BITS & PIECES : INVESTIGATING THE PROCEDURAL IDEATION OF ARCHITECTURAL MASSING FOR EARLY STAGE DESIGN

BY

JACOB RHODES-ROBINSON

A 120-point thesis
submitted to the Victoria University of Wellington
in partial fulfilment of the requirements for the
degree of Master of Architecture (Professional)

Victoria University of Wellington
School of Architecture

2018

Fig x.01.   Image of Concept Design - Main grid of iterations*

*All figures marked with a * were graded in the 75% design examination in November 2017.*

BITS + PIECES

INVESTIGATING THE PROCEDURAL IDEATION OF ARCHITECTURAL MASSING FOR EARLY STAGE DESIGN

Jacob Rhodes-Robinson

BITS + PIECES

ABSTRACT

The architectural discipline is constantly experiencing change to the way in which its practitioners operate. The continual evolution of computing hardware and the substantial development of Computer Aided Architectural Design (CAAD) has seen Architecture shift from a discipline of predominantly analogue techniques to one that relies almost entirely on the digital medium. As a result, the role of the practicing architect has seen considerable change. Architecture, once a discipline of pencil and paper, now shares creative techniques and tools with Computer Science, Film, Visual Effects, Interactive Media, Robotics, and Computer programming. Such new partners are providing alternative views of what it is to be a creative practitioner, challenging the discipline of architecture to step beyond the preconceived boundaries and means of operating embodied within conventional practice. Architects now have the opportunity to adopt new methods for the production of the built environment.

This research engages with developing computational techniques designed for film and interactive media and explores how they can be utilised to augment the way in which architecture may be produced. This body of researches adopts the technique procedural generation as a vehicle for this investigation; a technique used for content creation in interactive media and game design. This research also adopts the use of a computational design software called Houdini - an industry standard procedural software used widely within film and game. Through an architectural lens, it explores the re-purposing of this software and procedural design, developing an understanding for how they can both aid in the ideation of built form during the infancy of the design process.

This research initially addressed the question: 'how can conventional architectural practices be augmented by procedural computational design techniques, to further explore the impacts of opportunity and ideation on architectural design?' As a result of refinement, it came around to focus on asking 'how can the application of procedural generation design techniques augment the ideation of architectural massing for early stage design?' It identifies how procedural techniques can be used in the process of ideating architecture and aims to investigate how procedural generation offers an alternative methodology to the production of architecture in early design stages. It explores, through computational design, the limitations and constraints that occur in the process of mastering design orientated procedural techniques. It subsequently develops, through computational design, an understanding of how procedural techniques can be applied to the early stage design of architecture. Finally, through architectural design, it examines how procedural design techniques can be partnered with specific architectural conditions such as site, function, and form, in order to augment the architectural ideation process.

BITS + PIECES

PREFACE

A significant portion of my life has been spent playing computer and console games thanks to those I grew up with; my Father, my friends, and my colleagues. As both a gamer and creative designer, I became fascinated with the possible crossover between the creation of Architecture and content creation in game design. The following thesis is an investigation into that fascination.

BITS + PIECES

DEDICATION

Dad, you are the body and soul of this research thesis. Your inspiration, enthusiasm, and passion for what you do in life has formed the foundations for who I am today.

I know we may not spend much time together these days, but gaming has always been a platform in which we have found common ground. Let this thesis between architecture and gaming become another common ground between us.

Thank you for always being there for me.

I dedicate this to you.

BITS + PIECES

RECOGNITION

I must first acknowledge the immense support and dedication of my thesis supervisor Tane Moleta. Not only is he my tutor and mentor, but he is my friend. The past nine months have been tough, but with his guidance and patience, we have reached the finish line together. He has been my role model throughout this entire body of research, pushing me to trust myself and putting his faith in my ability to produce the best possible result. With all the banter, laughs and grumbles, this would not have been possible without him. Thank you Tane.

To my friends and colleagues, inside the School of Architecture and out, thank you for all of the support you have given me - not only for this last twelve months but for the last five years of my education here at Victoria University. I would not be where I am today if it weren't for the support and love of those I surround myself with. Here's to a few beers after this is all done and dusted!

To Mum and Dad, thank you for putting up with my architectural nonsense over the past five years. To the Rhodes', the Homans', Ogilvies' and the Sbromas', thank you all for the love, support and encouragement you have always given me throughout my studies.

To my partner, Hannah, you are my rock and I am grateful for everything you have done for me over the last six years. Your love and enthusiasm for what I do has pushed me to become who I am today. I don't know where I would be without you. For this, I thank you.

Thank you so much.

Fig x.02.   Image of final review


EVAL-
UATE
BITS +
PIECES
Investigating the Procedural Ideation of Architectural Massing for Early Stage Design

BITS + PIECES

CONTENTS

BITS + PIECES

CONTENTS

BITS + PIECES

INTRODUCTION

BITS + PIECES

INTRODUCTION

Introduction

The architectural discipline is changing. Whether for better or for worse, the continual development of computer hardware and computational software has seen the discipline evolve over past decades. The role of the 'Architect' no longer lies restricted to the boundaries of pencil and paper. Instead, architects and designers now have an ever-growing set of physical and digital tools to turn to; computational processes, three-dimensional visualisation, virtual reality, and robotics to name a few. These new partners are providing alternative views of what it is to be a creative designer. With this, both students and masters of the discipline are learning in tandem from the opportunities arising with the introduction of new methods, developing and refining the way in which we as a profession come to produce the built environment.

With this evolution, it is essential to continually discover and investigate alternative design-ideating methods. This thesis, heavily influenced by game design, delves into the world of content creation for interactive media and investigates what the tool of 'procedural generation' has to offer for the discipline of architecture and the ideation of built form. It is an investigation of trial and error - a learning process within a learning process - aiming to bring to light how procedural design may crossover to the architectural field. It explores the application of procedural generation within the early stage design stage of architectural practice. It is a visual portfolio and a document of my own learning of procedural techniques, explored within the vehicle of Houdini - a procedural software used for visual effects in the film and game industries.

In order to investigate this avenue of research, this thesis asks: 'how can the application of procedural generation design techniques augment the ideation of architectural massing for early stage design?' It aims to identify how procedural techniques can be used in the process of ideating architecture and identify how procedural generation may offer an alternative methodology to the production of architecture in early design stages. It does this by first exploring, through computational design, the limitations and constraints that occur in the process of mastering design orientated procedural techniques. It then develops, through computational design, an understanding of how procedural techniques can be applied to the early stage design of architecture. Finally, through architectural design, it examines how procedural design techniques can be partnered with specific architectural conditions such as site, function, and form, in order to augment the architectural ideation process.

The research question, aims and objectives above form the foundation for the procedural design decisions made throughout this thesis. They guide the direction for this body of research and provide a reflective criteria for myself as a Master in this field. This research is also an extension of research undertaken by Welch & Moleta (2012), Balzalo & Moleta (2015) and Richards & Moleta (2016). It exists as a fragment of a wider body of thought that explores and critically reflects upon the use of computational tools within the Architectural discipline.

BITS + PIECES

INTRODUCTION

Research Scope

With the addition of new technology and design tools in the Architectural discipline, traditional techniques are often challenged and in many ways, such as with the introduction of computational processes, sought after to be 'surpassed' in some way.

However, this research thesis does not attempt to replace an existing architectural process nor does it aim to produce a resolved architectural design. Instead, by testing the capabilities of procedural generation through an architectural lens, this thesis investigates the argument that procedural generation may offer an alternative means to the ideation of space during early stage design. I have aspired to understand, through an amalgamation of design-led research and the adoption of a new software, how to best investigate procedural design and develop a knowledge for how it can be applied to conventional architectural practice. The result is a thesis that is driven heavily by process, rather than a focus on the final built outcome. The final design phase of this thesis is thus not a completely resolved building, but rather an experiment that seeks to evaluate the transition between procedural design methods and architectural practice. It encompasses a focus towards computational design processes and the opportunities that arise with them and thus will not conclude with a final building design.

THIS THESIS IS:

- An exploration into the IDEATION of architectural massing through the vehicle of design-orientated procedural techniques.

- A MASTERY of design-orientated procedural techniques applied to the early stage design of architectural form and function.

- A mastery of HOUDINI, adopting technical skills relevant to procedural design processes.

- An exploration into the OPPORTUNITY that occurs within a pseudo-random, automated design process.

Fig 1.01. What this thesis aims to be.*

THIS THESIS IS NOT:

- An attempt to REPLACE conventional methods used in the early stage design of architectural form and function.

- A system or tool that SIMULATES the effect of external extremities on the production architectural form and function.

- A tool for finding an OPTIMUM result.

Fig 1.02. What this thesis does not aim to be.*

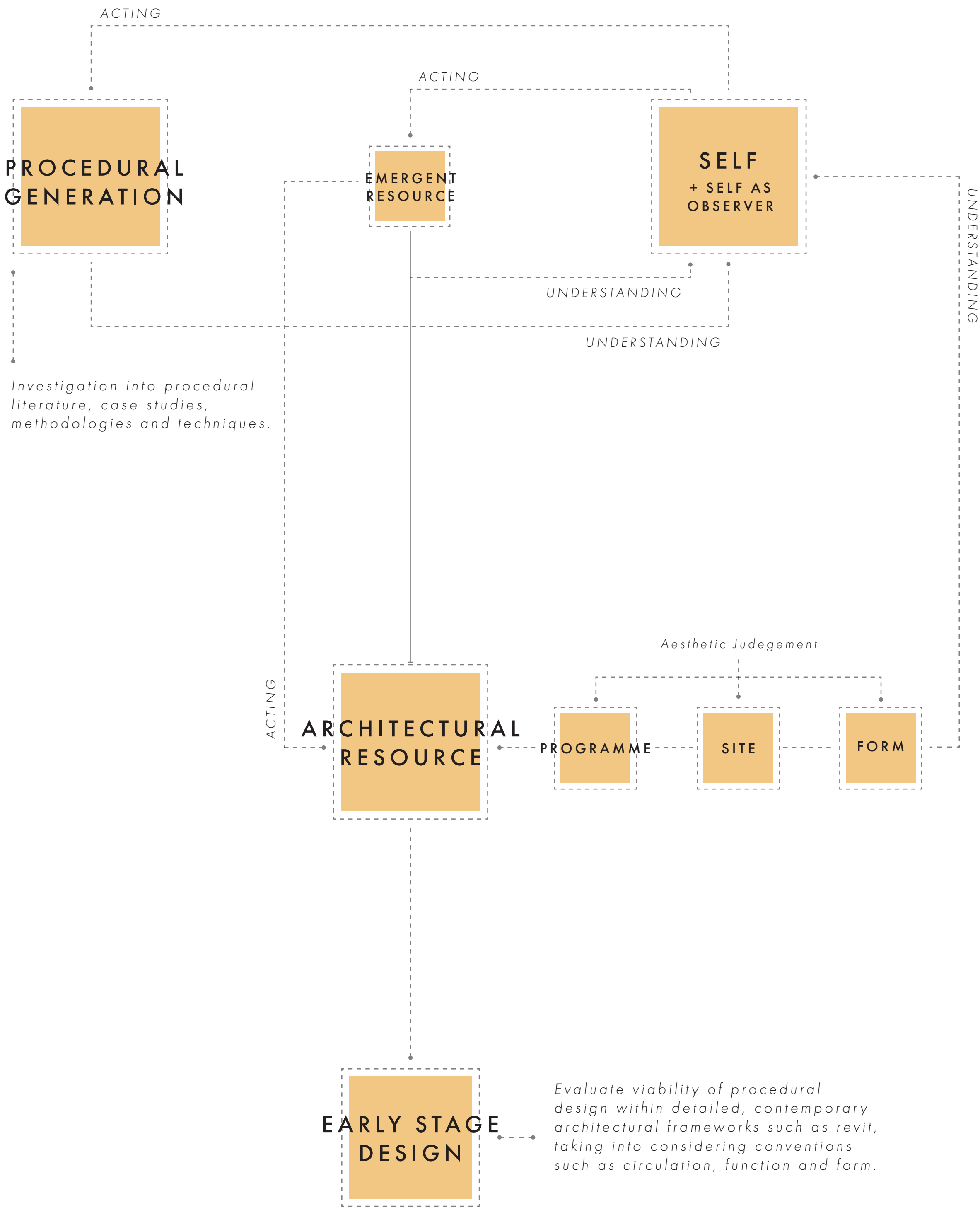


Fig 1.03. Diagram illustrating research methodology adopted from Thomas Fischer's keynote presentation.*

BITS + PIECES

INTRODUCTION

Method and Process

This thesis employed a two stage research methodology in order to investigate the overall research question. The first part, research for design, investigates current theoretical positions on procedural design and case studies involved in the relationship between architecture and procedural generation. The second part, research led design, employs an iterative design process specific to computational design and unique to this thesis alone, treating the tool of procedural generation as an actor (the observed) and myself, the designer, as the director (the observer). This methodology was adopted and adapted from a keynote presentation presented by Thomas Fischer, an Associate Professor at the Xi'an Jiaotong-Liverpool University, where he spoke about the importance of "observational research when concerned with artificial intelligence and robotics" (Fischer, 2017). This method of research provides the designer with the ability to take a back-seat to the possibilities of a computational process, allowing for the observation of its outcomes and a reflection of how said outcomes may be translated into an application within their respective field. Fischer referred to these findings as the 'emergent resource'.

A significant portion of the findings in this document were discovered through a critical reflection of what was possible from the Houdini software in regards to its capabilities; outcomes referred to as the 'emergent resource' throughout this investigation. Going about the research methodology in this way allowed for me as the designer to reflect upon the emergent resources at the concept, preliminary and developed design stages and draw conclusions as to how each stage was meeting the aims and objectives of the research itself. This reflective process offered a unique approach to exploring the research question at each stage in the research. It identifies a possible solution at each stage in relation to the research goals and opens up a series of reflective questions and opportunities leading into each subsequent phase of the thesis.

BITS + PIECES

INTRODUCTION

Thesis Structure

01 | INTRODUCTION

This chapter highlights why this avenue of research is relevant to the architectural discipline through a problem statement and introduces the research question, aims and objectives. It establishes the scope of the research and outlines the research methodology and thesis structure.

02 | BACKGROUND

This chapter introduces the core concepts of procedural design techniques and discusses their use and importance within the context of interactive media and game design. It establishes an understanding of how Houdini acts a vehicle for testing and implementing procedural techniques and outlines how the Houdini interface and procedural systems are to be treated throughout the rest of the research.

PART ONE: RESEARCH FOR DESIGN

03 | THEORETICAL POSITIONING

This chapter outlines the key theoretical arguments surrounding procedural generation in interactive media and architectural design. The theoretical context of this research is built upon the arguments summarised in this chapter and the theoretical influences for procedural design decisions moving forward are established here.

04 | CASE STUDIES

This chapter identifies specific case studies in interactive media and film that have investigated the relationship between procedural generation and architecture in the past and at present. This chapter provides practical context to this research, establishing a foundation for procedural design experiments moving forward.

PART TWO: RESEARCH LED DESIGN

05 | DISCOVERY | CONCEPT DESIGN

This chapter introduces the very first procedural design component of this research. It investigates the capabilities of procedural generation through a very broad architectural lens, identifying what the tool itself is capable of producing. It consists of two initial design experiments, a reflection of the limitations and findings that arose in those experiments and a revised concept design, introducing the first relationship between the procedural and architectural fields.

06 | REFINEMENT | PRELIMINARY DESIGN

This chapter of the thesis first reflects upon the findings and critique of the concept design investigation. It builds upon the emergent resource from the concept phase and develops a refined procedural system, delving into the area of recursive design and preference to further strengthen the relationship between the procedural and architectural fields.

07 | APPLICATION | DEVELOPED DESIGN

This chapter builds upon the investigations of the previous two chapters and uses their emergent resources to explore how a procedural logic may be applied to the architectural discipline. It integrates the architectural conditions of site and program to evaluate whether procedural generation can be translated to a contemporary architectural context. Ultimately, this chapter ties the previous investigations of this thesis together and brings them in line with a conventional architectural workflow.

PART THREE: REFLECTION

08 | CONCLUSIONS + CRITICAL REFLECTION

This chapter provides a critical reflection on the research as a whole, discussing how well it addresses the research question. It highlights the key knowledge discovered throughout the process of undergoing this design-led research thesis and reflects upon the validity of the research in relation to the theoretical arguments discussed in the theoretical review.

09 | REFERENCES

10 | LIST OF FIGURES

11 | APPENDIX

BITS + PIECES

BACKGROUND

BITS + PIECES

BACKGROUND

Procedural Generation

> *"Procedural generation is a type of design methodology in which components or elements are handcrafted and each given unique properties. These components are then fed into a database which generates forms based on a series of rules and regulations put into place by the designers that dictate what components and elements are used, combined, and generated." – (Maguid, 2016)*

> *"Procedural modelling deals with (semi-)automatic content generation by means of a program or procedure. Among other advantages, its data compression and the potential to generate a large variety of detailed content with reduced human intervention, have made procedural modelling attractive for creating virtual environments increasingly used in movies, games and simulations." – (Smelik et al., 2014)*

Procedural modelling is most commonly used within the film and game industries, acting as an efficient method of content creation for both. As Maguid states, it is the process of developing a series of handcrafted components, with unique properties, that then interrelate to eachother through a set of procedural rules and regulations (Maguid, 2016). It provides designers with the ability to produce content at a large scale by focussing the majority of their development time in the creation of a procedural logic or system and assets. The output stage of procedural generation is often much shorter than the input, making it a viable method for the production of content where variety is needed en masse. To support this, within film and game, Mueller et al. wrote that "procedural modelling techniques allow for the detailed production of architectural models for game and film environments, at lower costs" (Muller et al., 2006). Once a procedural logic is set up, rules and regulations can be altered with new values, entirely changing the end output of the system. Smelik et al. noted that the potential variety that arises with procedural modelling is one of its many advantages, including its flexibility (Smelik et al., 2014). It is this notion of flexibility that, for me, built an interest around whether or not the method of generation modelling is one that can be applied to conventional architectural practice.

IMAGES REMOVED DUE TO COPYRIGHT

Fig 2.01. Compilation of images illustrating the vast quantity of unique planets available to players in No Man's Sky. Planets were generated using a procedural system designed by Hello Games.

BITS + PIECES

BACKGROUND

No Man's Sky

For the gaming and design industries, the initial announcement of Hello Games' 'No Man's Sky' on the 7th of December 2013 was one of excitement and enthrallment. Hello Games' promised that their new development was to be the first to fully utilise procedural generation as both its main method of content creation and gameplay experience. The game was to adopt procedural techniques as a means of creating spatial experiences larger than anyone had seen from interactive media ever before. It was to use the method to create a multitude of planets, creatures and vehicles, allowing Hello Games to promise a 'never-ending' experience of space and fantasy. Upon release, No Man's Sky was marketed to be a game-changer within the industry, but it unfortunately flopped. Hello Games' relied too much upon the procedural system being able to sustain an exciting gameplay experience and once the novelty of the 'never-ending' universe wore off for many gamers, including myself, the game became another rusty trophy on the shelf.

However, a few months later, KillScreen media published an article titled: 'What Can Architects Learn from No Man's Sky?', written and edited by Youssef Maguid. This article was to become one of the few reasons why I would go on to undergo this avenue of research. In the article, Maguid wrote about procedural systems in modern content creation;

> *"Forms are not designed, they are generated...the script and constraints are designed, but they are designed to serve themselves, not the final form. " – (Maguid, 2016)*

The ability to generate 'multiple outputs' by implementing and manipulating a 'series of inputs' gives new light what it is to be a creative designer. It changes the way in which we think about the end product, with the core of the development time spent in the initial creation of the procedural system. Interestingly enough, Maguid went on to further express his intrigue with the relationship between procedural systems and architecture;

> *"[Architects] are able to rethink and reconfigure the rules and conditions that enable and inform them in the first place...It is no longer about designing a final object or a product, but about designing or configuring the system or the process of their formation—the underlying code, algorithm, or procedure that can generate not just one but multiple outcomes." – (Maguid, 2016)*

It was with this article that the foundations of this thesis were set in 2016. Maguid's thoughts on the current re-fabrication architecture is going through in regards to how it is created and designed sparked an interest that would see this research come to fruition. I started asking myself - what if one day the Architect no longer gave his client two or three possibilities to choose from during concept design, but thousands? What would this methodology look like and how would it function?

BITS + PIECES

BACKGROUND

Procedural Design Tools

Houdini is an industry standard software when it comes to visual effects in film and content creation in game design. Developed by SideFX, it is a powerful program that employs a procedural workflow in parallel with a visual programming interface. The software acts as a vehicle for realising the users' intent and is often referred to as being only limited by their imagination. The software is a complete platform, providing tools for procedural programming, physics simulations, rendering and as of late, acts a third-party link to real-time virtual engines. As seen with No Man's Sky, many developers choose to produce their own procedural tools for the development of their films and games. However, Houdini is seen as the go-to, out-of-the-box product for amateur and semi-professional developers and designers who wish to engage with procedural systems. The software gives designers the ability to make changes to their programming on the fly; changes that trickle down the procedural workflow and allow for the alteration of the final outcome. However, Houdini is also considered extremely hard to pick-up and learn without significant time investment. This was taken on as a challenge in this body of research and it thus attempts to utilise as much of Houdini's procedural power as it can, exploring the capabilities of procedural systems through an architectural lens.

BITS + PIECES

BACKGROUND

Procedural Design Documentation

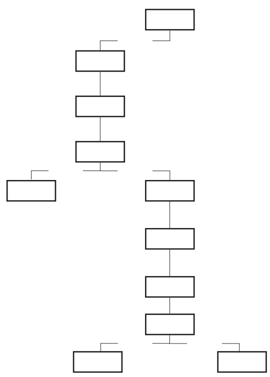


Fig 2.02. Documenting Houdini through screenshots of nodal interface.*

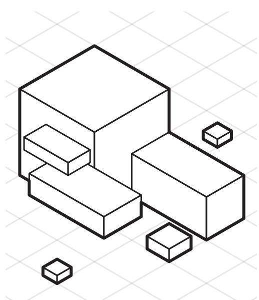


Fig 2.03. Documenting Houdini through communicative diagrams.*

When it comes to documenting the workflow of the Houdini software, as with any other digital program, there are a multitude of possibilities. This research looks to two methods in which the workflow of Houdini could be documented, as follows;

> **Method 1:** Raw Houdini workspace screenshots.
>
> **Method 2:** Communicative diagrams.

Each method has its pros and cons. The screenshot method provides an ease of creating and communicating the workflow of Houdini and the steps taken at each stage of the investigation. It also provides a means of common ground for those familiar with the Houdini interface, but becomes difficult to understand without prior understanding of the software itself. The diagram method allows for the negation of bias and communicates Houdini systems clearly, but also results in a longer time-frame to complete and can often result in the oversimplification of the Houdini interface and procedural systems.

This thesis therefore adopts the use of 'Method 2: Communicative diagrams'. At each stage in this investigation, any Houdini logics used to produce content are explained by a series of diagrams that communicate the systems present. A small body of text accompanies each set of diagrams, briefly explaining the function of the system and ensuring the systems aren't oversimplified. By documenting the Houdini interface and procedural systems in this way, myself as the designer am able to avoid revealing my workflow preferences, workarounds and biases. The method allows for the Houdini interface to be understood by individuals of the Architectural discipline - those not currently familiar with procedural generation. Also, by documenting the Houdini interface in this way, this thesis avoids communicating the overwhelming nature of the Houdini node-based workflow that can, and does, reach up to one thousand components used in a single system at any given time.

PART ONE:

Research for Design

THEORETICAL

Theoretical Positioning

BITS + PIECES

THEORETICAL

Introduction

This body of research aims to understand how the computational technique of procedural generation could be adopted at the infancy of the architectural design process. It investigates the relationship between procedural methodologies and architectural practice, and through design, explores how the technique may be integrated into the architectural workflow.

This chapter discusses some of the key thinkers involved in both procedural generation and computational architecture, highlighting the core arguments for and against the use of procedural generation within the discourse of architectural practice. It first discusses the main arguments surrounding 'Procedural and Generative Design', highlighting broader theories about generative design techniques. It then moves on to discuss 'Procedural Design and the Designer', highlighting arguments surrounding the use of procedural techniques in the overall field of creative design. Finally, it moves on to discuss 'Procedural Design and the Architect', outlining arguments made by theorists and practitioners of the Architectural discipline itself, positioning this body of research within a disciplinary context. This chapter locates the research within a wider theoretical body of thought surrounding procedural design and architecture, exploring how the thoughts of many intersect and how they can be reflected upon to contextualise this investigation.

BITS + PIECES

THEORETICAL

Procedural + Generative Design

Procedural techniques are generally used within interactive media and game design in order to mass-produce content at the tail-end of a development process. This achieved by the time investment of the workflow revolving around the coding and back-end development of the procedural logic, which can be littered with numerous parameters and rules. As a result, the logic can be interacted with long after it has been established and procedures can be altered to change the possible outcomes. Patrick Janssen stated that the inherent nature of procedural design results in it being a “highly interactive environment; an impressionable ecosystem that is susceptible to manipulation” (Janssen, 2001). Procedural design thus becomes a system that - with heavy investment into its initial code - is capable of producing content in immense quantities, where each product is unique.

It is important to note here that the design, or aesthetic, for the outcomes of a procedural logic are generally not considered when developing the system. It is the system that is designed and its outcomes are complementary. This is often the case with game design, where coders work to design a system that will ultimately result in a series of unique products. This process flips the notion of traditional architectural design on its head, where instead of continually developing a current ‘iteration’ for a building, the procedural system is instead continually developed to refine its output; a series of ‘variations’ of the same brief. This is supported by Youssef Maguid, who stated that within procedural systems, “forms are not designed, they are generated…the script and constraints are designed, but they are designed to serve themselves, not the final form” (Maguid, 2016). Once a procedural system, its rules and its constraints are established, it can be manipulated at the tail-end of its workflow through a series of inputs or parameters, altering the variables for its rules and resulting in a new, unique series of outputs.

The introduction of procedural systems in creative design fields has given designers the ability to pseudo-automate the production of content, allowing them to mass-customise and mass-produce digital built form for worlds of varying scales. The concern here arises as to whether or not this type of design methodology is seeking to replace the traditional process of hand-crafted content. This instigates a conversation around the benefits of hand-crafted content vs generated content. Triple A titles such as Activision Blizzard’s Overwatch will always benefit from the former, where the finer details are essential to the game experience. The same can be said for Architectural practice, where detail is essential to the execution of a successful building. With this conversation is a worry that generative techniques such as procedural design can be seen to be ‘bettering’ an existing design methodology. However, Patrick Janssen’s theoretical belief counters this concern, stating that:

> *"[Procedural design] does not aim to solve some 'problem' with an existing design methodology. Instead, such an environment is one aspect of a new possibility. More specifically, this new possibility does not attempt to support or emulate the existing design methodology. Rather, it relies on a modified type of design methodology in tandem with a new type of computational environment." – (Jannsen, 2001)*

Janssen's beliefs provide a critical positioning for this body of research. It positions procedural design as a methodology not seeking to replace an existing architectural design process but instead one seeking to work in tandem with it. The 'modified' type of design methodology that Janssen speaks of is one that this body of research seeks to investigate - identifying the possibility of a design process that considers an amalgamation of architectural practice and procedural techniques.

BITS + PIECES

THEORETICAL

Procedural Design + The Designer

With an underlying understanding of procedural generation established in previous sections, it is important to begin understanding how the technique benefits creative designers and how creative designers are expected to respond to its capabilities. Although procedural generation acts as an alternative design methodology, it is essential it acts as one that adds to the creative process in some way. The inherent nature of procedural design means its ability to produce, en masse, a series of unique answers to a specific brief provides designers with a new way of selecting design decisions. Once a procedural system is established and capable of producing plausible outputs, the designer is presented with a multitude of possibilities that respond to their very brief at the beginning of a project. This range of possibilities gives the designer unique responses to their design-problem, allowing them to react to and evaluate each variation at a conceptual level. Patrick Janssen expressed that the benefit of a system of this type is that “users of the system are able to to give preference to certain design proposals over others, save some design types from extinction, change prediction criteria in mid run, tweak formatives, and so forth” (Janssen, 2001). As stated, the adoption of the procedural system gives the designer the ability to alter the original rules of the logic as per their needs, changing the outputs of the system entirely. By giving preference to certain outputs and adjusting the procedural rules, designers are able to refine their system and thus refine its outcomes. Involved with traditional architectural practice, Christopher Alexander emphasised the importance of designers having the ability to ‘choose’ from a set of rules, or outcomes. When discussing the rules established in A Pattern Language, he noted that the purpose of establishing a set of rules that were flexible meant that “you can solve the problem for yourself, in your own way, by adapting it to your preferences, and the local conditions at the place where you are making it” (Alexander, 1977).

A consequence of the procedural system capable of producing a multitude of content is that many of the variations will contain defects or errors that deviate from the original design brief. As will be made apparent in coming chapters of this research, the pseudo-random nature of the procedural system often results in variations that fail to meet aesthetic and functional requirements. With this, the variation becomes ambiguous and the design answer is not obvious. This body of research argues that these ‘defects’ are not to be seen as failures, but instead an opportunity for answering the designer’s brief from a different angle. It argues that the designer should accept the defects of the procedural system as a challenge and attempt to use them in a design response, treating ambiguity as a progressive possibility. Winger Sei-Wo Tseng supported this notion when discussing ambiguity and creative designers, stating that “designers are highly sensitive to the ambiguity inherent in visual stimuli, with such ambiguity having a major impact on their interpretative processing and idea production during conceptual design development" (Winger Sei-Wo Tseng, 2017).

Tseng's beliefs illustrate that designers have an ability to respond ambiguity and interpret it creatively, but this thesis argues that in order for architects and architectural designers to respond to the ambiguity inherent in procedural design, this needs to be pushed one step further. In order for architects to adopt the use of the procedural methodology in their workflow, they need to learn to create the system themselves in the first place and learn how to alter its outcomes.

BITS + PIECES

THEORETICAL

Procedural Design + The Architect

In order to understand how architects can benefit from adopting a computational process such as procedural generation, it is important to understand the limitations of doing so. With the intersection of tools used in game design and architecture becoming more common, as seen with virtual reality and real-time virtual engines, it is becoming apparent that such tools will never quite address the inherent principles of the physical world architects are required to respond to in practice. Iman Ansari echoed this, stating that "despite all the possibilities that digital tools provide, the architect always has to scale, trim, or orient the object to accommodate and situate it in within the real physical realm" (Maguid, 2016). Ansari argued that these techniques are ingrained into architectural practice and that architecture can never fully renounce its physical roots. Maguid supported Ansari's views, noting that "for architects, there is always a given: a physical, material, legal, and functional environment that architecture has to respond to" (Maguid, 2016). These beliefs are supported by Daniel Davis of WeWork, who states:

> *"In many ways, the inability to automate architecture hones in on the divide between people and machines. Architects themselves are vital despite the fact that computers are assuming many of their menial and repetitive tasks. What remains is the core competency of the architect, which is defined not in terms of what an architect is, but in terms of what a computer can't do." – (Davis, 2009)*

However, this thesis argues that procedural systems can possibly be integrated within the architectural workflow as an early stage design tool, acting as a method for ideation rather than a tool for detailing. When discussing the use of computational techniques for his architectural project 'The Embryological House', Greg Lynn noted that "[Procedural design] rethinks the notion of the manufactured house...the goal is to design and manufacture houses that exhibit variety based on shared regulating principles—a 'mass customization' to allow the mass production of individually unique products" (Lynn, 2009). Lynn's beliefs support the uniqueness that occurs in the procedural design process and the ambiguity supported by Tseng noted earlier in this chapter. With this theoretical positioning, procedural design becomes a tool for ideation, where the partnership of pseudo-randomness and en masse content results in the production of a alternative 'emergent resource'; a design method to be used for the conceptualisation of architecture as not a single design respond, but multiple.

BITS + PIECES

THEORETICAL

Reflection

This body of research aims to understand how the computational technique of procedural generation could be adopted at the infancy of the architectural design process. This chapter has discussed some of the key thinkers involved in this avenue of research, using their theoretical arguments to position this thesis within a wider body of thought. This body of research aligns itself with the theories of the practitioners and academics discussed in this chapter, arguing that procedural generation should be proposed as a tool for design ideation. In order to do so, it positions itself within the two following points:

1. *Procedural generation is a technique that gives designers the ability to produce, en masse, pseudo-random outputs that respond to a specific design brief. It is a tool for content creation that encourages the adoption of ambiguity during the creative design process.*

2. *Architecture as a practice will always be grounded upon the constraints of the physical world and it has a set of ingrained techniques that make it what it is. Thus, for procedural generation to be an effective design methodology, it needs to find avenues that add uniqueness to the architectural workflow instead of ones that seek to better it.*

These points have been shaped by the key arguments discussed in this chapter. They locate the thesis within an area of research concerned with the amalgamation of architecture and procedural techniques and identify where it can contribute to a wider field of knowledge. Subsequently, they act as theoretical objectives for the design decisions made in the future chapters of this body of research.

PRACTICAL

Case Studies

BITS + PIECES

PRACTICAL

Introduction

Whilst the previous chapter established and discussed the theoretical context in which this body of research is positioned, this chapter identifies and discusses two case studies which ground the research within a practical context. The two case studies, 'Brick Block' and 'Houdini Lake Houses', have been selected from the disciplines of interactive media and film to investigate how non-architectural disciplines are using procedural generation to ideate built form, albeit digital in the case of the two. Belonging to the 'research for design' section of this research, the investigations into these case studies are used to inform design decisions moving forward.

This body of research aims to understand how the computational technique of procedural generation could be adopted at the infancy of the architectural design process. For both case studies, this section discusses how each adopts the use of procedural techniques and how this research could build upon the use of the technique through an architectural lens. Each case study discussion covers the following questions:

> **Design Understanding:** How are procedural techniques used? How is built form generated?
>
> **Design Relevance:** What new views does the designer offer when considering the amalgamation of architecture and procedural practices?

The first part of this section will examine and discuss Oskar Stålberg's 'Brick Block', which illustrates how the pairing of procedural techniques and user interaction can give authority to the user, or client, involved in the creative process. The second will examine and discuss Anastasia Opara's 'Houdini Lake Houses', which illustrates how procedural techniques can be used to produce a multitude of formal outcomes that belong the same architectural language, with considerable architectural detail.

IMAGES REMOVED DUE TO COPYRIGHT

Fig 4.01. Compilation of variations extracted from Oskar Stalberg's 'Brick Block' case study. Each variation exhibits a unique variation of the aesthetic Stalberg set up within his procedural system.

BITS + PIECES

P R A C T I C A L

Brick Block | Oskar Stålberg

Title: Brick Block
Designer: Oskar Stålberg
Location: Digital
Year: 2015

Brick Block is a piece of interactive media developed by Oskar Stålberg - a freelance game and graphic designer. Developed within Unity 5, the game behaves much like a three-dimensional LEGO set, allowing the user to interactively add or subtract predefined components to form a building with the click of a mouse. With this interaction, the game's code takes note of the position of a each placed component and recognises its surrounding context, adapting its original form to establish a relationship with any surrounding components. Made possible by procedural techniques, the game is able to respond to the user's interaction in real-time, manipulating a component based on the location in which it is placed. An example of this is where two components are placed on different levels but located in close vicinity to eachother. The game's code recognises this as a relationship that requires a circulation connection and procedurally generates a stair between the two. This process can be repeated in many different variations, forming railings, porches, verandas and structure. There are subtle hints to the architectural discipline here too, where in some cases downpiping and facade treatments change depending on the complexity of the user's decisions. Built into this case study is also the possibility to randomly generate a complete configuration of multiple components, opening up the possibility for an infinite number of outcomes. This is made possible through the implementation of a 'seed' value, where each variation of the random function is given a different seed variable, resulting in a completely different arrangement of components and a unique final product.

Design Relevance:

This project illustrates how procedural techniques can be used in the creation of digitally built form. Purposed as a piece of interactive media, or game, Stålberg's project begins to explore how procedural techniques may in fact be used in an architectural context. Brick Block provides insight into how procedurality, paired with user interaction, can be used to augment - or change - the contemporary process of ideating a habitable space. By doing so, it gives understanding to one of the research aims of this thesis: 'identify how procedural generation may offer an alternative methodology to the production of architecture in early stage design.' Although rudimentary in nature in regards to its architectural aesthetic, the project highlights one of the strengths of procedural techniques when concerning architectural practice; the ability to give the user, or client, total authority over the the final outcome by giving them access to a multitude of possible opportunities.

IMAGES REMOVED DUE TO COPYRIGHT

Fig 4.02. Compilation of variations extracted from Anastasia Opara's 'Houdini Lake House' case study. Each variation exhibits a unique variation of the aesthetic Opara set up within her procedural system.

BITS + PIECES

PRACTICAL

Houdini Lake House | Anastasia Opara

Title: Houdini Procedural Lake Houses
Designer: Anastasia Opara
Location: Digital
Year: 2016-2017

Anastasia Opara is a freelance digital designer with a background in creative programming and photographics. Her career has seen her produce three-dimensional built environments for game and film design, mainly through the vehicle of Side FX's Houdini. In 2016, Opara published a series of five tutorials investigating the capabilities of procedural systems within the Houdini software, illustrating how procedural techniques can be used to generate architectural content for film and interactive media. The project, titled Procedural Lake Houses, used procedural functions created through the VEX programming language native to Houdini and established a procedural systemthat capable of generating digital buildings to a very detailed architectural aesthetic. Similar to Brick Block, Opara's project is capable of producing multiple outcomes of the same architectural language, including subtle details such as structure, weathering of materials and a visual sense of gravity. However, what Opara's project lacks is a sense of inhabitation. The system used to produce the series of variations is not capable of understanding relationships between internal spaces and instead relies of a crude understanding of visual architectural aesthetics in order to produce a built form. Through investigation, this weakness is isolated to the scope of the procedural system itself and would have been possible with significantly more time investment. Again, as stated earlier, Houdini is possible of producing almost anything and is only restricted by the designer's time constraints, as made clear with this project.

Design Relevance:

The tutorials and outcomes of Opara's work provide valuable insight into the procedural capabilities of the Houdini software and act as the initial foundation for this body of research. The Houdini Lake Houses project illustrates how procedural techniques can be used to 'mass-customise' architecture at a rudimentary level. However, as stated above, the challenge lies in understanding how architecture practices such as understanding function can be implemented into an automated system. Opara's work aligns itself with the first aim of this research: 'identify how procedural techniques can be used in the process of ideating architecture' and the first research objective: 'explore, through computational design, the limitations and constraints that occur in the process of mastering design orientated procedural techniques'. The initial design investigations in this body of research will involve understanding Opara's methodology in her tutorials and will seek to challenge them in an alternative way, made possible through a series of design experiments that occur from the learning process.

BITS + PIECES

PRACTICAL

Reflection

This body of research aims to understand how the computational technique of procedural generation could be adopted at the infancy of the architectural design process. This chapter has discussed two case studies that illustrate how procedural techniques have been used in interactive media and film in order to ideate built form. Each case study adopts procedural generation in a unique way, amalgamating it with a knowledge of architectural aesthetic in order to create unique procedural-design processes. Although similar in nature, each case study offers a unique perspective on how procedural techniques may be implemented into architectural practice. The following findings align themselves with the position of this thesis and will be used to inform the design process moving forward.

- *Stålberg's project combines user interaction and complex procedural logics to create rudimentary architectural forms that begin to speak of a real-world architectural aesthetic. It gives authority to the user, or client, involved in the creative design process. The challenge lies in understanding where the architect and user/client differentiate themselves within the procedural workflow.*

- *Opara's project illustrates the possibility of a procedural logic that can produces complex, detailed architectural outcomes. However, its lack of implementing an architectural function or relationship between spaces proves to be an interesting avenue to take this research.*

These case studies are acknowledged and referred to throughout this body of research during the design process. Their methodologies, strengths and weaknesses are the foundation for the practical context of this work.

PART TWO:

Research Led Design

BITS + PIECES

DISCOVERY

Concept Design

BITS + PIECES

DISCOVERY

Introduction

This chapter contains the first 'research led design' investigation for this body of research. It acts as a phase of discovery, exploring how procedural techniques might begin to amalgamate with an architectural workflow through learning and understanding the possibilities of the Houdini software. The design explorations in this chapter are not restricted by architectural site or program, allowing for an investigation into procedural techniques that is not limited by constraints of the physical world. The result, as will be apparent, is a raw, ambiguous 'emergent resource' that begins to respond to an architectural aesthetic.

This phase of the research consists of three procedural design experiments. The first two experiments, titled 'procedural experiment one' and 'procedural experiment two', investigate procedural techniques that closely relate to the Oskar Stalberg case study 'Brick Block', encapsulating his design methodology of using random-number generation to explore procedural forms. The third design experiment, the 'concept design', evolves from the findings of the first two experiments and acts as the main design experiment for this chapter. It aims to investigate how a procedural system can begin to respond to architectural tectonics such as walls, floors and structure and results in an 'emergent resource' that begins to illustrate how proceduralism can be used as an architectural tool.

Aim:

As the first set of 'research led design' experiments for this body of research, the design explorations in this chapter aim to develop a better understanding of procedural techniques within the Houdini software and act as the very first engagement this research has with developing a procedural system. It is the first investigation into the amalgamation of architecture and procedural design.

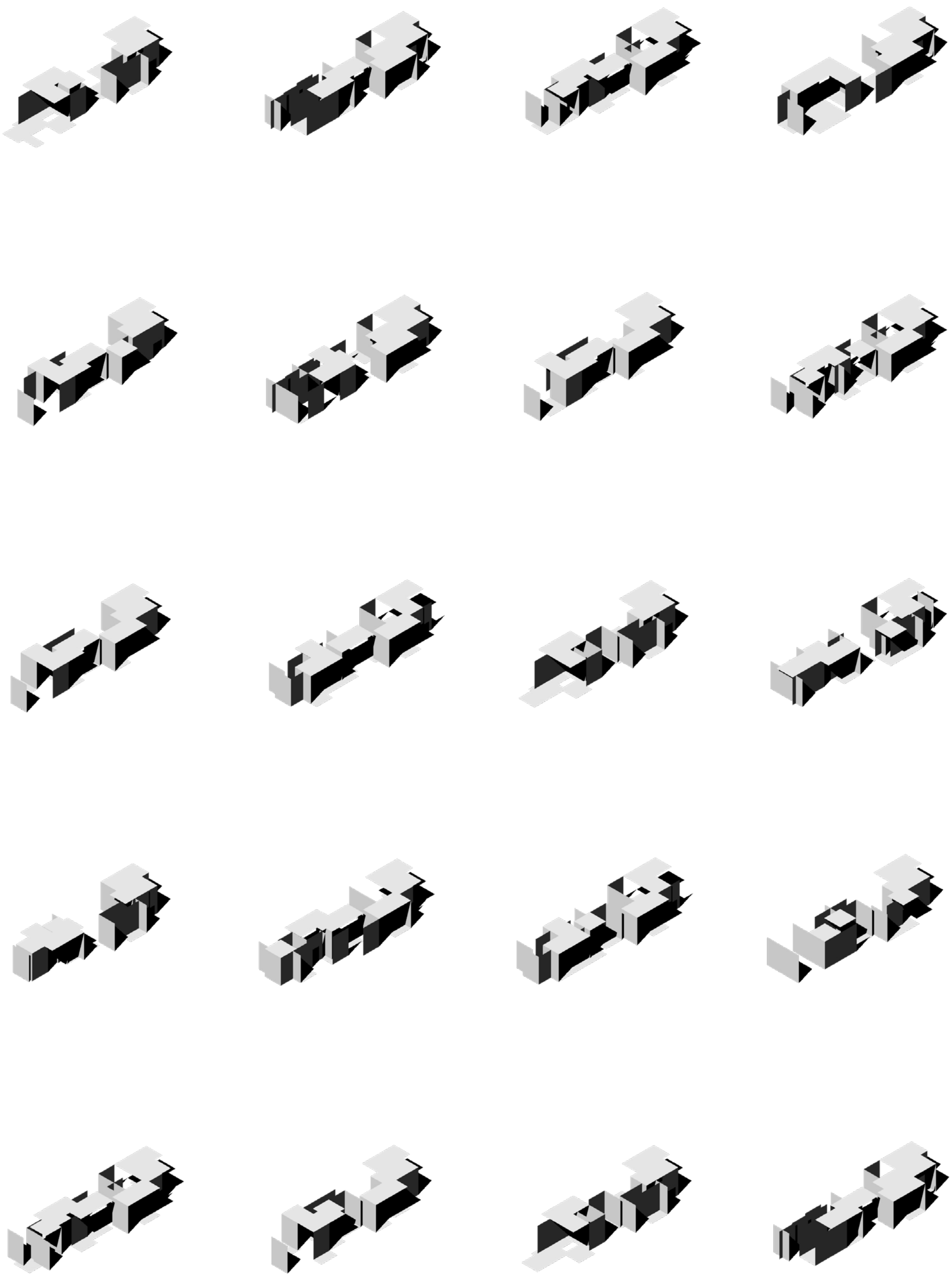


Fig 5.01. Compilation of variations extracted from procedural system developed during the first procedural experiment. Each variation illustrates a series of unique translations as a result of the rules set for the system.*

BITS + PIECES

DISCOVERY

Procedural Experiment One

The first design experiment in this chapter explored how basic procedural systems in Houdini can create digital massings that relate to each other within a bounding area. For this exercise a brief was set to design a system that could take a component, which this in this case was a 'wall', and create ambiguous outputs that did not consider inhabitation or user-interaction. The system focussed on creating a collection of walls that would relate to each other spatially without the constraints of the physical world. Procedural techniques were used to create multiple copies of the wall at given locations and manipulate the placement and orientation of each. The procedural system itself used a random-number generator to process the manipulations and for each variation this random-number was altered. The result was a series of outputs that were unique from eachother, but are still a product of the same system.

This design investigation acted as the first engagement this research had with the use of a procedural system within Houdini in relation to a simple brief. It was intended for this exercise to be entirely removed from the physical world and thus was solely as a digital exploration. Architectural site, program and function were irrelevant to this exercise and the aesthetic produced was nothing but a product of the system designed. This investigation echoed the theoretical position of Youssef Maguid discussed earlier, illustrating the importance of designing the procedural system to serve itself and leaving its 'emergent resource' or series of outputs to be generated as a result; "forms are not designed, they are generated…the script and constraints are designed, but they are designed to serve themselves, not the final form" (Maguid, 2016).

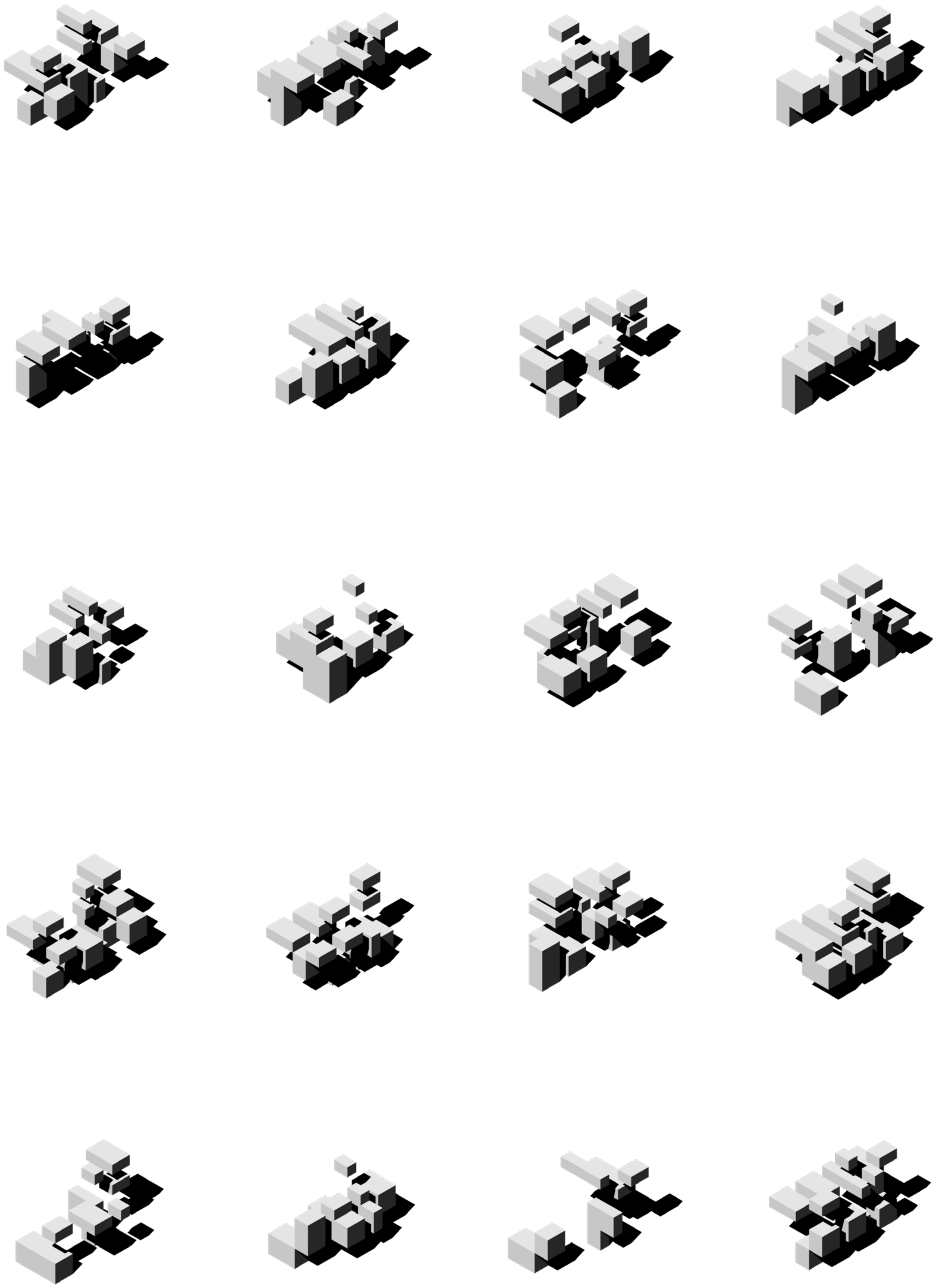


Fig 5.02. Compilation of variations extracted from procedural system developed during the second procedural experiment. Each variation illustrates a series of unique translations as a result of the rules set for the system.*

BITS + PIECES

DISCOVERY

Procedural Experiment Two

The second design experiment in this chapter built upon the findings and techniques developed previously, whilst attempting to emulate the methodologies present in the Oskar Stalberg case study 'Brick Block'. The brief for this experiment was to design a procedural system that would produce a series of outputs made from components that relate to each other spatially through proximity checks. The component used for this system was determined to be a 1x1x1 block for ease of processing. Attempting to echo the methods present in Stalberg's 'Brick Block', the way in which the procedural system was designed meant that the initial component was located at a random location in digital space and assigned a x, y and z value. A variable amount of copies of the component were made and as the procedural system processed each copy, it assigned them a unique random location that had not yet been assigned, within a given bounding box. Once this process was complete, it was repeated a variable number of times over with different random-number values in order to produce the varying outputs evident here.

This experiment did not consider the implementation of user interaction as a variable as seen in the case study. Its intention was to instead begin understanding how procedural systems can adopt simple architectural techniques such as a consideration for proximity between building elements. Although rudimentary at this stage in the research, this experiment and its predecessor were both intended to be learning experiences that would establish an initial understanding for proceduralism within the Houdini software.

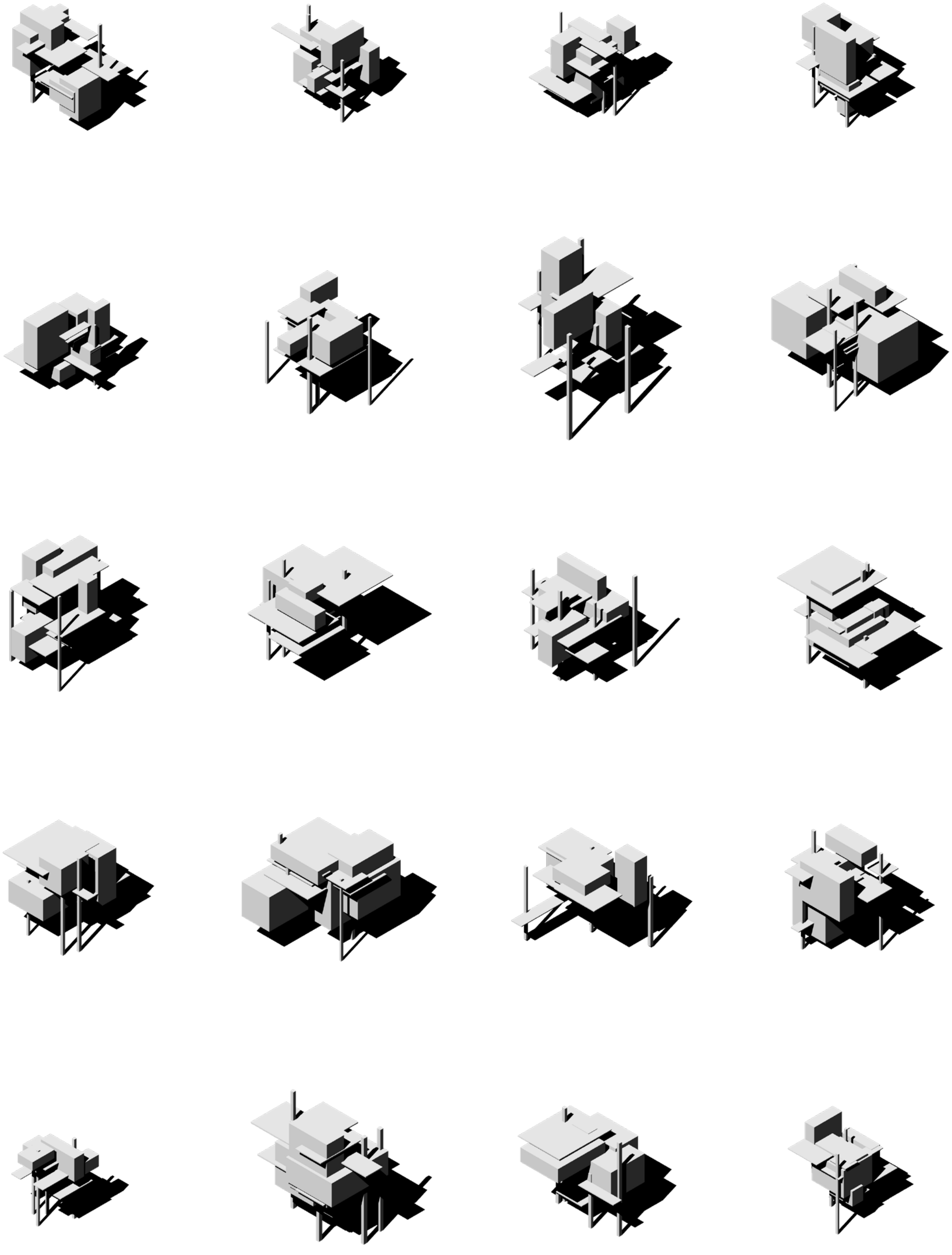


Fig 5.03. Compilation of variations extracted from procedural system developed during the concept design of this research. Each variation illustrates a series of unique translations as a result of the rules set for the system.*

BITS + PIECES

D I S C O V E R Y

Concept Design

Acting as the final design investigation for this chapter, the 'concept design' for this body of research consisted of an 'emergent resource' that began to respond to architectural tectonics such as walls, floors and structure. The procedural system for this investigation was designed specifically to include the findings and techniques used in the two previous design experiments, whilst including a refined procedural logic that sought to reduce the amount of randomness occurring between components. The design of this procedural system also illustrated the shear capabilities of procedural generation with the system capable of producing 10,000 unique varying products that respond to the same brief. The result was a procedural system that was capable of producing individually unique outcomes that began to respond to an architectural aesthetic and language.

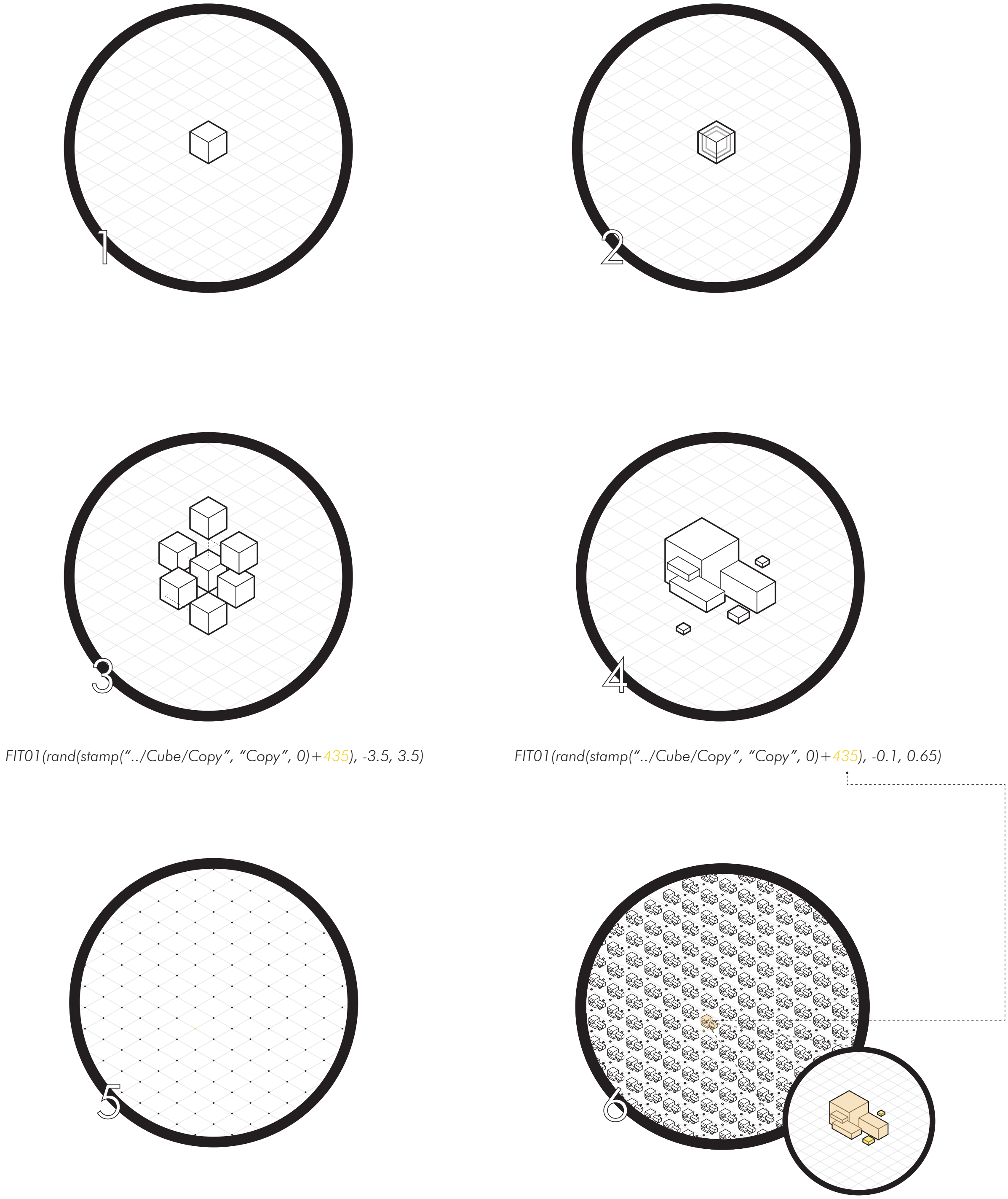


Fig 5.04. Diagram illustrating the procedural system developed for the concept phase of this research. The system uses a space displacement logic, where each component is treated uniquely within digital space.*

BITS + PIECES

DISCOVERY

Concept Design | The System

The design of this procedural system was achieved through a combination of various procedural techniques, including proximity checks and random-number generation. It also introduced the ability to translate and scale components by changing parameters in the procedural code, allowing the system to generate outcomes that were not only unique in regards to the location of components, but also unique in regards to the size of them. Once this part of the system was set up and tested on a single outcome, it was integrated with a copy-stamp process within Houdini. The copy-stamp process allows the designer to apply a procedural system to a range of values, where each acts as a variable within the system and thus change the output. At this stage of this research, the range of values was set to 0-10,000 in order to explore how much variance could be produced between each output. The rules set up for the system in this stage of the research are summarised below:

1 | Generate cube at base plane and assign dimensions to x, y and z parameters.

2 | Generate 'x' number of copies of cube at base plane - no scale/ transformation.

3 | Transform cube on x, y and z axis a random dimension between -3.5m and 3.5m.

4 | Scale cube along x, y and z axis of primitives between 10% and 65% of original size. '

5 | Assign grid points to base plane and number them from 1 through to n=maximum number of points.

6 | Mirror number of grid points to a given 'seed' value and feed these into translation/scale expressions to inform randomised output. Copy base procedure to each point on grid and generate 'x' number of variations: determined by number of points on grid.

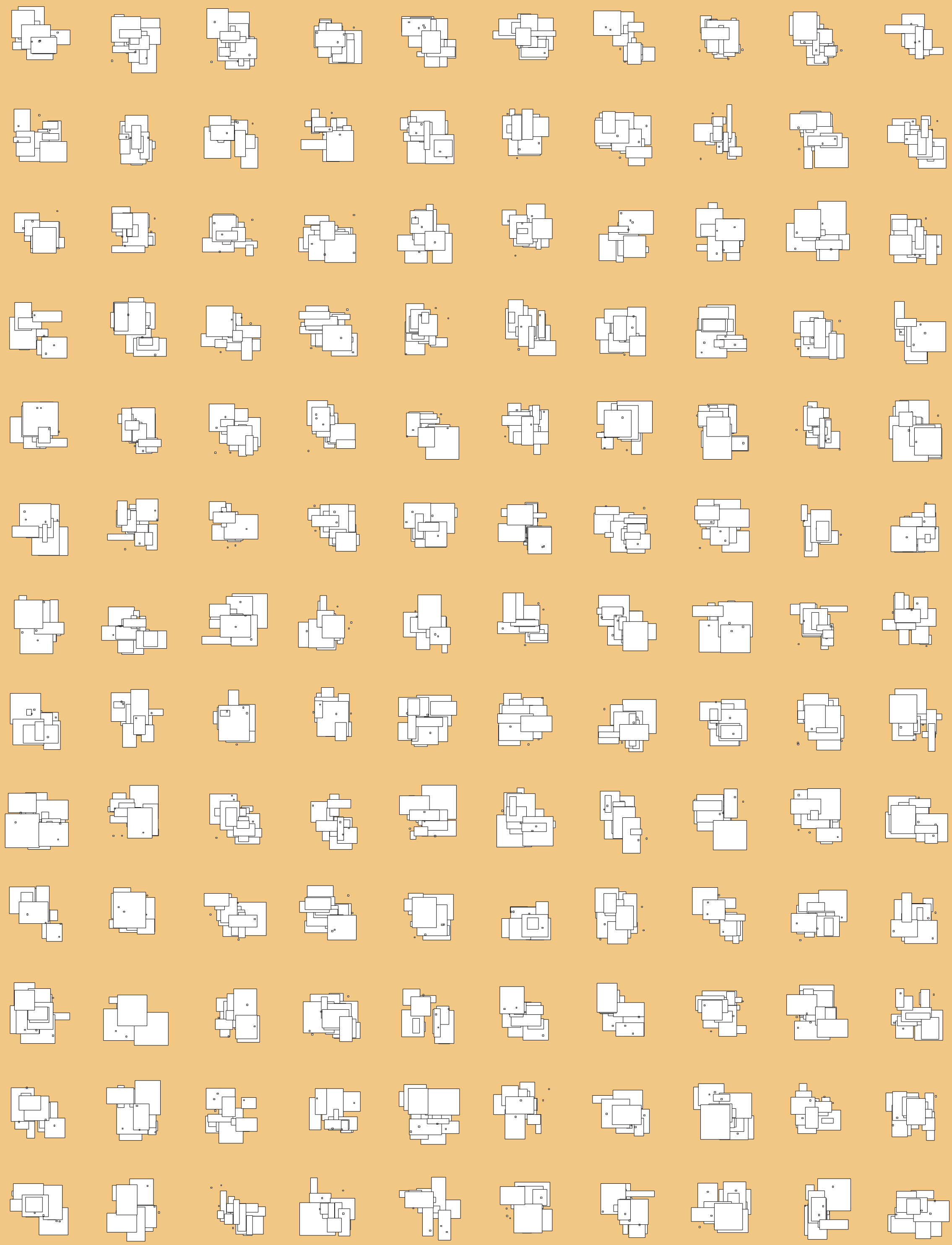


Fig 5.05. Compilation of variations, in plan, extracted from procedural system developed during the concept phase of this research.*

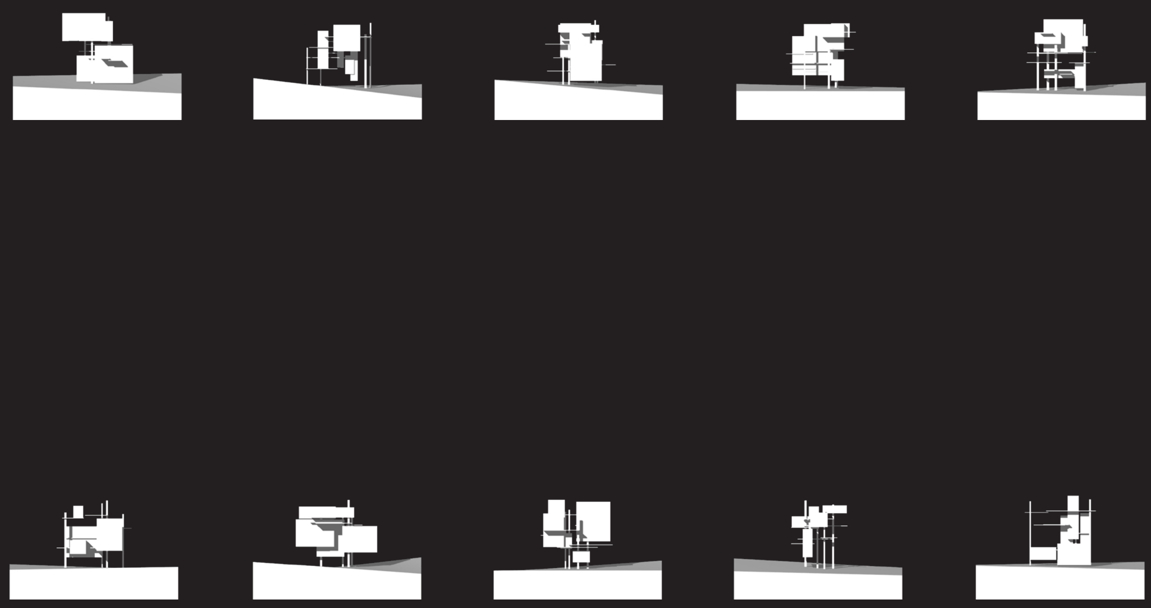
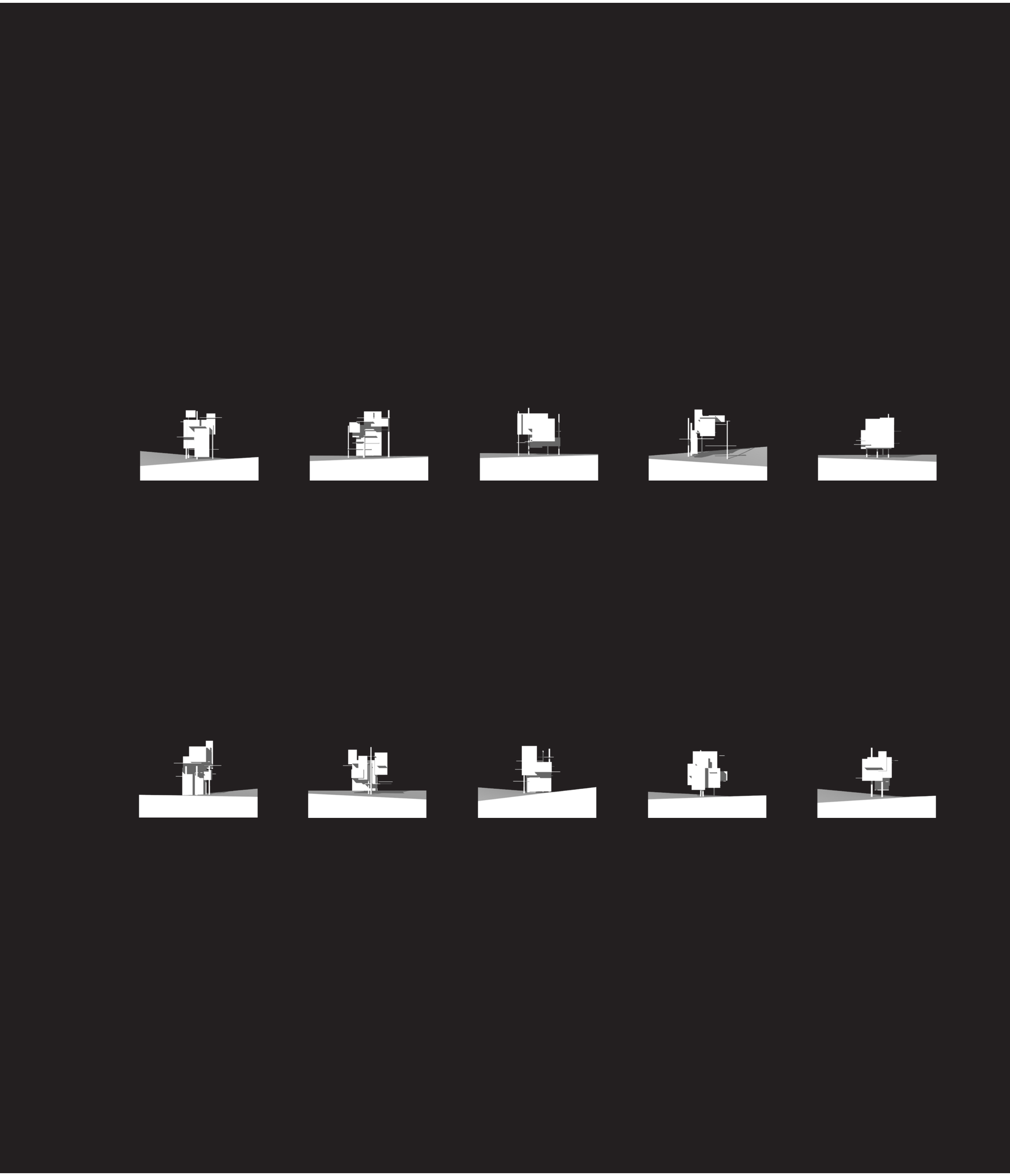


Fig 5.06. Compilation of variations, in elevation, extracted from procedural system developed during the concept phase of this research.*



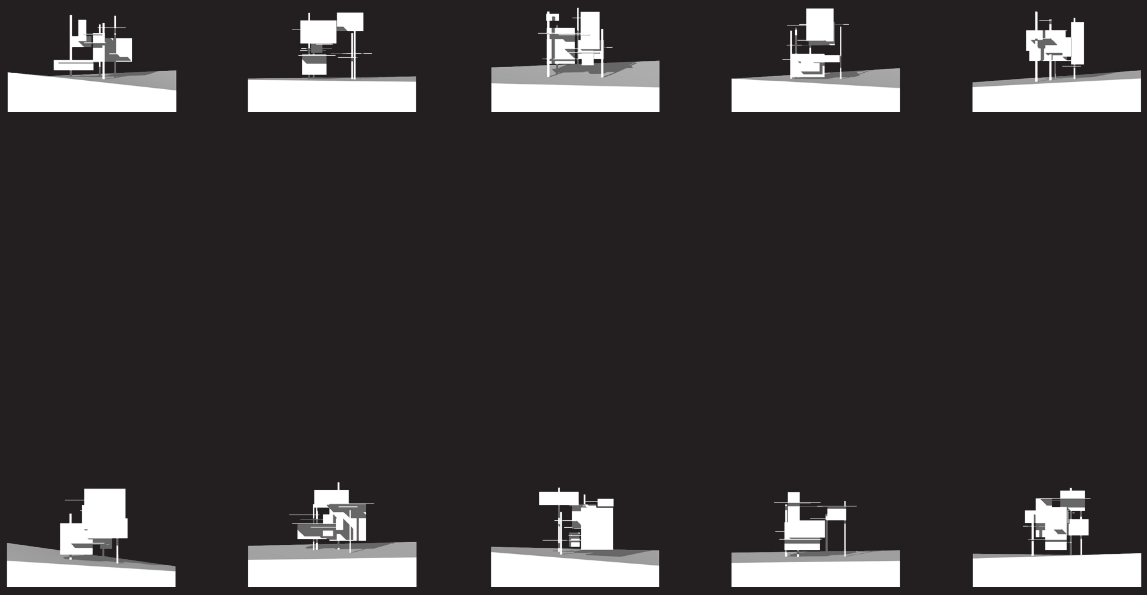
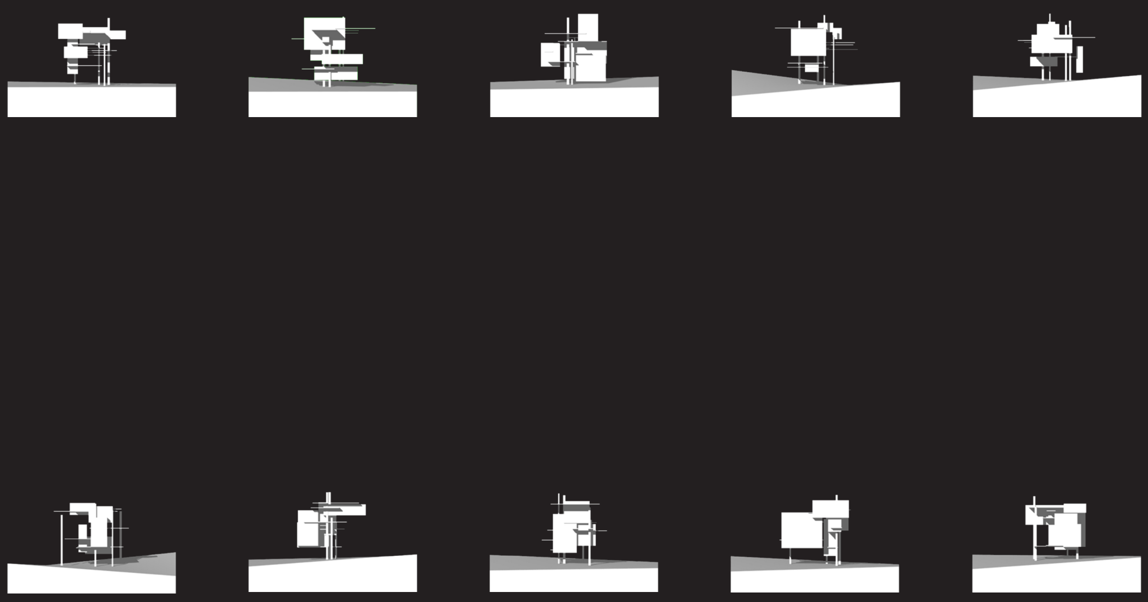


Fig 5.07. Compilation of variations, in elevation, extracted from procedural system developed during the concept phase of this research.*



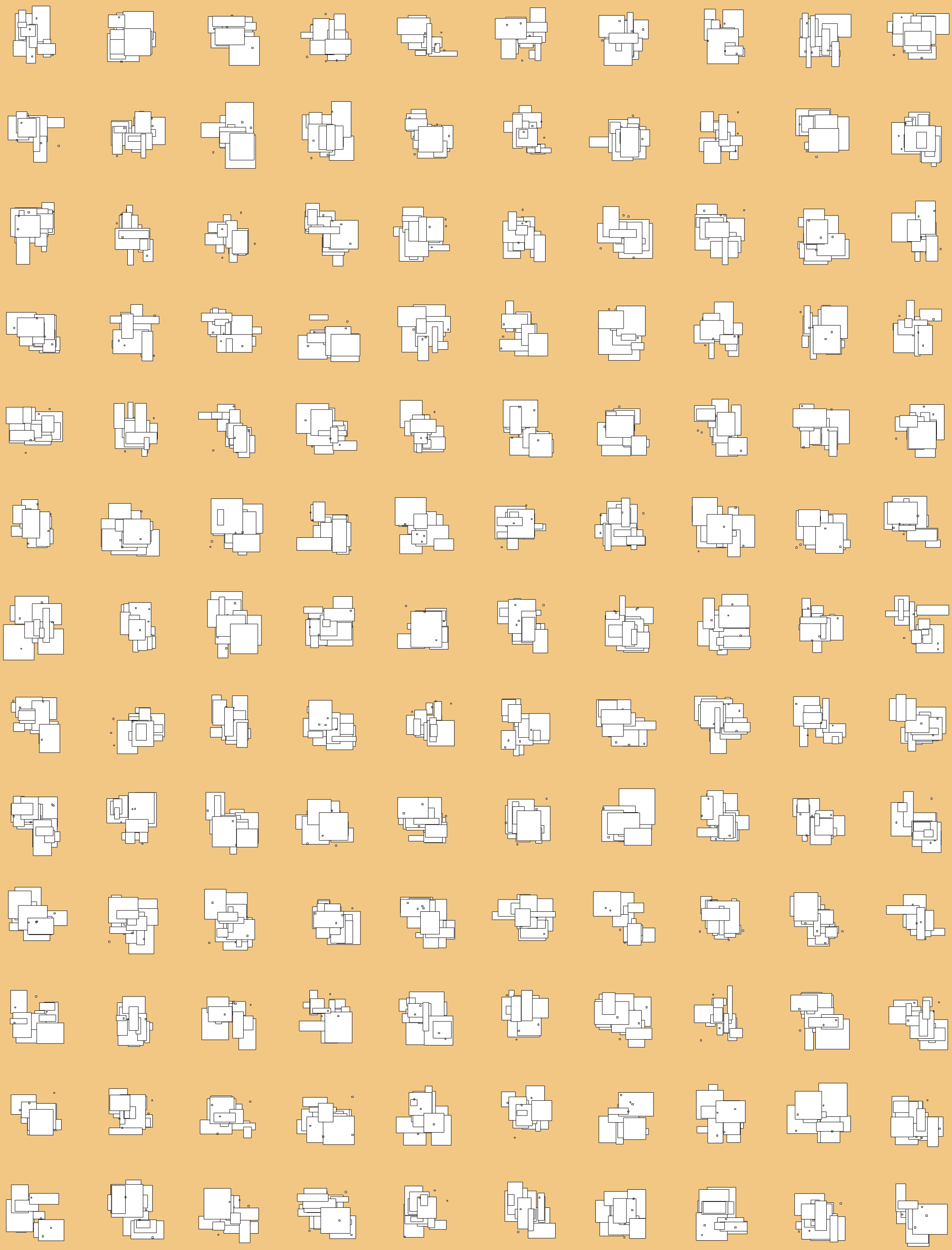
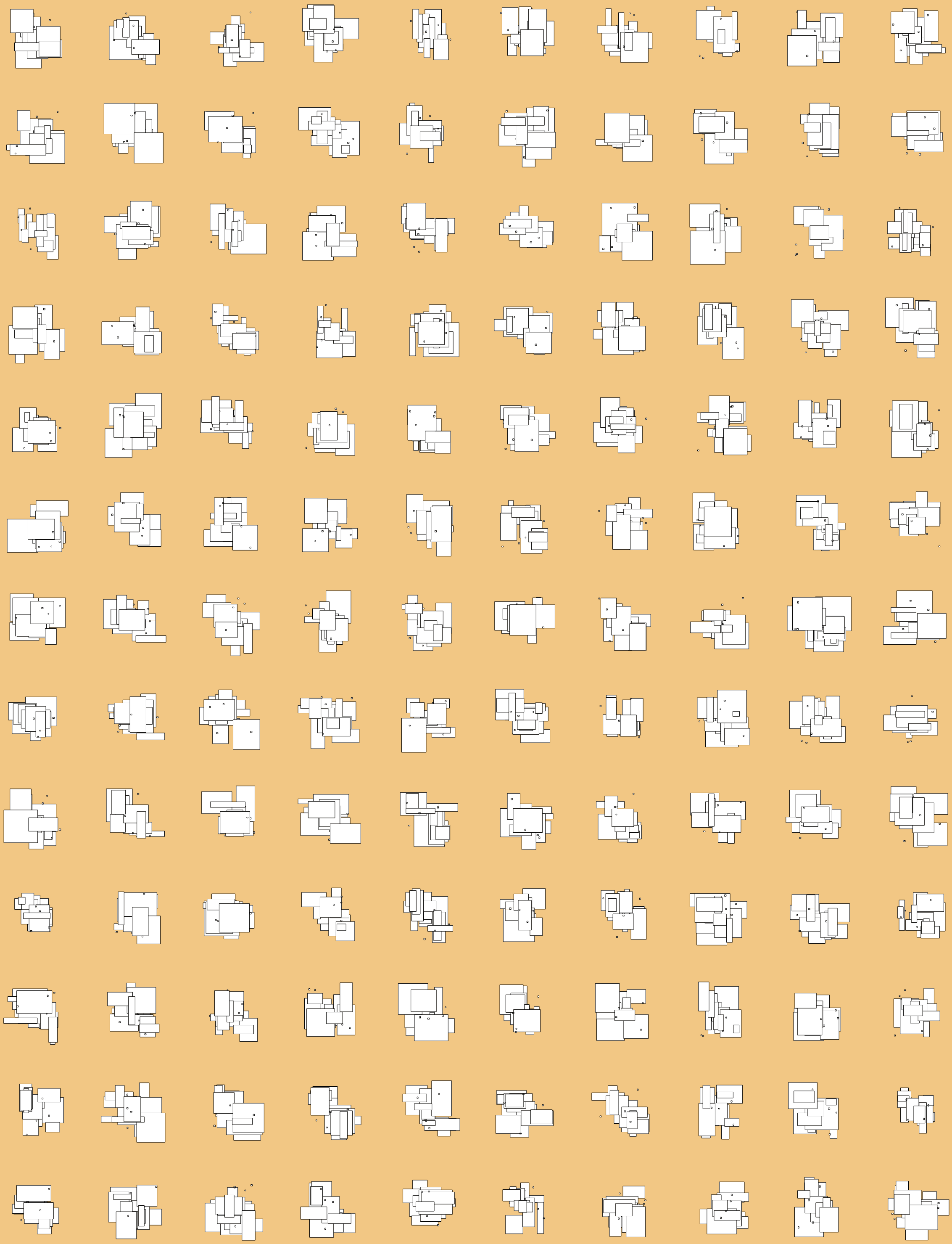


Fig 5.08. Compilation of variations, in plan, extracted from procedural system developed during the concept phase of this research.*



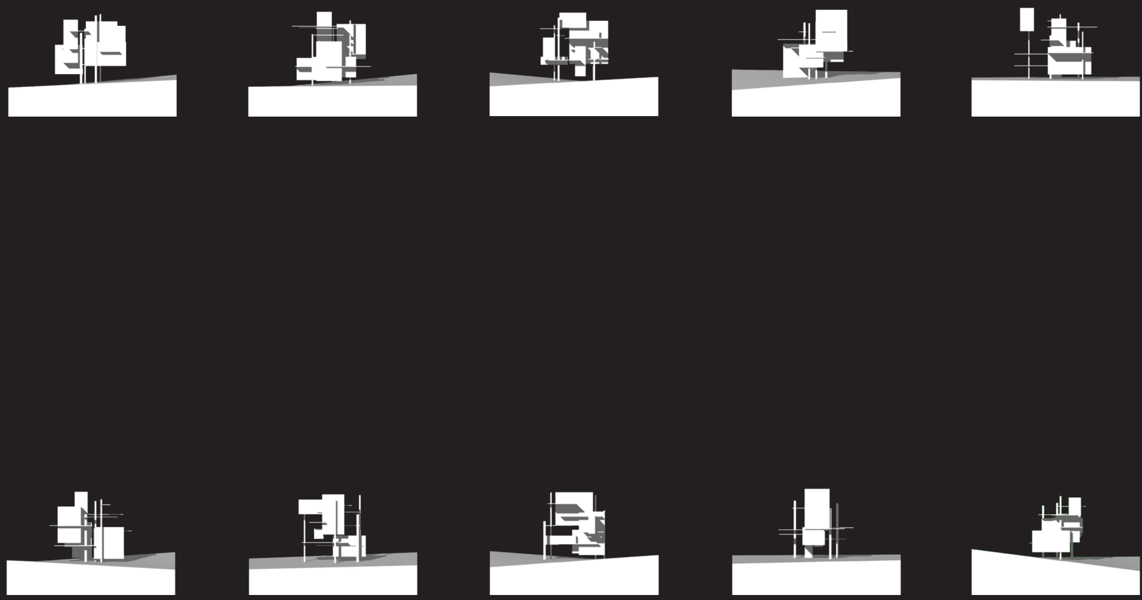
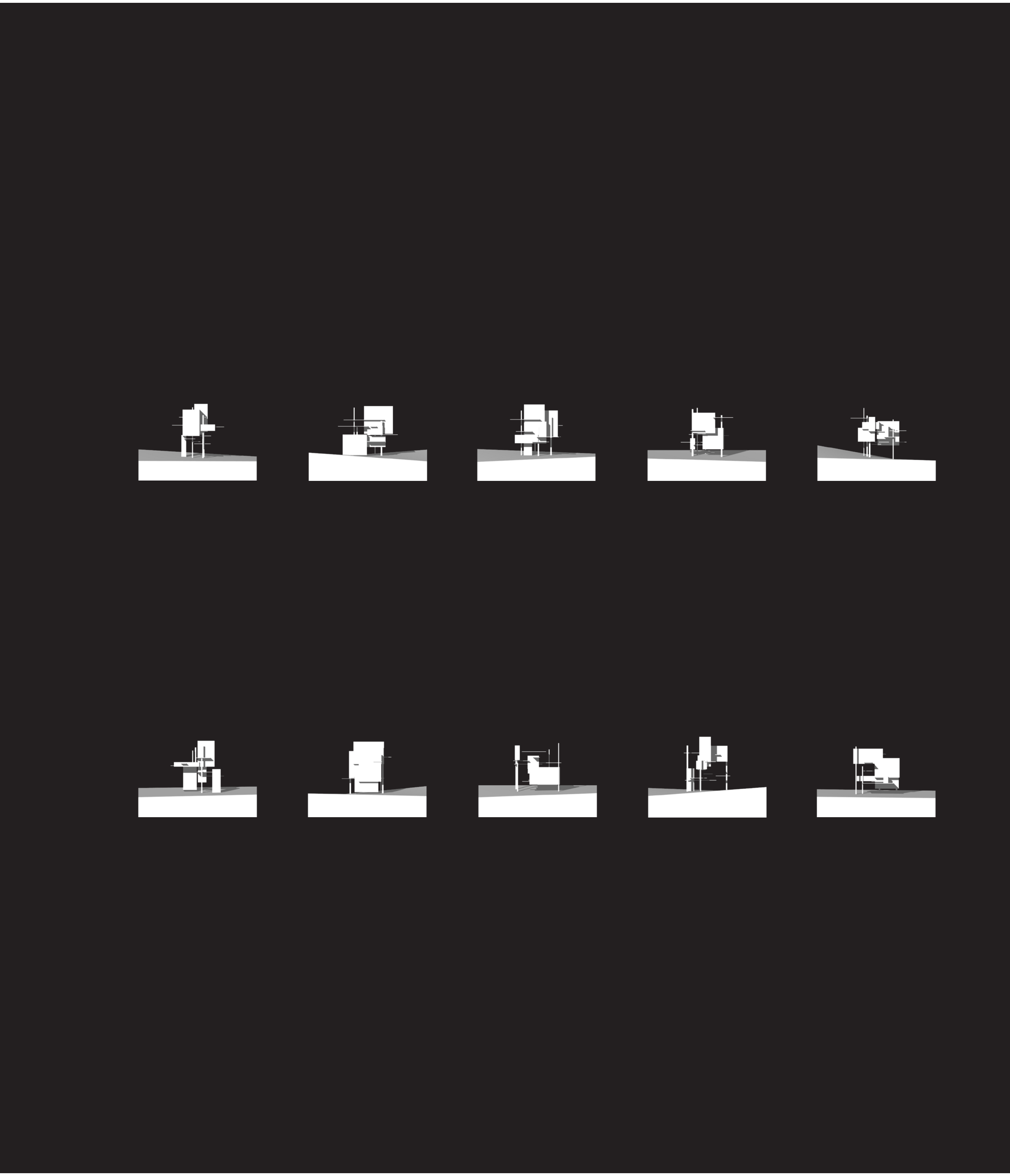


Fig 5.09. Compilation of variations, in elevation, extracted from procedural system developed during the concept phase of this research.*



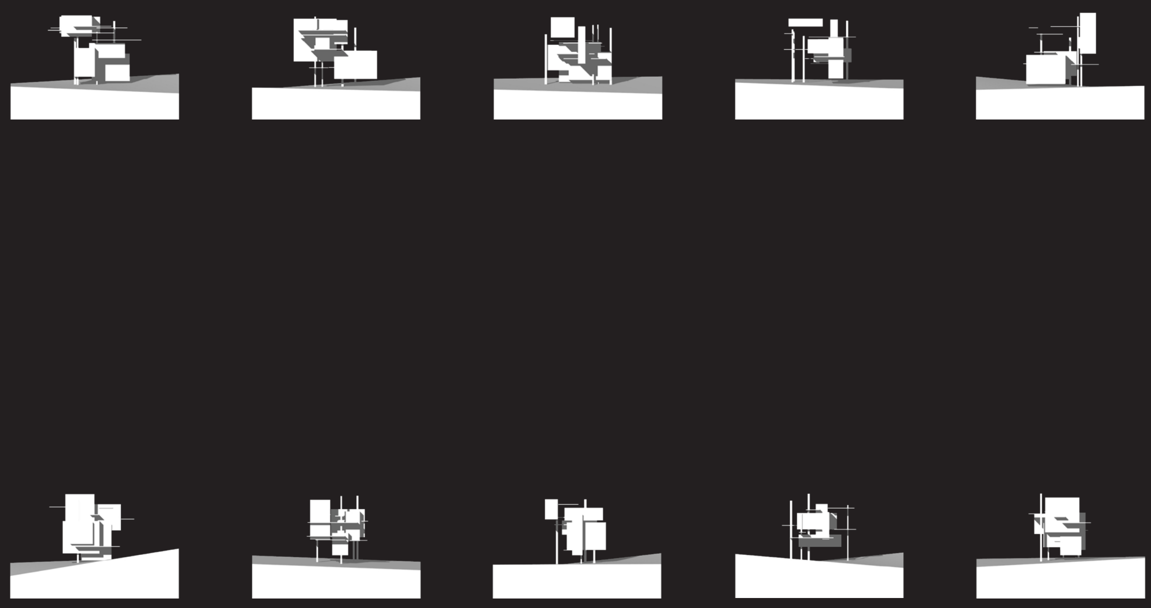
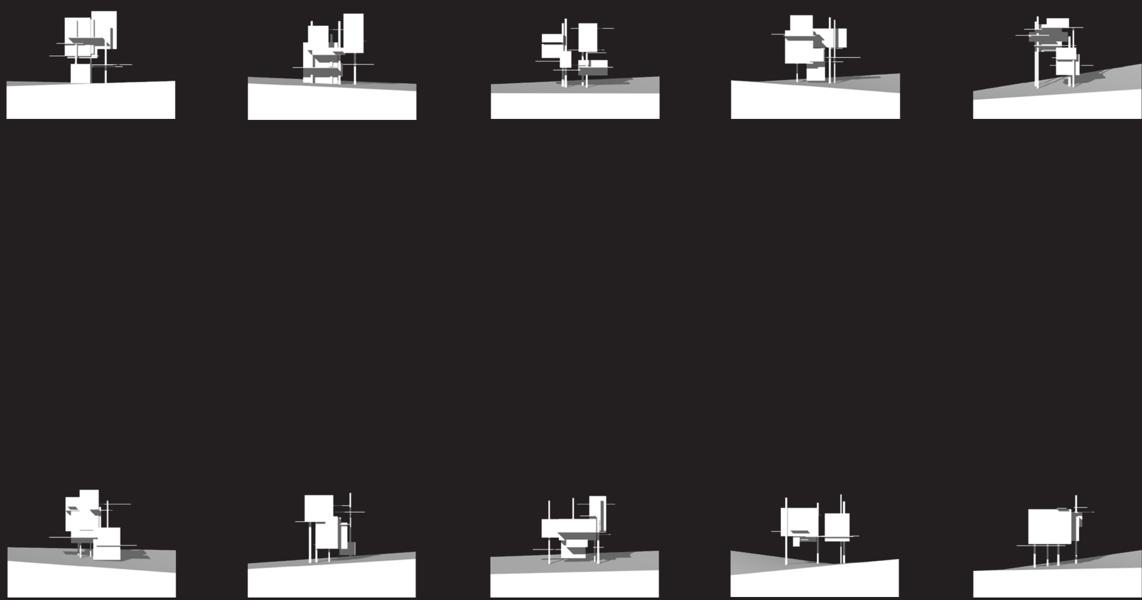
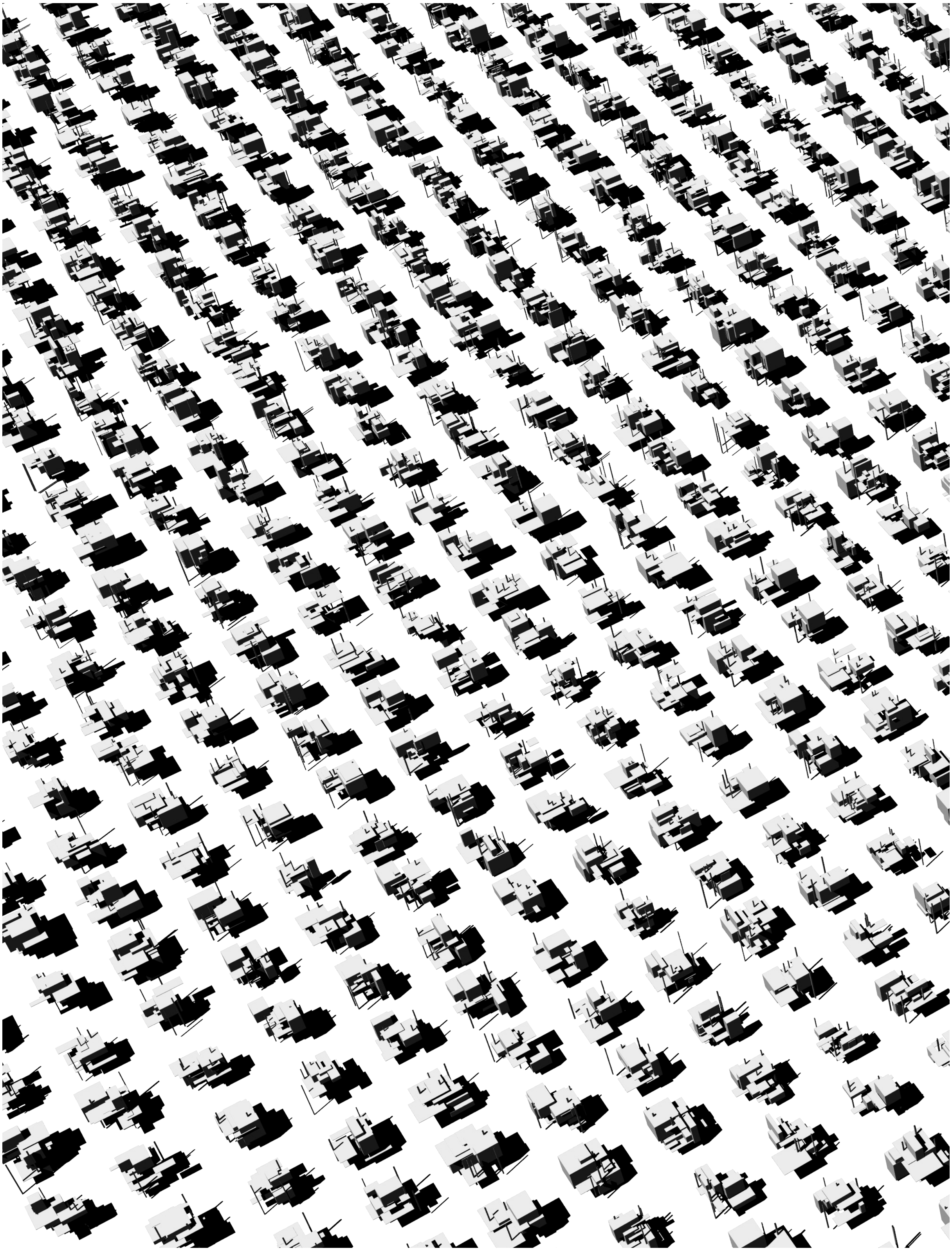


Fig 5.10. Compilation of variations, in elevation, extracted from procedural system developed during the concept phase of this research.*





BITS + PIECES

DISCOVERY

Concept Design | Positioning

Acting as the final engagement this research had with procedural techniques in Houdini in this chapter, this design investigation explored how the initial design of a procedural system could be altered in order to change its outputs. It was investigated this way in order to begin exploring how a procedural system could possibly add to the architectural workflow and thus was intended to be a rather ambiguous process, separated from the physical world. By exploring the capabilities of a procedural system in relation to how much it could allow for output manipulation, this design investigation began to echo what Patrick Janssen referred to as a 'highly interactive environment' (Jannsen, 2001).

Fig 5.11. [Left] 3D isometric perspective of variations extracted from procedural system developed in concept phase of this research.*

BITS + PIECES

DISCOVERY

Reflection

This phase of the research set out to develop an understanding for how procedural generation can be performed within the Houdini software through a series of three design investigations. The first two explored the application of novel procedural techniques within Houdini, whilst the third and core investigation increased the complexity of the procedural systems. This section illustrates the ability for procedural systems to produce content at a large scale, ranging from an individual output right through to 10,000 outputs. While the procedural system is rudimentary at this stage and does not exhibit an acknowledgement of architectural site, program or function, it does engage with the ambiguous and pseudo-random nature of procedural design. The investigations present in this chapter illustrate how procedural techniques can be used to create architectural massings, rather than detailed, architectural models. Upon presenting this phase of the research at the first design review, the following comments were made:

> *'You don't want to impose a city upon people. You want people to have the ability to choose. It's really important to let people design, in a real way, their sense of home – and this system will let you do that.'*
> *– Tom Kluyskens*

> *'It's hard to illustrate a liveable building, in terms of its interior and inhabitation, in a kind of way that communicates the variation and difference to the scale in which such an algorithm can produce. All of this 'stuff' can get a bit uncomfortable because it's not about architecture, it's about representation about form and detail. At some point you need to move beyond the grid and think more towards the inhabitation.'*
> *– Michael Dudding*

Moving onto the subsequent phases in this research, it will be integral to critically reflect and consider the amalgamation of the findings present in this chapter and the ability for user interaction to take place within the procedural system. Although Michael Dudding's comment holds true when considering how this research could add to the architectural workflow, further development must first take place around the existing procedural system. With that said, Tom Kluyskens is correct in expressing the importance of letting people design a sense of home, even when using an automated process such as procedural design. The upcoming phase in this research will thus look at refining the procedural system developed in this section and will investigate the implementation of recursive design; providing the user with the notion of choice.

BITS + PIECES

REFINEMENT

Preliminary Design

BITS + PIECES

REFINEMENT

Introduction

This chapter contains the second 'research led design' investigation for this body of research. It predominantly builds upon the procedural system established in the previous chapter, exploring how the design of the system can be developed and refined to allow for a detailed manipulation of its outputs. It subsequently investigates the ability to have the designer of the system and, in turn its user, choose a specific outcome and re-run the procedural process, introducing the ability to specify preference of one outcome over others. As before, the design explorations in this chapter are not restricted by architectural site or program, removing the constraints of the physical world. What it does do however is begin to explore how the design of a procedural system can be altered to allow for its outcomes to mimic a given architectural aesthetic. The result, as will be apparent, is an 'emergent resource' that still retains ambiguity, but begins to adopt architectural languages evidenced from real-world exemplars.

Aim:

As the second 'research led design' experiment for this body of research, the following investigation aims to build upon the findings and critique of the previous. It explores notions of refinement and choice and aims to further the understanding of procedural techniques within the Houdini software. It also aims to further build upon understanding where procedural design may sit within the contemporary architectural workflow.

BITS + PIECES

REFINEMENT

Refinement of Form

Refining the design of the procedural system established in the previous chapter became the first objective for this phase in the research. Through initially investigating procedural techniques in Houdini it very quickly became apparent that, although possible of producing ambiguous architectural massing en masse, the differences between each output of the system were sporadic and difficult to control. Upon investigating procedural techniques in Houdini further, it was clear that the original coding of the procedural system was resulting in outputs that, although sharing similar formal principles, were sitting at either end of a spectrum and not illustrating a sample of the possibilities between each extreme. It also meant that, although producing extremely ambiguous results, the procedural system was producing outputs that shared random, uncontrolled formal properties. Thus, the design investigation in this chapter seeked to develop the ability to give tighter formal controls to the outputs of the system, allowing for the designer to adapt and improve the aesthetic being produced.

BITS + PIECES

REFINEMENT

Recursive Design

The second half of the design investigation present in this chapter consisted of the implementation of a procedural system that allows the designer to give preference to certain outputs over others. As discussed earlier this development came as a result of input from Tom Kluyskens and very quickly became the core development for this stage of the research. The development of a procedural system providing choice over the outputs of a procedural system that already relies on pseudo-random content generation was a challenge, pushing the understanding of Houdini to its limits. It asked for an ambitious overlapping of procedural techniques, allowing the designer to be able to pick a specific output from the initial system and have the 'preference' system produce a new set of outputs. This came to be referred to as the ability to 'recursively design', where the designer would be able achieve what Alexander stated as "solving the problem for yourself, by adapting it to your preferences" (Alexander, 1977) and will be expanded upon in the upcoming preliminary design section.

BITS + PIECES

REFINEMENT

Preliminary Design | Refinement of Form

In order to refine the initial procedural system established in the concept phase of this research, emphasis had to be put on adjusting its code to allow for tighter controls at the production stage of the procedural workflow. This was explored by attempting to create three separate systems, resulting in three mini-procedural design experiments within this investigation. The outputs of each system were expected to share a unique set of 'aesthetic principles' that would correspond to formal studies of real-life examples in built architecture. For the sake of time-keeping, these three mini case-studies were used and treated as solely aesthetic influences. It is important to note that these studies were not to be treated as an architectural investigation for this research, but rather a formal study that, regardless of scale, would inform the development of the procedural system. The procedural code design for each mini-experiment was developed in a specific manner to attempt creating a series of outputs that would thus share similar formal properties with the real-world examples. It was at this stage in the research that 'designing the system and not its outcomes' became difficult, as it was easy to venture beyond the procedural code itself and refine it in parallel with the outputs instead.

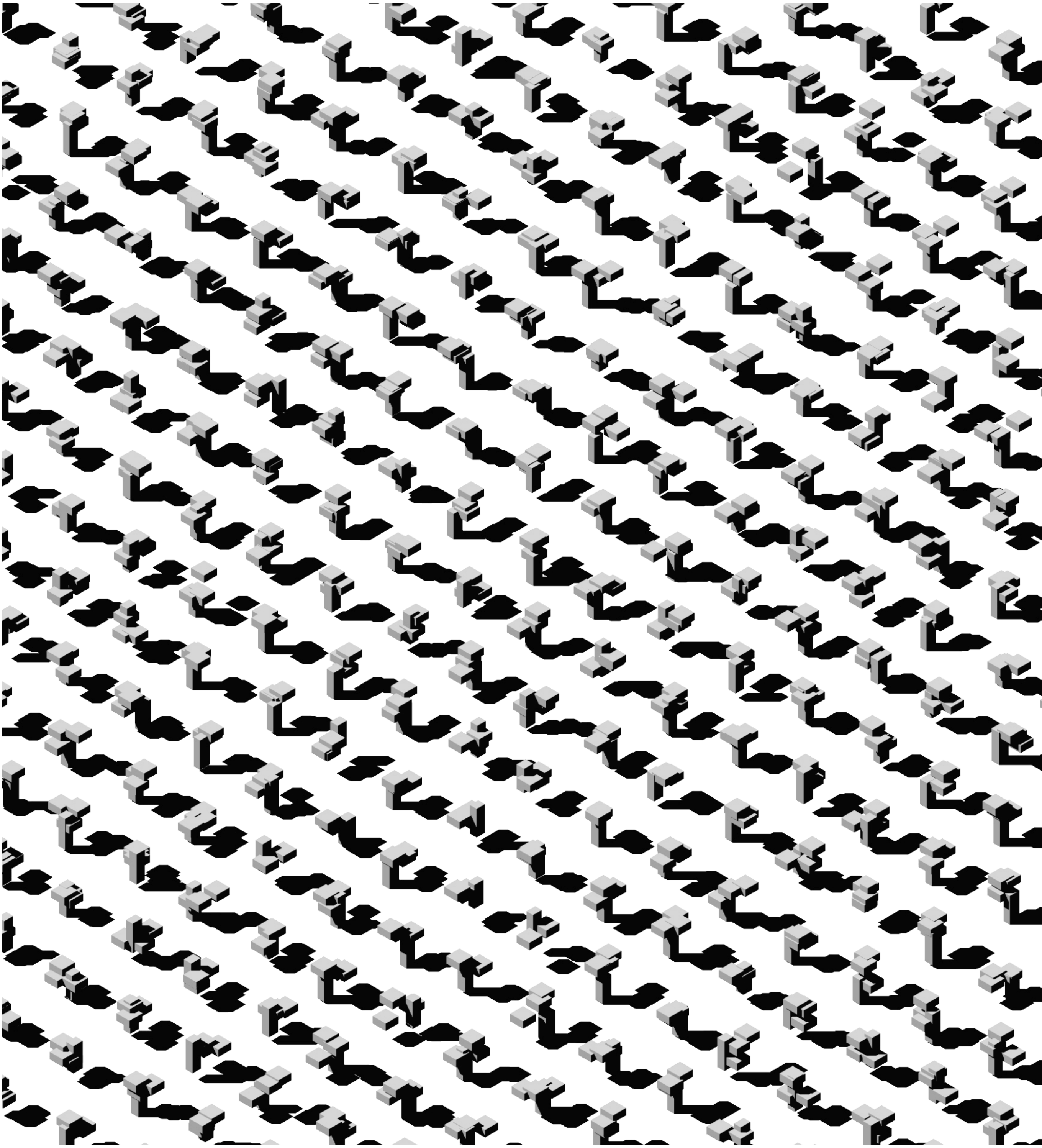
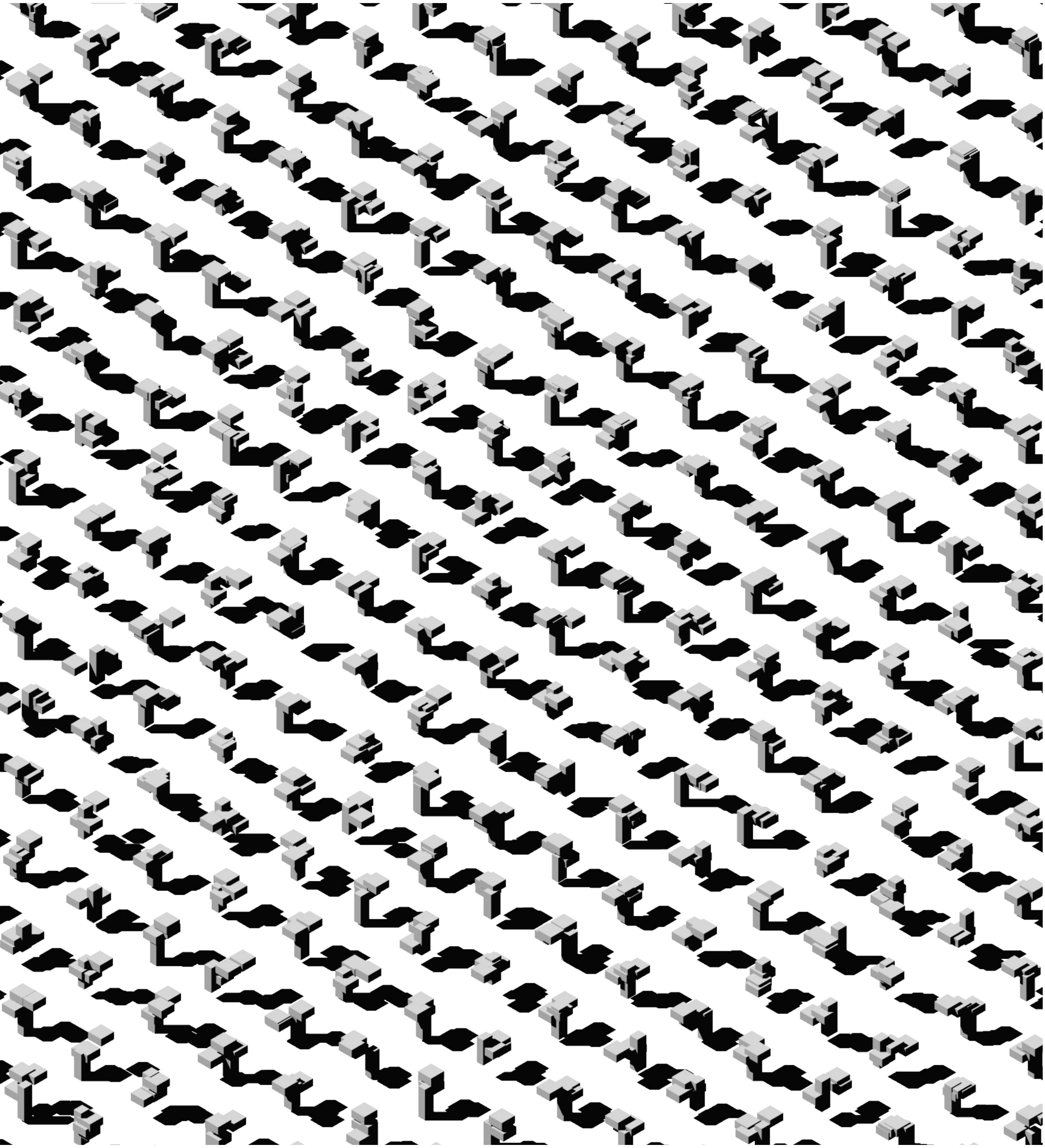


Fig 6.01. Aesthetic family one | compilation of variations extracted from the procedural system established n the concept phase, with the addition of a set of rules to refine the formal aesthetic of the system's outputs.*



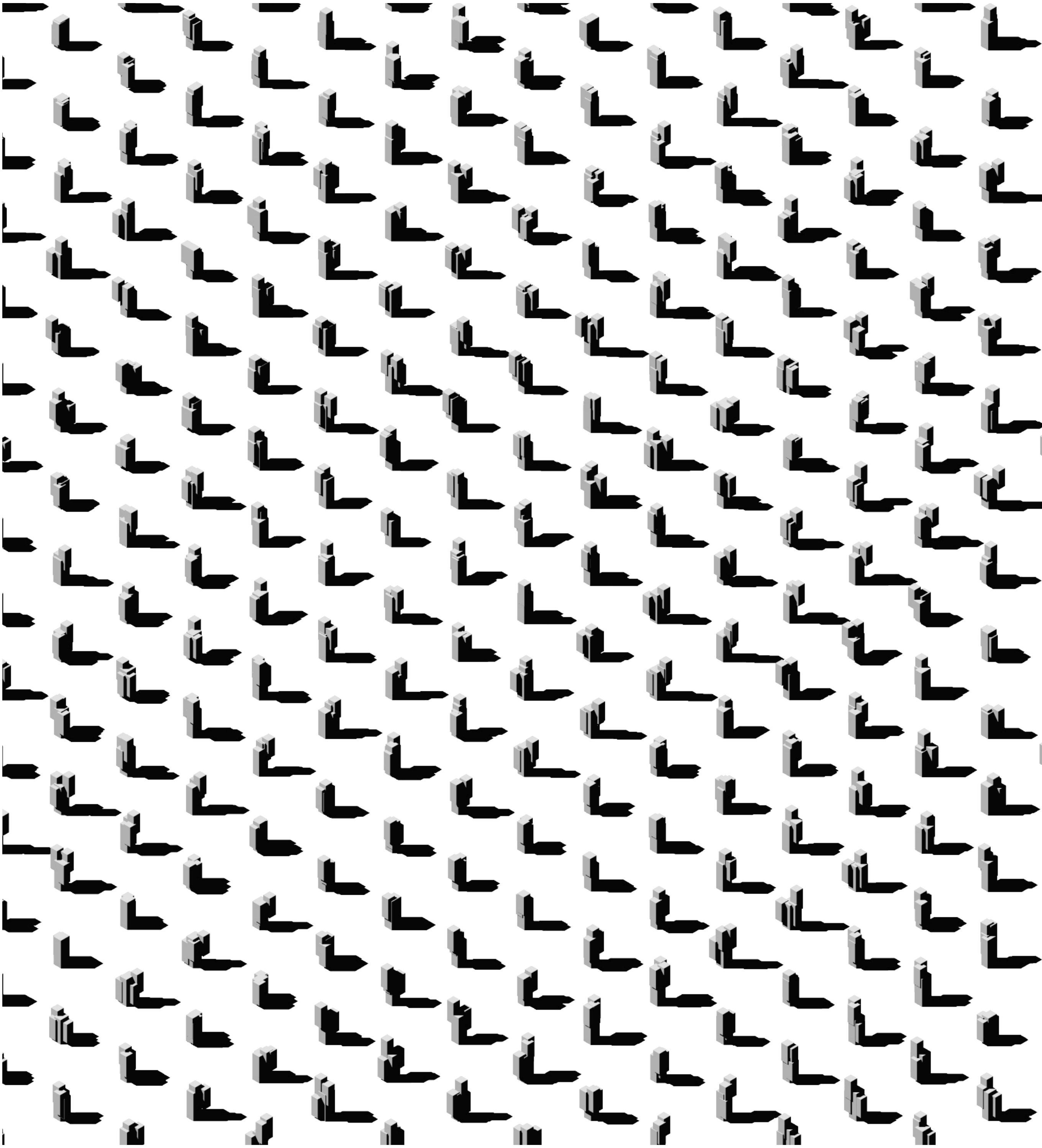
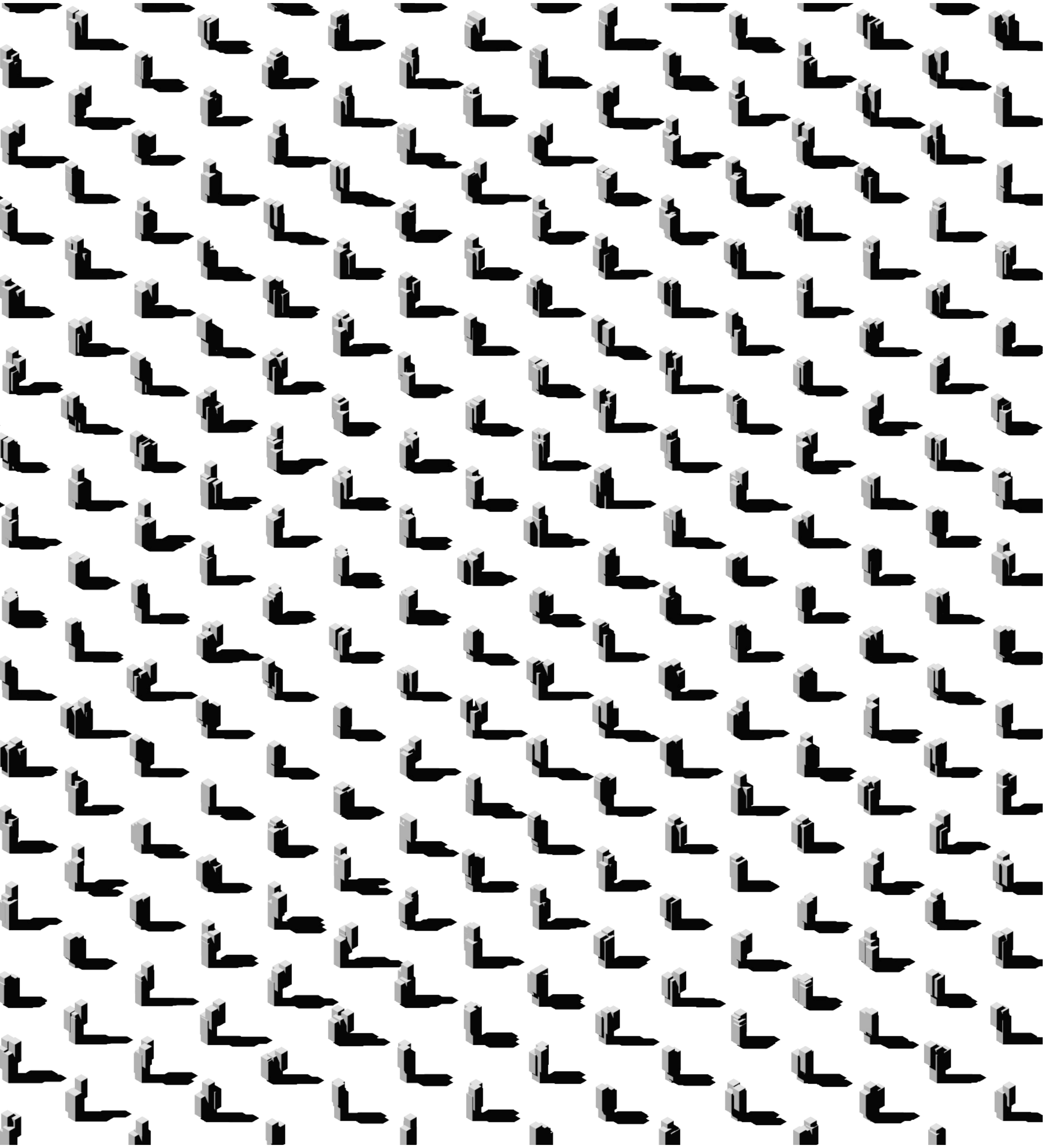


Fig 6.02. Aesthetic family two | compilation of variations extracted from the procedural system established n the concept phase, with the addition of a set of rules to refine the formal aesthetic of the system's outputs.*



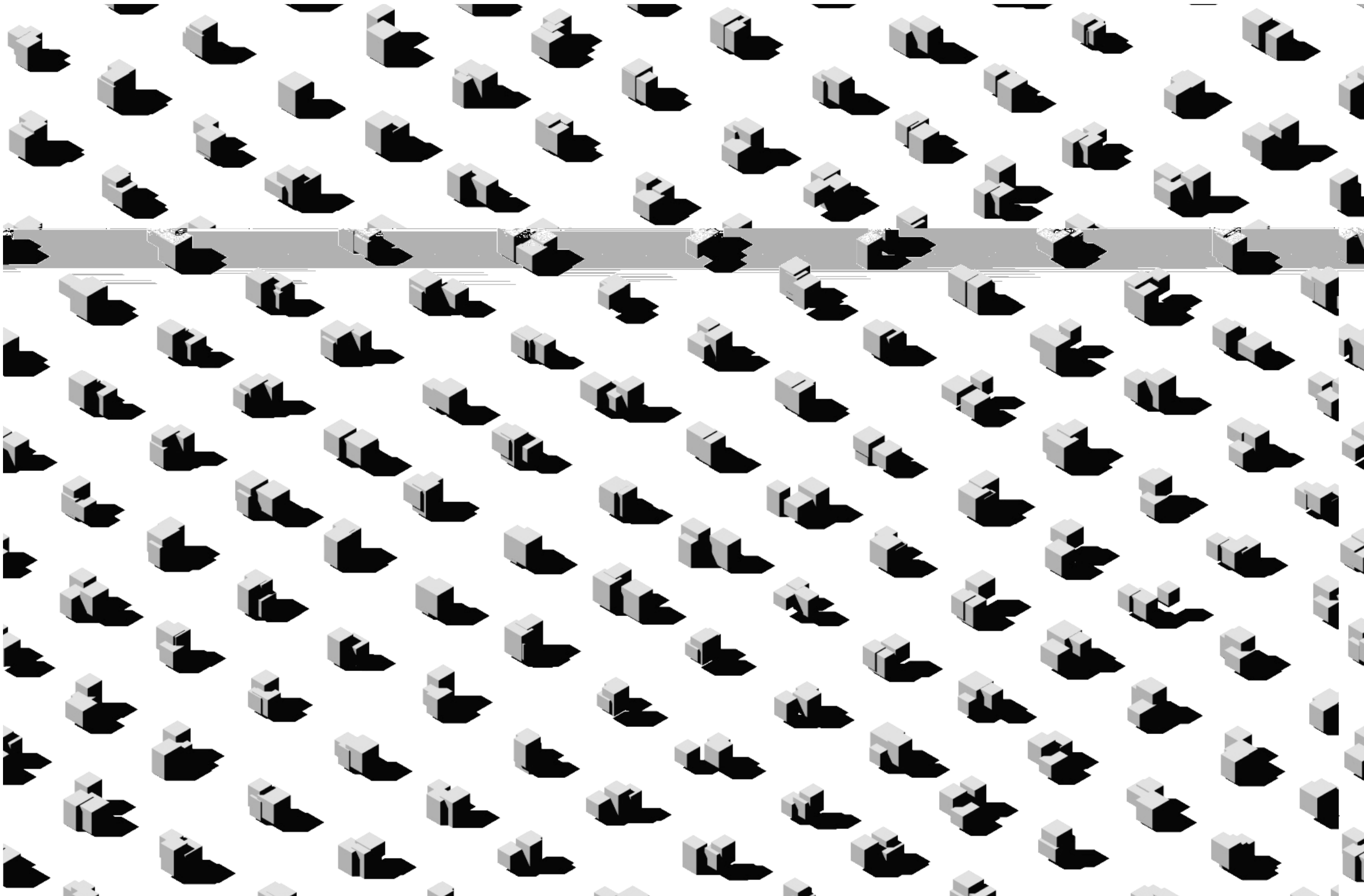
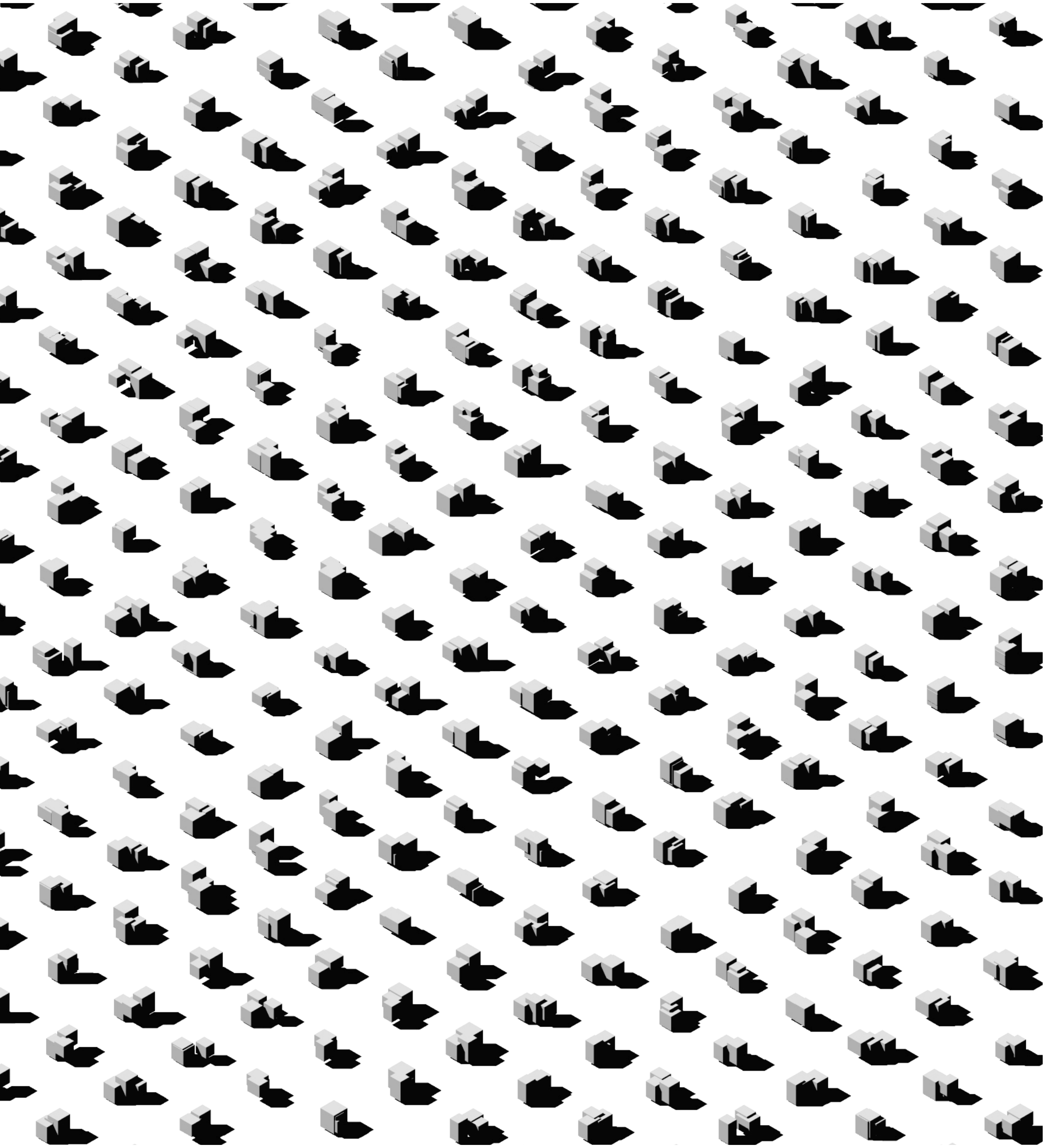


Fig 6.03. Aesthetic family three | compilation of variations extracted from the procedural system established n the concept phase, with the addition of a set of rules to refine the formal aesthetic of the system's outputs.*



BITS + PIECES

REFINEMENT

Preliminary Design | Recursive Design

After refining the formal properties of the procedural system, it was time to begin the development and implementation of the 'recursive' system mentioned earlier. This was explored by creating a procedural system that acted on its own - one that would be amalgamated with the existing system once it had been tested. When mentioning the ability to recursive design, this research refers to the act of the designer choosing an output form the system, isolating its components and their parameters, and re-running another procedural system on top of it. As will be apparent, this resulted in the possibility for the designer to pick an output, re-run the system to produce ten outputs that share very similar formal properties to the initial, and then repeat this process with a single choice form the ten to produce three tighter outputs. Confusing at this stage, the following images will bring clarity this process.

At this stage in this research it quickly became apparent that the design investigations shared an overarching purpose of 'refining' the procedural system established in the previous phase. The process of tightening the formal properties of the procedural system and the implementation of recursive design techniques resulted in the refinement of both the procedural code and an understanding for procedural techniques within Houdini. Both these investigations highlighted the flexible nature of proceduralism, illustrating that it is possible to tighten the massive production possibilities of a procedural system whilst still retaining some notions of ambiguity. With this, it became clear that the amalgamation of architecture and procedural techniques may be possible with further investigation.

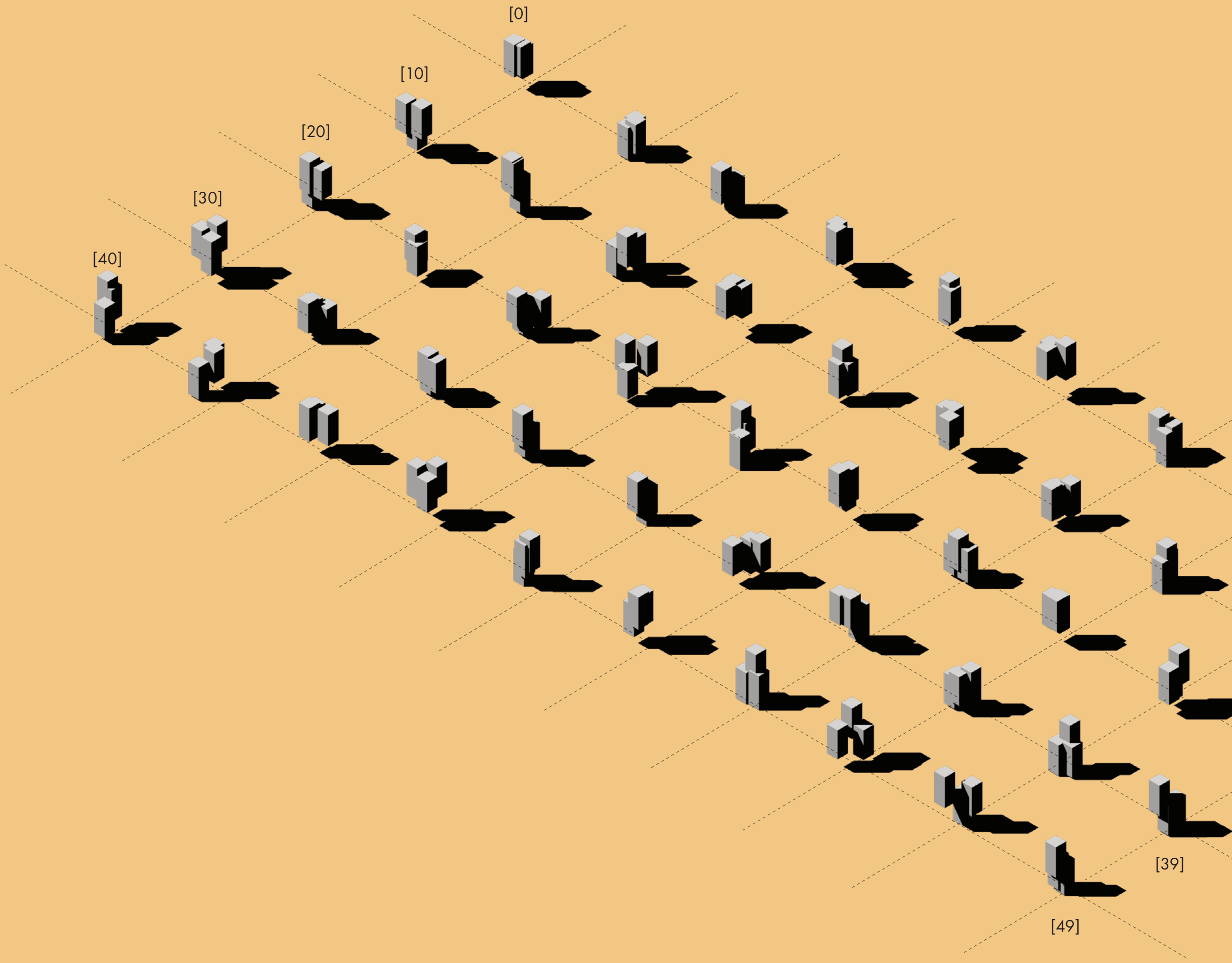
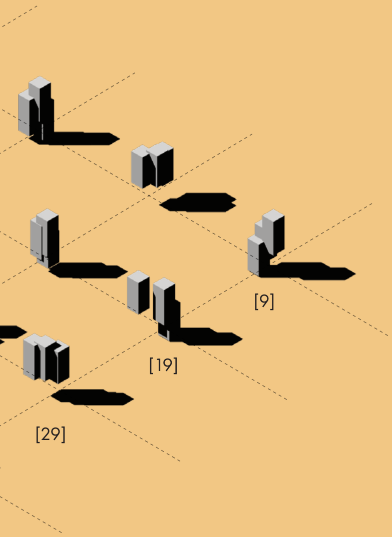
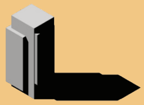


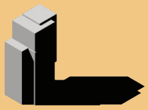
Fig 6.04. Recursive design stage one | establish a family of 50 variations and give user ability to select one.*


[9]
[19]
[29]

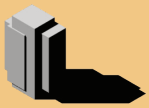
[0].[A]



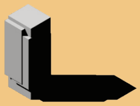
[0].[B]



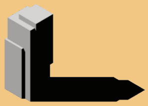
[0].[C]



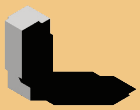
[0].[D]



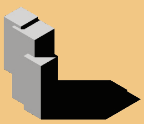
[0].[E]



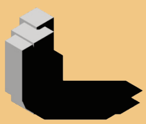
[0].[F]



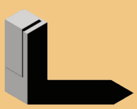
[0].[G]



[0].[H]



[0].[I]



[0].[J]

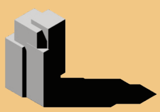
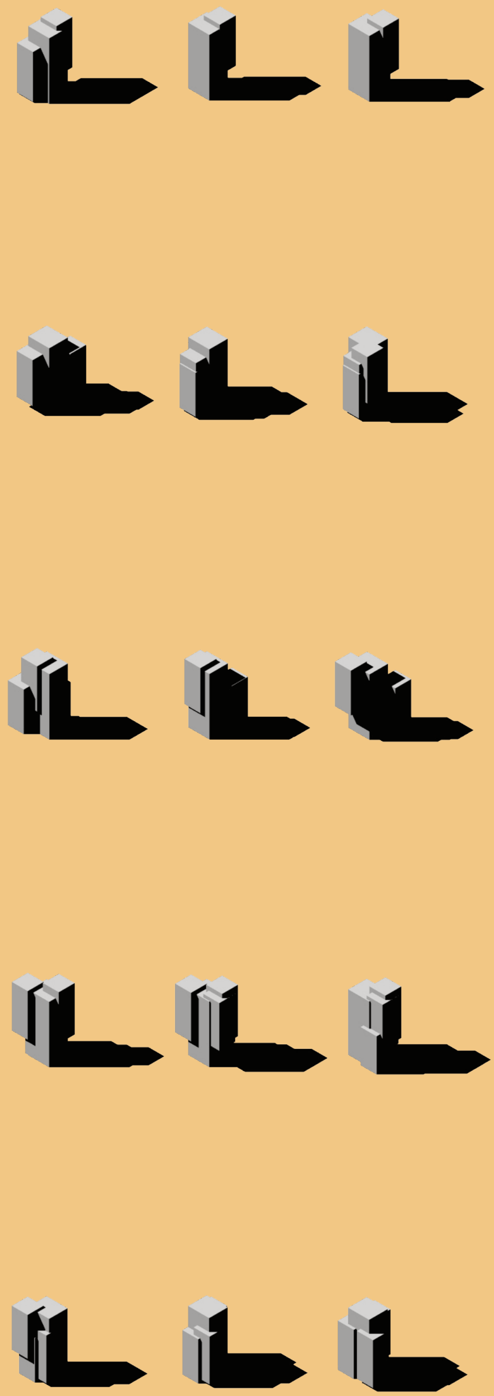


Fig 6.05. Recursive design stage two | from the intitial selection [0], procedurally generate 10 variations that share similar parameters to the initial selection.*



[0].[A] | [i] - [iii]

[0].[C] | [i] - [iii]

[0].[E] | [i] - [iii]

[0].[G] | [i] - [iii]

[0].[I] | [i] - [iii]

Fig 6.06. Recursive design stage three | from the second selection [0.A] etc, procedurally generate 3 variations that share very similar parameters to the initial selection.*

BITS + PIECES

REFINEMENT

Reflection

This phase of the research set out to refine the initial procedural system established in the concept phase. It aimed to explore notions of refinement and choice and further the understanding of procedural techniques within the Houdini software. It also aimed to further build upon understanding where procedural design may sit within the contemporary architectural workflow. It achieved these aims by further exploring procedural techniques in Houdini, adopting intermediate skills in order to develop the initial procedural system to include tighter controls on the aesthetic properties of its outputs and the ability to recursively design. While the procedural system is still considerably rudimentary at this stage in this research and is not influenced by architectural site, program or function, the investigations in this section illustrated the possibility of using built architectural influences as a driver in the design of the system itself. With this, it became apparent that it is possible to create a procedural system that is designed to respond to an architectural aesthetic whilst retaining the ambiguous nature of non-disciplinary procedural methods. However, moving toward to the final design investigation of this research the time has come to explore the implementation of architectural conditions. This was echoed during the design review for this design investigation, where Tim Lovell of Lovell and O'Connel Architects noted:

> *"I had a client come in once - he was a programmer - who asked why he couldn't generate his own building himself through code. It sparked an interesting discussion about 'experience' and 'spatial relationships' - terms closely linked with the Architect and the discipline. It would be interesting to see how someone involved in both visual programming and a design-centric discipline could challenge this."* – Tim Lovell

Lovell's observation highlighted the importance of transitioning the current design investigations into developed design, taking into consideration the amalgamation of architecture and procedural design techniques. Up until now, it was evident that the use of procedural techniques in the generation of architectural massing would often result in random formations that did not exhibit 'spatial relationships' often be produced in built architecture. This was too a result of architectural program and function being left out form previous investigations. Moving forward, the following chapter in this body of research will introduce the architectural conditions of site and program, investigating how they can be implemented into the code of a procedural system. It will identify a methodology of translating the real-world properties of site and program into an automated digital process and will established a method in which procedural generation can be amalgamated within the architectural workflow during early stage design.

BITS + PIECES

APPLICATION

Developed Design

BITS + PIECES

APPLICATION

Introduction

The previous two chapters of this research explored the use of procedural techniques within Houdini, developing an understanding for the software and its capabilities. As a result, they introduced a conceptual procedural system and saw it refined to accommodate for tighter controls over its outputs and the ability for the designer to give preference over certain outcomes. These two investigations illustrated the possibilities of using a procedural system as a massing tool, without the influence of architectural site, program or function.

This chapter contains the final 'research led design' investigation for this body of research. It adopts and builds upon the investigations presented in the previous chapters, exploring how the conditions of site and program can be integrated into an existing procedural system. The intention for this chapter is to establish an 'emergent resource' that can be used for architectural massing in early stage design It subsequently evaluates the plausibility of the resource within an industry standard software, Revit. Unlike before, the design explorations in this chapter are influenced by architectural site and program, allowing the constraints of the physical world to influence the outcomes of the procedural system and thus tightening the gap between architectural and procedural design.

Aim:

As the final 'research led design' experiment for this body of research, the following investigation aims to define how procedural systems can add to the architectural workflow. It develops an understanding for advanced procedural techniques within the Houdini software and investigates how they can be amalgamated with architectural site and program to produce an 'emergent resource'.

Fig 7.01. Aerial map of Mt. Crawford, illustrating area of land that was used integrated into procedural system (dashed circle) to evaluate site for slope data.*

BITS + PIECES

APPLICATION

Procedural Site

In order to integrate site conditions into a procedural system, it was important to first consider the limitations of the Houdini software, the constraints of time and the scope of knowledge in regards to procedural techniques. To avoid complication but still produce a procedural system that in some way mimicked an architectural analysis of site, it was decided that the system itself would be developed to evaluate the slope of a site. By doing so, it became possible to use that data to inform the transformations that take place in the already established procedural system, allowing its parameters to respond to site in a formal way. This decision required developing the procedural system to understand site in regards to its topographic conditions, which in this case was decided to be three extremes; 0-19°, 20-34° and 35°+. Once these conditions were coded, a site in Wellington City was chosen and ran through the system. The site selected was the eastern side of Mt. Crawford, looking out towards Kau Bay. It is important to note that the site was not chosen to be analysed for its historical importance or location to local amenities, but was solely analysed in regards to its topographic conditions to extract data for the procedural system. The sets of data extracted from the conditions on site were to be used as parameters for the design of the procedural system and not the design of an individual building.

Fig 7.02. Vector map illustrating sections of site injected into procedural system designed in Houdini. The topographic properties of these sections were analysed for slope.*

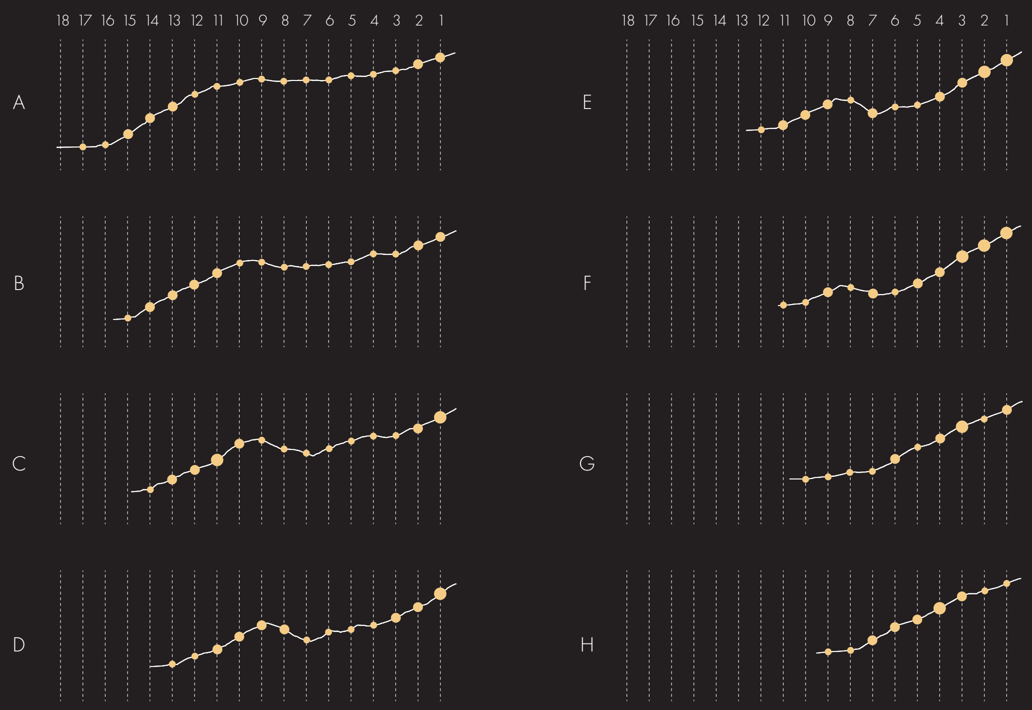
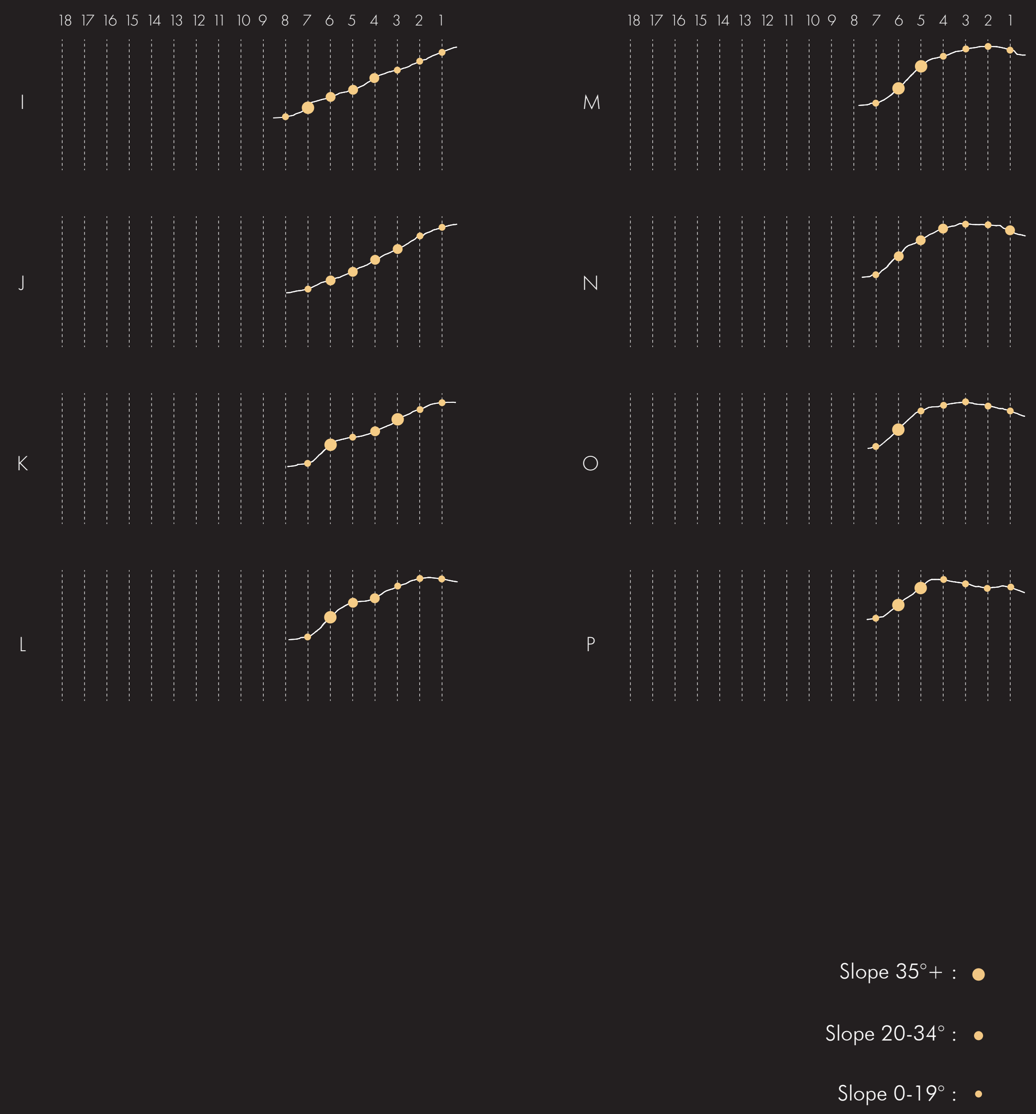


Fig 7.03. Series of data points extracted from analysis of slope in Houdini. These data points were used to inform formal transformations of components on a vertical axis.*


18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
I
J
K
L
18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
M
N
O
P
Slope 35°+ :
Slope 20-34° :
Slope 0-19° :

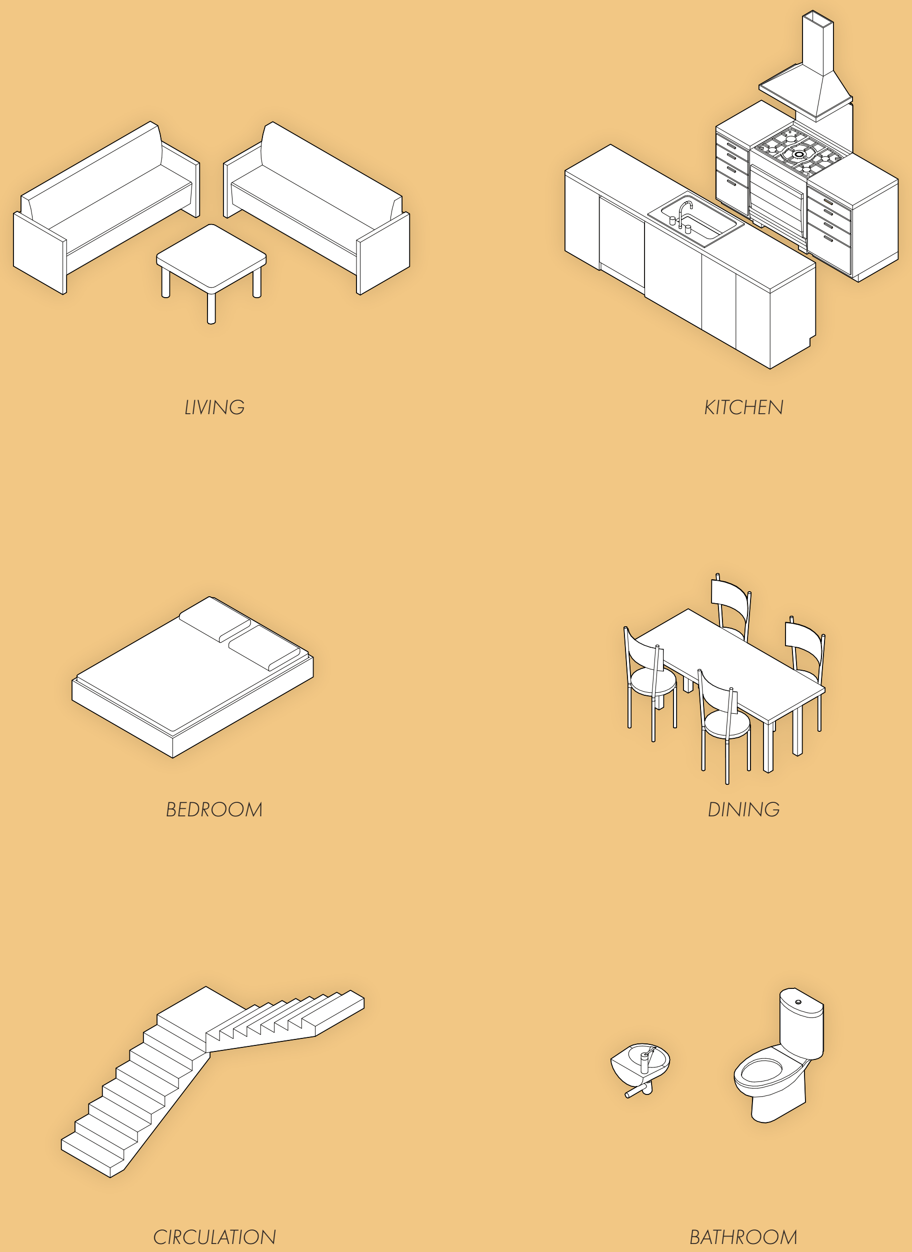


Fig 7.04. Diagrams illustrating choice of programmatic functions to be integrated into procedural system.*

BITS + PIECES

APPLICATION

Procedural Program

Once the procedural system had the ability to analyse the slope of a site and use it to influence the formal properties of its outputs, it came time to integrate architectural program. The choice of implementing an architectural program was two-fold; it was the critical disconnection between architectural and procedural practices in the Houdini Lake House case study; and thus it is vital to creating a link between the two disciplines. The first decision made was the choice of which program to use as a core variable. Although program was not a key factor for this research in the beginning, it was important to consider one that would benefit from the ambiguous nature of the procedural system - one that did not rely on rigorous spatial programming during early stage design. Short-stay accommodation, or airbnb, was selected as the architectural function to be implemented into the procedural system, due to the 'uniqueness' present in the airbnb experience. Airbnb by nature is often observed to be about the experience of the architectural space, rather than the overall function of it, serving as a perfect choice for use within this investigation. Thus, the functional components for the system were evaluated to be; living, kitchen, dining, bedroom, bathroom and circulation. It is essential to understand that these functions were not purposed to produce an individual building, but instead were transferred to code that was integrated into the design of the procedural system. The intention here was to investigate a method in which spatial programming could be translated into procedural coding to be used in an architectural massing tool.

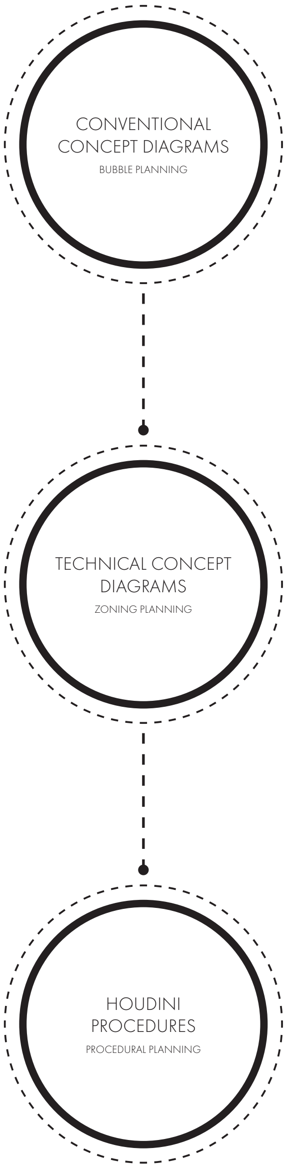
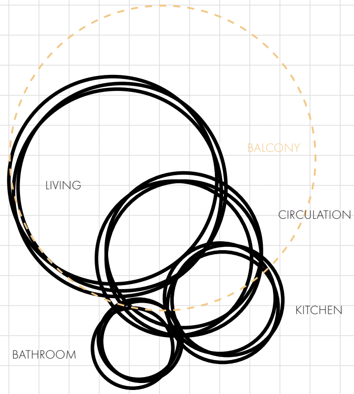
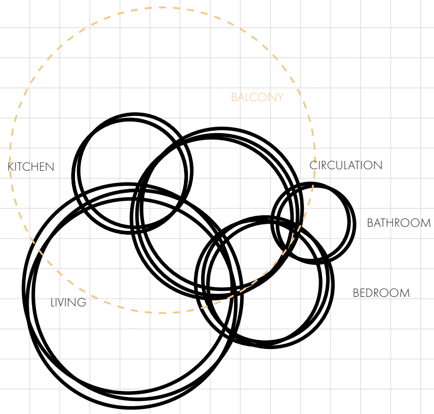


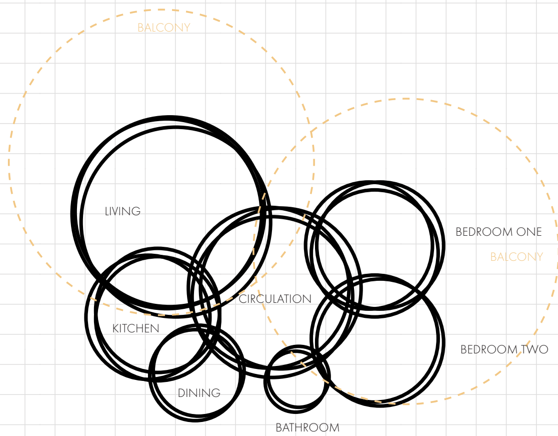
Fig 7.05. Process used for translating architectural function in procedural code.*



$45m^2$ SPATIAL ARRANGEMENT

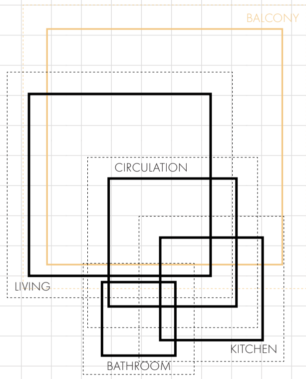


$90m^2$ SPATIAL ARRANGEMENT

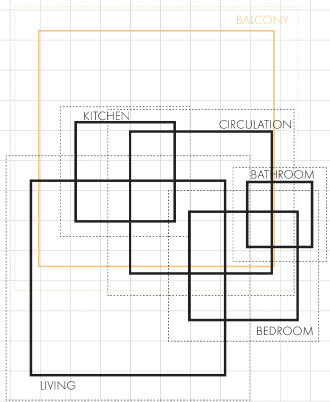


$120m^2$ SPATIAL ARRANGEMENT

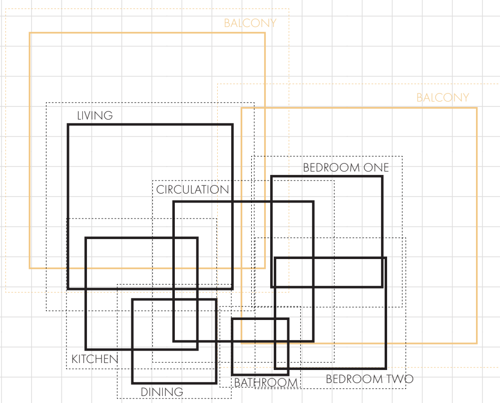
Fig 7.06. Diagram illustrating how function was first implemented into Houdini, mimicing conventional bubble diagrams.*



45m^2 SPATIAL ARRANGEMENT



90m^2 SPATIAL ARRANGEMENT



120m^2 SPATIAL ARRANGEMENT

Fig 7.07. Diagram illustrating how conventional bubble diagrams were translated into a language the procedural system could understand.*

BITS + PIECES

APPLICATION

Developed Design | Putting it all Together

At this stage in the research, it was important to test whether the integration of site and program would result in an 'emergent resource' that resembled a form of architectural aesthetic. The final design investigation began by undertaking a process similar to the preliminary design, where three design investigations took place, testing the procedural system against three different 'typologies'. This was achieved by redesigning the system to include a parameter for 'total square meters', allowing for a variable to be chosen that would affect the overall size of the outputs. For this investigation, 45m2, 90m2 and 120m2 were used as variables. Once these variables were set up, the procedural systems from all three design chapters were integrated into one, forming a system that considered all the investigations in this research. This resulted in a system that allowed for tighter controls over formal properties, recursive preference, architectural site, architectural program and the ability to give preference over the size of outputs.

Fig 7.08. Family one, 45m^2 spatial arrangements | Series of variations extracted from procedural system designed for a 45m^2 architectural mass.*

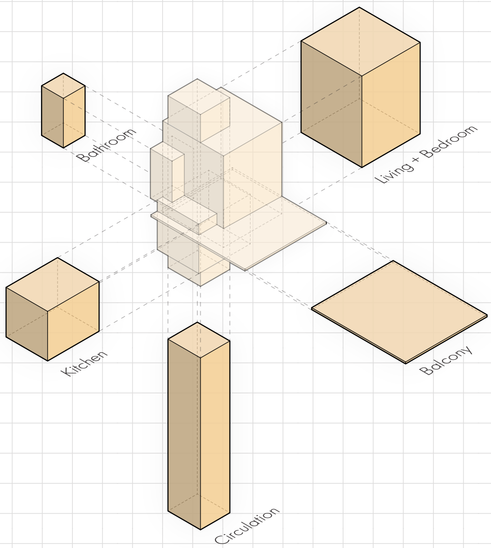


Fig 7.09. Diagrams illustrating functional components of a single architectural mass extracted from the 45m² procedural system.*

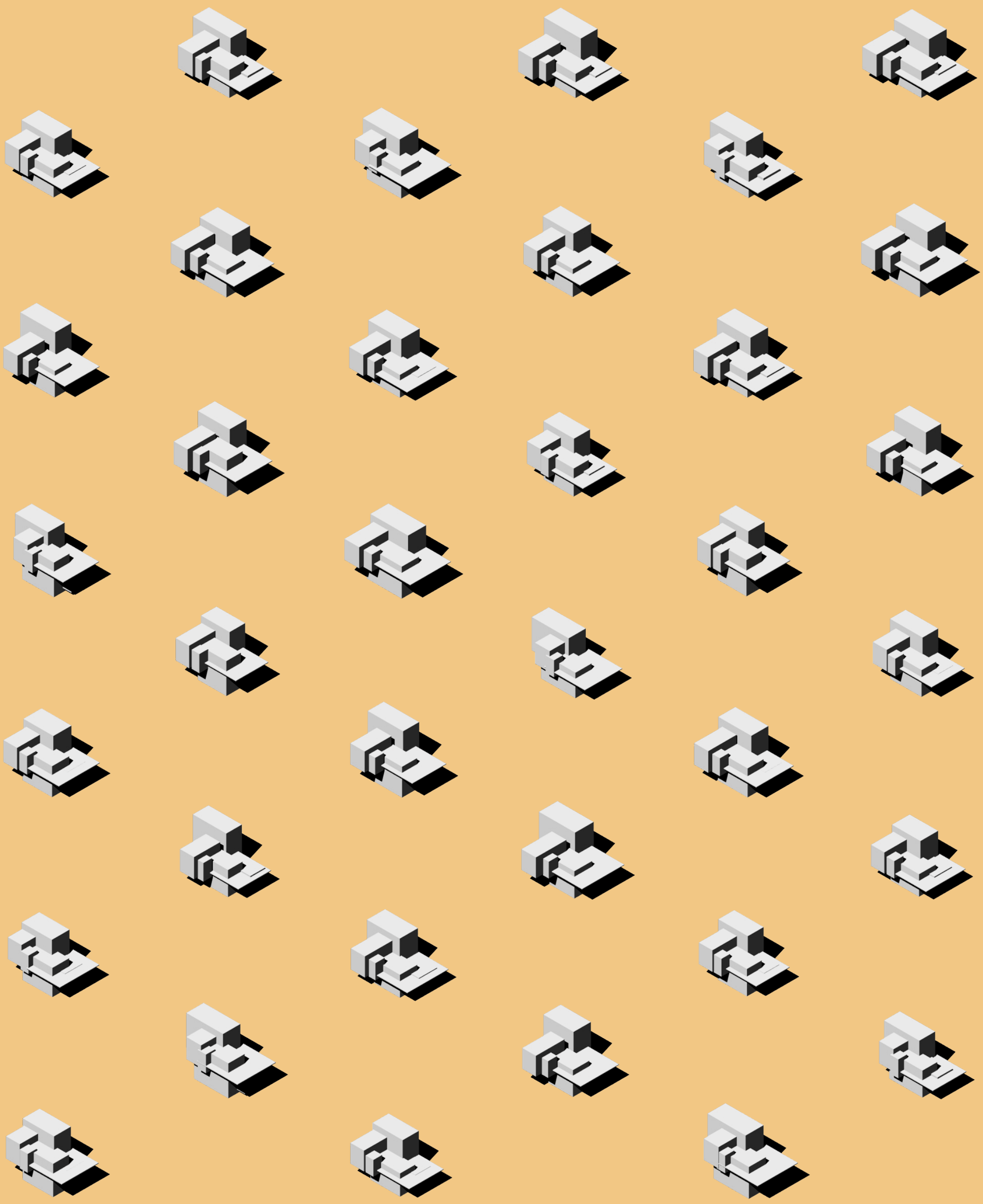


Fig 7.10. Family one, 90m^2 spatial arrangements | Series of variations extracted from procedural system designed for a 90m^2 architectural mass.*

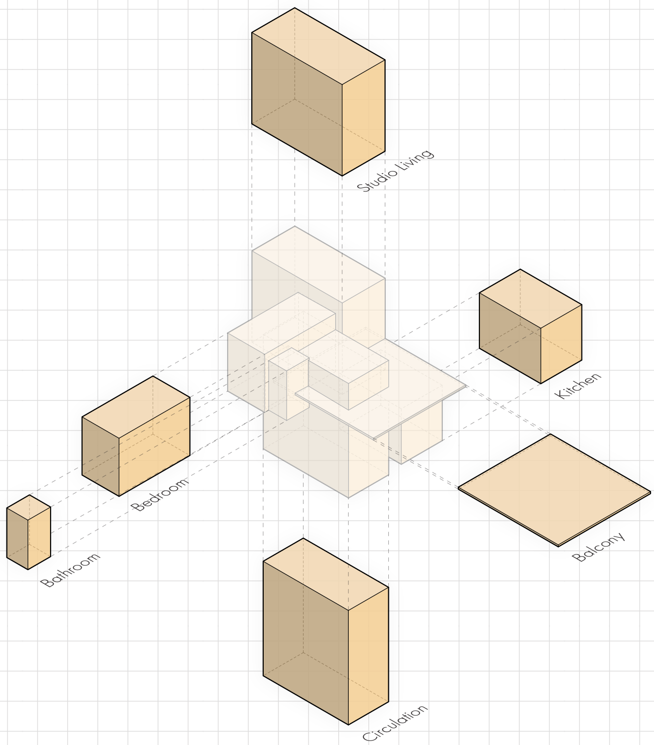


Fig 7.11. Diagrams illustrating functional components of a single architectural mass extracted from the 90m^2 procedural system.*

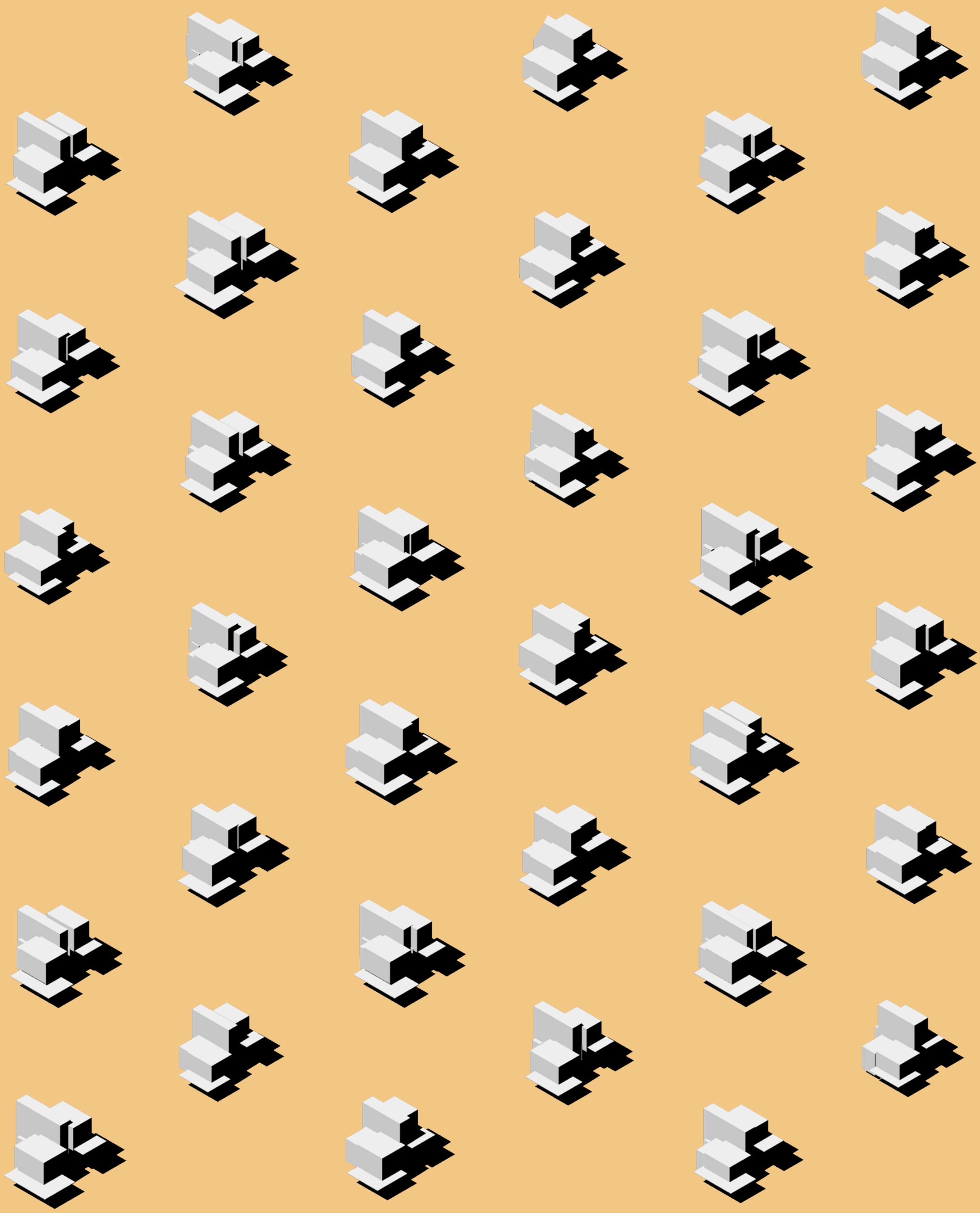


Fig 7.12. Family one, 120m^2 spatial arrangements | Series of variations extracted from procedural system designed for a 120m^2 architectural mass.*

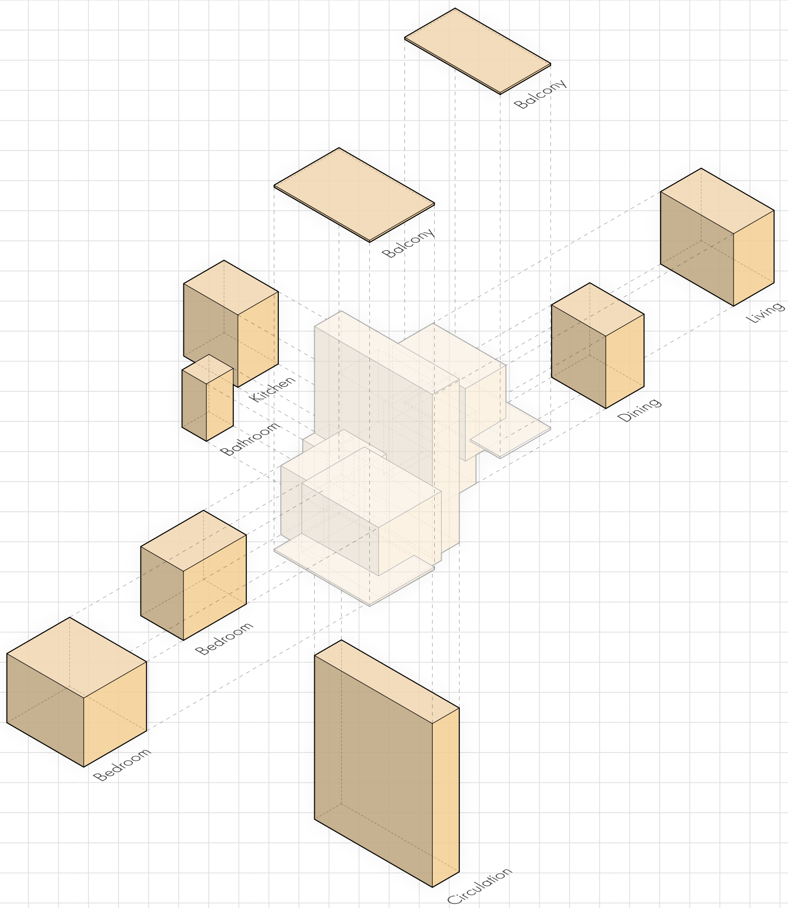


Fig 7.13. Diagrams illustrating functional components of a single architectural mass extracted from the 120m² procedural system.*

BITS + PIECES

APPLICATION

Developed Design | Evaluating

Upon developing a final procedural system capable of producing an 'emergent resource' that exhibited formal properties of 'architecture', this research identified that it would be worth briefly investigating whether or not the resource could be used within an industry standard software such as Revit. By doing so, it allowed for the evaluation of the site and program systems established earlier. It provided a point of reference to determine whether or not an architectural mass was being produced that required minimal change. As a sort of 'mini-experiment', a mass from the system was chosen at random and imported into the Revit software (at this stage in the research, any mass could have been picked). Architectural conditions were then tested, evaluating whether circulation and placement of program were efficient within the original mass.

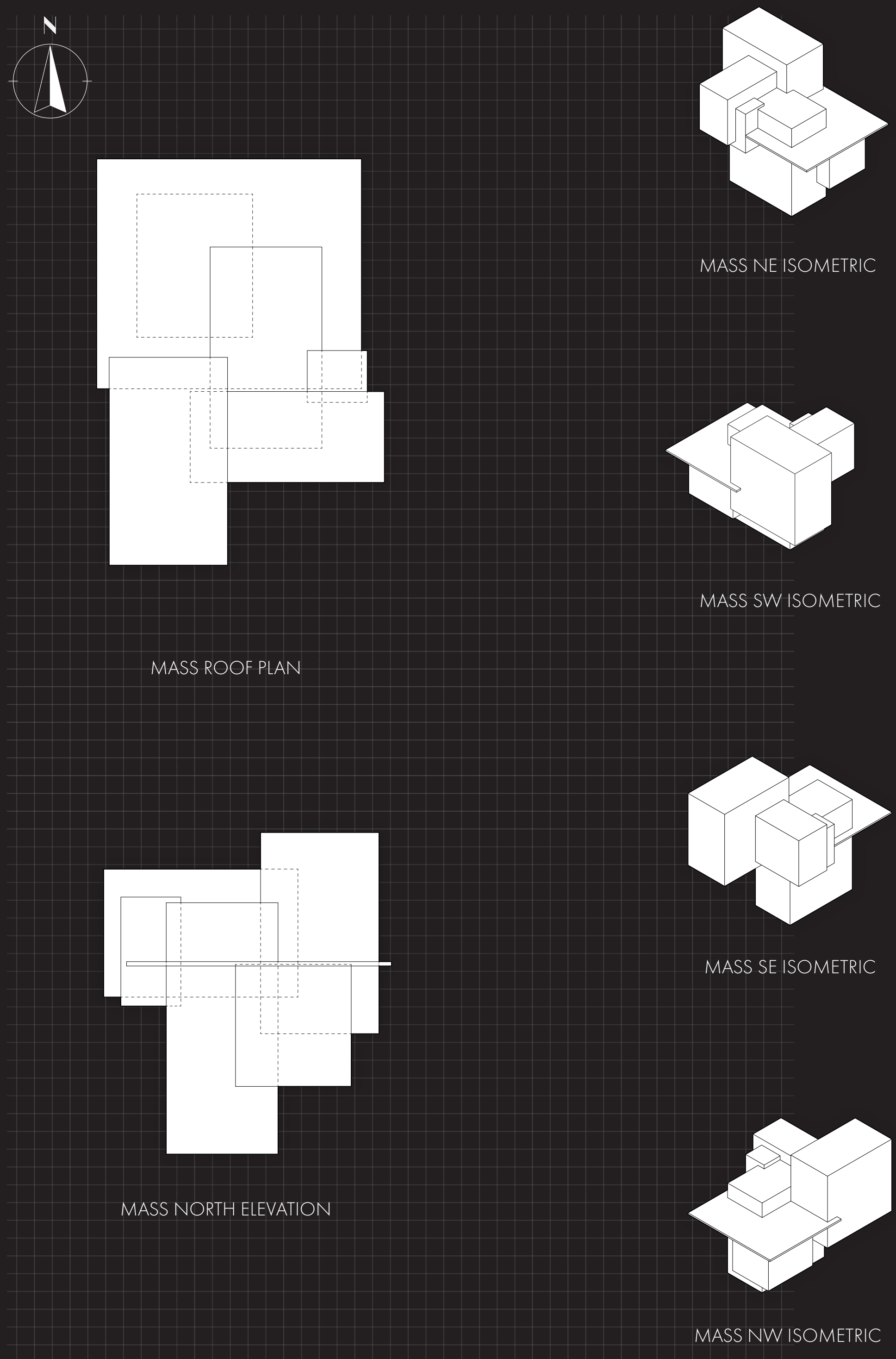


Fig 7.14. Architectural diagrams illustrating a single archtiectural mass extracted from 45m² procedural system.*

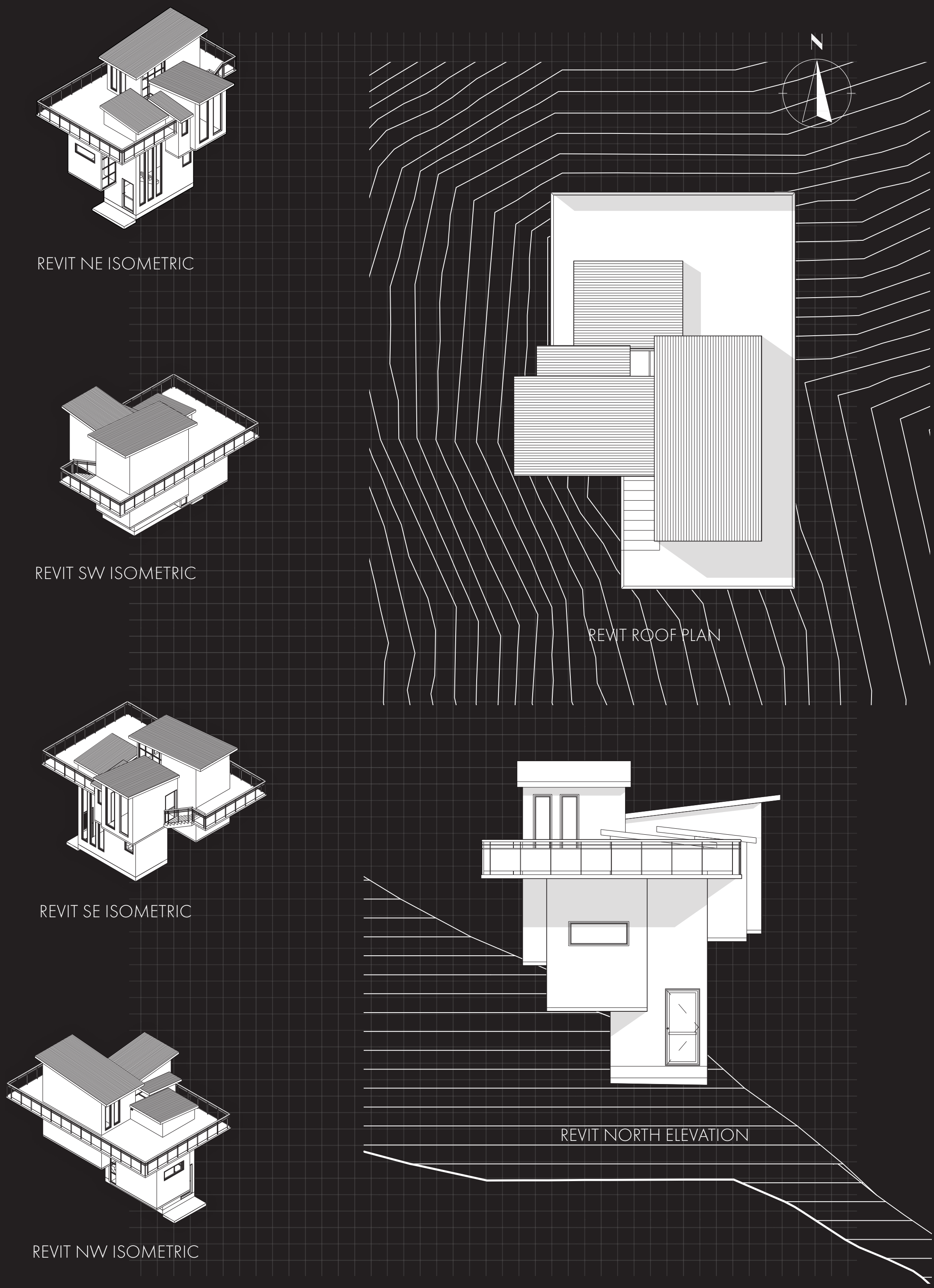


Fig 7.15. Architectural diagrams illustrating a single architectural mass worked up in Autodesk Revit using windows, walls, roofs, ballustrades, doors and floors.*

BITS + PIECES

APPLICATION

Reflection

This final phase of the research set out to adopt and build upon investigations presented in previous chapters, exploring how the architectural conditions of site and program can be integrated into an existing procedural system. It intended to explore an 'emergent resource' that could be used for architectural massing in early stage design. It established a method for integrating the conditions of site and program into a procedural workflow and resulted in the design of a procedural system that could output refined architectural massing. The design investigations in this chapter moved beyond the rudimentary findings of the previous, allowing real-world constraints to influence the parameters of the system, much like they influence the physical parameters of the built environment. The investigation present in this chapter illustrates how, through a consideration for multiple parameters and variables, that procedural techniques offer a unique path towards the production of built architecture. With more time and thought, it would have been interesting to further understand how architectural detail could be added to the procedural system - much like the detail present in Anastasia Opara's lake houses case study. However, the comparison of this research and Opara's work can also be seen as two extremes; Opara's focus was on detailed architectural models for game design where inhabitation and function was irrelevant, but this research explores the opposite. This research investigated how procedural techniques could be used within a real-world architectural workflow, where function and site are important conditions for the 'emergent resource' to exhibit. This is what has been achieved here - a resource for early stage design; an architectural massing tool that the designer can pick up, adopt, and learn from to explore the ideation of built form through the en masse creation of content.

CONCLUSION

Discussion, Conclusions and Critical Reflection

BITS + PIECES

CONCLUSION

Discussion, Conclusions and Critical Reflection

The architectural discipline is constantly experiencing change to the way in which its practitioners operate. Architecture, once a discipline of pencil and paper, now adopts and develops a myriad of techniques from various creative partners, giving its professionals an opportunity to adopt and evolve novel methods towards the production and understanding of built form. As a result of this, this thesis sought out to investigate one of these methods. It asked the question: 'how can the application of procedural generation design techniques augment the ideation of architectural massing for early stage design?'. By doing so, it explored how the technique of procedural generation may become amalgamated with the ideation of architecture, specifically during the conceptual design process. It undertook a series of design investigations, exploring how procedural techniques and the design of a procedural system can be developed into a tool for the creation of architectural massing.

The procedural design experiments presented in this research encompass a detailed investigation into what the technique of procedural generation has to offer the architectural discipline. With the first aim of the research being to identify how procedural techniques can be used in the process of ideating architecture, it quickly became apparent that the technique capable of being used as an ideation tool. During the concept phase, it became clear that the pseudo-random nature of procedural design reflects in the fact that its 'emergent resource' will always embody a sense of ambiguity. It is never perfect, let alone entirely refined. The technique evokes randomness and thus does not reflect the traditional architectural design process. However, it does offer a new method for the production of built form. It allows for ideation to occur through a reflection of ambiguity, giving the designer the ability to give preference to certain options over others. Tied in with a traditional workflow, procedural generation offers the discipline a new tool for the ideation of space, echoed by Patrick Janssen stating that procedural design is an 'environment of new possibility' (Jannsen, 2001).

However, with this new environment of ambiguity influencing ideation, it became evident in the design investigations that the process of adopting procedural methods was not an easy road. Procedural design is often coded by professionals with years of experience in information technologies, rather than the profession of architecture. There is a tremendous learning curve involved in developing knowledge for procedural generation; one that would likely require time to upskill outside of the professional environment. Software and tools such as Houdini make this transition easier for creative designers, giving them the ability to translate code into a visual networking interface, but that path still requires significant time investment. It is also important to consider that "despite all the possibilities that digital tools provide, the architect always has to scale, trim, or orient the object to accommodate and situate it in within the real physical realm" (Maguid, 2016). The question arises here whether or not procedural generation is a tool worth adopting, where instead the Architect could simply return to pencil and paper to competently ideate building concepts. Due to procedural generation being severely limited by the imagination and time investment of the designer. If the designer is preoccupied with other tasks, it is possible that the capabilities of a procedural system would not out weight the benefits of retaining a traditional workflow.

> *"Architecture is not meant to be free from constraints, it is meant to adapt to them, reason with them, and respond to them." – Maguid 2016*

Through investigating procedural systems and how they situate themselves in relation to the architectural workflow, it became apparent that the technique of procedural generation is not one that benefits the built environment in its entirety. It is safe to assume that architecture could never be fully automated, as the architect is vital to the production of the built environment (Davis, 2009). It may be possible that one day we see portions of the architectural process automated by computers, but it is certain that the architect will be the driving force behind that automation. The investigations present in this research illustrate that, although attempting to automate a phase in the architectural workflow, the architect (or creative designer) becomes the designer of the procedural system. The discipline's professionals have undergone studies that no programmer has - we understand spatial relationships, experience and function like no other. This is what the Architect brings to the table in the production of built form, but it can also be what they bring to the table in the production of a procedural system. What would happen if the Architect designed a procedural system, providing them with an emergent resource capable of seeing a project through to practical completion? This is where future investigations take place in this avenue of research. Although this body of research has positioned itself within early stage design, the opportunity arises for future work to investigate how procedural generation can aid the Architect in other areas of their workflow. Maybe there is nothing there and procedural generation should be contained to its origin; game and film. However, the possibility is also there that one day, the architectural discipline automates a significant portion of its workflow with an en masse content creator; one that is driven not by a programmer, but by a programming Architect who learns to adapt and respond to this new methodology for ideating architecture. In this future practice, the 'programming Architect' maintains all the skills and sensibilities of his or her training, but becomes equipped with this new tool set, enabled by a far more rigorous and reflective design practice; one that engages with ambiguity as a design-driver during the infancy of the architectural design process.

REFERENCES

BITS + PIECES

Alexander, C. (1979). The Timeless Way of Building. New York: Oxford University Press.

Alexander, C., Ishikawa, S., & Sukverstein, M. (1977). A Pattern Language : towns, buildings, construction. New York: Oxford University Press.

Baker, C. (2016). 'No Man's Sky': How Games Are Building Themselves: Rolling Stone. Retrieved from http://www.rollingstone.com/culture/news/no-mans-sky-how-games-Are-building-themselves-w433492

Bazalo, F., & Moleta, T. J. (2015). Responsive Algorithms. Paper presented at the Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA 2015), Daegu. Retrieved from http://papers.cumincad.org/cgi-bin/works/Show?caadria2015_237

CityEngineTV (2016). Esri CityEngine 2016 Highlights [Video file]. Retrieved from https://www.youtube.com/watch?v=mD0yml8LbPY

Davis, D. (2010). Untangling Grasshopper – Part 1: Design patterns. Retrieved from http://www.danieldavis.com/untangling-Grasshopper-part-1-design-patterns/

Davis, D. (2013). Modelled on Software Engineering: Flexible Parametric Models in the Practice of Architecture. (Doctor of Philosophy), RMIT University. Retrieved from http://www.danieldavis.com/papers/danieldavis_thesis.pdf

Esri. (2017). Esri CityEngine - 3D Modeling Software for Urban Environments. Retrieved from http://www.esri.com/software/cityengine

Fischer, T. (2017). From "Failure is not an Option" to "Try Again. Fail Again. Fail Better". Lecture, Victoria University of Wellington, New Zealand.

Hello Games (2016). About Us: Hello Games. Retrieved from http://www.hellogames.org/about-us/

Hussain, T. (2016). No Man's Sky's 18 Quintillion Planets Take Up Just 6 GB on Disc. Retrieved from https://www.gamespot.com/articles/no-mans-skys-18-quintillion-planets-take-up-just-6/1100-6441663/

Janssen, P. (2012). Staff CV - Patrick Janssen. Retrieved 29 March, 2017, from https://www.arch.nus.edu.sg/people/cv/pj.htm

Janssen, P., Frazer, J., & Tang, M. (2001). Generating-predicting soup: A conceptual framework for a design environment. Paper presented at the CAADRIA 2001, Sydney.

Johnson, M. (2016). Generation Next, Part 4: Procedural Generation's Future. Retrieved from https://www.rockpapershotgun.com/2016/09/16/the-future-of-procedural-generation/

Kilkelly, M. (2016). 5 Ways Computational Design Will Change the Way You Work: ArchDaily. Retrieved from http://www.archdaily.com/785602/5-ways-computational-design-will-change-the-way-you-work

Kluyskens, T. (2016, September 29). September Meetup: Tristan Bunn talking vectors + a Houdini introduction. Lecture presented at Victoria University of Wellington. Wellington

Lynn, G. (2016). Origins of the Digital - Embryological House. Cca.qc.ca. Retrieved 24 May 2017, from https://www.cca.qc.ca/en/issues/4/origins-of-the-digital/5/embryological-house

Maguid, Y. (2016). What can architects learn from No Man's Sky? - Kill Screen. Kill Screen. Retrieved 25 April 2017, from https://killscreen.com/articles/can-architects-learn-no-mans-sky/

Mamo, N. (2017). The Feature Lens: Procedural Generation. Retrieved from https://www.eyeforgames.com/TR-efgblogs/the-feature-lens-procedural-generation/

Manaugh, G. (2009). Procedural Destruction and the Algorithmic Fiction of the City. Retrieved March 10th, 2017, from http://www.bldgblog.com/2009/08/procedural-destruction-and-the-algorithmic-fiction-of-the-city/

Manaugh, G. (2014). Procedural Brutalism. Retrieved April 1st, 2017, rom http://www.bldgblog.com/2014/09/procedural-brutalism/

Manaugh, G. (2016). Sometimes the house you come out of isn't the same one you went into. Retrieved March 18th, 2017, from http://www.bldgblog.com/2016/03/sometimes-the-house-you-come-out-of-isnt-the-same-one-you-went-into/

Miller, N. (2016). Should architects learn to code?: Building Design + Construction Network. Retrieved from https://www.bdcnetwork.com/blog/should-architects-learn-code

Muller, P., Wonka, P., Haegler, S., Ulmer, A., & Gool, L. V. (2006). Procedural Modeling of Buildings. Paper presented at the SIGGRAPH '06, Boston.

No Man's Sky. (2016). No Man's Sky Gallery. [online] Available at: https://www.nomanssky.com/press/#images [Accessed 24 Mar. 2017].

Opara, A. (2016). Realistic Procedural Architecture for Games. Retrieved March 20th, 2017, from https://80.lv/articles/realistic-procedural-architecture-for-games/

Opara, A. (2017). Houdini Procedural Lake Houses. [online] Gumroad. Available at: https://gumroad.com/l/qaEZ [Accessed 25 Mar. 2017].

REFERENCES

BITS + PIECES

Rutten, D. (2012). What are random seed values? Retrieved from http://www.Grasshopper3d.com/forum/topics/what-are-random-seed-values

Smelik, R., Tutenel, T., Bidarra, R., & Benes, B. (2014). A Survey on Procedural Modelling for Virtual World. Computer Graphics Forum, 33(6), 31-50. doi: 10.1111/cgf.122/6

Stålberg, O. (n.d.). Brick Block. [online] Oskarstalberg.com. Available at: http://oskarstalberg.com/game/house/index.html [Accessed 11 Mar. 2017].

Stiny, G. (2008). Shape: Talking about Seeing and Doing: The MIT Press.

Welch, C., Moleta, T. J., & Moloney, J. (2014). Selective Interference: Emergent complexity informed by programmatic, social and performative criteria. Paper presented at the ACADIA 14: Design Agency, Los Angeles. Retrieved from http://papers.cumincad.org/data/works/att/acadia14_719.content.pdf

FIGURES

BITS + PIECES

Sources of Figures

*All figures marked with a * were graded in the 75% review.*

Fig. x.01*	*Image by Author*
Fig. x.02	*Image by Author*
Fig. 1.01*	*Image by Author*
Fig. 1.02*	*Image by Author*
Fig. 1.03*	*Image by Author*
Fig. 2.01	*No Man's Sky. (2016). No Man's Sky Gallery. [online] Available at: https://www.nomanssky.com/press/#images [Accessed 24 Mar. 2017].*
Fig. 2.02*	*Image by Author*
Fig. 2.03*	*Image by Author*
Fig. 4.01	*Opara, A. (2017). Houdini Procedural Lake Houses. [online] Gumroad. Available at: https://gumroad.com/l/qaEZ [Accessed 25 Mar. 2017].*
Fig. 4.02	*Stålberg, O. (n.d.). Brick Block. [online] Oskarstalberg.com. Available at: http://oskarstalberg.com/game/house/index.html [Accessed 11 Mar. 2017].*
Fig. 5.01*	*Image by Author*
Fig. 5.02*	*Image by Author*
Fig. 5.03*	*Image by Author*
Fig. 5.04*	*Image by Author*
Fig. 5.05*	*Image by Author*
Fig. 5.06*	*Image by Author*
Fig. 5.07*	*Image by Author*
Fig. 5.08*	*Image by Author*
Fig. 5.09*	*Image by Author*
Fig. 5.10*	*Image by Author*

FIN.