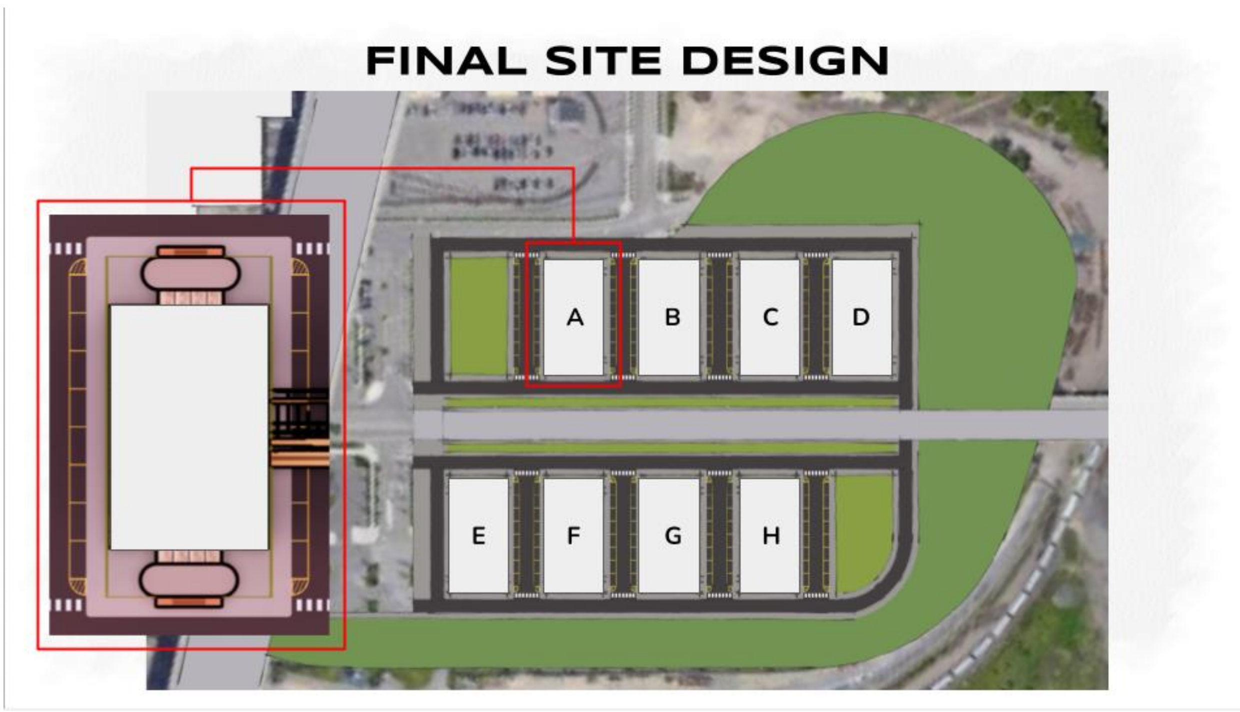
The St. Paul Union Depot has a large parking lot that is usually only sparsely populated by employees but the space could be redesigned to accommodate a moderately sized modular housing development.

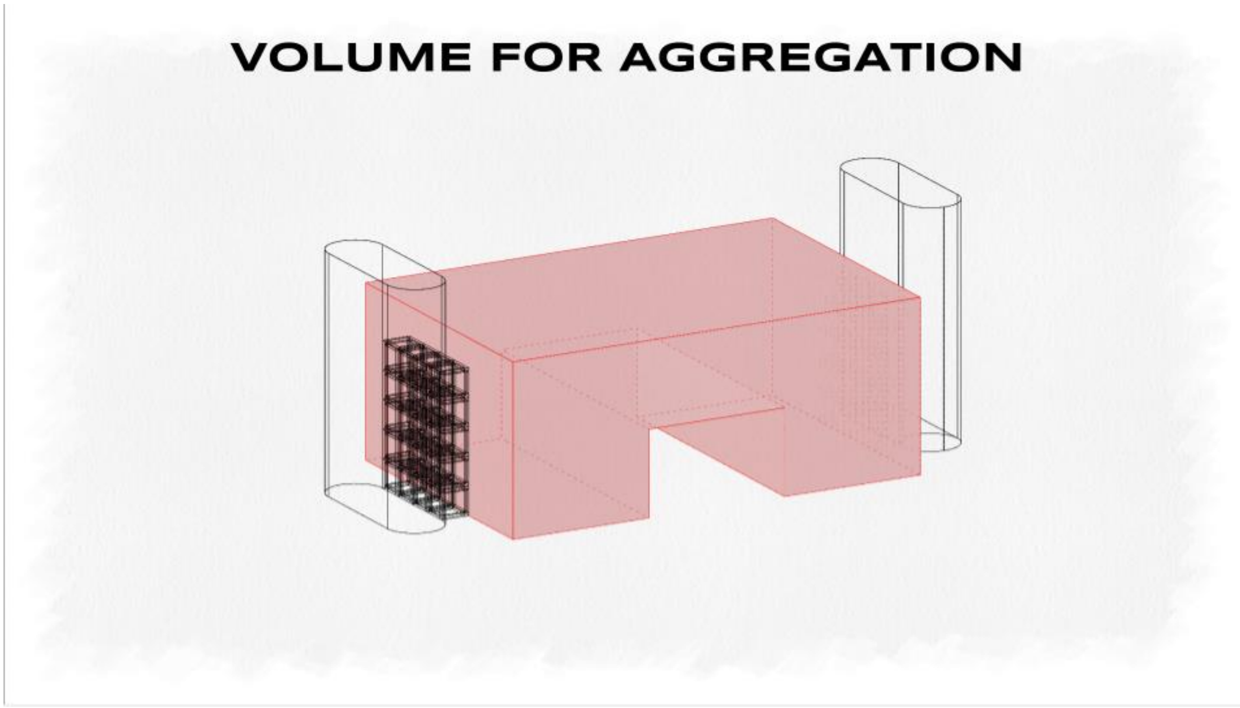
The Site is unique in its locale that it has a challenging road configuration but the goal is that through our testing constraints we will be able to come up with an optimized unit configuration that accounts for these factors.

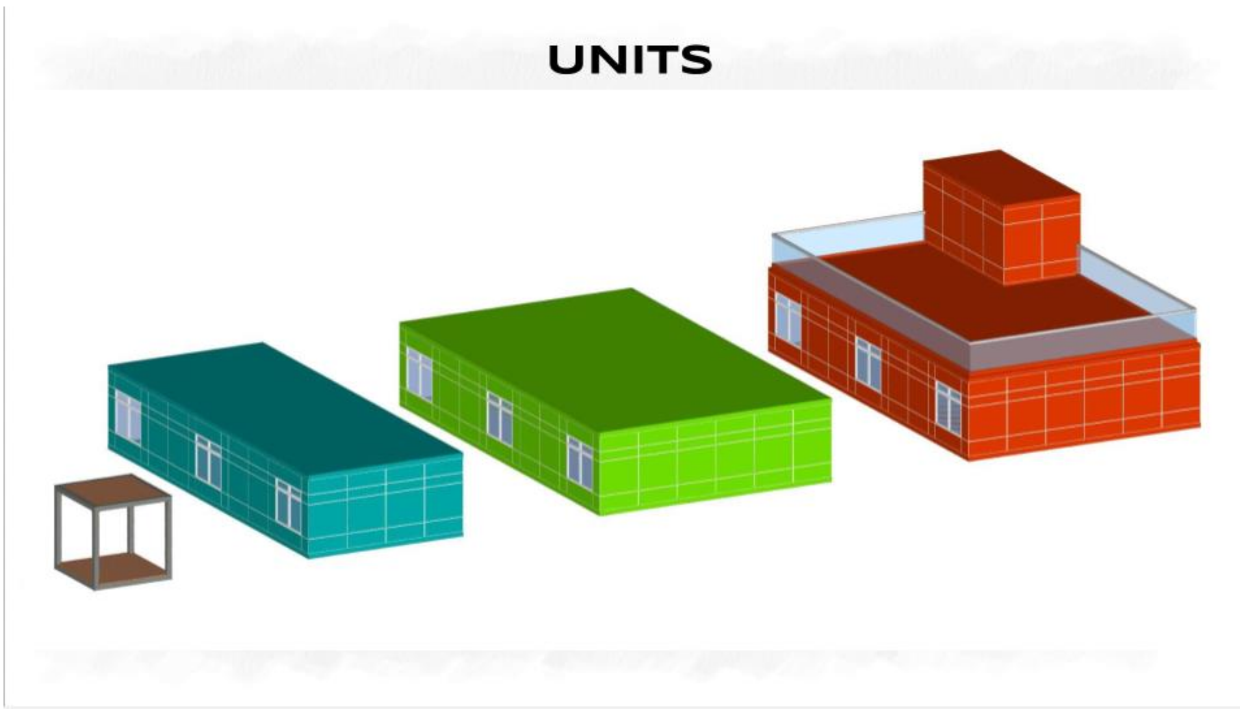

DESIGN INSPIRATION FROM CASE STUDIES
HABITAT 67
Semi-Random Configuration of Units
NAKAGIN
CAPSULE
TOWER
Modularity of Units For Ease of Production and Assembly
ALEXANDRA
ROAD
ESTATE
Unit Volumes are spaced apart and can be walked underneath


SITE DESIGN CONCEPTS

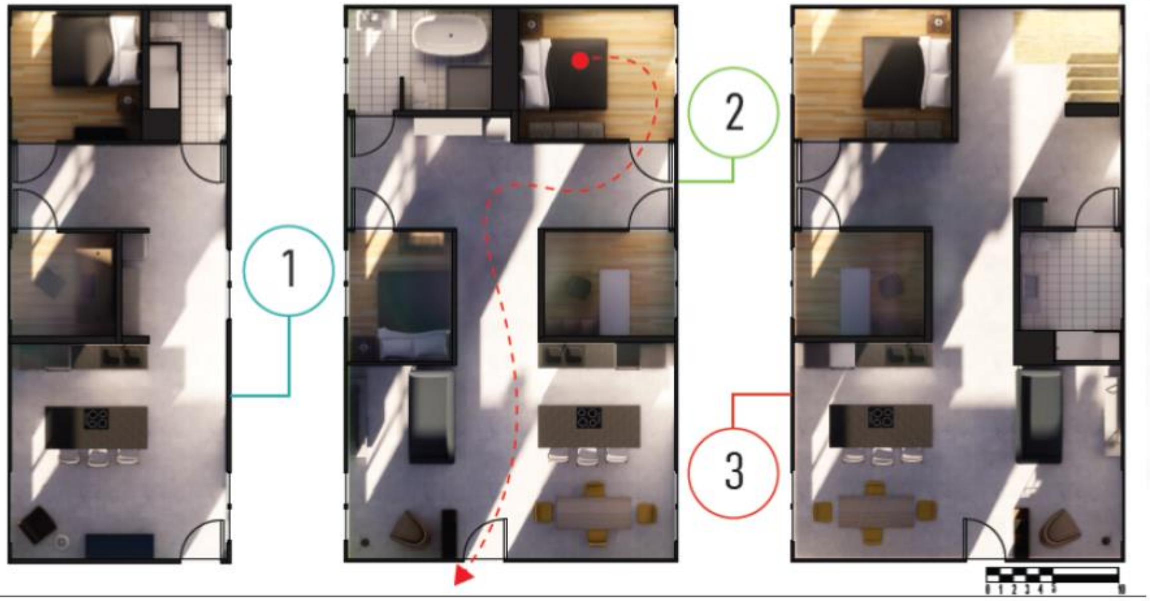

SITE DESIGN CONCEPTS


FINAL SITE DESIGN
A
B
C
D
E
F
G
H


VOLUME FOR AGGREGATION

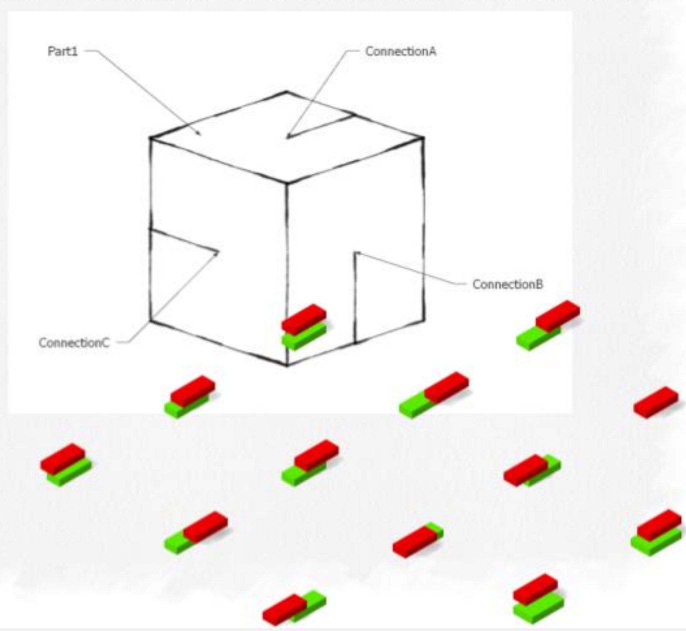

UNITS

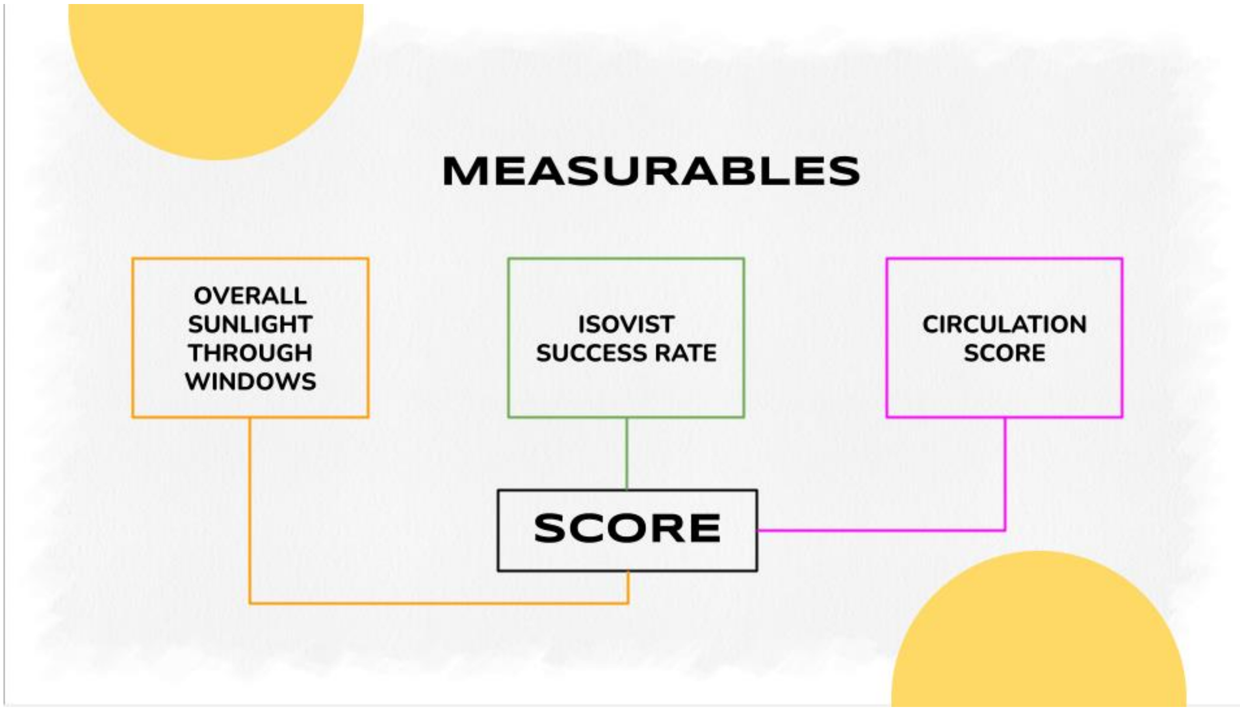
UNIT INTERIORS

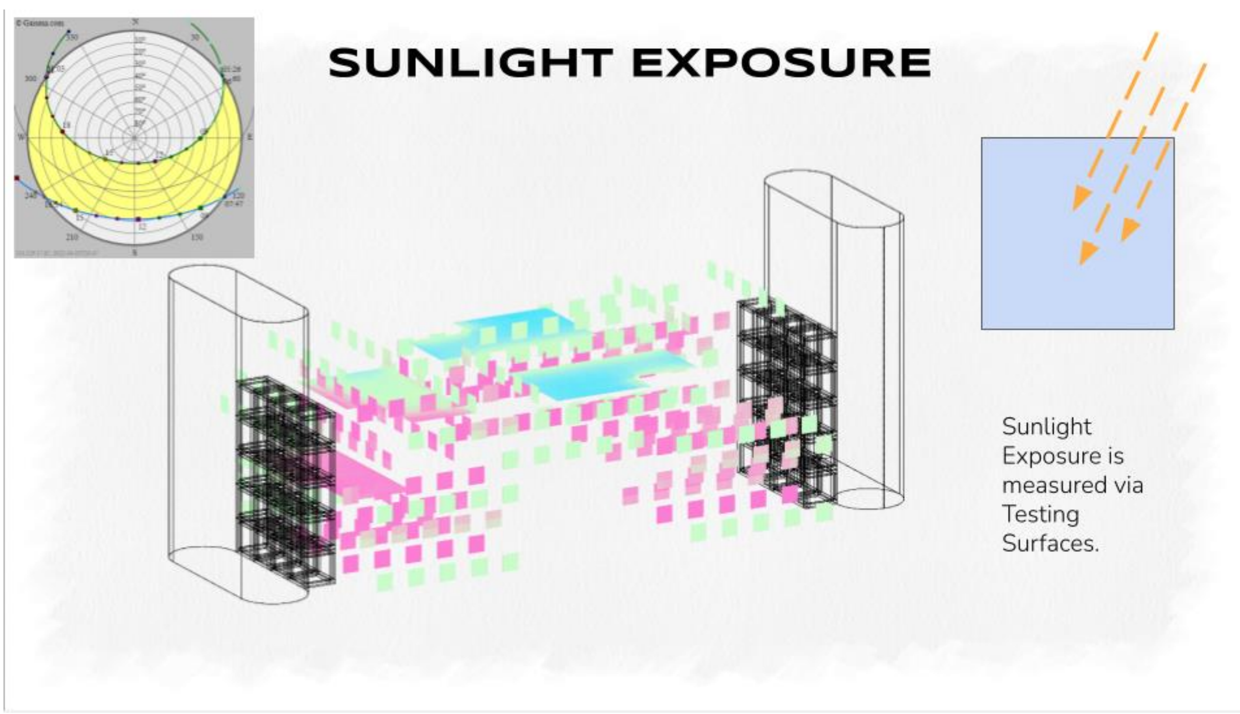


RULES. RULES. RULES. AND MORE RULES.

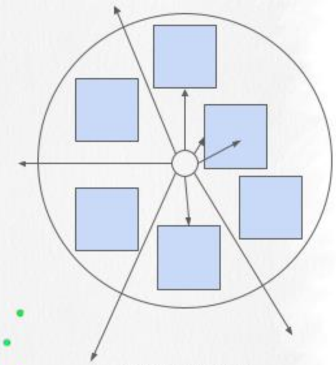
	A
988	WaspRule [UNITSPACER|3_UNIT42.5X25.5|26]
989	WaspRule [UNITSPACER|3_UNIT42.5X25.5WALKONROOF|3]
990	WaspRule [UNITSPACER|3_UNIT42.5X25.5WALKONROOF|5]
991	WaspRule [UNITSPACER|3_UNIT42.5X25.5WALKONROOF|7]
992	WaspRule [UNITSPACER|3_UNIT42.5X25.5WALKONROOF|9]
993	WaspRule [UNITSPACER|3_UNIT42.5X25.5WALKONROOF|11]
994	WaspRule [UNITSPACER|0_UNITSPACER|5]
995	WaspRule [UNITSPACER|1_UNIT42.5X17|22]
996	WaspRule [UNITSPACER|1_UNIT42.5X17|23]
997	WaspRule [UNITSPACER|4_UNIT42.5X17|10]
998	WaspRule [UNITSPACER|4_UNIT42.5X17|11]
999	WaspRule [UNITSPACER|1_UNIT42.5X25.5|28]
1000	WaspRule [UNITSPACER|1_UNIT42.5X25.5|29]
1001	WaspRule [UNITSPACER|1_UNIT42.5X25.5|30]
1002	WaspRule [UNITSPACER|4_UNIT42.5X25.5|15]
1003	WaspRule [UNITSPACER|4_UNIT42.5X25.5|16]
1004	WaspRule [UNITSPACER|4_UNIT42.5X25.5|17]
1005	WaspRule [UNITSPACER|1_UNIT42.5X25.5WALKONROOF|13]
1006	WaspRule [UNITSPACER|1_UNIT42.5X25.5WALKONROOF|14]
1007	WaspRule [UNITSPACER|1_UNIT42.5X25.5WALKONROOF|15]
1008	WaspRule [UNITSPACER|4_UNIT42.5X25.5WALKONROOF|0]
1009	WaspRule [UNITSPACER|4_UNIT42.5X25.5WALKONROOF|1]
1010	WaspRule [UNITSPACER|4_UNIT42.5X25.5WALKONROOF|2]




MEASURABLES
OVERALL SUNLIGHT THROUGH WINDOWS
ISOVIST SUCCESS RATE
CIRCULATION SCORE
SCORE


SUNLIGHT EXPOSURE
Sunlight Exposure is measured via Testing Surfaces.

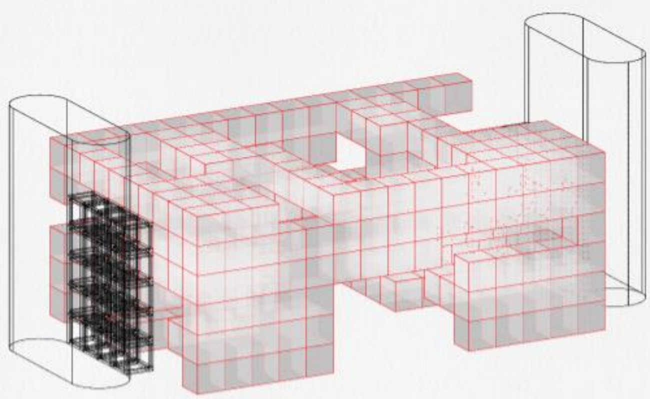
ISOVIST TESTING



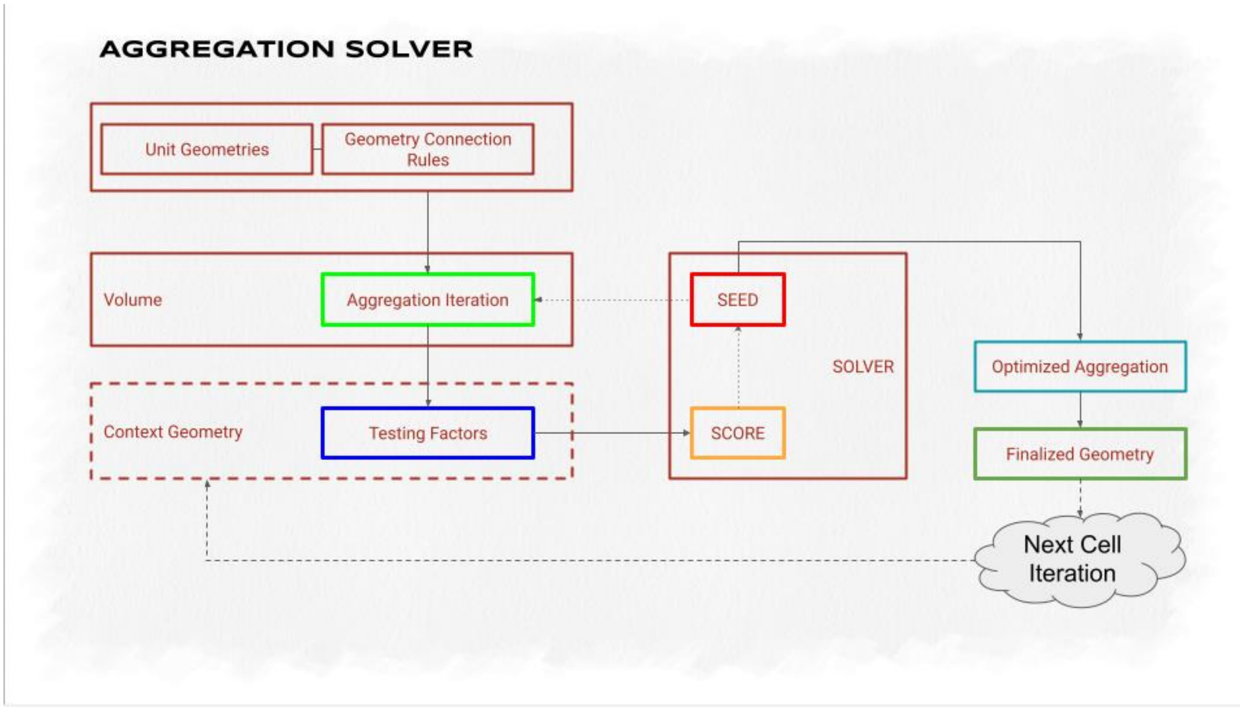
Isovist. How far a point in space can "see".

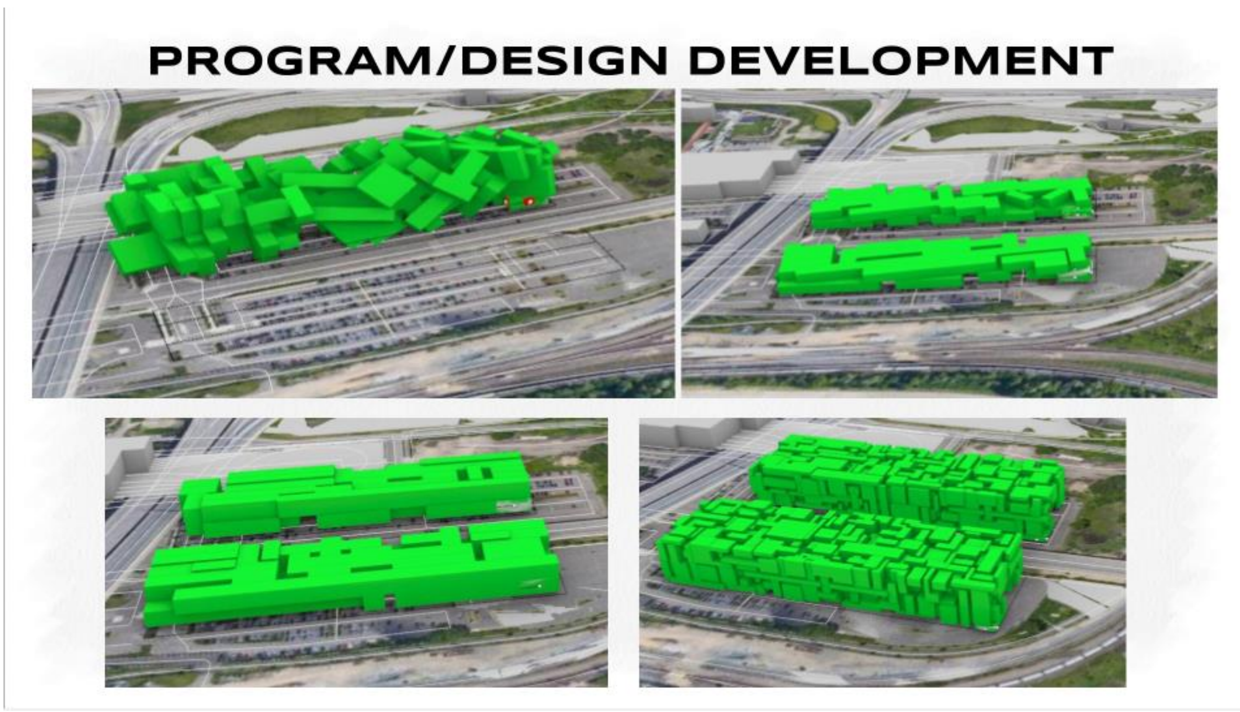
Can be used to Maximize Views.

CIRCULATION SCORE



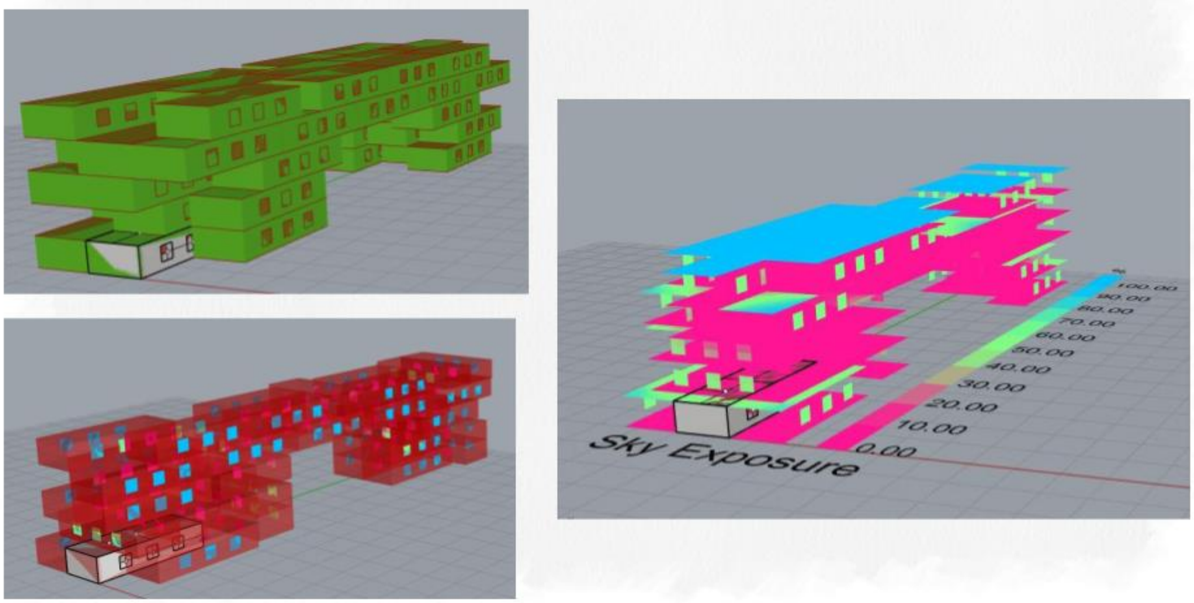
How "well" the Circulation Geometry Propagates throughout the Volume.


AGGREGATION SOLVER
Unit Geometries
Geometry Connection Rules
Volume
Aggregation Iteration
Context Geometry
Testing Factors
SEED
SOLVER
SCORE
Optimized Aggregation
Finalized Geometry
Next Cell Iteration

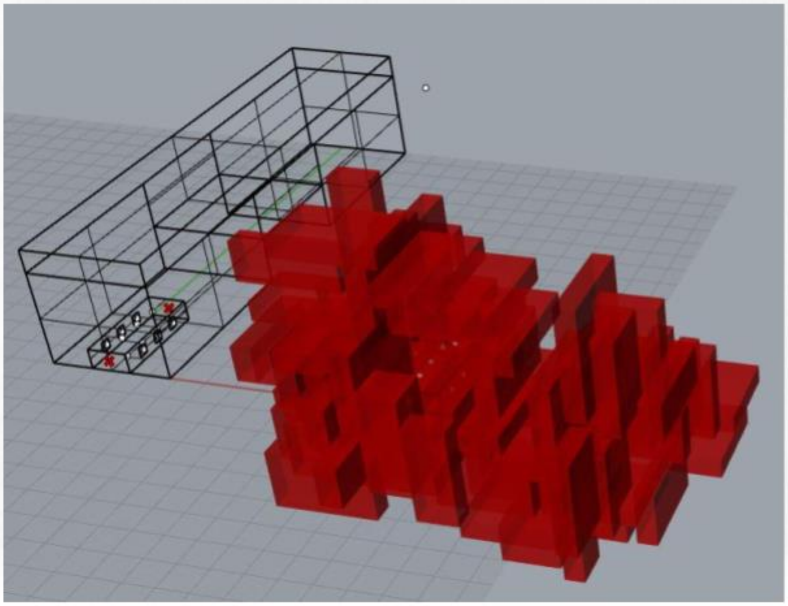

PROGRAM/DESIGN DEVELOPMENT

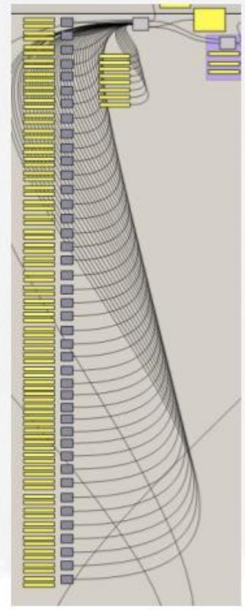
PROGRAM/DESIGN DEVELOPMENT

PROGRAM/DESIGN DEVELOPMENT

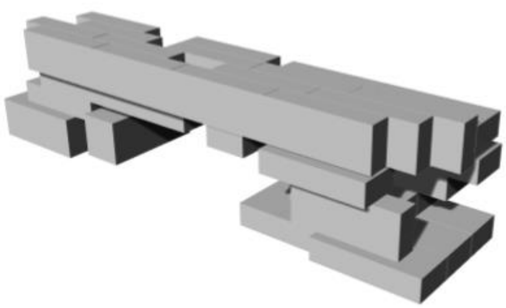


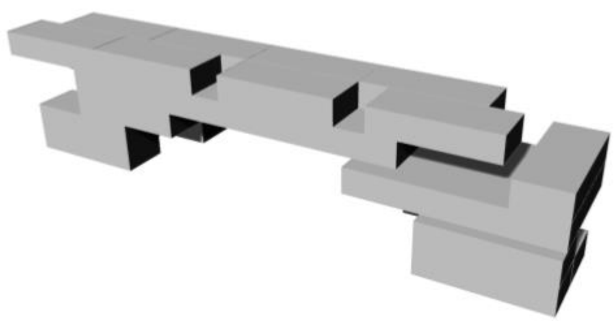
PROGRAM/DESIGN DEVELOPMENT

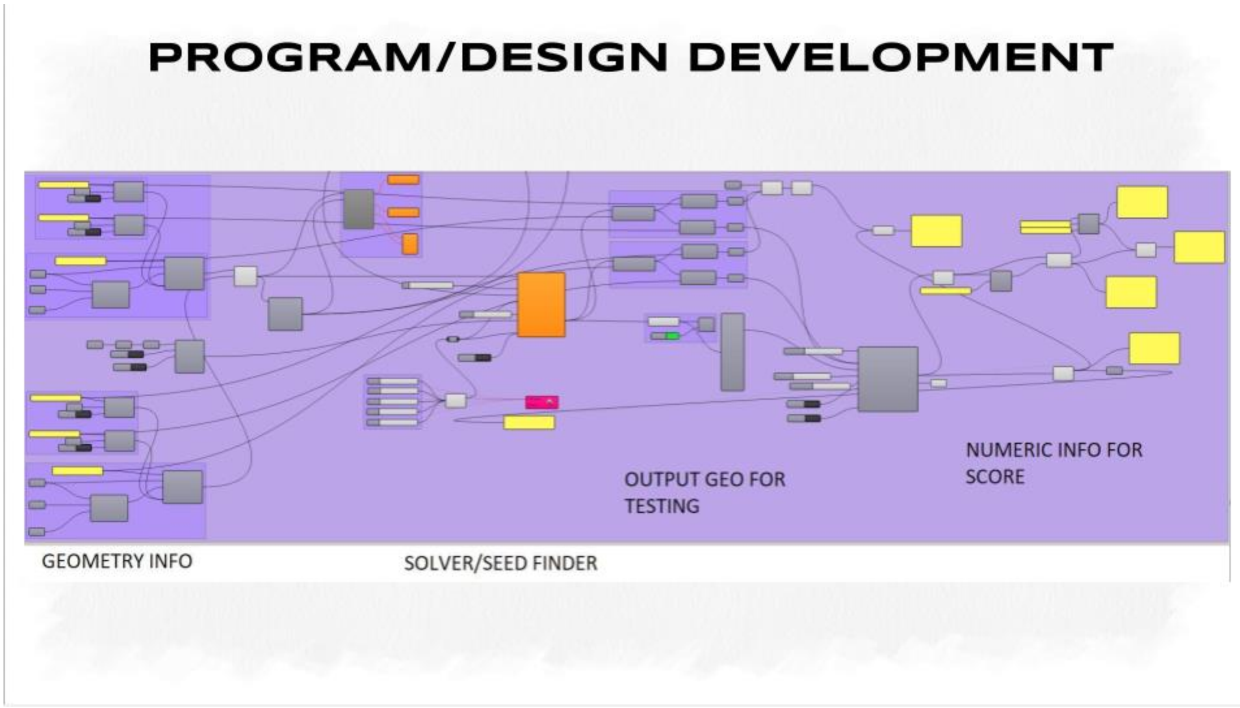


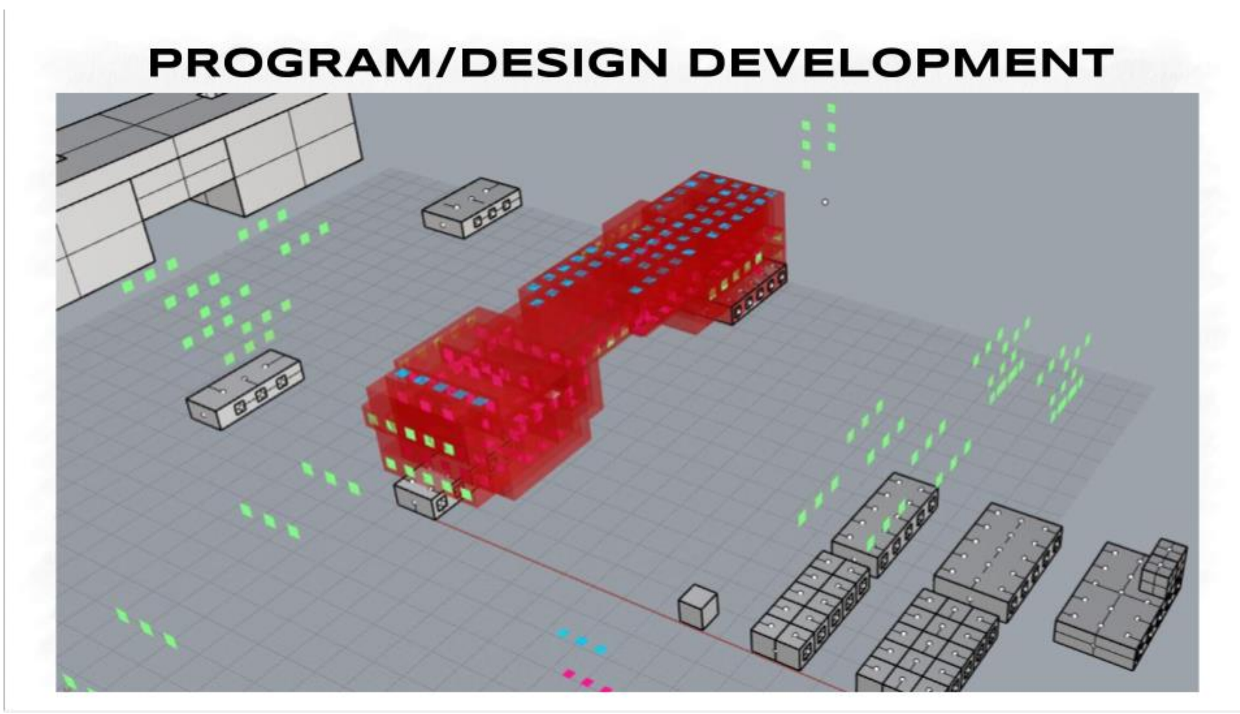


PROGRAM/DESIGN DEVELOPMENT

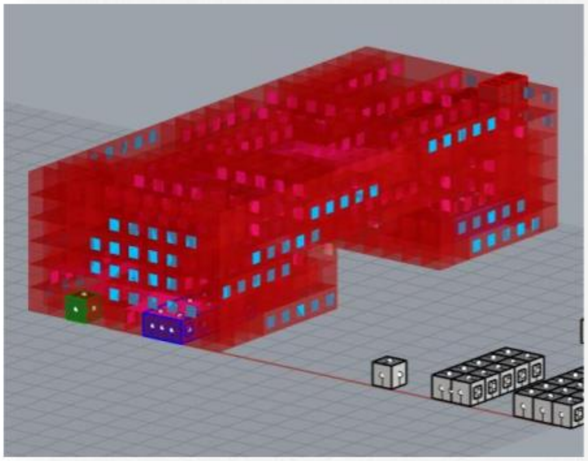


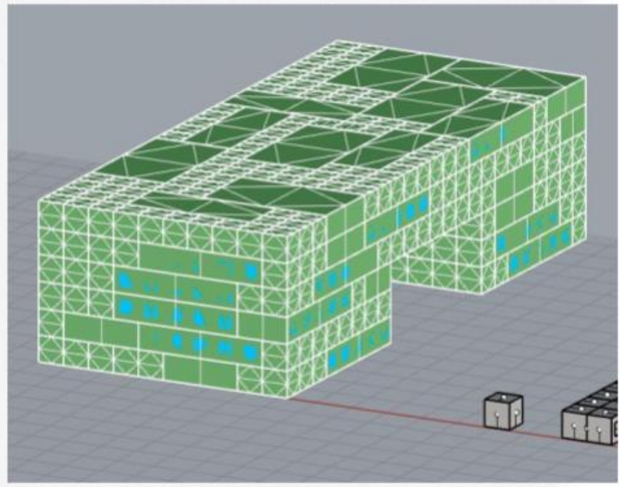



PROGRAM/DESIGN DEVELOPMENT
NUMERIC INFO FOR
SCORE
OUTPUT GEO FOR
TESTING
GEOMETRY INFO
SOLVER/SEED FINDER

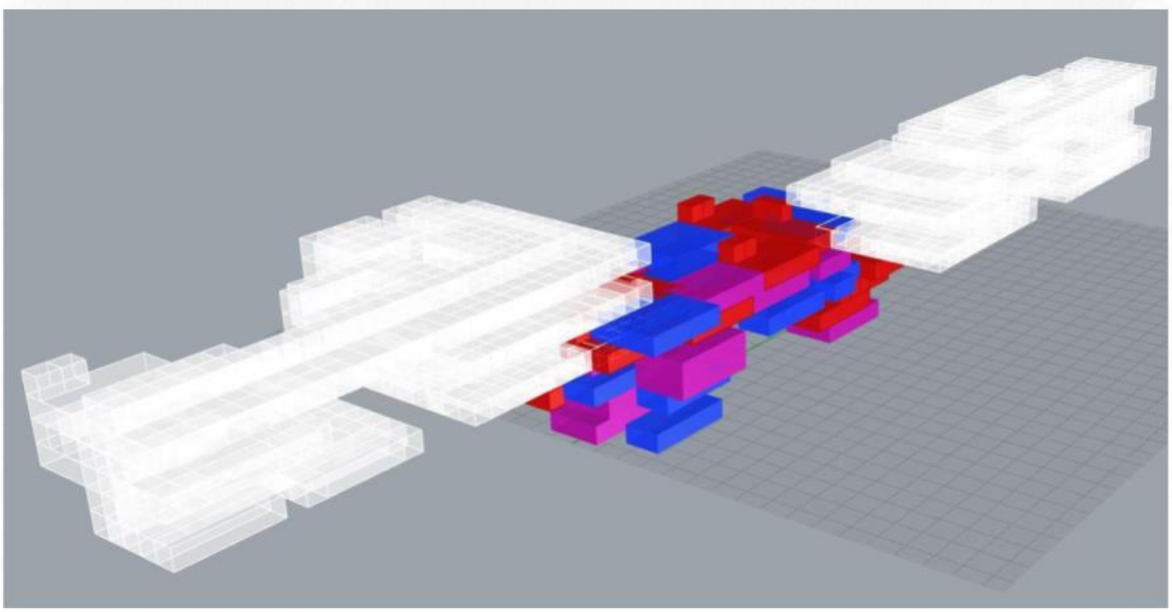

PROGRAM/DESIGN DEVELOPMENT

PROGRAM/DESIGN DEVELOPMENT

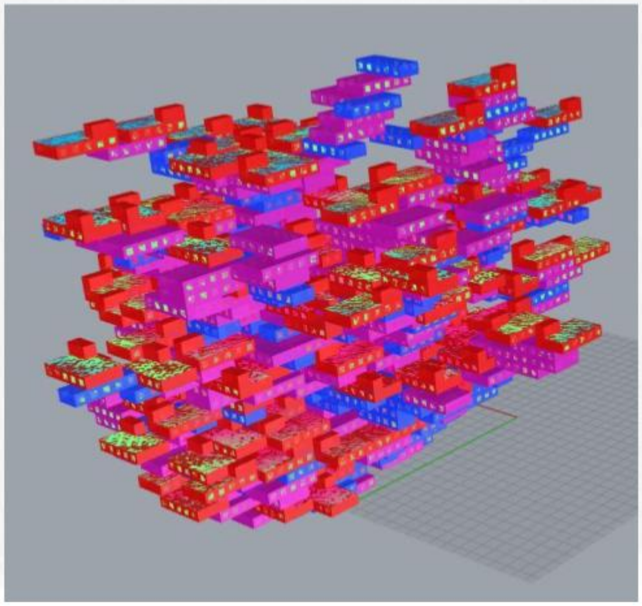




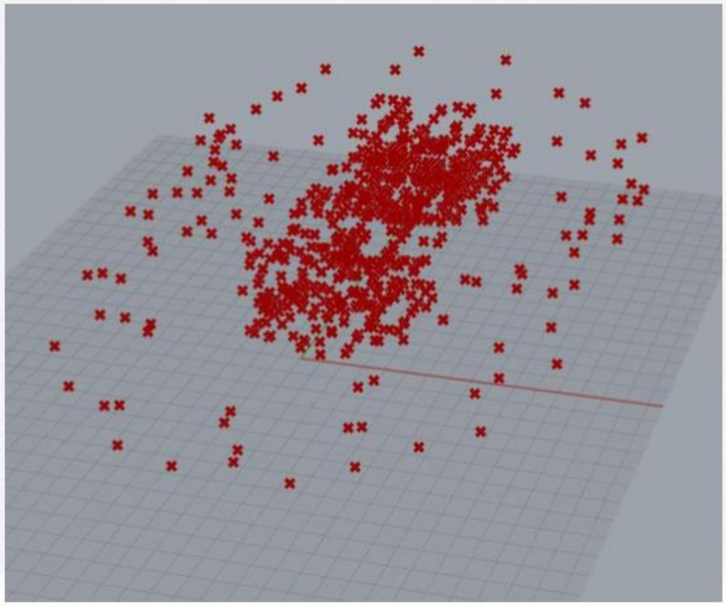
PROGRAM/DESIGN DEVELOPMENT



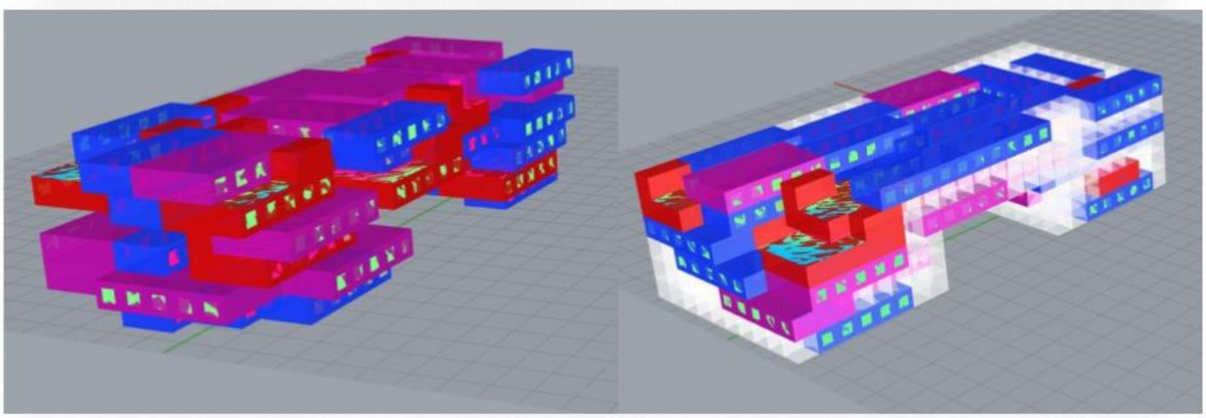
PROGRAM/DESIGN DEVELOPMENT



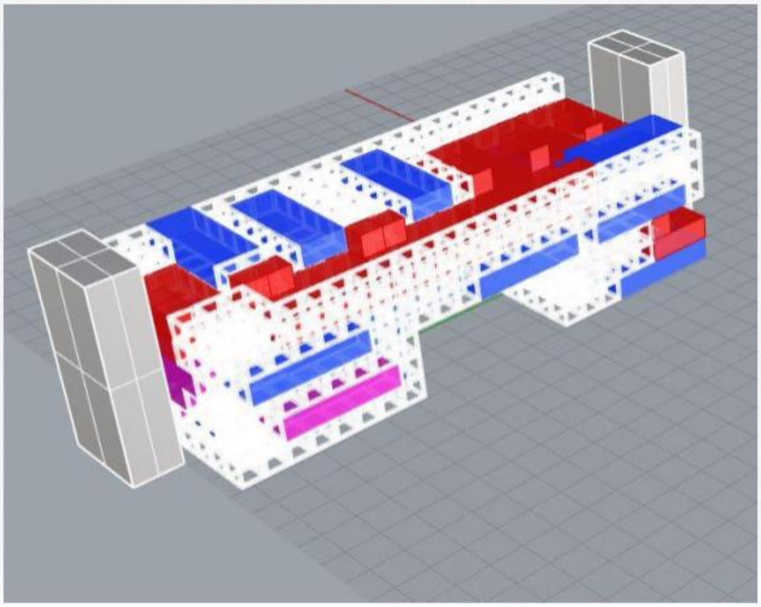
PROGRAM/DESIGN DEVELOPMENT



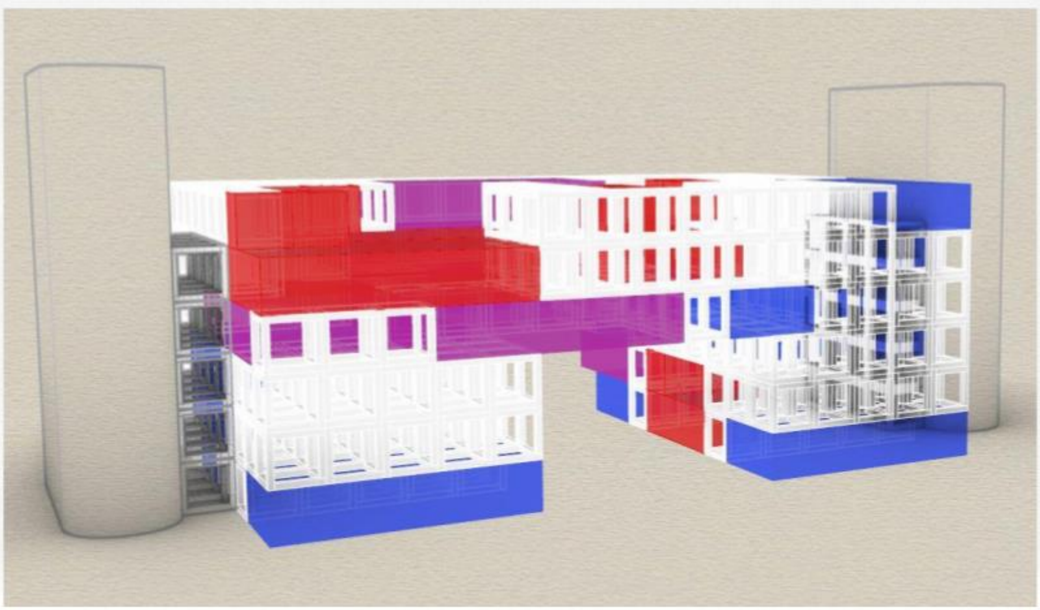
PROGRAM/DESIGN DEVELOPMENT



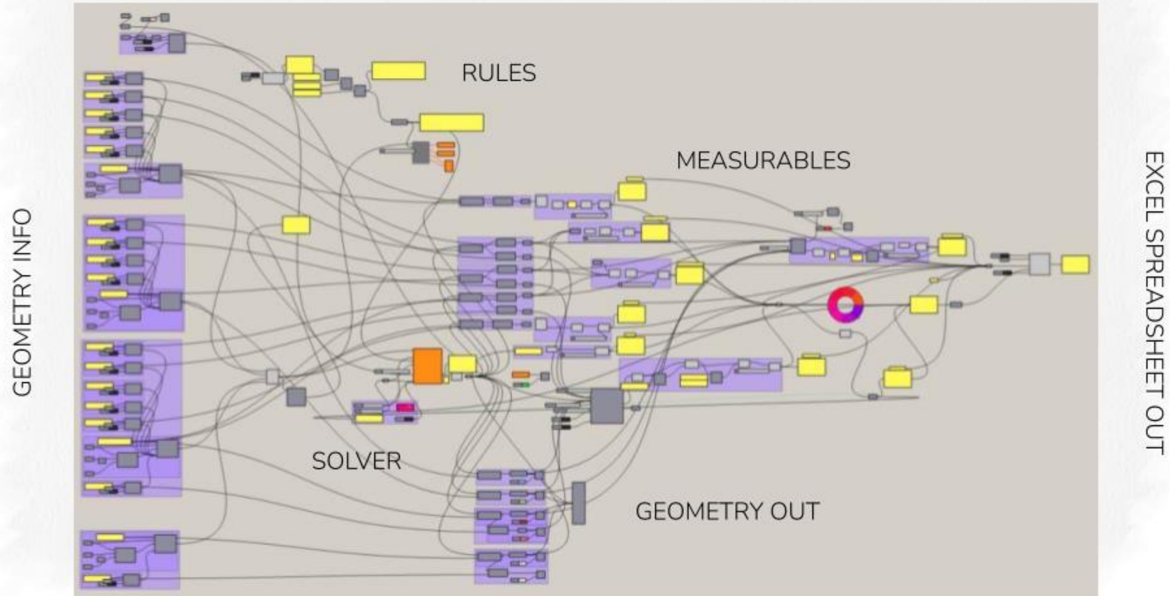
PROGRAM/DESIGN DEVELOPMENT



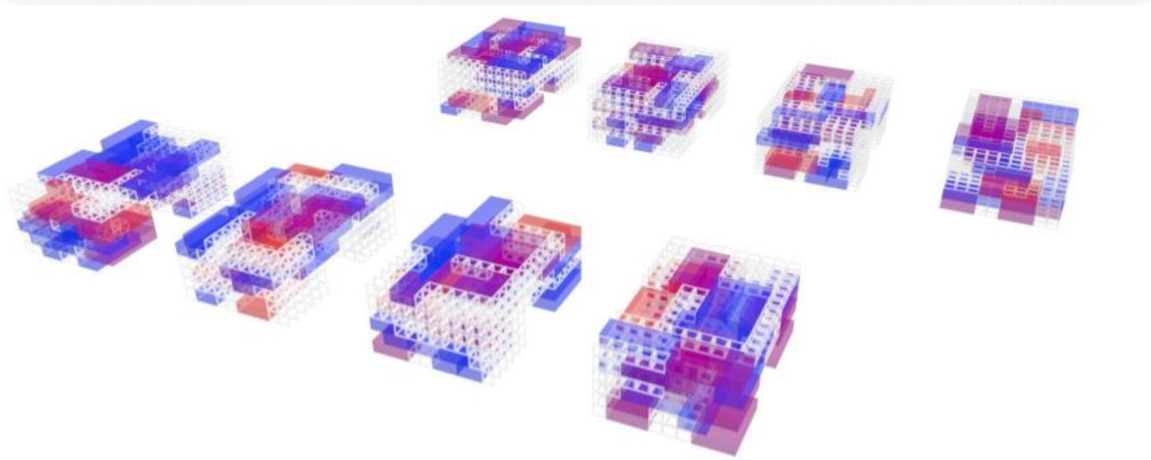
PROGRAM/DESIGN DEVELOPMENT



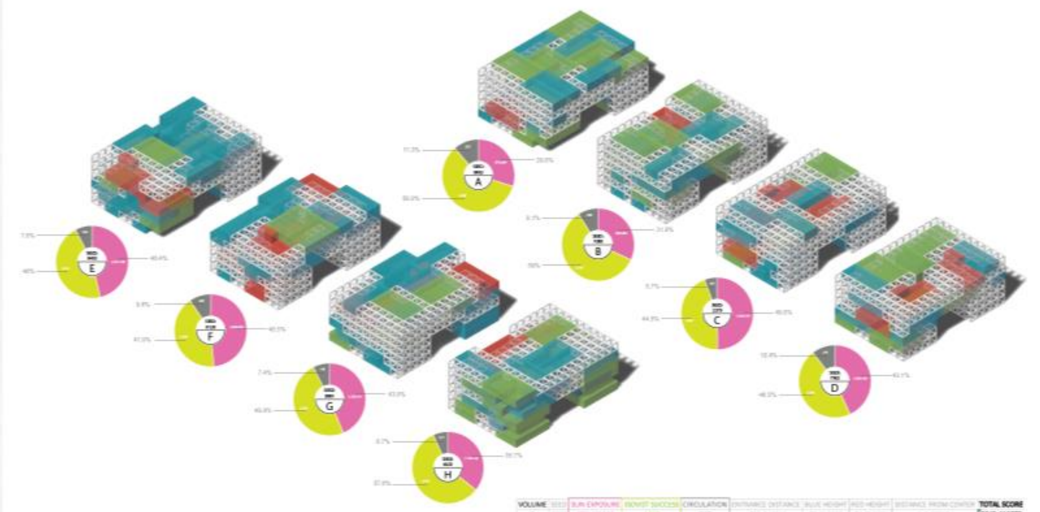
FINAL PROGRAM



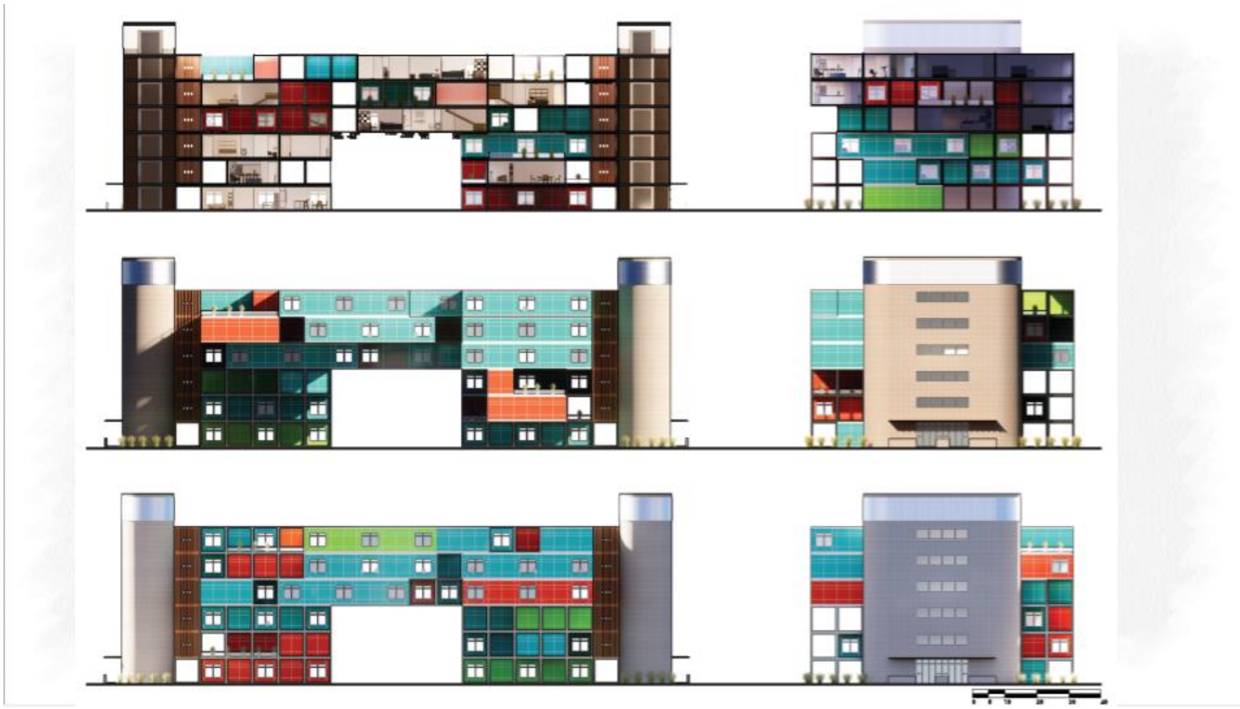
VOLUME GENERATIONS A-H

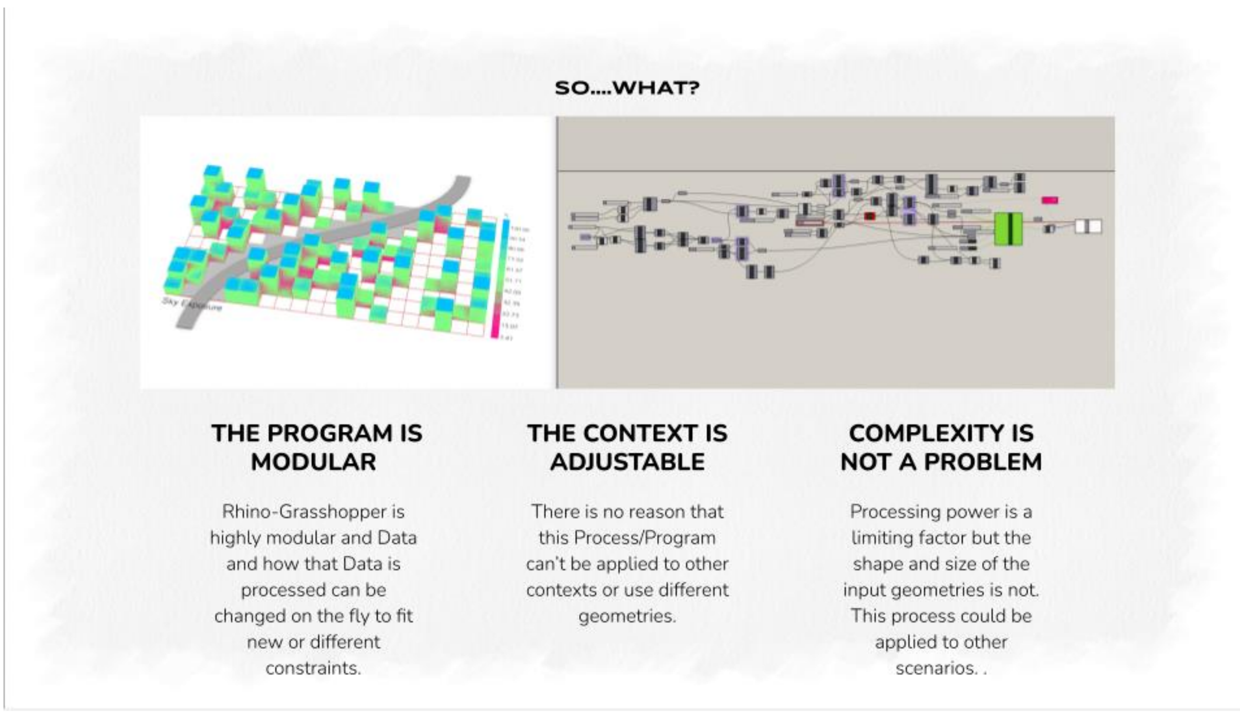


OUTPUT/PERFORMANCE ANALYSIS



VOLUME	[illegible]	SUN EXPOSURE	[illegible] SUCCESS	CIRCULATION	[illegible] DISTANCE	BLUE HEIGHT	RED HEIGHT	DISTANCE FROM CENTER	TOTAL SCORE
A	[illegible]	676.405888	1340	257	[illegible]	[illegible]	[illegible]	[illegible]	3842.410079
B	[illegible]	938.999494	1740	268	[illegible]	[illegible]	[illegible]	[illegible]	5094.13076
C	[illegible]	1444.704705	1290	165	[illegible]	[illegible]	[illegible]	[illegible]	4642.322788
D	[illegible]	1280.302152	1380	358	[illegible]	[illegible]	[illegible]	[illegible]	4760.341667
E	[illegible]	1221.16859	1210	200	[illegible]	[illegible]	[illegible]	[illegible]	5386.823657
F	[illegible]	[illegible]	1260	288	[illegible]	[illegible]	[illegible]	[illegible]	4848.386919
G	[illegible]	[illegible]	1290	195	[illegible]	[illegible]	[illegible]	[illegible]	4728.29864
H	[illegible]	[illegible]	[illegible]	217	[illegible]	[illegible]	[illegible]	[illegible]	5687.279233




SO....WHAT?
THE PROGRAM IS MODULAR
Rhino-Grasshopper is highly modular and Data and how that Data is processed can be changed on the fly to fit new or different constraints.
THE CONTEXT IS ADJUSTABLE
There is no reason that this Process/Program can't be applied to other contexts or use different geometries.
COMPLEXITY IS NOT A PROBLEM
Processing power is a limiting factor but the shape and size of the input geometries is not. This process could be applied to other scenarios. .

"There is no real ending. It's just the place where you stop the story."

— Frank Herbert

REFERENCES

Anna Fogel. (n.d.). *Hab67-section.jpeg*. The Polis Blog. photograph, Montreal, Quebec, Canada. Retrieved May 11, 2022, from https://www.thepolisblog.org/2012/03/habitat-67-cube-housing-in-montreal.html.

Barnaby, J. (2020, December 31). *Alexandra Road Estate - London's Architectural Gem*. London x London. Retrieved May 11, 2022, from https://www.londonxlondon.com/alexandra-road-estate/

Brown, N. (1978). *10257505/1312167_Orig.jpg*. (J. Abbott, Ed.)Modernism In Metro Land. photograph, London Borough of Camden. Retrieved May 11, 2022, from https://www.modernism-in-metroland.co.uk/blog/in-house-part-1-camden.

Davidson, S. (n.d.). *Grasshopper*. Grasshopper - ALGORITHMIC MODELING FOR RHINO. Retrieved May 11, 2022, from https://www.grasshopper3d.com/

Gourde, P. B. (Ed.). (2022, February 23). *About Habitat 67*. Habitat 67. Retrieved May 11, 2022, from https://www.habitat67.com/en/about/

Keith, L. (2016). *Alexandra-road-estate.jpg*. No Place Like It. photograph, London Borough of Camden; Lauren Keith 2022. Retrieved May 11, 2022, from https://www.noplacelike.it/highlights-open-house-london/alexandra-road-estate/.

Kisho Kurokawa Architect & Associates. (n.d.). *Nakagin Capsule Tower*. Kisho Kurokawa Architect & Associates. Retrieved May 11, 2022, from https://www.kisho.co.jp/page/209.html

Kurokawa, K. (2021). *Nakagin-capsule-tower-tokyo-kisho-kurokawa-archeyes-18.jpg*. Arch Eyes. photograph, Shimbashi, Tokyo, Japan; Arch Eyes. Retrieved May 11, 2022, from https://archeyes.com/nakagin-capsule-tower-kisho-kurokawa/.

Llach, D. C. (2021). Between form and information: Early philosophies of Computer-Aided Design. *Nexus Network Journal*, *23*(4), 933–943. https://doi.org/10.1007/s00004-021-00581-w

Marc Cramer + Thomas-Miau. (n.d.). The Spaces. photograph, Montreal; Jamie Robinson. Retrieved May 11, 2022, from https://thespaces.com/moshe-sadfie-habitat-67-montreal-open-to-public/.

Nakagin.jpg. (n.d.). Voice of East Anglia. photograph, Shimbashi, Tokyo, Japan. Retrieved May 11, 2022, from http://www.voicesofeastanglia.com/2013/07/nagakin-capsule-tower.html.

Pacheco, A. (2019, November 18). *L.A. architects are embracing modular multi-family housing*. Archinect. Retrieved May 11, 2022, from https://archinect.com/news/article/150170940/l-a-architects-are-embracing-modular-multi-family-housing

Robert McNeel & Associates. (n.d.). *Rhinoceros 3D*. www.rhino3d.com. Retrieved May 11, 2022, from https://www.rhino3d.com/

Rossi, A. (n.d.). Wasp. Retrieved May 11, 2022, from https://www.grasshopper3d.com/group/wasp

Rutten, D. (n.d.). *Galapagos*. Galapagos - Evolutionary Solver. Retrieved May 11, 2022, from https://www.grasshopper3d.com/group/galapagos

User: Concierge.2C. (2011). *File:Montréal - Habitat 67 (04).jpg*. photograph, Montreal.

User: 準建築人手札網站 Forgemind ArchiMedia. (2009). *Kisho Kurokawa 黑川紀章 - 中銀膠囊大樓 Nakagin Capsule Tower - Photo 07*. Flickr. photograph, Shimbashi, Tokyo, Japan; Flickr. Retrieved May 11, 2022, from https://www.flickr.com/photos/eager/19051920494/in/photolist-vYY6rH-vYyYkD-v2y5cQ-vWfMnA-vXWFtS-vFXeJW-vXWH3o-vYyY3V-v2GuaT-vYyXMe-vFXt1N-vXWxNw-vYXWzp-v2y44N-vXWwCL-v2GtPH-vYyXRc-vG5msx.

Wyatt, D. F., Wynn, D. C., Jarrett, J. P., & Clarkson, P. J. (2011). Supporting product architecture design using computational design synthesis with network structure constraints. *Research in Engineering Design*, *23*(1), 17–52. https://doi.org/10.1007/s00163-011-0112-y