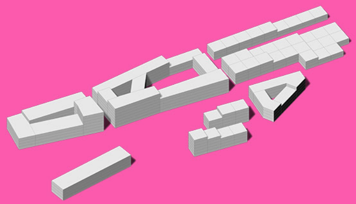
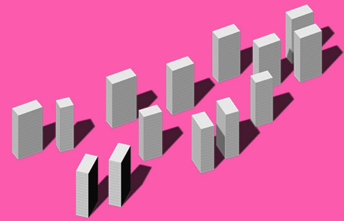
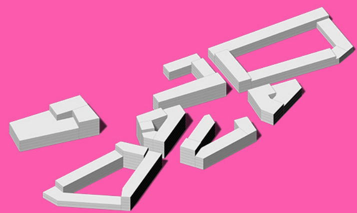
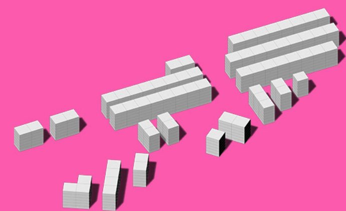


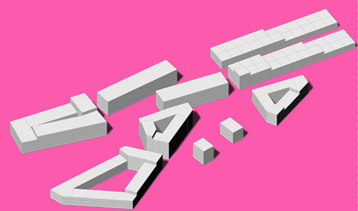




EVOLUTIONARY ALGORITHMS:

THEIR CREATIVE APPLICATIONS in the EARLY DESIGN PHASE of URBAN DESIGN & ARCHITECTURE





EVOLUTIONARY ALGORITHMS: THEIR CREATIVE APPLICATIONS in the EARLY DESIGN PHASE of URBAN DESIGN & ARCHITECTURE

Politecnico di Torino
MSc, Architecture for Sustainability Design

Master Thesis

Supervisor
Robiglio Matteo

Co-Relator
Barioglio Caterina

Company
Brains Digital SRL

Authors
Kuş Büşra
Özgen Berkay
Şimşek Uğur Yıldırım

July, 2023

Politecnico di Torino

PREFACE

This thesis is the result of a remarkable collaboration that brought together minds from different backgrounds and locations, striving to create something meaningful and impactful. Despite the significant gaps, both in terms of mental perspectives and physical distances, our shared passion for knowledge and discovery brought us on a journey that spanned from Izmir to Torino and the most recently to Copenhagen.

Throughout this vibrant journey, we encountered numerous stakeholders, each with their own unique background and expertise, who generously contributed their valuable insights and perspectives. While some of these individuals may not be officially recognized as part of this thesis, their contributions played an indispensable role in enriching our collective experience.

As we present this thesis, we acknowledge that it is not the end of our journey but rather a stepping stone towards further discovery and growth. We hope that the insights shared within these pages inspire others and contribute to the ongoing discourse in the field.

B.U.B.

"We are talking about a symbiosis that is a cohabitation of two intelligent species"

regarding to human-computer relationship in the book of The Architecture Machine (1970, The MIT Press) by Nicholas Negroponte

CONTENT

ABSTRACT

> The field of urban design and architecture is continually evolving, driven by advancements in technology and the need to address complex challenges in the built environment. As cities grow, there is an increasing demand for sustainable, efficient, and aesthetically pleasing urban spaces. To meet these demands, designers and urban planners are constantly seeking innovative approaches that can enhance the design process, optimize spatial configurations, and promote creative solutions.

Evolutionary algorithms, a subset of artificial intelligence(AI) and computational design, have emerged as powerful tools with the potential to revolutionize the way we approach urban design and architecture. These algorithms, inspired by natural selection and evolutionary principles, simulate iterative processes of variation, selection, and adaptation. By harnessing the power of evolutionary algorithms, designers can generate and evaluate a vast range of design solutions, uncovering novel and optimized urban design strategies.

Despite the promising potential of evolutionary algorithms in urban design and architecture, there remains a gap between their theoretical applications and their practical implementation in real-world projects. The field still lacks a comprehensive understanding of how to effectively integrate these algorithms into the design workflow and how to maximize their creative applications to tackle complex urban design challenges.

Furthermore, the creative potential of evolutionary algorithms in urban design and architecture is yet to be fully explored. While these algorithms are commonly used for optimization and problem-solving tasks, their capacity to inspire and generate innovative design solutions is often overlooked. There is a need to explore the creative capabilities of evolutionary algorithms and understand how they can push the boundaries of traditional design thinking in urban contexts. Therefore, the motivation of this thesis is to investigate the evolutionary algorithms and their creative applications in urban design and architecture. The research aims to bridge the gap between theory and practice by developing a comprehensive framework that enables designers to effectively utilize evolutionary algorithms as creative tools in the urban and architecture design process. By exploring the potential of these algorithms to generate unique and contextually responsive design solutions, this research seeks to contribute to the advancement of practice in the field.

Keywords: Evolutionary Design, Urban Form, Parametric Design, Multi-Objective Optimization

... 0.1.1. Problem Definition

> In recent years, the field of architecture and urban design has become increasingly complex due to the dynamic interplay of environmental, social, and economic factors. Every project face numerous challenges arising from the lack of coordination among various components involved in the design process. Limitations related to sustainability, socio-economic considerations, and stakeholder collaboration often impede the achievement of high-quality design outcomes. While engineering disciplines have witnessed significant innovations in their design methodologies, the field of architecture and urban design has yet to fully embrace transformative approaches.

The thesis aims to develop a creative framework (tool) that utilizes machine learning algorithms to optimize the balance between architectural and planning components in the early stages of design. By harnessing the power of evolutionary algorithms and parametric design techniques, the tool aims to enhance the overall quality and efficiency of sustainable design solutions.

The primary objective of the tool is to enable designers to explore a diverse range of architectural typologies and livable urban typologies, generating and evaluating multiple design alternatives by considering key performance criteria such as; sustainability, socio-economic impact, and time while balancing various project constraints.

Moreover, the tool aims to enhance collaboration among stakeholders by providing a shared platform for communication and decision-making. Clients, urban planners, architects, engineers, and other relevant stakeholders actively engage and participate in the design process, fostering a comprehensive understanding of project objectives and priorities. This collaborative approach ensures that resulting design solutions align with the needs and aspirations of the community.

To further improve the efficiency of the design process, the tool significantly reduces the time required to generate design alternatives. By providing a wide range of solutions within hours, as opposed to weeks, designers can explore numerous possibilities and make informed decisions in a timely manner.

... 0.1.2. Research Questions

How can evolutionary algorithms be effectively integrated into the urban design and architecture workflow to optimize the balance between architectural and planning components?

How can evolutionary algorithms contribute to the creative process in urban design and architecture, going beyond optimization and problem-solving tasks?

What are the contextual factors that need to be considered when applying evolutionary algorithms to generate contextually responsive and unique design solutions in urban environments?

How can the collaborative features of the genetic design tool enhance communication and decision-making among various stakeholders involved in urban design projects?

How can the tool integrate data-driven approaches, such as utilizing urban analytics and real-time data, to inform and enhance the design decision-making process?

How can the tool incorporate feedback loops and iterative processes to enable designers to continuously refine and improve design solutions throughout the project lifecycle?

What are the practical implications and potential challenges of integrating generative design methodologies into the traditional urban design workflow?

How does the use of the generative design tool influence the final design outcomes in terms of creating vibrant public spaces, fostering community engagement, and enhancing the overall livability of urban areas?

These research questions will guide the investigation into the application of genetic design and optimization in the field of urban design, providing insights into the potential benefits, challenges, and implications of using such tools in practice.

... 0.2. Methodology

... 0.2.1. Theoretical Chapters

> In a brief overview, the thesis is divided into two chapters: theoretical and design. In the theoretical chapters, the thesis provides background information to the reader about the pillars of genetic design principles and how artificial intelligence has developed over the years, enabling us to perform such activities. This is followed by a bibliographic research on urban density, its classification, and a discussion about livable urban density criteria. Additionally, through collaborative research conducted with Brains Digital company, the thesis investigates current design and analysis tools with a generative design mindset in the architecture and urban design industry.

Next, the design chapters aim to consolidate the discussions highlighted in the theoretical chapters and propose a creative framework to address the issues discussed. Specifically, we have developed scripts in Rhinoceros/ Grasshopper to parametrize the early design process. While creating these scripts, we have either utilized our custom-written scripts or gained valuable insights from existing research, plug-ins, platforms, etc. To execute our creative workflow and run design simulations, we have used the evolutionary design engine called Wallacei, (Makki, & Showkatbakhsh, 2018).

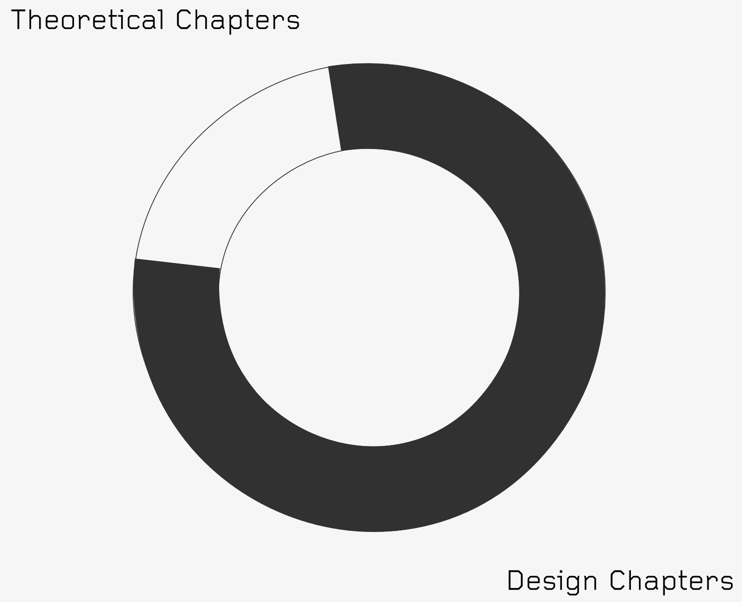
... 0.2.2. Design Chapters

In more detail, in the design chapters, we conducted three experiments using the tool we have developed in the thesis. In the first chapter of the design part, we focus on a case study located in the Milan region since the proposed project aims to design Italy's first carbon-neutral social housing project, thereby sharing similar environmental concerns with ours. The objective in this scenario is to enhance the performance values of the design generations in terms of environmental aspects while maximizing the typology varience in the built area. In order to compare generated design solutions with case study we keep floor area ratio (FAR) as a constant in this experiment. Once we get the design solutions from the simulation we sort them according to their performance values and make a comparison with existing project proposal. This experiment is aims to discuss the potential usage of the machine learning algorithms and parametric design techniques in the creative process of livable urban density creation.

In the second experiment, the objective is to evaluate the effectiveness of the tool as an assistant to designers rather than a standalone creative tool. Therefore as designers, we use the tool to generate extreme urban scenarios in the same location while keeping the FAR constant. The focus here is on the variety of options that can be created rather than solely increasing the performance values of the design.

In the final part of the design chapters, the research shifts from the urban design process to the building scale to validate the effectiveness of the approach in a different context and scale. One of the selected phenotypes from the first experiment is chosen, and the focus is directed to one of the buildings within that design solution. Once we have a building, we generate apartment layout solutions using genetic design simulation and discuss the generated outcomes.

In summary, the aim of the paper is to create a comprehensive creative framework. We begin with a theoretical background research on the definitions and analyze the current design tools. After examining the potentials and limitations of those tools, we propose our original creative framework. To validate the effectiveness of the tool, we conduct experiments in a variety of scales and contexts. During the course of these experiments, we aim to discuss the current role of the designer and the potential future cooperation between human and artificial intelligence. We explore how the proposed framework can empower designers and facilitate their collaboration with AI technologies. The general scheme of the paper is illustrated on the right.



C-1: *Exploring the Creative Process of AI*	**C-2:** *Understanding Livable Urban Forms*	**C-3:** *Creating Master Plan*	**C-4:** *Built Environment & Spatial Configuration of AI*	**C-5:** *Creating Floor Plan*

... 0.2.3. Positioning the Thesis

> The thesis is positioned as a comprehensive exploration of the integration of machine learning algorithms and parametric design techniques in the fields of urban design and architecture. It aims to address the complex challenges faced in these disciplines by proposing a creative framework that optimizes the balance between architectural and planning components.

Throughout the research, there is an emphasis on collaboration among stakeholders, including clients, urban planners, architects, engineers, and other relevant parties.

The thesis also explores the potential role of the designer and the future cooperation between human intelligence and artificial intelligence. It examines how the framework empowers designers and encourages their active participation in the design process, while also leveraging the capabilities of machine learning algorithms.

Overall, the thesis positions itself as a contribution to the advancement of urban design and architecture by proposing a novel generative design tool tailored to urban contexts. It aims to optimize design outcomes in terms of sustainability, socio-economic considerations, stakeholder collaboration, and time efficiency.

chapter 01

... EXPLORING THE CREATIVE PROCESS OF AI

... 1.3.1. A Historical Perspective to Creative AI

> In 1950, Alan Turing, an exceptional polymath from Britain, established the initial framework for the investigation of artificial intelligence. He embarked on a mathematical exploration of machine intelligence, proposing that humans utilize accessible information and logical reasoning to solve problems and make decisions. This proposition led to a pivotal inquiry: Why can't machines do the same? Turing's groundbreaking work found its culmination in the influential paper titled "Computing Machinery and Intelligence." In this paper, he not only discussed the construction of intelligent machines but also outlined techniques for evaluating their intelligence

> In 1970, Nicholas Negroponte, published "The Architecture Machine." This groundbreaking work introduced visionary ideas that continue to shape the discourse on the human-AI relationship in architecture. Negroponte's exploration of the intersection between technology and design laid the foundation for integrating artificial intelligence into architectural practice. "The Architecture Machine" envisioned a future where computers and advanced technologies would revolutionize the architectural profession. Negroponte proposed that intelligent machines could augment human creativity, enabling architects to explore new design possibilities and enhance their decision-making processes. Computers, according to him, could serve as powerful tools for architectural analysis, simulation, and representation, leading to more efficient and innovative design practices. Negroponte's vision extended beyond the traditional one-architect-one-machine dialogue. He emphasized that an adaptable machine, which he referred to as an architecture machine, should establish two further contacts with the real world. Firstly, it should receive direct sensory information from the environment, being able to see, hear, read, and even take walks in the garden(Negroponte, 1970: 27-29).

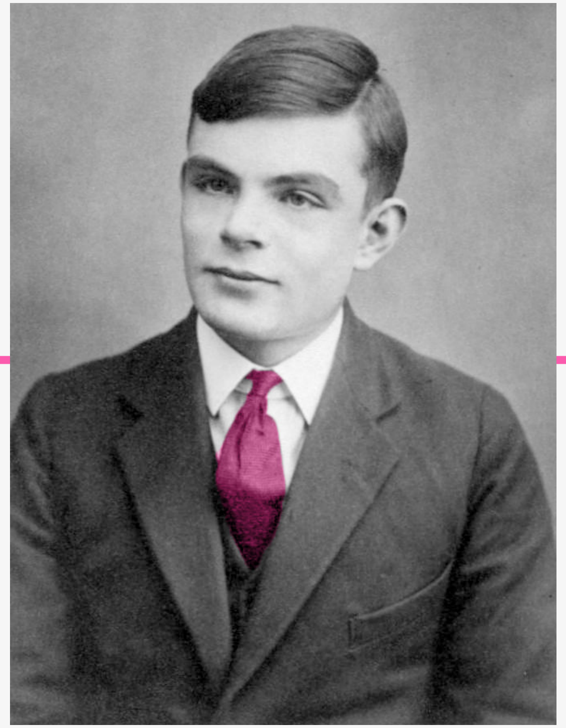


Figure 1: Portrait of Alan Turing

Figure 2: "URBAN 5" an early example of human-machine dialogue

> On May 11, 1997, a significant milestone in the advancement of artificial intelligence occurred when IBM Deep Blue, a computer system, achieved victory over the reigning world chess champion, Garry Kasparov. This momentous event unfolded during a six-game match, with IBM securing two wins, the champion securing one win, and three games ending in draws.

Figure 3: Garry Kasparov against Deep Blue

... 1.3.1. A Historical Perspective to Creative AI

> The artwork titled "Edmond de Belamy" has made history in 2018 by becoming the first piece created by artificial intelligence to be sold at auction.The creation of "Edmond de Belamy" series involved the utilization of a two-part algorithm, particularly GAN algorithms. To generate these artworks, the team behind the project, fed the algorithm 15,000 images of portraits from various time periods.

Figure 4: Edmond de Belamy, GAN generated painting

Figure 5: Nature Dreams, Refik Anadol

> Can machines dream? In 2019 Refik Anadol as an important figure to architecture and AI relationship, he created his artwork called "Machine Hallucinations" and series of other artworks using generative algorithms for the upcoming years with advecments and further researchs in his techinque.

> In 2022, a significant advancement in the field of AI emerged from OpenAI known as DALL-E 2, a hierarchical text-conditional image generator. DALL-E 2, along with similar image generators, is a neural network-based model capable of generating images from textual descriptions. These models exhibit a remarkable ability to comprehend and envision visual concepts based on textual prompts, resulting in the creation of highly realistic and novel images. These advancements have not only expanded the possibilities of generating visual content but have also paved the way for new forms of collaboration between humans and machines in the creative process.

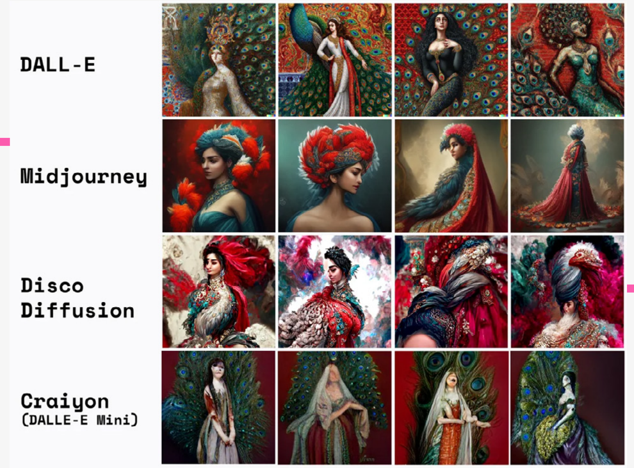


Figure 6: Text to Image Generators

... 1.3.2. Genetic Algorithms

> Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search(Goldberg, 1989).

The algorithm follows a series of steps to iteratively improve a population of potential design solutions. These steps include initialization, evaluation, selection, recombination, mutation, and termination. Initially, a population of design solutions is randomly generated. Each solution is evaluated based on predefined criteria or objectives. The selection process favors solutions with better fitness, which are more likely to produce desirable outcomes. Through recombination, selected solutions are combined to create new offspring with varied genetic information. Mutation introduces small random changes to the offspring, ensuring exploration of new design possibilities. The evaluation, selection, recombination, and mutation steps are repeated iteratively until a termination condition is met, such as reaching a specified number of generations or achieving a desired level of design performance.

As an example, we can consider a population of frogs living in a pond. The characteristics of each frog, such as its size, color, and jumping ability, can be represented as a set of parameters or variables in a genome. Through the processes of reproduction, mutation, and selection, the population of frogs can evolve over time to better adapt to their environment. For example, if there is a scarcity of flies in the pond, the frogs with the best jumping ability may have a higher chance of survival and reproduction, leading to an increase in the frequency of genes associated with good jumping ability in the population. This process is analogous to the way that genetic algorithms operate to find the optimal solution to a given problem.

Genetic algorithms are often used to solve optimization problems where the goal is to find the optimal solution to a given problem. They are particularly useful for problems that are too complex to be solved using traditional optimization techniques or for problems where the solution space is too large to be searched exhaustively. Some examples of problems that can be solved using genetic algorithms include finding the optimal parameters for a machine learning model, optimizing the design of a product or system, solving scheduling and resource allocation problems, and optimizing the layout of a manufacturing facility.

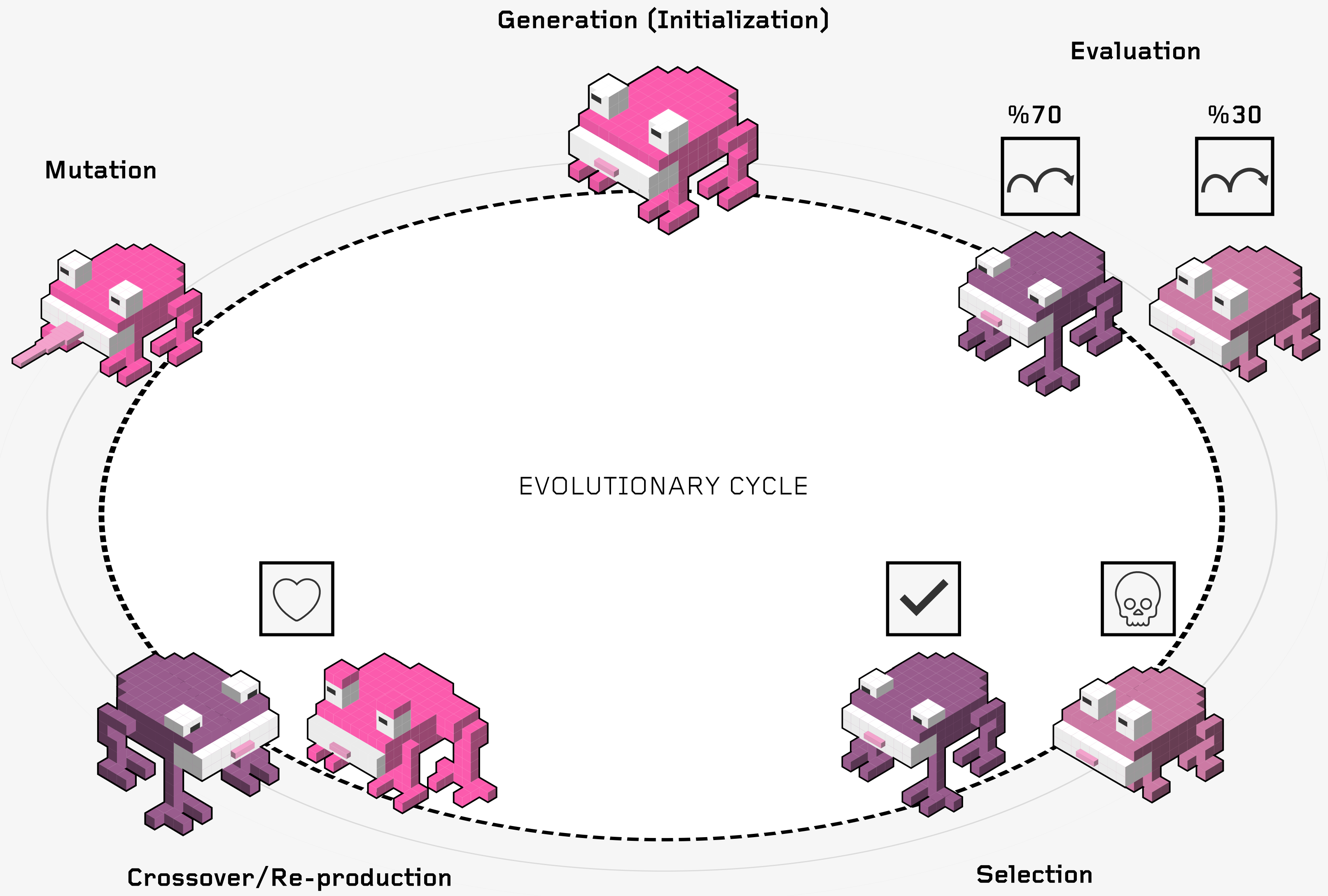


Figure 7: Natural Selection Cycle

a.Initialization

> Initialization is the process of creating the initial population of solutions that will be evolved over the course of the algorithm. This initial population is often created randomly, with each individual solution being a possible solution to the problem that the evolutionary algorithm is trying to solve.

The process of initializing a population in a genetic algorithm involves randomly generating a set of solutions, known as individuals or genomes, that represent potential solutions to the problem at hand. These individuals are then evaluated using a fitness value, which measures their quality or suitability as a solution to the problem.

Once the initial population has been generated and evaluated, the evolutionary algorithm can begin the process of evolving the population towards better solutions. This typically involves applying genetic operators, such as crossover and mutation, to the individuals in the population in order to generate new, potentially improved, solutions. The process continues until a satisfactory solution is found or a pre-defined stopping criteria is met. If we go back to frog example, which we are trying to evolve a population of frogs that can jump the furthest distance. To do this, we first need to generate the initial population of frogs that will be evolved over the course of the algorithm. In the initialization stage, we might randomly generate a set of 100 frogs, each with different characteristics such as body size, leg length, and muscle strength. These characteristics would be encoded in a genome which is a string of values that represents the genetic makeup of each individual frog. Next, we would evaluate each frog in the population using a fitness function that measures how far they can jump. This fitness function might take into account factors such as the frog's body size and leg length, as well as the force of their jump. Based on their fitness scores, we can rank the frogs in the population from best to worst in terms of their jumping ability. The top-performing frogs will then be selected to move on to the next stage of the algorithm, while the lower-performing frogs will be discarded. From this initial population, the evolutionary algorithm can begin to apply genetic operators such as crossover and mutation to generate new, potentially improved, solutions. Over time, the population of frogs will evolve and become better at jumping, with the best-performing individuals passing on their genes to the next generation. Eventually, the algorithm may converge on a population of frogs that are capable of jumping the furthest distance, solving our problem.

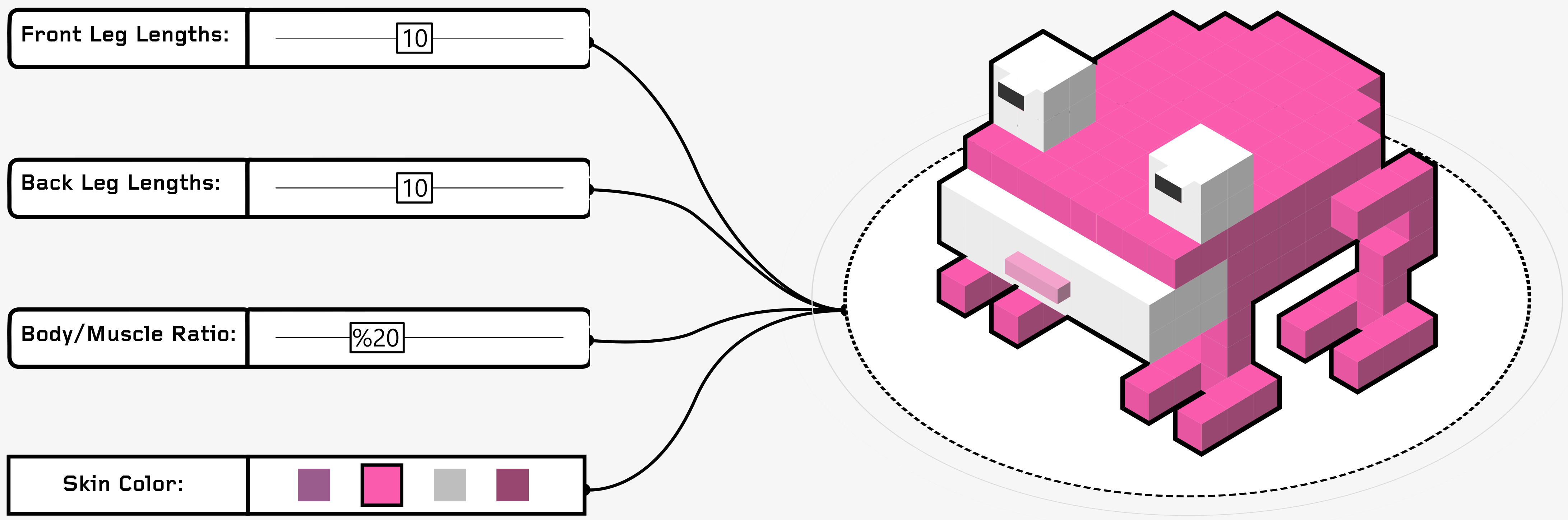


Figure 8: An example to initialization phase

b.Evaluation

> In a genetic algorithm, the evaluation phase is a the step where the fitness of each individual in the population is determined. The main goal of this phase is to assign a numerical value, known as the fitness value or fitness score. The fitness value of an individual is a measure of how well it performs a certain task or solves a given problem. It is usually calculated by comparing the results of the individual to some desired goal or target.

During the evaluation phase, each individual in the population is assigned a fitness score based on how closely it meets the desired criteria. In this case, the evaluation phase would involve measuring the jumping ability of each individual frog in the population using a fitness function.

The fitness function might take into account factors such as the frog's body size, leg length, and muscle strength, as well as the force of their jump. By applying this fitness function to each individual frog, the algorithm can calculate a numerical score that indicates how well that frog performs on the problem of jumping the furthest distance.

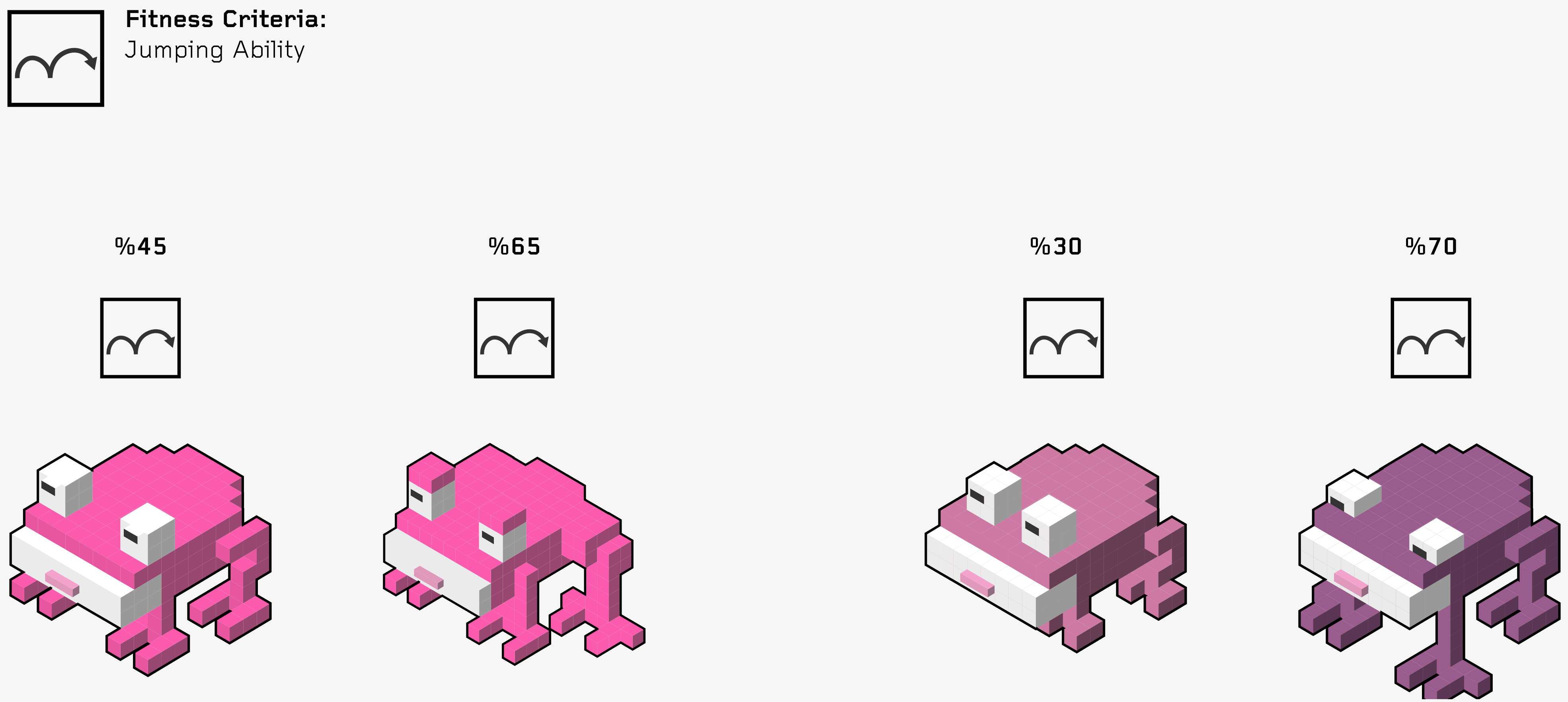


Figure 9: Comparison of fitness values of a population

For example, if we have a population of 100 frogs, the fitness function might produce scores that look something like this:

Frog-1: fitness score = 45
Frog-2: fitness score = 65
Frog-3: fitness score = 30
Frog-4: fitness score = 70
Frog-5: fitness score = 40
Frog-6: fitness score = 65
.
.
.
Frog-100: fitness score = 50

In this way, the evaluation phase helps the genetic algorithm to identify the best solutions in the population and guide the evolutionary process towards better solutions. By measuring the fitness of each individual in the population, the algorithm can determine which frogs are the most promising candidates for further evolution and focus its efforts on improving those solutions.

c.Selection

> The selection phase is the next step in the process of a genetic algorithm, following the evaluation phase. In this phase, the algorithm selects the best-performing individuals in the population to move on to the next stage of evolution. This is done using a selection method, which is a set of rules or criteria that determine which individuals will be selected for further evolution.

There are many different selection methods that can be used in genetic algorithms, and the specific method used will depend on the nature of the problem being solved and the goals of the algorithm. Regardless of the specific selection method used, the selection phase serves the important function of selecting the best solutions in the population and ensuring that they are passed on to the next generation. By selecting the best individuals to move on to the next stage of evolution, the algorithm can focus its efforts on improving those solutions and increasing the chances of finding a good solution to the problem at hand.

To continue with our example of frogs jumping the furthest distance, let's say that we have a population of 100 frogs and we have used the fitness function to rank them from best to worst in terms of their jumping ability. In the selection phase, we might apply a threshold survival to choose the top-performing frogs to move on to the next stage of evolution.

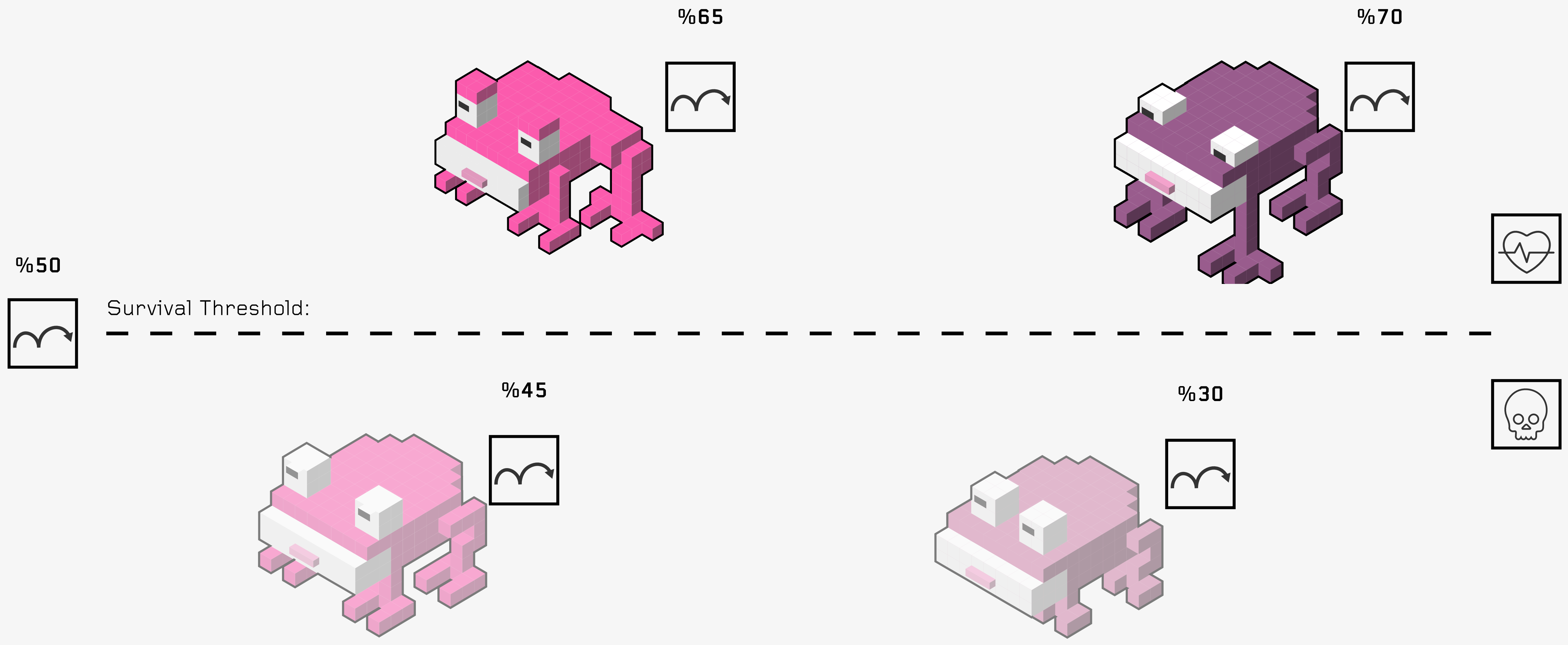


Figure 10: An example to natural elimination

If we say that the fitness value 50 is the critical threshold value for frogs to survive in pond, we can compare each frogs fitness value to given threshold value. Thus frogs with fitness values under 50 is going to be eliminated and the ones that are fullfiling the threshold value will survive and pass their genes to next generations.

In this way, the selection phase helps the genetic algorithm to identify the best solutions in the population and ensure that they are passed on to the next generation, where they can be improved through the application of genetic operators. By selecting the best individuals to move on to the next stage of evolution, the algorithm can focus its efforts on improving those solutions and increasing the chances of finding a good solution to the problem at hand.

d.Crossover/Re-production

> In this phase, the algorithm combines the genetic information of the selected individuals to produce new, potentially improved, solutions. This is done using a genetic operator called crossover, which is a process that combines the genetic information of two parent individuals to produce one or more offspring individuals. The crossover phase is an important step in the process of a genetic algorithm, as it allows the algorithm to explore new, potentially improved, solutions by combining the genetic information of the best-performing individuals in the population. By applying the crossover operator to the selected individuals, the algorithm can generate new solutions that inherit characteristics from their parents and potentially improve upon them.

Going back to our frog experiment about jumping the furthest distance, let's say that we have selected the top 10 performing frogs in the population and want to apply the crossover operator to produce new, potentially improved, solutions. We might use the two-point crossover method to combine the genetic information of the selected frogs and produce two offspring frogs with a combination of characteristics from their parents. For example, let's say that we have two parent frogs with the following chromosomes:

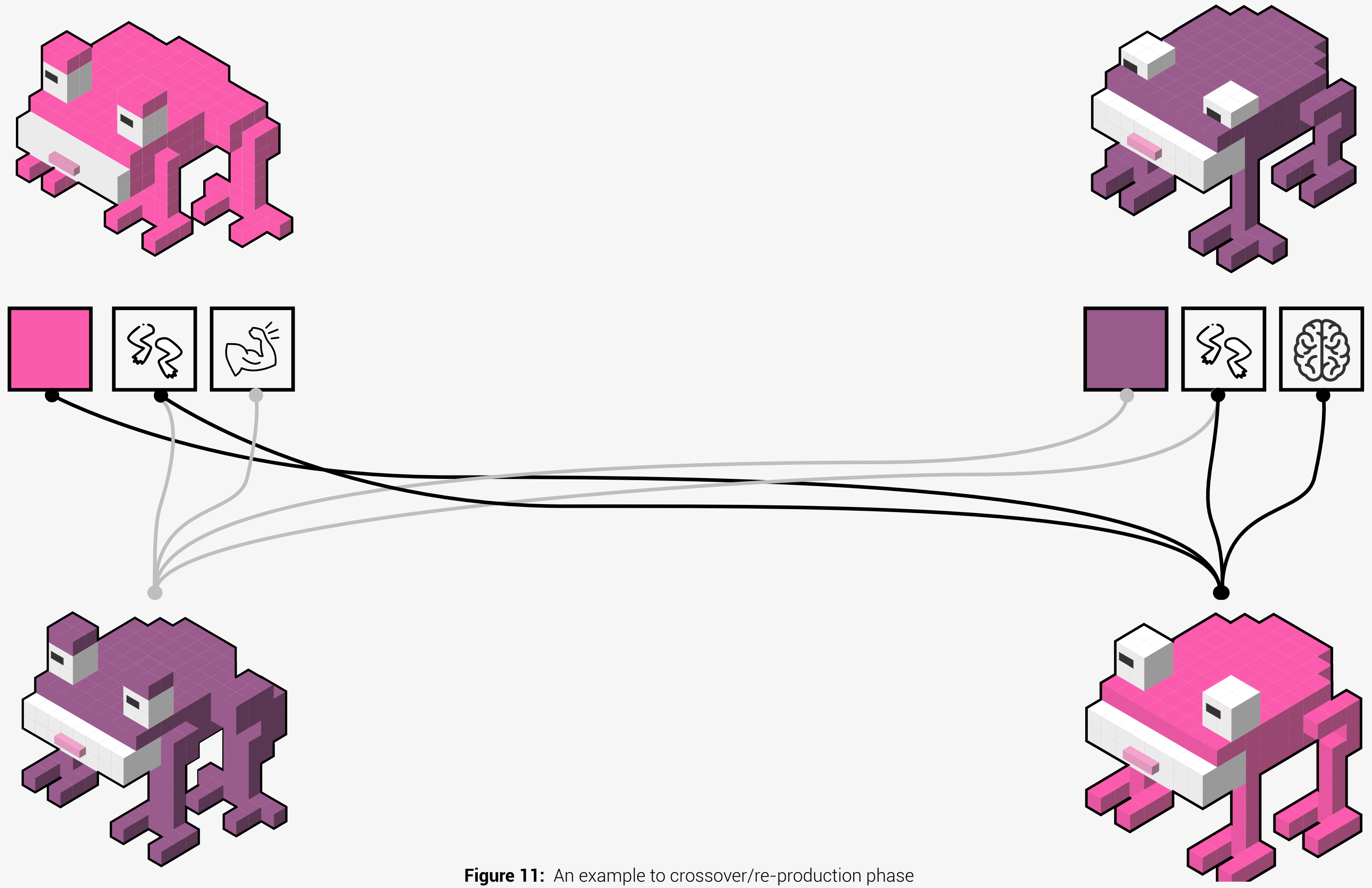


Figure 11: An example to crossover/re-production phase

Parent-1: [0.5, **0.7**, 0.2, 0.9, 0.1]
Parent-2: [0.8, 0.4, **0.6**, 0.3, 0.5]

We would select two crossover points at random along the length of the chromosomes, such as the 2nd and 3rd positions: The genetic information between the two crossover points (positions 2 and 3) would be taken from one parent, while the genetic information outside of those points would be taken from the other parent. This would produce the following two offspring frogs:

Offspring-1: [0.5, 0.4, **0.6**, 0.9, 0.1]
Offspring-2: [0.8, **0.7**, 0.2, 0.3, 0.5]

As we can see, the offspring frogs have a combination of characteristics from their parents, with some traits taken from one parent and other traits taken from the other parent. This allows the algorithm to explore new, potentially improved, solutions by combining the genetic information of the selected individuals.

e.Mutation

> In genetic algorithms, the mutation phase is a process in which the genetic material of individuals in the population is randomly altered in order to introduce new genetic variation. This process is an important part of the evolutionary algorithm, as it allows the population to adapt and evolve over time in response to changes in the environment or the problem being solved.

In the context of evolving a population of frogs that can jump the furthest distance, the mutation phase might involve randomly altering the DNA sequence of each frog's genes. This could result in the development of new traits that could potentially improve the frog's jumping ability, such as stronger legs or a more streamlined body shape.

For example, if a mutation caused a frog to express a gene that regulates the development of its leg muscles in a different way, this could result in the frog having stronger legs and an improved jumping ability. Alternatively, if a mutation caused a frog to have a longer tongue, even though it retains the same jumping capability, the longer tongue may enable it to catch more flies without jumping. Consequently, it increases its chances of survival in a pond by acquiring a new skill.

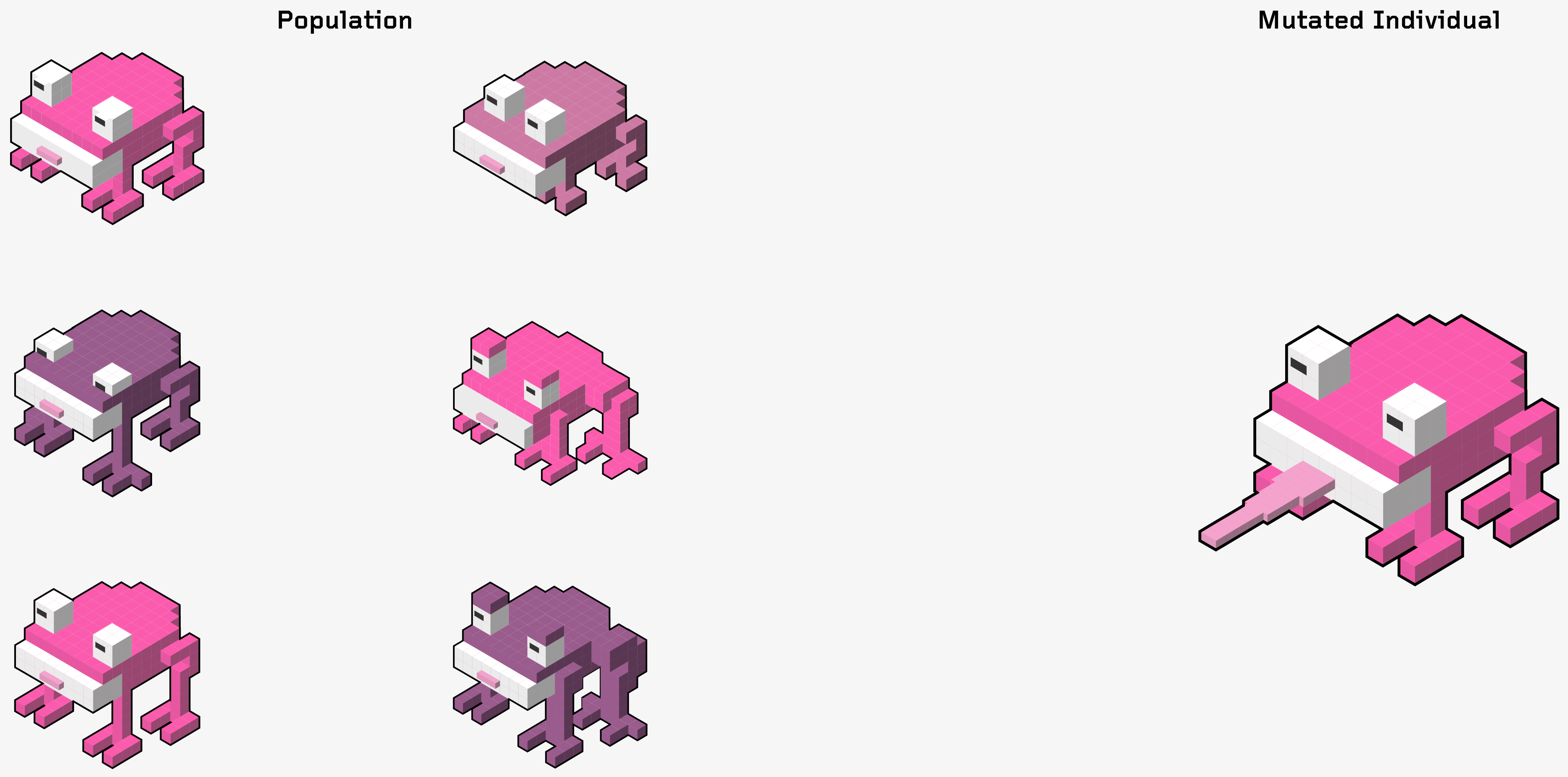


Figure 12: Initial population and mutated individual

The specific mutation rate, or the probability that a particular gene will be mutated, is typically chosen by the user and can vary depending on the problem being solved and the goals of the evolutionary algorithm. A higher mutation rate can introduce more genetic variation into the population, but may also result in a higher number of less fit individuals. A lower mutation rate can reduce the number of less fit individuals, but may also slow down the rate of evolution in the population.

... 1.3.3. Multi-Objective Optimization

> In the frog pond example, we have introduced a single-objective evolution where we consider the jumping distance as the fitness value and evaluate their survival skills related to that skill. However, as we introduced another unrelated feature during the mutation phase, such as the length of the tongue, it may drastically increase the chances of survival. Therefore, when we have **more than one competing objective** to optimize, we refer to it as multi-objective optimization. These problems involve finding a set of solutions that represent the best trade-off between multiple conflicting objectives, rather than simply finding a single solution that optimizes a single objective.

To give another example of multi-objective optimization using frogs, we can imagine a scenario in which we are trying to evolve a population of frogs that can jump the furthest distance while also minimizing the amount of energy required to jump. This would involve balancing the trade-off between jumping distance and energy efficiency, as it may not be possible to achieve the optimal solution for both objectives simultaneously.

In this case, the multi-objective optimization problem would involve finding a set of solutions that represent the best trade-off between jumping distance and energy efficiency. To solve this problem using a genetic algorithm, it would be necessary to define a fitness function that captures both objectives. This fitness function could take into account factors such as the frog's body size, leg length, muscle strength, and the force of its jump, as well as the energy required to jump.

By applying this fitness function to each individual frog in the population, the algorithm can evaluate the quality or suitability of each solution and select the fittest individuals to move on to the next generation. Over time, the population of frogs will evolve and become better at balancing the trade-off between jumping distance and energy efficiency, with the best-performing individuals passing on their genes to the next generation.

The Pareto front represents the set of solutions that are not dominated by any other solution in the population, in terms of their performance on the multiple objectives being optimized. To understand the Pareto front, it is helpful to consider a two-dimensional plot in which one objective is plotted on the x-axis and the other objective is plotted on the y-axis. In this plot, each solution in the population is represented by a point, with the x-coordinate representing the performance on the first objective and the y-coordinate representing the performance on the second objective.

The Pareto front is then defined as the set of points that lie on or below the curve formed by connecting the points in the population. Any point on the Pareto front represents a solution that is not dominated by any other solution in the population, in terms of its performance on the two objectives being plotted(Pareto, 1906/2020).

In the image on the right, two Pareto front solutions are illustrated to provide a better understanding of the matter. The fitness values that are closer to the center of the polygon are considered the fitter objects, indicating better performance. While the purple frog demonstrates better fitness values in terms of body/muscle ratio and leg length, the pink frog exhibits superior values in neural capacity and tongue length. Since all of these values contribute to increased survival chances, we can conclude that these solutions are non-dominant to each other. Relating this concept to architecture, when designing a building, there are often multiple design objectives to consider, including maximizing natural light, optimizing energy efficiency, and ensuring aesthetic appeal. These objectives may conflict with each other, necessitating trade-offs to achieve the desired outcomes.

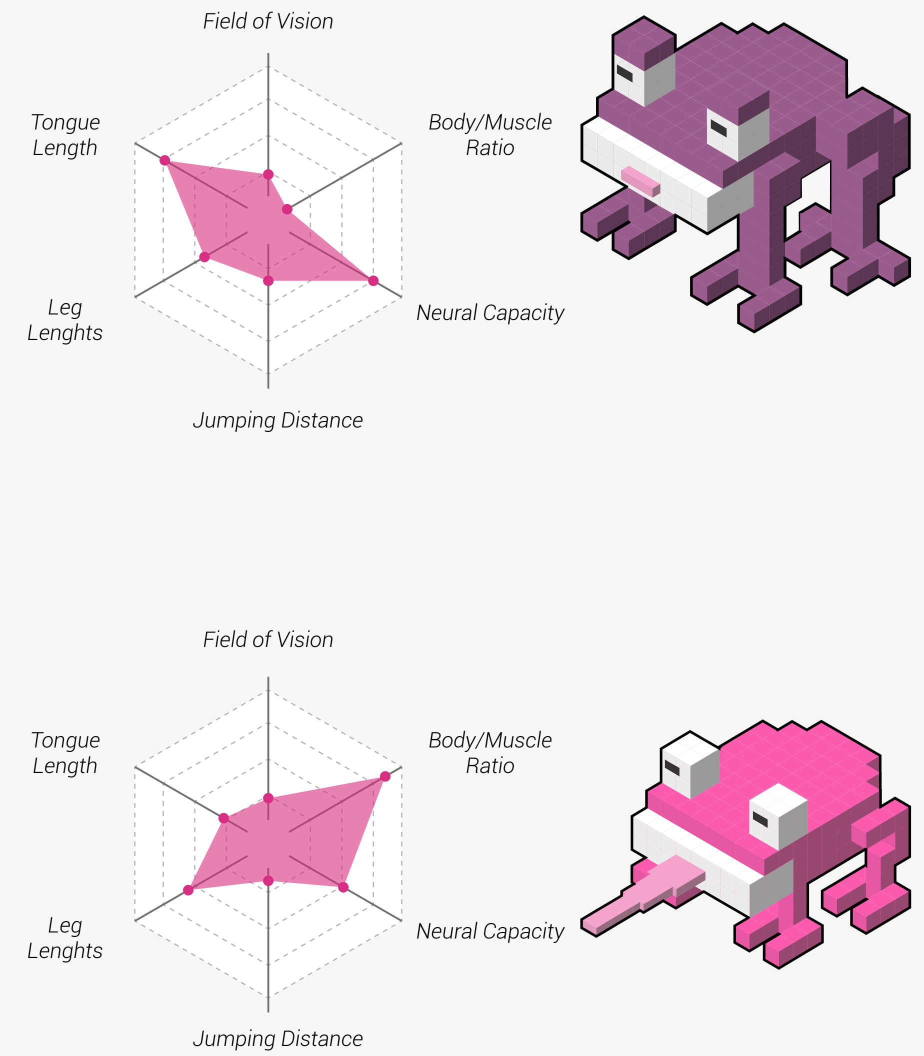


Figure 13: Comparison of two individuals based on multiple-criteria

chapter 02

... UNDERSTANDING LIVABLE URBAN FORMS

chapter 02

... P-01 DECODING URBAN FORMS THROUGH DENSITY

> Within this section, the thesis delves into the multifaceted aspects of urban density, with a primary focus on comprehending the implications and considerations associated with creating livable and sustainable cities. Through a comprehensive analysis, this section aims to provide insights into effective urban design strategies and the creation of urban formations that promote a softer and more livable urban density.

> Density is a concept that encompasses the concentration of individuals or physical structures within a given geographical unit. Physical density represents a quantitative and objective measure, whereas perceived density is subjective and influenced by an individual's perception of the number of people in a specific area, as well as the organization and presence of open space. While the physical characteristics of space play a significant role in the perception of density, the interaction between individuals and their environment also assumes a crucial role.

Achieving similar levels of built density is possible through the utilization of different building typologies, including high-rises, mid-rises, townhouses, or detached houses. This implies that although the physical appearances and configurations of these buildings may differ considerably, the ratio of constructed space to land area can be equivalent.

In this chapter, the doctoral thesis titled "Space, Density and Urban Form" is used as a foundational supplementary resource for investigating the concept of density and its associated measurement methods(Yolanda & Haupt, 2009). The thesis critically examines the concept of urban density and practical implications in urban planning and design. The aim is to address the limitations of existing density measurement methods and propose a new approach, known as the Spacematrix method, that incorporates multiple variables and scales. By redefining density as a multivariable and multi-scalar phenomenon, the study seeks to provide a comprehensive framework for understanding and utilizing density as a guiding tool in urbanism, integrating quantitative and qualitative considerations.

By incorporating the knowledge and findings presented in the thesis, the research aims to enhance the comprehension and evaluation of density as a crucial aspect of urban planning and design.

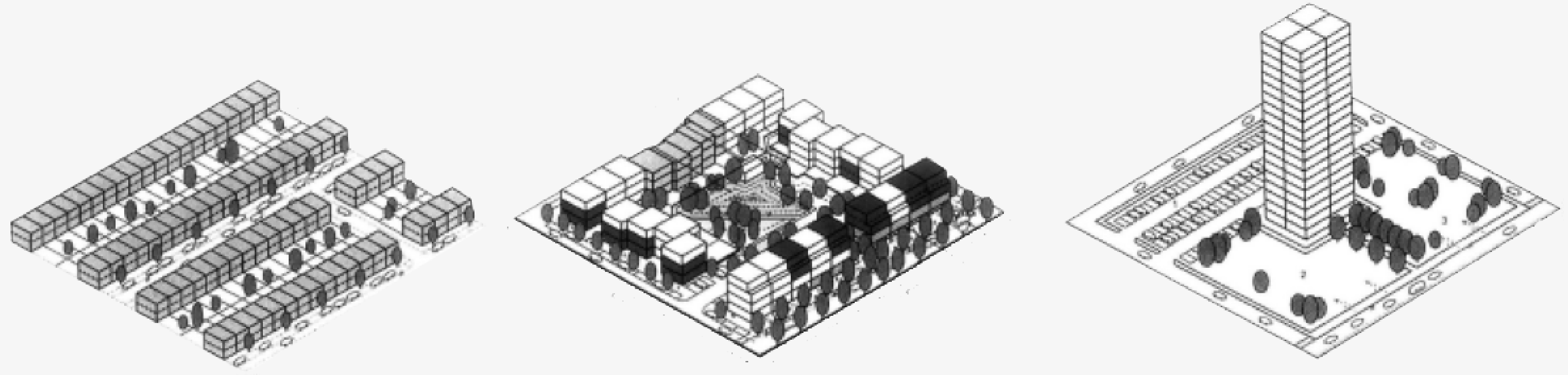


Figure 14: Three urban forms characterized by a density of 75 dwellings per hectare

a. Population and Dwelling Density

Population density and housing density are fundamental metrics utilized to analyze the concentration of individuals and dwellings within a specific region. Population density quantifies the quantity of individuals or households residing in a designated area, while residential density focuses on the number of residential units within that same area.

b. Landuse Intensity (FAR)

Land use density, commonly referred to as the floor area ratio (FAR), serves as a zoning regulation to control a development density and the permissible amount of floor area that can be constructed on a specific land parcel. It plays a pivotal role in establishing the volumetric, dimensional, and density characteristics of buildings within a designated area. Typically expressed as a ratio or percentage, FAR represents the allowable floor area in relation to the total land area.

c. Coverage (GSI)

Coverage, alternatively referred to as the Gross Area Index (GAI), serves as an evaluative metric employed to assess the intensity of development or land utilization within a specific locality. It quantifies the ratio between the aggregate floor area of constructed buildings and the total land area, encompassing both built structures and open spaces.

d. Building Height

Building height pertains to the vertical extent of a structure, conventionally quantified in terms of the number of stories or floors it encompasses. It is a significant determinant influencing the physical configuration and density of urban environments. Building height regulations are commonly implemented through ordinances or zoning regulations to uphold the envisioned urban character and guarantee harmonious integration with the surrounding context.

e. Spaciousness (Open Space Ratio)

The Open Space Ratio (OSR), also known as the Spaciousness Open Area Ratio, serves as an evaluative metric within the realm of urban planning, utilized to gauge the relationship between open areas and built floor areas in development. It facilitates the assessment of the equilibrium between optimizing building volume and ensuring the provision of sufficient open space.

Bijlmer, Netherlands
Inhabitants per hectare: 112
Dwellings per hectare: 44

Betondorp, Netherlands
Inhabitants per hectare: 108
Dwellings per hectare: 74

Zuidwest Kwadrant, Netherlands
FAR: 0.78
Dwellings per hectare: 51

Nieuw Sloten, Netherlands
FAR: 0.77
Dwellings per hectare: 36

Berlage, Plan Zuid, Netherlands
Coverage(GSI): 0,37

Noorderhof, Netherlands
Coverage(GSI): 0,39

Nieuw Sloten, Netherlands
FAR: 0.77
The average number of floors: 2.3

Bijlmer, Netherlands
FAR: 0.76
The average number of floors: 7.9

Venserpolder, Netherlands
Spaciousness(OSR): 0.63
The average number of floors: 4.5

Noorderhof, Netherlands
Spaciousness(OSR): 0.60
The average number of floors: 2.5

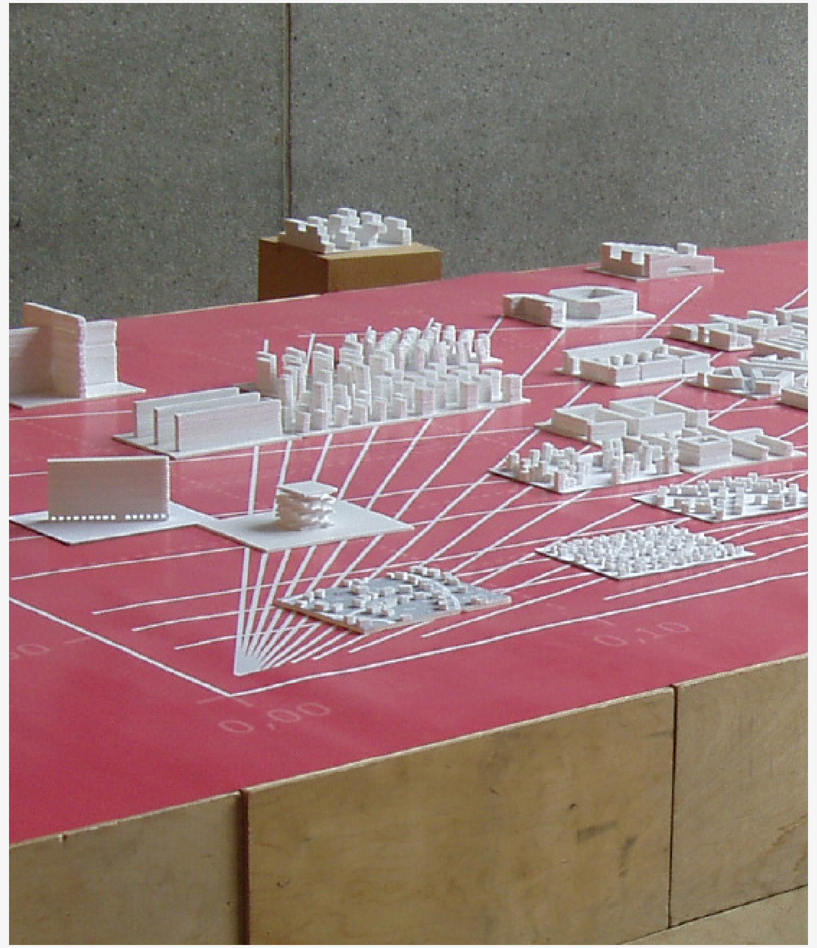
Figure 15: Density and urban form relations

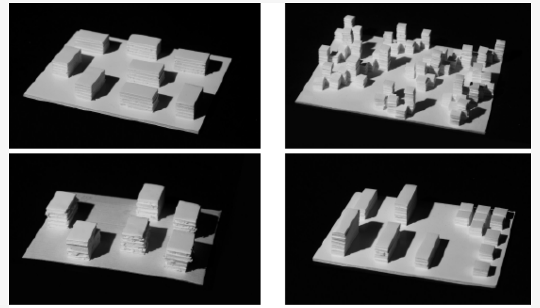
> Urban structures encompass various types of urban environments and built forms that constitute a city or town. These may include different types of neighborhoods, streets, buildings, and public spaces, along with their arrangement and functional characteristics. Examples of urban typologies include:

- Residential areas typically consist of houses and apartment buildings designed for human habitation.
- Commercial districts typically comprise stores and offices designed for work and shopping purposes.
- Industrial zones are generally composed of factories, warehouses, and other industrial buildings designed for manufacturing and other industrial activities.
- Mixed-use areas combine different land uses such as residential, commercial, and industrial.
- Historic neighborhoods are characterized by a concentration of buildings and structures of historical or architectural significance.

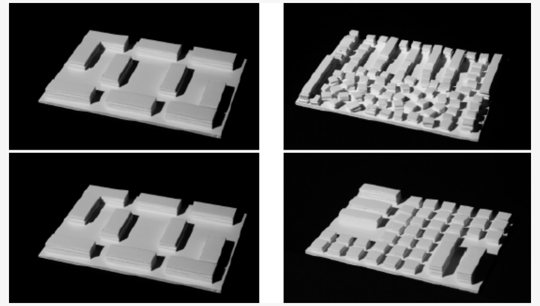
This section aims to investigate the relationship between density and urban form through the application of physical density measurement methods. By examining the primary impacts of diverse density conditions on the built environment, it seeks to explore potential variations that may arise in urban design. In this research, Urban textures are categorized into three distinct morphological archetypes: point, block, and stripe development. These archetypes are further subdivided into low, medium, and high-intensity conditions, resulting in a total of nine morphological categories.

These morphological types offer a conceptual framework for comprehending and categorizing various urban forms based on their spatial characteristics. Each type exhibits distinct building arrangements, densities, and connectivity patterns, which can significantly influence the overall character and functionality of an urban area. Through the analysis and classification of urban forms according to these typologies under appropriate density conditions, valuable insights can be gained regarding the spatial organization and structure of cities. This, in turn, informs decision-making processes related to urban design and land use planning.





The models represent different urban form characteristics under a specific floor space index (FSI) of 1.0 and a gross site index (GSI) of 0.2.



The models represent different urban form characteristics under a specific floor space index (FSI) of 1.0 and a gross site index (GSI) of 0.4.

Figure 16: Models generation workshop conducted at TU Delft 2004, to evaluate and explore various density criteria

... 2.2.1. Morphological Classification

> The research employs morphological classifications such as point, strip, and block types to describe distinct urban forms or building layouts. These classifications serve as a means to categorize and analyze various urban form typologies based on their density, building composition, and spatial organization.There may exist hybrid forms that defy clear categorization and exhibit characteristics that blur the boundaries between these classifications. The research elucidates the urban fabrics that is described by these classifications as follows:

- The Point & Cluster type is characterized by a central focal point or cluster, around which buildings are densely grouped.
- The Strip classification refers to a linear configuration of buildings or structures, typically elongated in shape. These developments commonly align the buildings along a street or open space, resulting in a continuous line or elongated cluster formation.
- The Block type, signifies a compact and enclosed arrangement of buildings, often forming a square or rectangular block shape. Block developments frequently feature a well-defined perimeter, with buildings surrounding an inner courtyard or open space.

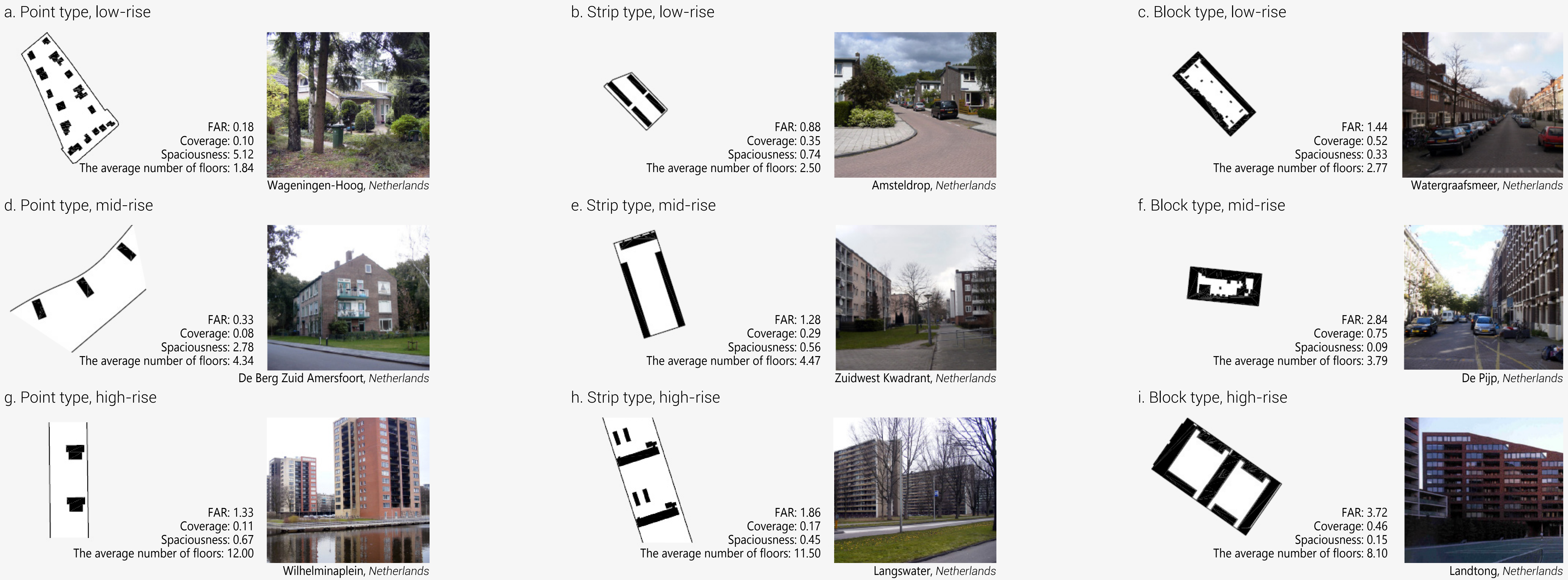


Figure 17: Morphological classifications and density relations

... 2.2.2. Examination of existing urban forms

> This section investigates spatial configurations and their form's effect on densities within urban areas by examining various city blocks through an analysis of their Floor Area Ratio (FAR) and coverage values. This part of the research uses a research ' 50 urban blocks' which is a collection of 50 urban block designs, including building height, coverage, floor area ratio value, and detailed measurement of the built area and occupied floor(A+t Architecture, 2017). A selection of existing urban forms are visually compared based on their Ground Area Ratio (FAR) and coverage (GSI) values, drawing upon the information derived from an examination of 50 urban blocks. The urban forms are classified into categories of poor, moderate, intense, and extreme based on their respective FAR values. This classification enables a systematic visual comparison, evaluation of the blocks and also creates a foundation library for generating urban forms in the design chapters.

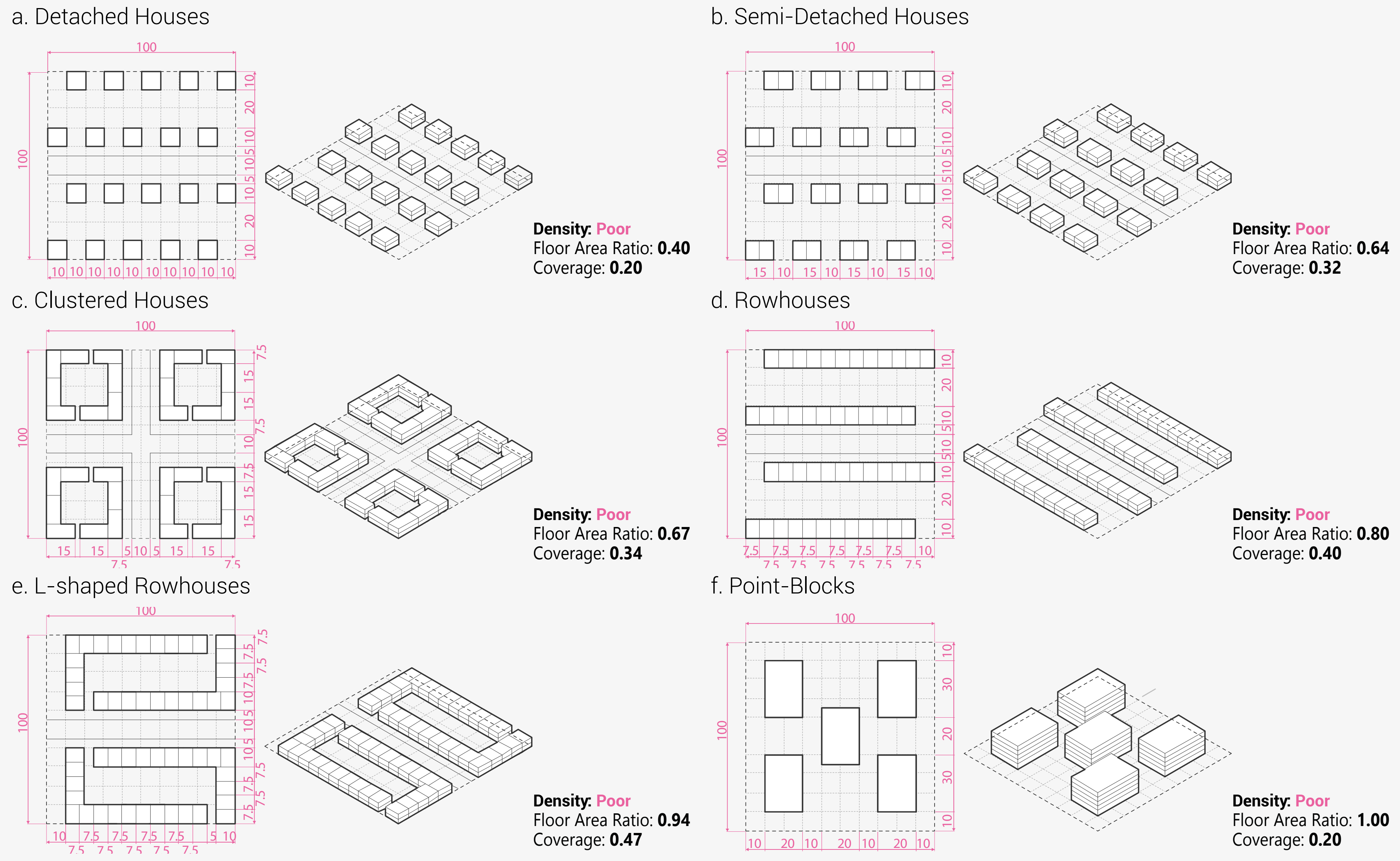


Figure 18: Urban typology toolkit

... 2.2.2. Examination of existing urban forms

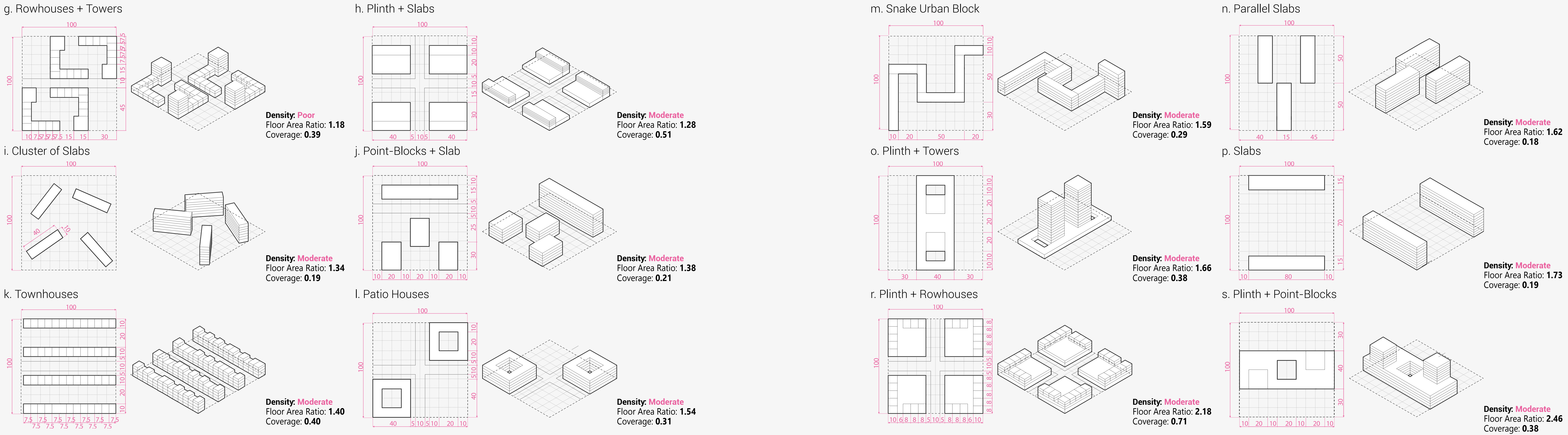
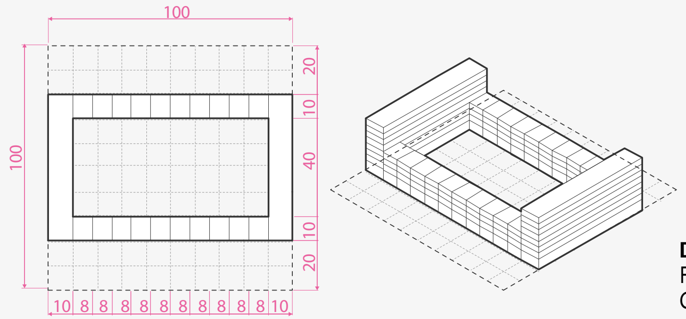


Figure 18: Urban typology toolkit

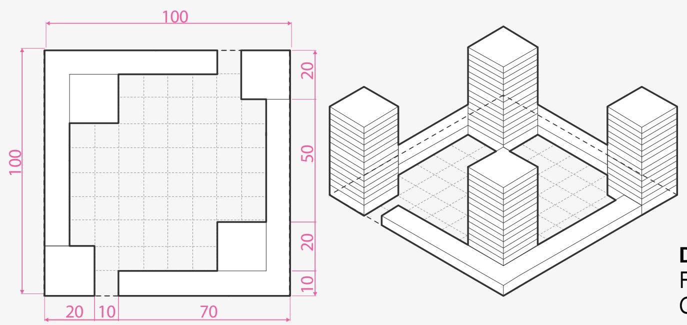
... 2.2.2. Examination of existing urban forms

t. Hybrid Urban Block



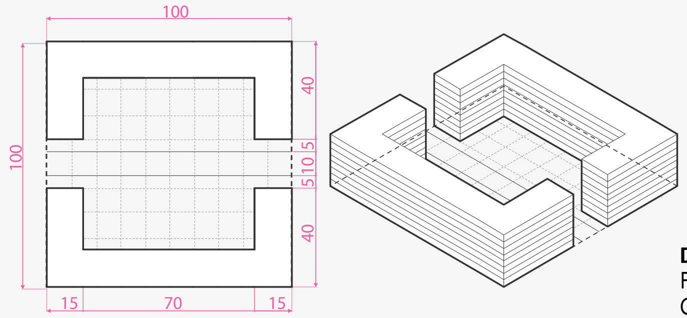
Density: Intense
Floor Area Ratio: **2.52**
Coverage: **0.42**

u. Open Block + Towers



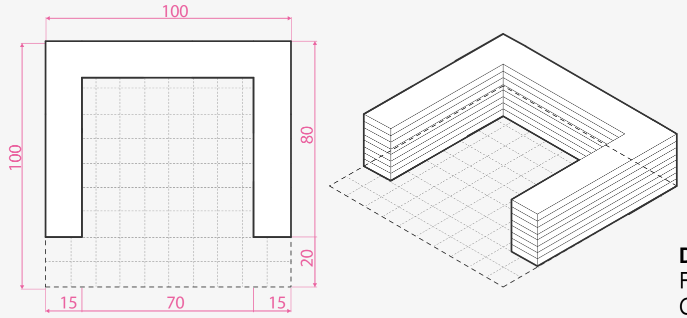
Density: Intense
Floor Area Ratio: **2.65**
Coverage: **0.57**

v. Half Urban Block



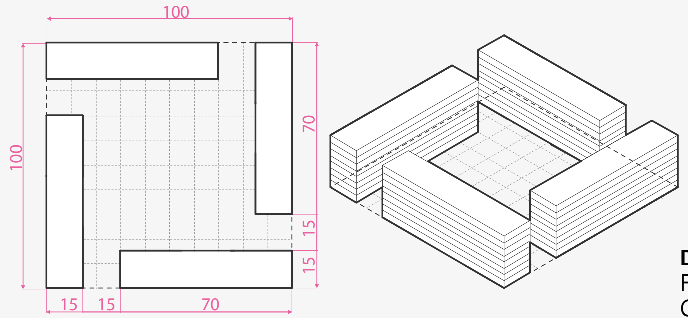
Density: Intense
Floor Area Ratio: **2.80**
Coverage: **0.35**

w. 1/4 Open Urban Block



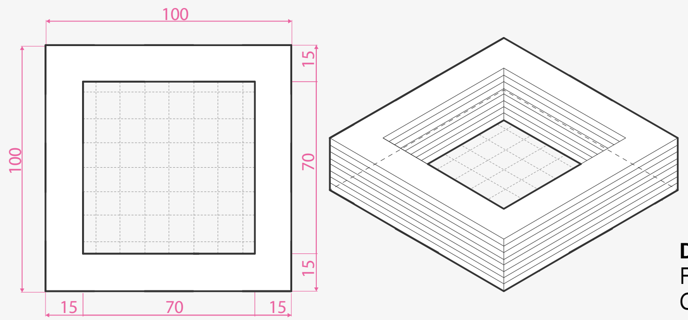
Density: Intense
Floor Area Ratio: **2.88**
Coverage: **0.36**

x. Open Urban Block



Density: Intense
Floor Area Ratio: **3.36**
Coverage: **0.42**

y. Closed Urban Block



Density: Extreme
Floor Area Ratio: **4.08**
Coverage: **0.51**

Figure 18: Urban typology toolkit

... 2.3. Density & Performance

... 2.3.1. Nine Criterias for Livable Urban Density: Soft City

> Conventional density measurements such as Floor Area Ratio (FAR), such as size or quantity, and may not provide a comprehensive understanding of the success of higher-density urban forms. To adequately assess the performance of high-density urban environments, a more intricate and comprehensive approach is required, utilizing qualitative criteria. It is essential to examine how the built form supports the everyday lives of residents and contributes to their overall well-being. A crucial aspect of this evaluation process should center on the relationships established between the built form and its surroundings.

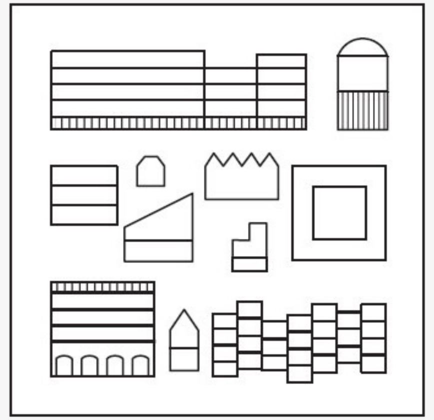
Livable density refers to the delicate equilibrium between density within a specific area and the quality of life experienced by individuals residing in that vicinity. Achieving livable density entails meticulous urban design, efficient utilization of space, and the provision of adequate infrastructure and services to cater to residents' needs. The objective is to forge vibrant, sustainable, and inclusive cities that afford a high standard of living to their inhabitants.

Livable density facilitates the efficient utilization of land within urban areas. By accommodating a higher concentration of people and activities within a confined space, it curtails urban sprawl and fosters the establishment of compact, walkable neighborhoods. Furthermore, higher-density environments foster social vitality and interaction among residents. Livable density brings individuals into closer proximity, nurturing opportunities for social connections, community engagement, and the formation of support networks.

"Soft City" explores the correlation between well-designed density and an enhanced quality of life in urban environments by examining various urban spaces and innovative architectural approaches. To evaluate and understand densities by performance, this section focuses on nine criteria for livable density in Soft City; diversity of built form, diversity of outdoor spaces, flexibility, human scale, walkability, a sense of control and identity, a pleasant microclimate, smaller carbon footprint and greater biodiversity(Sim, & Gehl, 2019).

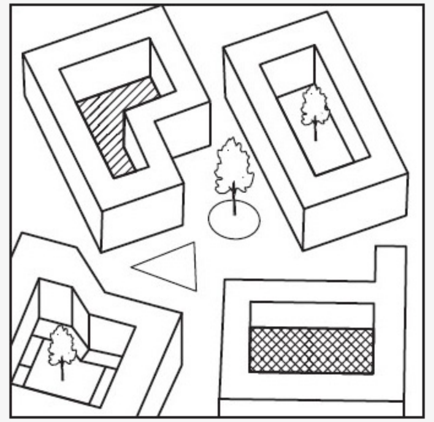
a. Diversity of Built Form

> Urban forms should support the convenience of daily life arising from the proximity of different activities. Living, working, and learning in close proximity to one another enable a more localized way of living. The use of spatial variations in building designs is significant for both functionality and visual diversity, as it can enhance the user experience and facilitate activities such as wayfinding and walking by increasing the recognizability of neighborhoods or streets. Therefore, considering different building types, uses and scales in urban planning and architectural design is a crucial step toward creating more sustainable and livable communities.



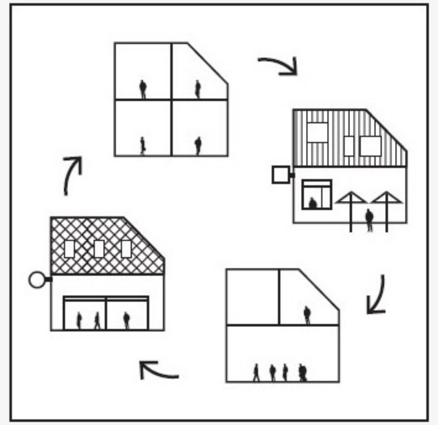
b. Diversity of Outdoor Spaces

> The outdoor spaces in cities play a significant role in expanding residents' living areas, facilitating diverse activities, and creating various gathering places. The more diverse the spatial allocation in cities, the greater the potential for different activities to take place. Making time spent outdoors easy and enjoyable facilitates connections between individuals and their surroundings. These factors, when associated with physical activity and social interaction, can contribute to improved physical and mental well-being. Also, walkways, sidewalks, and squares are important areas for people to gather, have conversations, participate in events, and foster community solidarity. Streets can strengthen neighborhood relationships and support community bonds.



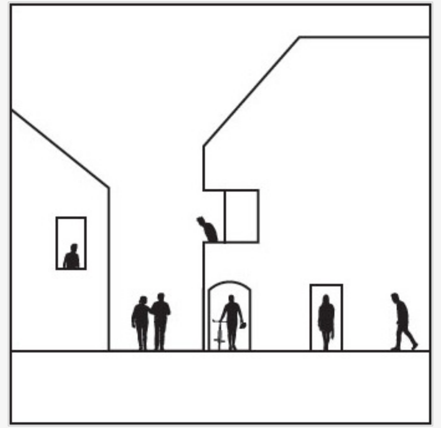
c. Flexibility

> The urban form of a city or settlement needs to exhibit flexibility to effectively accommodate the evolving dynamics of the community. Urban areas are subject to diverse changes in demography, economy, and social fabric, necessitating prompt adaptation. A flexible urban form, capable of responding to these changes, is integral to ensuring sustainability, usability, and livability. It can adjust the housing stock or convert existing structures for different uses in response to population fluctuations. Flexible spaces can swiftly respond to short-term changes and serve multiple purposes. This enables more effective and efficient utilization of spaces and the ability to meet the demands of diverse user groups.



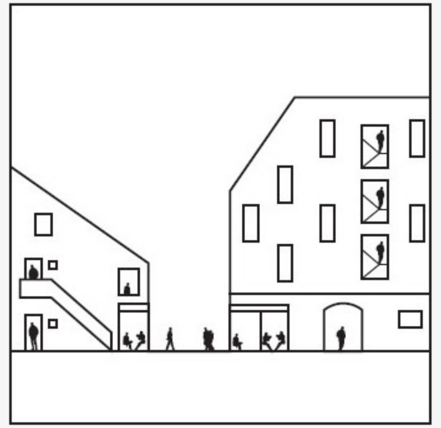
d. Human Scale

> Creating urban spaces that consider human needs aims to construct sustainable, livable, and human-centered cities. This approach seeks to establish spaces where individuals can meet their fundamental needs, in a safe, accessible, diverse, and socially interactive manner. Human scale is based on dimensions that are in harmony with human senses and behaviors, which entails smaller building elements and lower building heights. Eye-level continuity maintains visual and auditory experiences by connecting buildings. Spaces designed with consideration for human scale are ideal for individuals to spend time comfortably and safely, as well as interact with others. Such spaces foster social interaction by bringing people closer together and providing psychological relaxation.



e. Walkability

> Walkable cities aim to make public spaces accessible and usable for everyone and encourage individuals to engage in physical activity. Places where active modes of transportation such as walking, cycling, and public transit are preferred lead to increased physical movement, adoption of a healthy lifestyle, and contribute to the prevention of health issues. Promoting active transportation methods reduces air pollution and greenhouse gas emissions by reducing motor vehicle traffic, thus contributing to environmental sustainability goals. Also, it fosters interaction among people and strengthens social connections. Pedestrian-friendly streets, squares, and public spaces facilitate gatherings, participation in activities, and the development of community life.



f. A Sense of Control and Identity

> The urban form should consist of physically defined areas controlled by an individual or a group. These areas offer opportunities for customization and personalization. This ability to shape and personalize the immediate surroundings contributes to a sense of control and agency over the space, enhancing the overall sense of identity. These areas, which are defined as edge zones, refer to the transitional area between the public realm and private buildings in urban environments. It serves as a buffer between the interior of buildings and the external urban environment, facilitating interaction. Residents and users can express their unique identities through the design, and use of their respective edge zones.

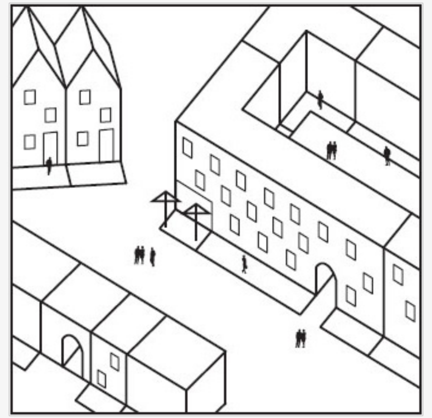
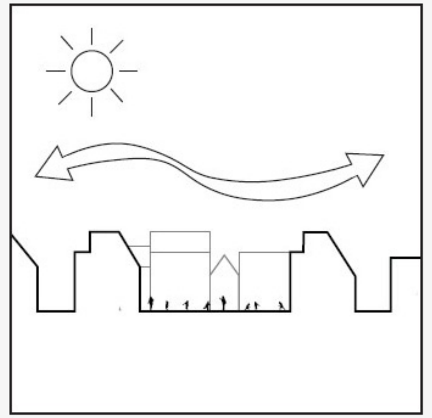


Figure 19-25: Soft City criterias

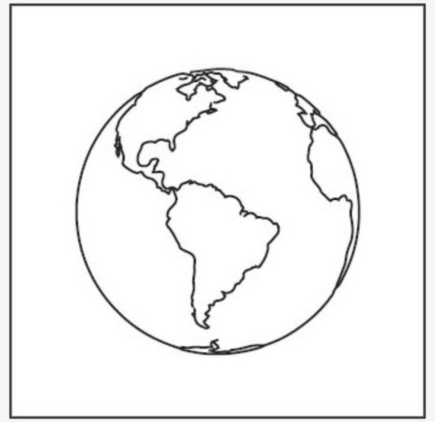
g. A Pleasent Micro-climate

> Designing cities based on the conditions brought by the microclimate is of great importance in improving public life, promoting outdoor activities, and encouraging sustainable behaviors. Cities that are designed taking into account the elements of the microclimate aim to integrate urban dwellers with the built environment, despite the conditions imposed by their climate, and foster a public life that encourages walking, cycling, and spending time outdoors. Overall, by prioritizing the creation of a pleasant microclimate throughout the urban fabric, starting right outside people's homes, it becomes possible to establish an environment that promotes active mobility, encourages outdoor activities, and enhances the overall quality of life in the neighborhood.



h. Smaller Carbon Footprint

> Environmentally friendly urban design is primarily achieved through energy savings obtained from daily activities in a walkable neighborhood. A walkable neighborhood refers to a pedestrian-friendly environment where people can access their daily needs without relying on cars. In a walkable neighborhood, individuals can walk or cycle to their workplaces, schools, parks, shops, and other daily services. This reduces car usage, resulting in lower fuel consumption and greenhouse gas emissions. Additionally, a more compact city layout allows for buildings with fewer surfaces that are exposed to solar radiation, reducing construction costs and, over time, individual building heating and cooling expenses.



i. Greater Biodiversity

> Vegetation contributes to the creation of natural habitats and ecosystems in cities. Trees, shrubs, grasses, and other plants provide habitat and support biodiversity and allow birds, insects, and other organisms to be part of the natural balance in urban areas. Moreover, vegetation aids in enhancing privacy by serving as a visual barrier around buildings and gardens. Trees and shrubs can mitigate traffic noise, urban sounds, and other intrusive noises, creating a more peaceful atmosphere in living spaces.

The design options in Chapter 1, aim to create sustainable cities that prioritize community well-being, foster diversity, and minimize negative environmental impacts in light of the principles of the soft city concept and the importance of density criteria. These performative density strategies explored in this chapter, contribute to the realization of more livable, inclusive, and environmentally conscious urban forms and environments.

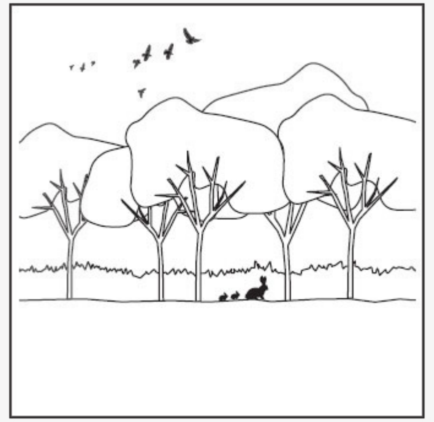


Figure 19-25: Soft City criterias

chapter 02

... P-02 ASSESSMENT OF URBAN DESIGN & ANALYSIS TOOLS

> The next part of the chapter 2 investigates the potentials, limitations, and effectiveness of generative design tools in urban design and architecture. Brains Digital company serves as a collaborator in this section of the thesis, providing valuable insights and contributions from their industry perspective.

> There has been many technologies, techniques, and concerns emerging in construction industry since the integration of computer. All these improvements have meant that the complexity of designing and building a project has increased more than ever before. Creating a project from scratch would mean that there is a need for more resources and many more people who bring different new aspects to consider and address. This chapter of the thesis, with the help of computational design, is going to tackle the sustainability aspect of this process, perhaps the most urgent aspect among many industries including the construction industry.

Design tools have followed the trend as they evolved around the designer's needs. One of the most interesting inclusions to a workflow of an architect could be programming skills. Burry demonstrated programming process influence upon work of the design studio(Burry, 1997).

A programmatic version of the design process is a cyclic process that gets evolved and is modified until the solution, unlike a rather linear architectural design process(Boeykens, & Neuckermans, 2013). Programming, algorithms and scripts allow designers to overcome the limitations of traditional CAD and 3d modelling. It helps designers reach a level which is beyond the human manual ability or takes an enormous amount of time(T-

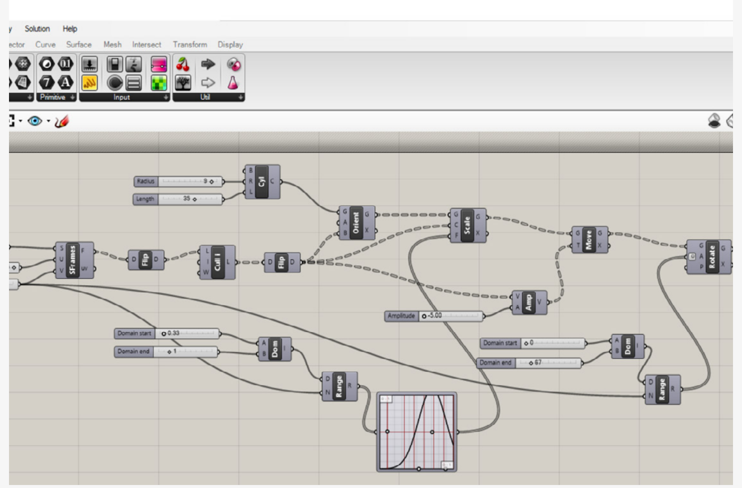


Figure 26: Grasshopper interface

edeschi, & Lombardi, 2018). This also is the case for designing sustainable buildings because of its complexity in design decision making. It is now possible to simulate real-world conditions in software to calculate necessary analysis to improve the design.

This chapter will be diving deep into how can environmental analysis can be implemented in a design computationally. Since it is a very broad topic, the chapter will be focusing on the Urban scale. More specifically it is going to consider the project site itself and it is immidiate surroundings, surroundings that can physically alter the site and vice-versa. It is going to investigate how to enhance the generative tool by focusing on the comfort aspects of a project, having a deeper look at the realm of environmental analysis and implementing the characteristic features of the context to the core of the design. The design will be enhanced with the quantitative support from environmental analysis. The ability to perform which analysis or simulation would still be dictated by the designer itself as different contexts may require different concerns.

In the construction industry, the effects of the nearby buildings on a project are something that is not considered on a deep level when designing, the possible effects of the buildings to its nearby buildings are considered even less. The research of already existing generative tools in the market shows that there is more attention that's being paid to automatization of repetetive processes such as creating car parking instantly or mass modelling. However, the focus has not been on environmental issues. The chapter will try to study and improve this aspect of the design process and improve the tool.

There are certain regulations that try to create this consideration by creating limits for designers. An example could be not being able to build after passing a certain height. But these regulations have existed many years before and are in need of deepening for the sake of environmental comfort. The chapter is going to try to create more complex and well-thought limitations. It is also going to be discussing how this information can be computationally used to design the site, creating tangible parameters and constraints for a well-educated design proposal in terms of environment and comfort.

The improvement on this tool is also designed to be adaptable to every situation and every location that is given. It is going to be able to extract the location information given and create analysis depending on it. Given the location information and weather data file, it will be able to quickly create the context of the site and extract weather data, ready to use for analysis.

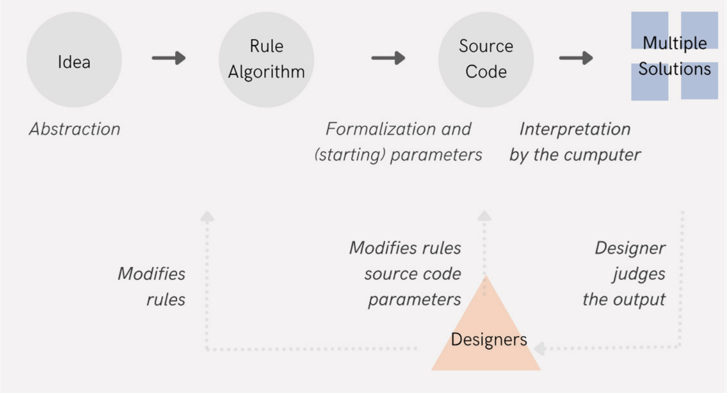


Figure 27: Generative design workflow

... 2.4.1. Enhancing Environmental Analysis at the Urban Scale

> It is possible to predict many environmental aspects of a given location such as the location of the sun, wind direction/speed, temperature, humidity, precipitation and so on. With this data, it also is possible to create relevant analysis regarding the specific requirements of the project and this can help the process of the design tremendously. Different type of context requires a different type of approach to the site, a different type of tools and a different type of building code. This leads to spending a lot of time trying to adjust and adapt to each project individually. The computational tools perhaps can not replace the designer but they can certainly assist the designer in this regard by giving them a headstart. They can do that by giving them a framework to work with and not letting them start from scratch on a project. The moment a designer would start a design, they would already have presented the sustainability convers of the context demonstrated in a clear and communicative way so that they can have a few ideas to start with their design while considering sustainability.

The designers can now get instant feedback with their project in the early stages of the design. This enables designing more environmental aware projects since the concerns are already issued at the beginning phase of the architectural process and will save time and money. If these concerns were handled towards the later stages of the process, some decisions would have been already taken and it would be diffcult and time consuming to return to the fundamentals of the project to achieve to the desired outcome.

The conventional design process tends to rely on trivial and not precise methods when it comes to environmental sustainability. Computational methods could possibly provide an alternative through quantitative and interactive means. Existing conventional quantitative approaches for environmental design, on the other hand, have the problem of language; usually, they are done with Excel and is difficult to read the information in a clean and efficient way.

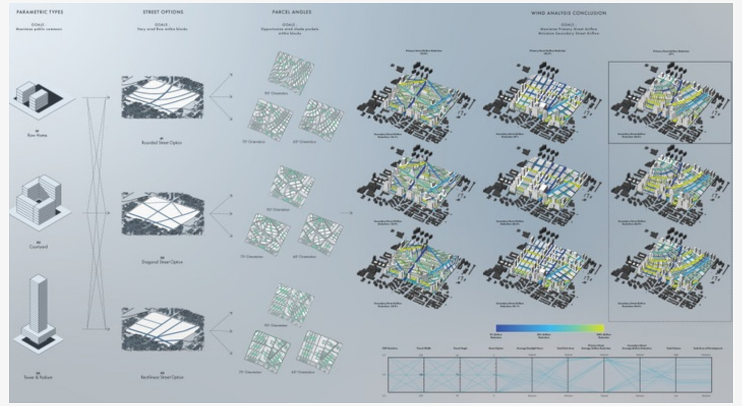


Figure 28: Computational design practices- columbia GSAPP

In the paper first theories and methods will be introduced. What type of tools are required, what type of information is needed and how to make it work for environmental analysis with computational design will be explained. Then the explained things will be put to the test through simulations with different working possibilities. Sun studies, view analysis and radiation analysis will be demonstrated. A case study project will be utilized and the found scripts will be performed on the site to determine its feasibility. Since it is possible to get results not only with visuals but also with numbers, it is possible to try many options and find the most optimal massing options with optimization. For analysis to work, there needs to be massing proposals, so the scripts will be used along with the tool proposed to get massing options to work on.

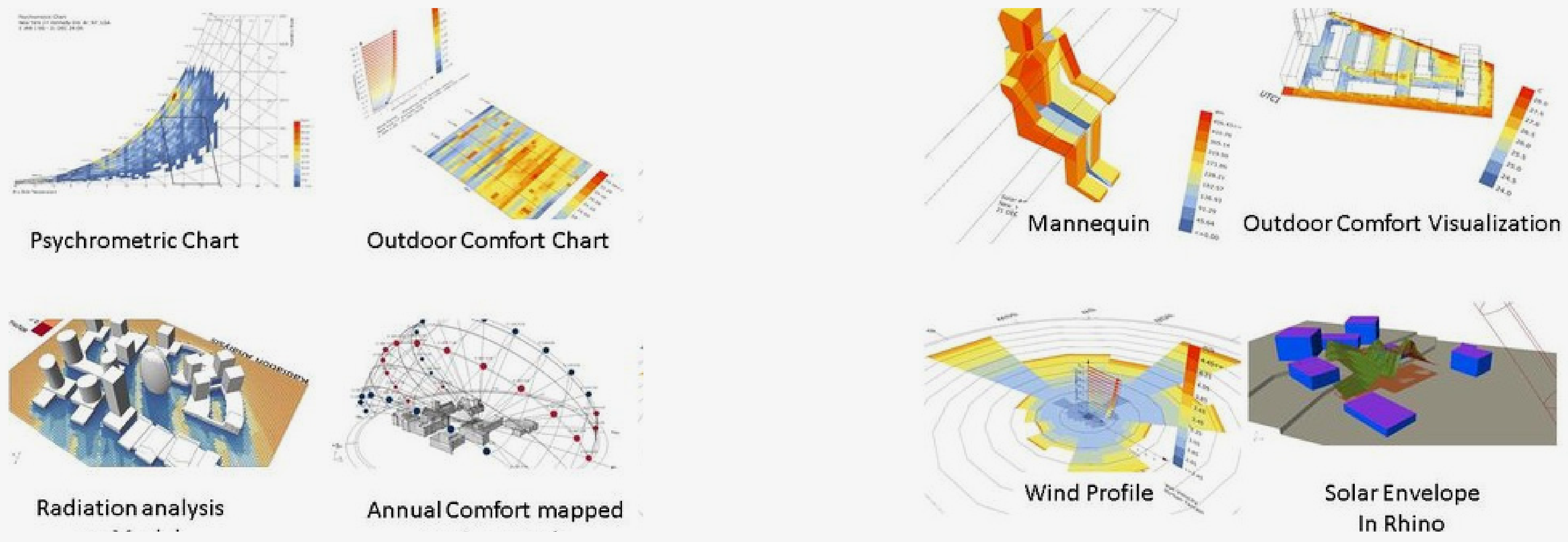


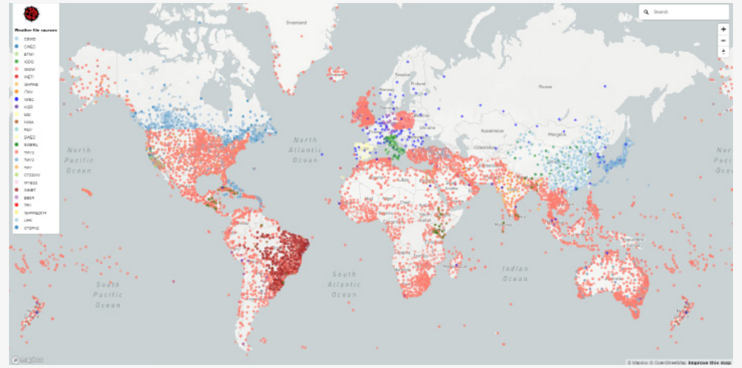
Figure 29: Various environmental analysis

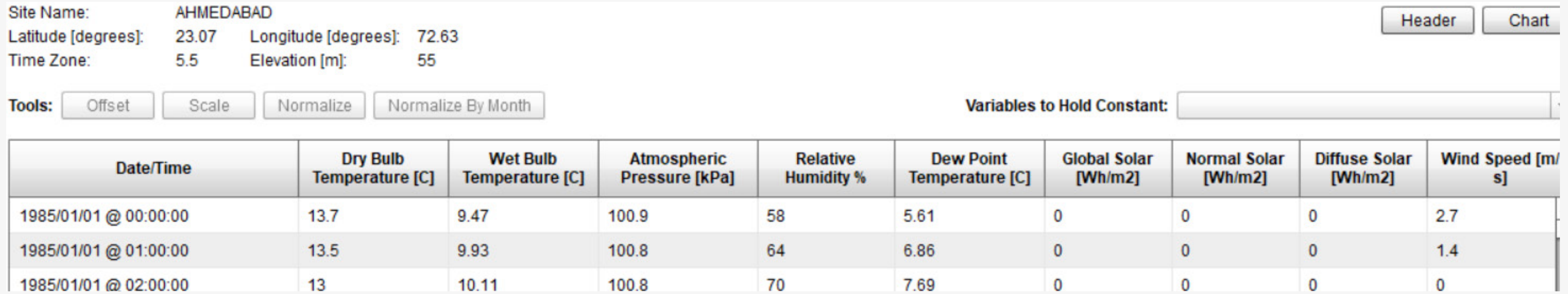
> As with any project, the first step in conceptualizing a building from an environmental standpoint is to begin studying the climate and site characteristics, and Ladybug is the plugin that allows, as previously stated, to study and show weather data. A weather data file is type of a file that contains various information regarding to the location it is analysed from. It contains weather information such as temperature, humidity, wind speed/direction and so on. This type of data is always required in order to conduct the analysis. One of the most commonly used weather data files is .EPW files. The EPW weather data format was created for use with two major simulation programs, EnergyPlus and ESP-r and has now been adopted as a standard format by a number of other building simulation tools(Sadeghipour, & Pak, 2013).

EPW files are generally associated with EnergyPlus by the United States Department of Energy. EPW file extensions are classified as data files that are related with the EnergyPlus energy simulator software. It is an open-source building energy simulation program in its entirety. This file format, established by the US Department of Energy, is accessible for commercial and academic use in executing simulation operations for location-specific weather data. The EPW files include the essential weather data for architects, engineers, and researchers to utilise for modeling electricity and water use.

The standard EnergyPlus format weather data files contain data for over 2100 locations in 100 countries around the world. These files are a free climate data library for building performance simulation. This database contains a huge amount of data gathered by data scientists and data collectors throughout the dacades for regions all around the world excluding Antartica.Thus, it is usually possible to find a location that one searches. Within each EPW weather file, the first eight lines or header establish fundamental location information such as longitude, latitude, time zone, elevation, yearly design conditions, monthly average ground temperatures, typical and extreme periods, holidays/daylight saving periods, and data periods included. There is also room for users to document any unique features or information about the file, such as data sources. Typically, the header is followed by 8760 lines of data, one line for each hour of the year.

Considering global warming and other factors, the fact that the epw file is up to date is relevant as well; an epw map that is dated in the 1970s will lead to different results than an epw file that is created during the 2010s. The case study is located in Milan therefore an epw file taken from Linate Airport between 2007-2021 will be conducted for the analysis. The most popular and typical location to find these files is on the official energy plus website at https://energyplus.net/weather. A world map indicates all of the locations that have this type of document, and a link is provided to immediately download them. Alternatively, the Ladybug Tools have an interactive website for epw files as well where it is possible to choose a location from a world map and copy the link to the epw file and paste it to the Grasshopper without having to download the files.





Date/Time	Dry Bulb Temperature [C]	Wet Bulb Temperature [C]	Atmospheric Pressure [kPa]	Relative Humidity %	Dew Point Temperature [C]	Global Solar [Wh/m2]	Normal Solar [Wh/m2]	Diffuse Solar [Wh/m2]	Wind Speed [m/s]
1985/01/01 @ 00:00:00	13.7	9.47	100.9	58	5.61	0	0	0	2.7
1985/01/01 @ 01:00:00	13.5	9.93	100.8	64	6.86	0	0	0	1.4
1985/01/01 @ 02:00:00	13	10.11	100.8	70	7.69	0	0	0	0

Figure 30: Epw data and properties

... 2.4.2. Environmental Analysis with Visual Programing

> Two of the most used software in the construction industry are Rhinoceros by Robert McNeel & Associates and Revit by Autodesk. Along with their main software, they also come with their visual programming tool, Grasshopper and Dynamo respectively. A visual programming language in computing is any programming language that allows users to develop programs by manipulating program elements graphically, rather than textually stating them. Because of their comprehensive features and wide range pool of plugins, Rhinoceros and Grasshopper, have been extensively used in this chapter. This type of choice is very suitable for this kind of work since it is possible to analyse and describe different aspects of the building as well as observe how the different features work together and if the expected results are promising or not.

In Grasshopper for environmental analysis, there are many plugins that allow creating environmental analysis on different scales spreading from city-scale to room-scale. Some of them work as a bridge to other environmental software, and some of them lets the possibility of doing calculations directly from Grasshopper. Before Ladybug, one of the most popular plugins for environmental analysis was GECO. As a plugin, it was working like the latter, you can import the data and the geometry to Autodesk Ecotect, allowing a direct workflow between the two software, making environmental analysis of the project feasible. Once the process is finished, all the data are pushed back to Grasshopper and are used to give information to geometry(Tedeschi, 2014).

However, Autodesk announced in 2015 that support for the Ecotect Analysis software would be phased out by the end of the year since such assessments would be performed within Autodesk Revit. Ladybug Tools was quickly and effortlessly established as the new reference point in this field. Ladybug Tools is a free Grasshopper plugin suite that works with free environmental and energy analysis tools (in a similar fashion to GECO). The difference between them is that Ladybug Tools is open source. This means that users can access the coding behind the plugin and either read or edit the content inside.

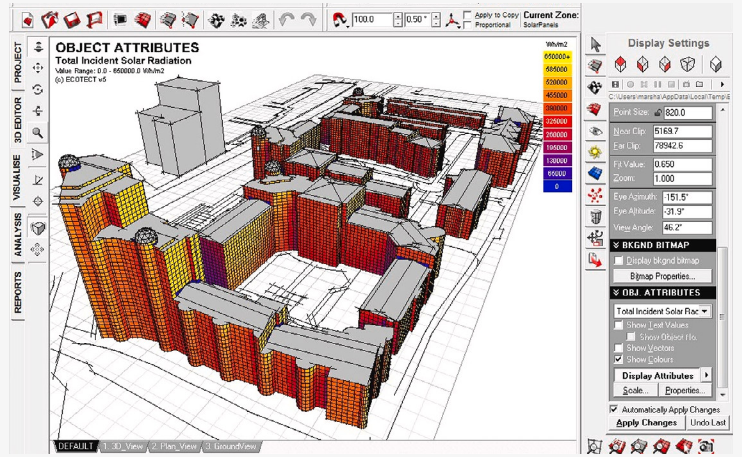


Figure 31: Autodesk Ecotect

This creates many possibilities for access and use of the plugin. Also, unlike GECO, it works inside the Grasshopper to contact analysis rather than creating a bridge between the Grasshopper and another software.

Ladybug Tools is a collection of four plugins: Ladybug, Honeybee, Butterfly, and Dragonfly. Ladybug conducts climate analyses using weather and meteorological data (EPW files). Only Ladybug was used for this study since the other plugins had more specialized uses. It can create 2D and 3D interactive graphs from the data provided, assisting decision-making at the project's concept phase. It's also feasible to directly link the result of the analyses to inform geometry and optimize it, thanks to the modular structure like with the other plugins in this suite(Crawley, Hand, & Lawrie, 1999).

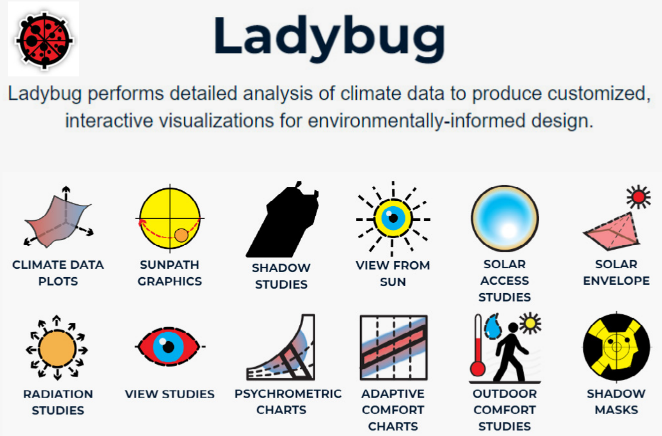


Figure 32: Ladybug components

Ladybug also assists with the evaluation of preliminary design ideas using solar radiation studies, view analysis, sunlight-hours modeling, and other methods. Integration with visual programming environments such as Grasshopper enables real-time feedback on design changes and a high level of flexibility.

Along with clear visual results, every analysis conducted also provides numeric data as an output, either as a total number or a number for each point of analysis. With these numeric results, it is possible to do optimization and try many design options and choose best performing ones. The amount of design options to be tested can be choosen by the user. However, the more design options the longer the process takes to finish. Also factors such as the computer performance and quality of analysis done should be considered as they influance how long the process lasts greatly.

... 2.4.3. Environmental Analysis Methods

> View analysis in Ladybug Tools allows designers to assess the quality of views from different vantage points. It helps evaluate visibility, openness, and scenic qualities in building and urban environments. One aspect of view analysis is calculating viewsheds. It determines the visible area from a specific location, considering obstructions and terrain. This helps identify obstructed views and optimize window and design element placement. Additionally, Ladybug Tools evaluates view blockage in urban settings, finding views that are impeded by neighboring structures. Using this knowledge, designers can preserve significant views by strategically placing buildings.

> This analysis allows you to compute the amount of radiation that falls on any surface, such as a facade, a roof, or the ground. This form of radiation is also helpful for surfaces such as windows, where you may be interested in solar heat gain, or solar panels, where you may be interested in energy collection. Radiation is a significant aspect of measuring outdoor thermal comfort since it is the primary generator of thermal variability across the urban area during the day. As a result, average radiation can be used to estimate the coolness or warmth of a specific microclimate during the day.

> Direct sunlight analysis, a fundamental aspect of daylighting studies, serves the purpose of calculating the intensity and distribution of direct sunlight that illuminates surfaces, particularly in outdoor environments. This analysis holds great significance in the design process, as it directly influences both the thermal and visual comfort aspects of a space. This knowledge can be used to improve outdoor space and create more comfortable environments for users. Designers can identify locations prone to excessive heating or shading and apply suitable shading or landscape elements to alleviate thermal discomfort by analyzing the distribution of direct sunlight.

These analyses aid in identifying the presence of favorable or unfavorable areas concerning visual and thermal comfort within the design of an outdoor space. These analyses assist in identifying areas within the design of an outdoor space that either enhance or hinder visual and thermal comfort. When visual comfort is poor, it can make it difficult to function and reside in the space. However, when visual comfort is optimal, it can greatly improve mood, and efficiency, and even enhance the value of the house economically. On the other hand, well-designed thermal comfort enhances occupants' well-being, energy efficiency, and comfort, while poor design leads to discomfort, inconvenience, energy wastage, and increased expenses. Using Ladybug tools, conducting these analyses has become remarkably simple and time-efficient, requiring just a single click and a few seconds to obtain comprehensive results. This streamlined process empowers designers to swiftly evaluate visual and thermal comfort, facilitating informed decision-making and efficient optimization of outdoor space design.

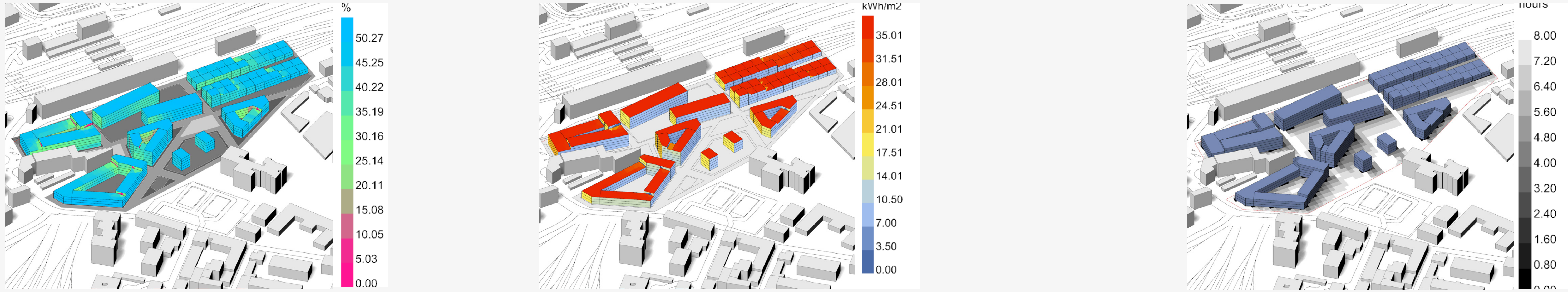


Figure 33: View range, solar radiation and annual ground shadow analysis with grasshopper

... 2.4.4. A Comparative Analysis of Generative Architectural Design Software

> In recent years, the field of architecture has witnessed a revolution with the emergence of generative design tools and advancements in cloud computing. These developments have opened up new possibilities for exploring innovative design solutions. This part aims to investigate the feasibility of various generative architectural design cloud websites for practical applications and assess their potential impact on the design process.

The research focuses on conducting a comparative analysis of selected generative architectural design software and cloud websites, with an emphasis on examining their capabilities in terms of input data, performance, and output data. By dividing the evaluation into these three key aspects, a comprehensive understanding of the strengths and limitations of each platform can be achieved. The primary objective of this research is to explore the suitability of the selected tools for real-world case studies. The functionality, usability, and performance of these platforms are examined to gain insights into their potential to support designers in their creative endeavors. Moreover, this research is motivated by the development of a new tool. Insights gained from the strengths and weaknesses identified in existing generative architectural design cloud websites inform the design and development of this new tool, incorporating the lessons learned from these platforms.

To achieve the objectives of this study, a series of tryouts was conducted on the selected generative architectural design cloud websites. The tryouts were divided into three main parts: input data, performance, and output data. The input data section focused on assessing the ease of importing and manipulating architectural data into the generative design platforms. Evaluation included examining the compatibility of different file formats, the efficiency of data processing, and the user-friendliness of the importation process. The performance section involved evaluating the computational capabilities and responsiveness of the generative design tools. Assessment criteria included the speed, accuracy, and stability of the platforms in generating design alternatives based on user-defined parameters.

Lastly, the output data part investigated the quality and diversity of design outputs generated by each platform. The subsequent sections of this study will provide an overview of the selected generative architectural design cloud websites, analyze their features, compare their functionalities, and present conclusions based on the research findings. Furthermore, the discussion will explore how the insights gained from this study can influence the design and development of future generative design tools, highlighting the key considerations and lessons learned throughout the process.

Figure 34: Test platform logos: Digiwo, Testfit, Architecthtures and Hypar

... 2.4.4. A Comparative Analysis of Generative Architectural Design Software

Digiwo

> Digiwo is a grasshopper plugin that provides a number of potent tools for parametric design and digital fabrication While quick automation generation of a large number of diverse design options, pre-filter of the nonsensical variants and ability of the user to influence the generation flow can be counted as advantages, on the other hand, the possible exclusion of good solutions in advance and diversity of the variants depends fully on the predefined heuristics are the limitations of Digiwo. Digiwo can be highly useful for creating the massing of a site. It offers a wide range of actions that can seamlessly integrate and combine with each other, resulting in numerous design options. Additionally, it provides interesting sun studies where the massing is strategically generated.

Architechtures

> AI-powered platform Architechtures helps designers and architects with their creative processes. It creates an opportunity for the designer to have complete control over the quantitative aspects of the project in order to be able to focus on decision making and improve the added value. The platform does not primarily focus on massing studies at the master plan scale. It primarily emphasizes the concept and schematic design of individual buildings.The software does not currently offer sustainability analysis features. However, it focuses on providing solutions for building design, including interior planning considerations such as the placement of vertical and horizontal circulations, and balconies.

Hypar

> Hypar blends cloud computing with the strength of parametric modeling and automation to give users the ability to build, iterate, and optimize intricate architectural designs in real-time. Professionals can enter limitations and design criteria into Hypar, and the platform will then produce intelligent, data-driven design solutions. It offers powerful tools, such as dynamic 3D models, automated drawings, and performance simulations, for visualizing and analyzing designs. While Hypar provides flexibility in using a variety of tools for systems of a single unit, natively it does not support designs with multiple blocks or has any features for designing the accessibility to a plot or within the plot. When it comes to systems of a single unit, it has many features like core, structure, grid, façade systems and fastens their process quickly. It also provides real-time feedback based on changes instantly.

Testfit

> TestFit is a software tool for architectural space planning and feasibility studies. Based on user inputs and limitations, it generates a variety of design choices using automation and parametric modeling. TestFit offers speedy design iteration and optimization by providing immediate input on parameters like unit count and financial performance. It is a useful tool for architects and real estate professionals to save time, cut expenses, and make educated decisions because of its visualization features, which improve collaboration. When it comes to relationships to building zoning regulations TestFit takes more initiatives. Since the presets that are predefined can be easily adapted with building zoning regulations, it allows the designer to compare different typologies and effects of the design decisions rapidly. On the other hand, TestFit doesn't provide any environmental analyzes.

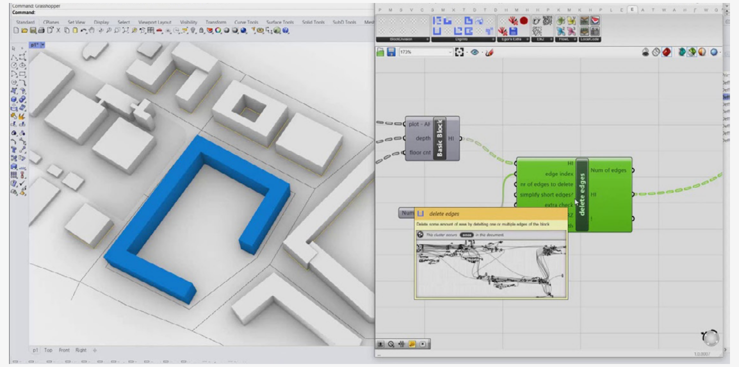


Figure 35: Digiwo and grasshopper interface

Figure 36: Architechtures cloud interface

Figure 37: Hypar and functions

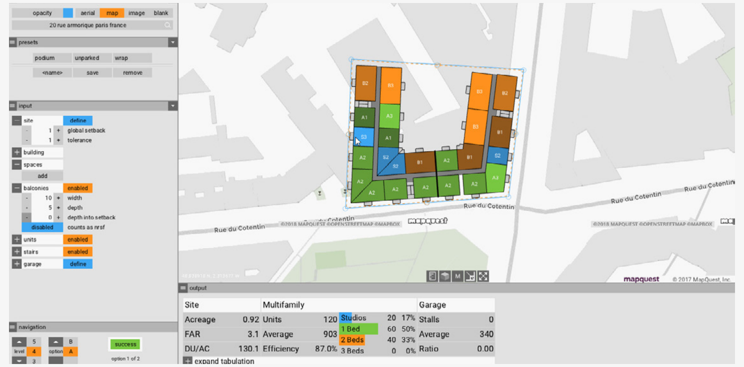


Figure 38: Testfit and plan features

... 2.4.4. A Comparative Analysis of Generative Architectural Design Software

> The comparison table of the tools are highlighted on the right side of the page. It compares the tools in terms of inputs, data types, performances, outputs and a brief comment regarding their state.

In order to ensure the successful execution of architectural projects, it is essential to address their economic, social, and sustainable aspects. However, upon evaluating existing tools like Hypar, Testfit, Architectures, and Digiwo, it becomes apparent that they may not fully encompass the complexity of the challenges at hand when considered collectively. Some of these tools prioritize ease of use but may lack the necessary depth, while others offer complexity without addressing a wide range of problems. Therefore, exploring the development of a customized tool that is more suitable for overcoming these limitations becomes a possibility.

In the upcoming chapters, we will discuss our creative design tool, which gains valuable insights from the analysis of existing tools and introduces a comprehensive approach that aims to tackle the complex challenges of the early design phase.

	Input Data	Input File Type	Performance	Output	Output File Type	Evaluation
tf	Location, and or Custom Raster Input	JPEG	Testfit can analyze the performance of "Acreage, Floor Area Ratio, Residential dwelling units per acre," and other cost components simultaneously using the tabulation section.	Site, building and parking info, model of the mass, an extended tabulation which gives a breakdown of different area takeoffs.	PDF, DXF , SKP , CS, RSD(Testfit file), GLTF(3D model)	Efficient in considering relationships with building zoning regulations. It offers various initiatives, including easily adaptable presets that align with these regulations. This enables designers to swiftly compare different typologies.
HYPAR	Outlines of the building, volume of the building and location	3dm, RVT , DXF	It can easily adapt predefined presets to align with building zoning regulations, allowing for quick comparison of different typologies and design decisions. It also provides analysis of property data from CSV files and offers basic environmental features.	Facade system, structural systems, core systems, grids etc.	IFC, JSON, glTF, RVT	Known for its accessibility and ease of sharing, thanks to its open-source system. However, it has limitations when it comes to creating multiple masses and generating a comprehensive master planand there are a few tools available for arhitectural planning.
ARCHITECHTURES	Outlines of the building, volume of the building and location	DWG	Provides detailed real-time information on land, covered area, permeable area, and cost parameters of buildings. It also offers the flexibility to switch to a manual mode when needed, allowing users to make tweaks.	Provides extensive measurements, including housing programs and general built surfaces by uses and typology. Offers unit prices for each item, facilitating cost analysis.	IFC, DWG, XLS	The main focus is on the building scale, with a particular emphasis on detailed information and analysis. However, it also offers intriguing capabilities in interior design through its machine learning abilities.
	Building geometry (curves), material information, environmental factors, design parameters	Works inside Grasshopper and Rhino, supports CAD files	Very useful in creation of massing of the site. Has different actions that can integrate and combine with each other endlessly thus creating a lot of different design options. It provides sun studies as well .	The output is a simple mass model of the building in Grasshopper / Rhino. It is possible to get data repots of performance according to the sun studies as well.	3dm, gh	Can enhance understanding of building performance and assist in making informed design decisions. Plugin itself has many prequisites like knowledge of Grasshopper and Rhinoceros as well as many other plugins to make it work. It requires commitment and not easy to use.

Table 1: Comparative analysis of software and tools

chapter 03
... CREATING MASTER PLAN

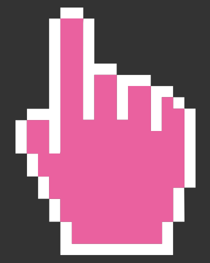
> Within this chapter the design part of the thesis is going to be introduced to reader. The chapter aims to integrate a computational creative framework to create a prototype solution for the arguments that are introduced in the first two chapters of the thesis.

design phase 01

... P-01 DECODING THE TOOL

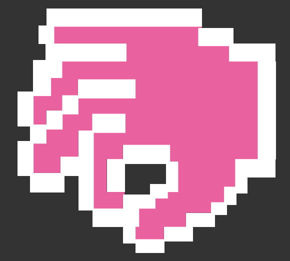
> In the following part, a comprehensive explanation is provided from a designer's point of view regarding the actions, components, and their utilization, as well as the intended outcomes achieved through their implementation. This phase aims to enrich the understanding of the processes involved masterplan designing and guide the reader through the practical application of these elements into the proposed creative framework.

Design Phase:1
... 3.1. Action Toolkit
... 3.1.1. Agents



CLICK & SLIDE AGENT

It allows user to navigate between pre-defined inputs by sliding the values or add custom integer inputs such as: offset distance, building thickness etc.



PICKING AGENT

It allows user to add custom inputs in design canvas such as: adding focal points for heatmap.

DRAWING AGENT

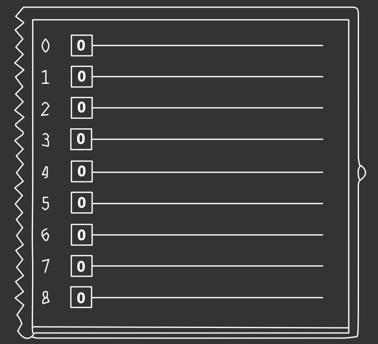
It allows user to draw 2d lines on design canvas, which allows to manipulate layout, such as: drawing public spaces.

COMPUTER AGENT

It represents the actions that are generated or manipulated by computer automatically, usualy user has a seed value to navigate between different generations/manipulations.

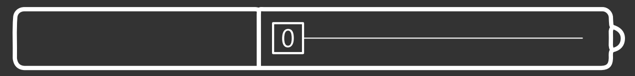
PANEL

It is a component that allows user to input any data that will be decided as constant, such as threshold angle, floor height etc.



GENE POOL

It is a component that allows user to navigate between multiple variables at the same time, such as: multiple building thicknesses etc.



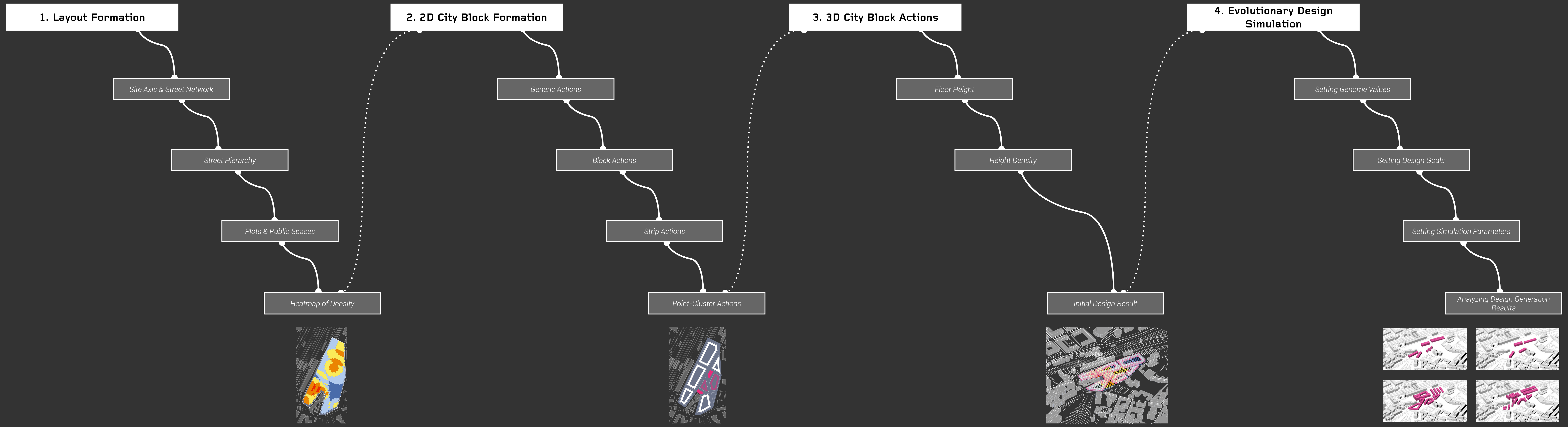
SLIDER

It is a component that allows user to navigate between single variable such as: adding or removing a design option etc.

Design Phase:1

... 3.1. Action Toolkit

... 3.1.2. Components: Keyplan

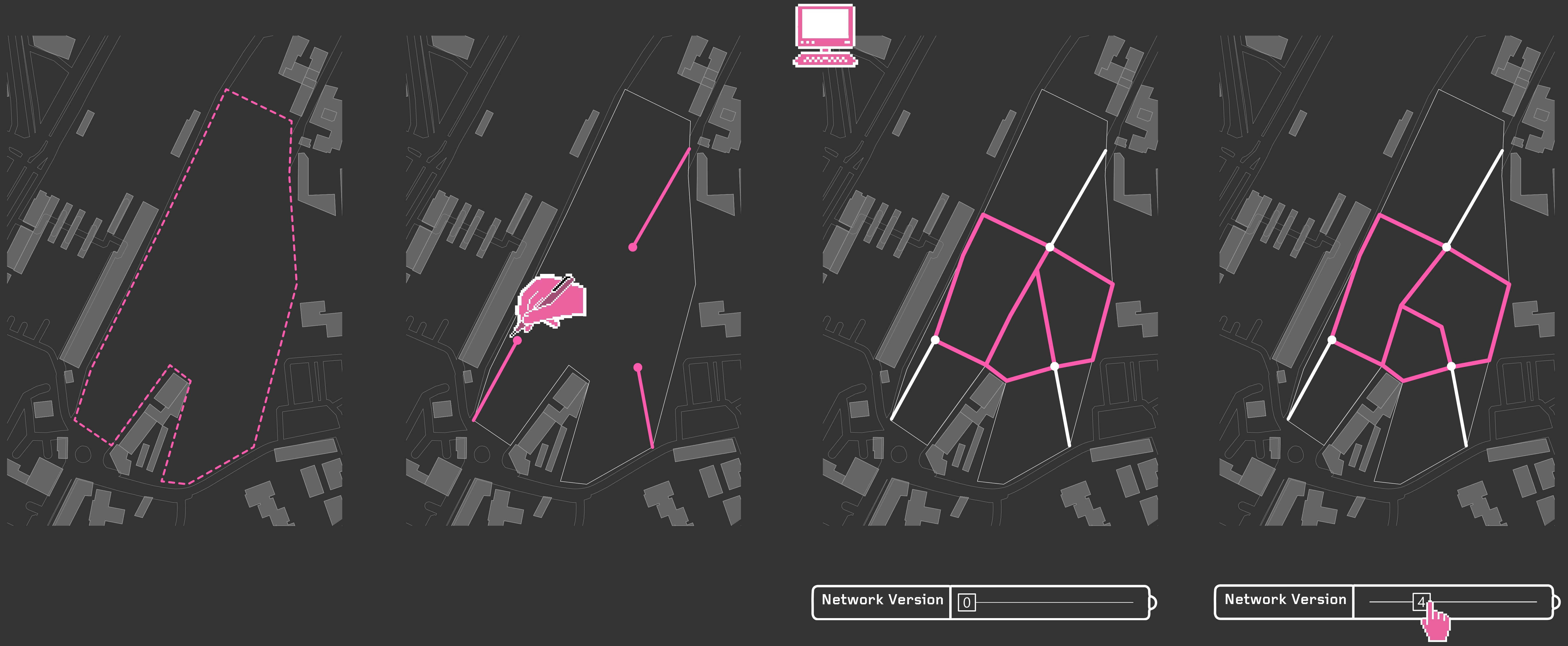


... 3.2. Layout Formation

... 3.2.1. Site Axis & Street Network

> Site axis is a linear feature or route within a city that connects significant places or landmarks and serves as a major reference point for navigation and wayfinding. Site axes can include streets, boulevards, parks, or other open spaces that are designed to be visually prominent and have a strong connection to the city's culture and history. In urban planning, a site axis is typically defined by its geographic location, its relationship to surrounding landmarks and buildings, and its role in connecting different parts of the city. Site axes can also be designed to serve specific functions, such as providing a major transportation corridor, promoting economic development, or supporting cultural and recreational activities.

In order to create site axis and street network in the comptutational environment, we ask user to use drawing agent and create curves that are intersecting with the existing site parcel. These curves are the main access routes for the pedestrian or in other scenarios for other mobility options. Once the main access routes has been drawed, the script creates a road network intersecting them by using Street Network Synthesis component created by DecodingSpacesToolbox (Osintseva, Koenig, Berst, Bielik, & Schneider, 2020)

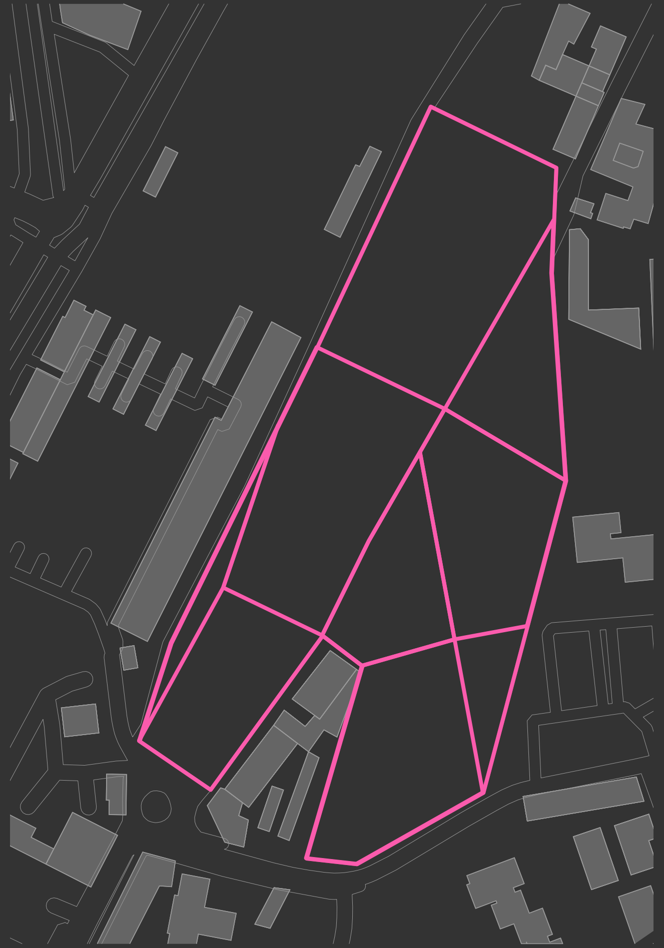


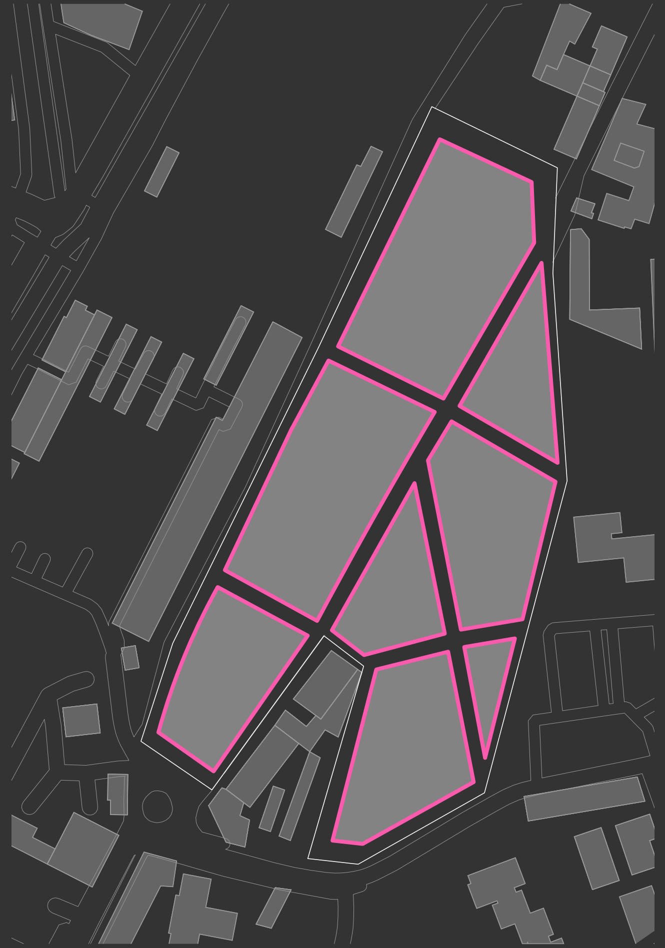
Design Phase:1

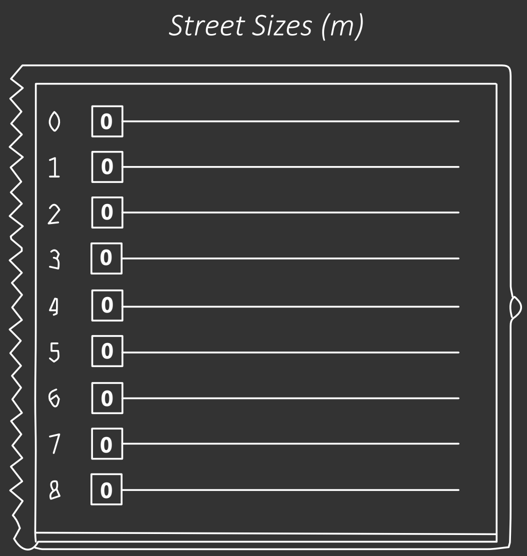
... 3.2. Layout Formation

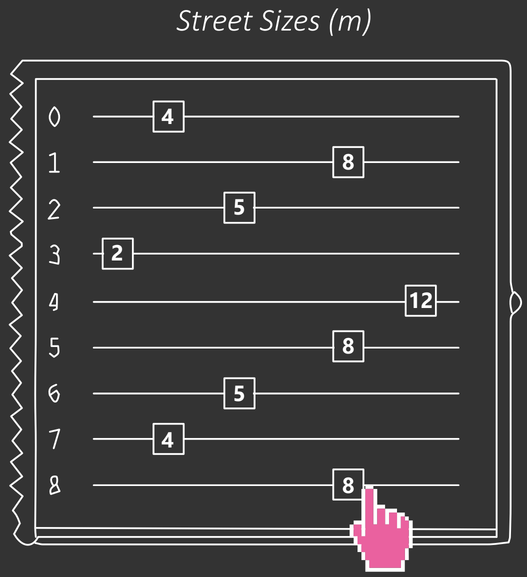
... 3.2.2. Street Hierarchy

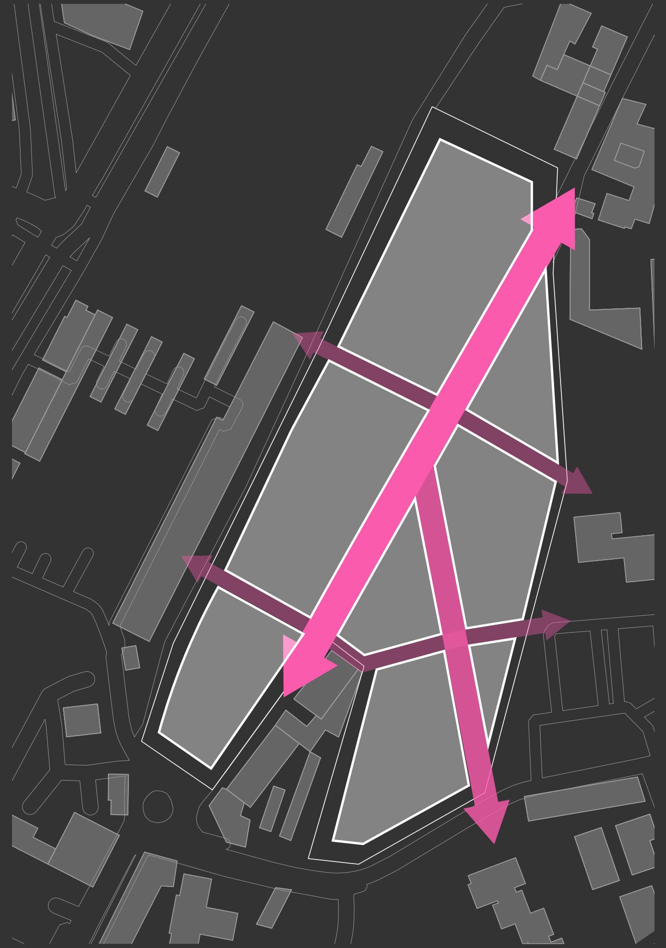
> Once the user is satisfied with the street axis network, the next step is to define the hierarchy of the generated streets. Street hierarchy involves determining the relative importance or size of each street within the network. To establish this hierarchy, the user is prompted to set a distance for the offset in both directions. This distance determines how far away from the centerline of the street axis the building plots and other features will be placed. By adjusting this offset, the user can control the dimensions and proportions of the streets in the network. The user is provided with tools like the gene pool component and the click & slide action agent to manipulate the hierarchy. The gene pool component allows the user to define the order of importance or size for the streets. This means that the user can designate certain streets, such as a main boulevard, to have a higher hierarchy or greater significance than smaller side streets. The click & slide action agent enables the user to interactively adjust the hierarchy by clicking on a street and sliding it left or right in the hierarchy. This feature gives the user precise control over the placement and importance of different streets within the network.











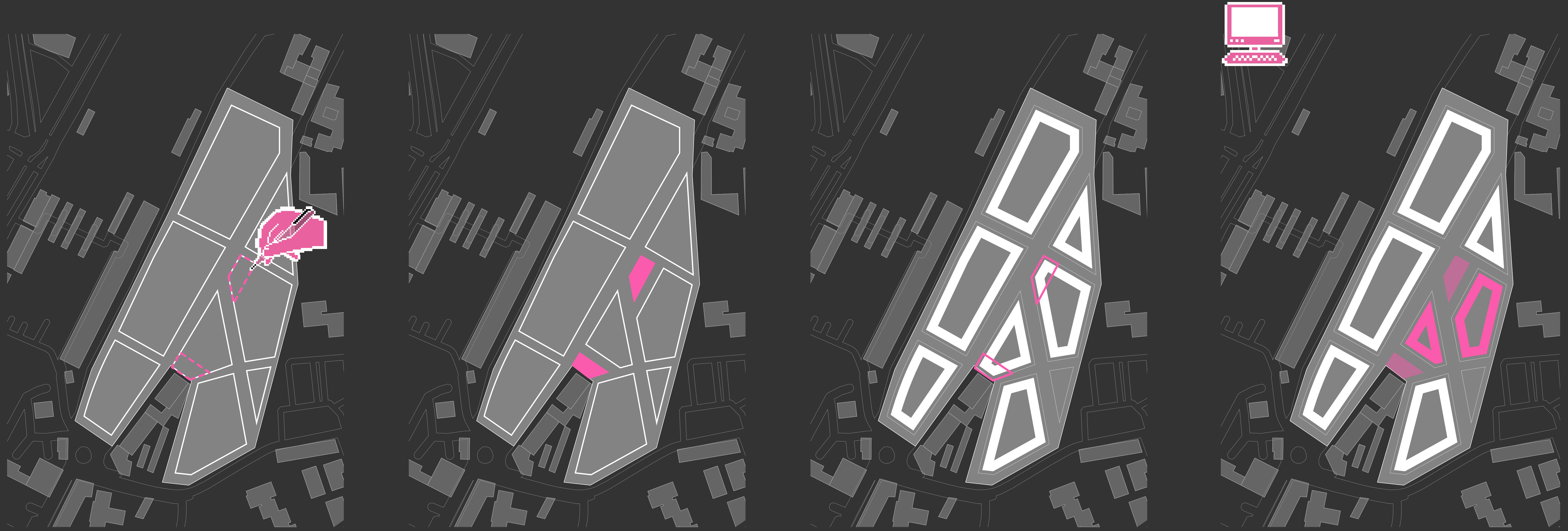
Design Phase:1

... 3.2. Layout Formation

... 3.2.3. Public Spaces & Plots

> One of the most crucial steps in designing a human-centric masterplan layout involves creating a well-organized spatial organization of public spaces. To create public spaces, the user is asked to create a 2D polygon on the site using a drawing action agent. By utilizing the drawing action agent, the user has the flexibility to define the boundaries and layout of the public space according to their vision and requirements.

Once the 2D polygon representing the public space is drawn, the street network system takes this information into account and adjusts its formation accordingly. The street network understands the location and extent of the public spaces, and it can revise its layout to integrate and connect with these spaces. When the newly generated street network fulfills the user's requirements, building plots are automatically generated by the computer.



Design Phase:1

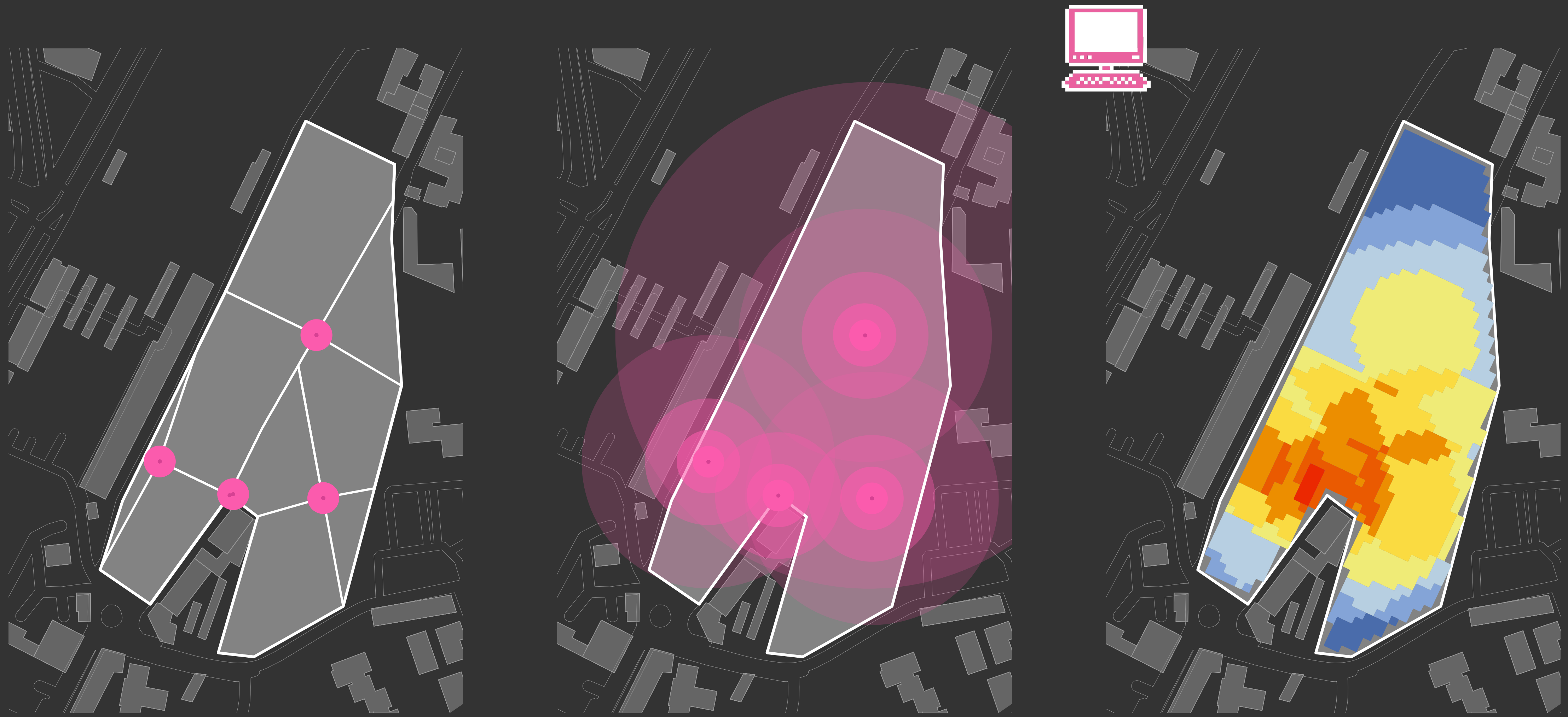
... 3.2. Layout Formation

... 3.2.4. Heatmap of Density

> In order to achieve a lively and sustainable urban environment that fosters social interaction, cultural vitality, and economic growth within the community, it is important to diagnose the behavior of the physical spatial configurations of the site. To guide users about which streets or physical spaces are more inclined to have denser relationships in terms of encountering and spontaneous interactions, a series of operations have been suggested by the computer.

First, all the intersection points are highlighted. Then, based on walkability distances, a set of circles are created and intersected with each other. Each intersection area receives a number of points based on the intersection count. By summing up all the intersections, a density map is created.

This process helps identify areas with higher levels of activity and potential for vibrant social interactions within the urban environment.

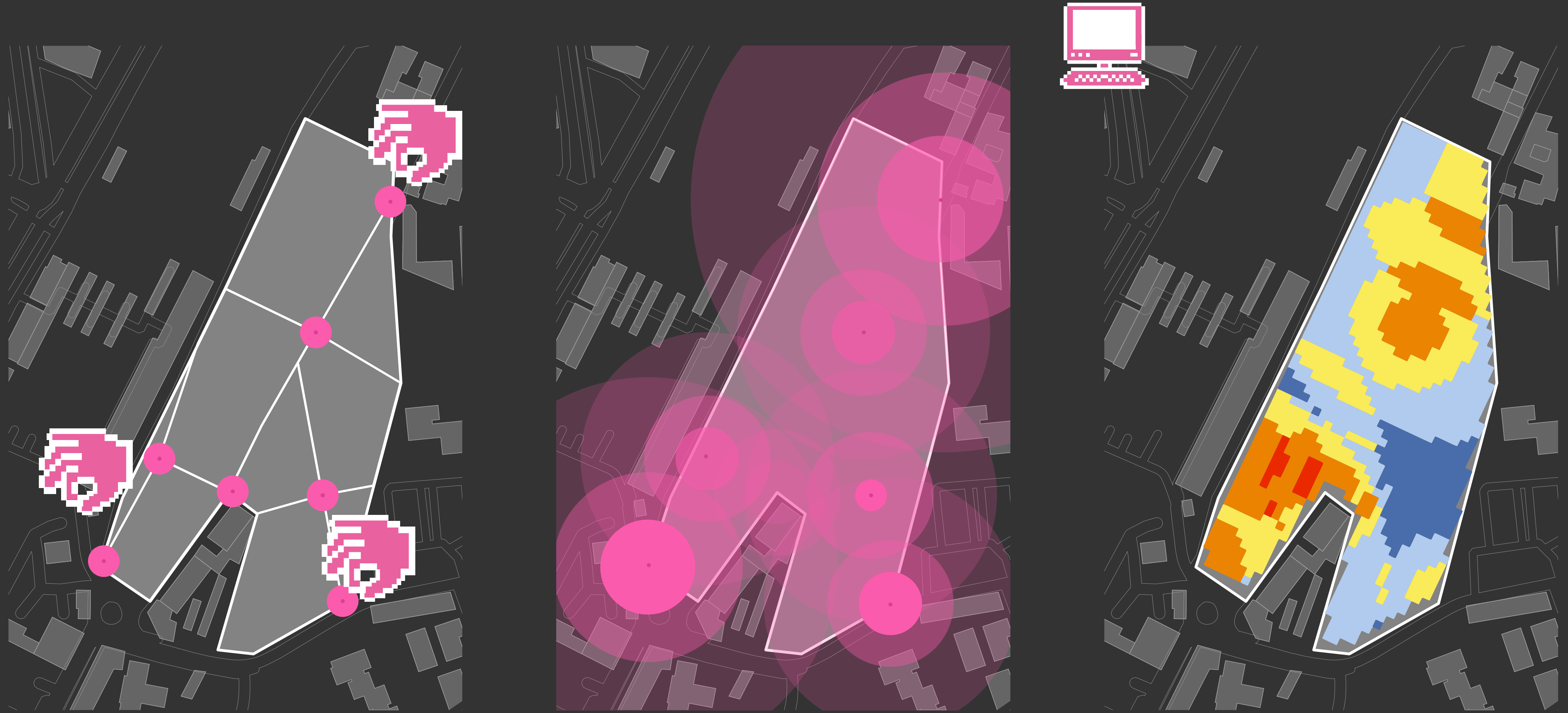


Design Phase:1
... 3.2. Layout Formation
... 3.2.4. Heatmap of Density

> After obtaining the suggested density heatmap, users are given the freedom to manipulate the map based on their own decisions. Since certain relationships are difficult to identify in a computational environment, designers may consider external factors that play a significant role in shaping the design. For example, if there is a prominent building in the neighboring area, the designer may want to treat it as a key actor in creating public spaces and define a focal point in that area. Conversely, if there are limiting factors, such as a train line as in the case of Innesto, the designer may prefer to locate quiet areas as far away as possible from that area.

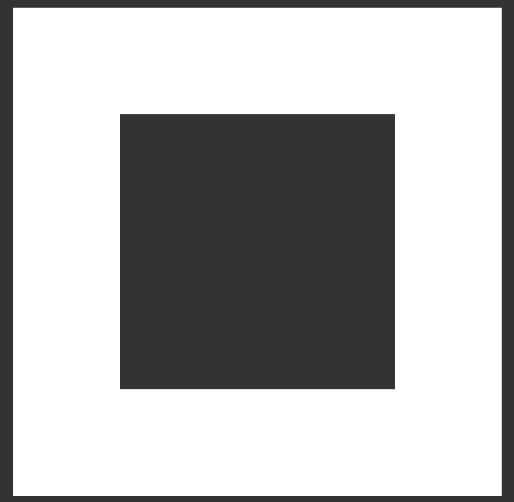
These complex relationships cannot always be accurately translated to a computational environment as we perceive them. Therefore, by using a picking agent, users are set free to add custom nodes to the map. Additionally, with the click and slide agent, users can manipulate various factors and create hierarchies between nodes. For instance, if there is a street intended to be the main entrance of the masterplan proposal, the user can increase the factor of that point, resulting in a higher density being inclined to form around that node.

This approach enables users to incorporate their manual interferences and local knowledge into the design process, allowing for a more nuanced and contextually responsive masterplan.



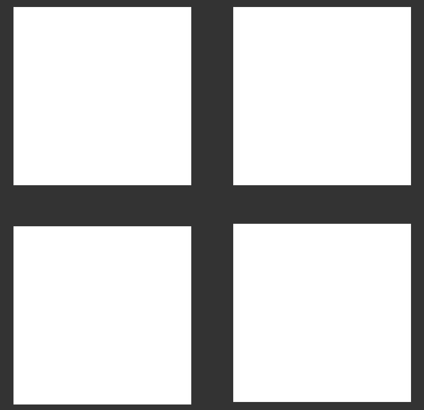
... 3.3. City Block Formation

... 3.3.1. 2D_Decoding: Generic Actions:



BLOCK

STRIP



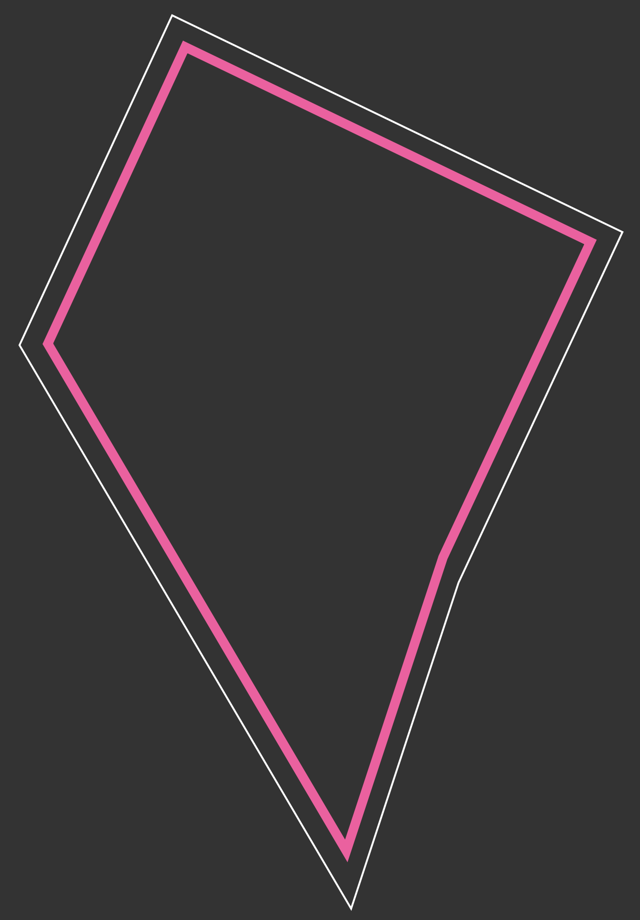
POINT & CLUSTER

> City Block Formation actions are classified into three classes and four action sets: generic, block, strip, and point & cluster actions. The generic actions are considered as master actions that serve as the foundation for the following actions and are shared by all the other sub-actions. On the other hand, sub-actions are only valid for the possessed class.

Design Phase:1
... 3.3. City Block Formation
... 3.3.1. 2D_Decoding: Generic Actions:
a. Setback

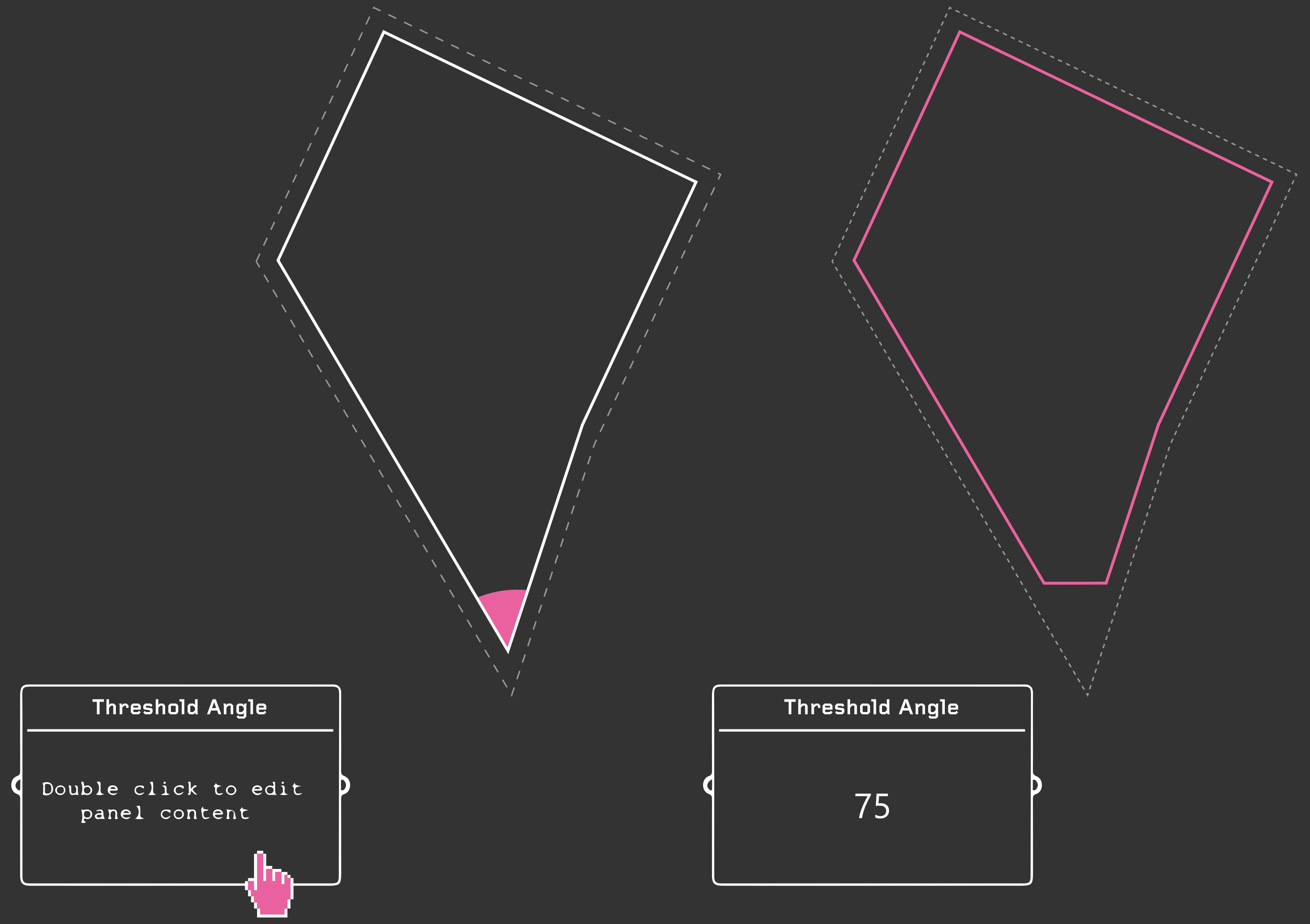
> Generic actions start after the user has created building plots in the layout formation phase. In this step, the design algorithm detects the generated plots designated for building construction. By applying an offset to the plots, with a distance specified inwards (typically determined by local zoning regulations or other design constraints), edge zones and footprints for the buildings are created according to the user's specifications.

The setback distance can be later adjusted by the user to define larger or narrower edge zones using the click & slide action agent, as well as the slider or gene pool component.



> TA (Threshold angle) controls the shape of the building plot. After the building plot is created, the user inserts the threshold angle using click & slide agent with a panel component. This angle is utilized to automatically remove any edges that exceed the specified limit, resulting in a more regular and suitable shape for the building plot.

The action is formed by decomposing it into its individual edges. The angle of each edge is calculated, and any edges that exceed the threshold angle are selected. The polygon is then reconstructed with these excessive edges removed, resulting in a new polygon with a shape that is optimized to create a more efficient building footprint.

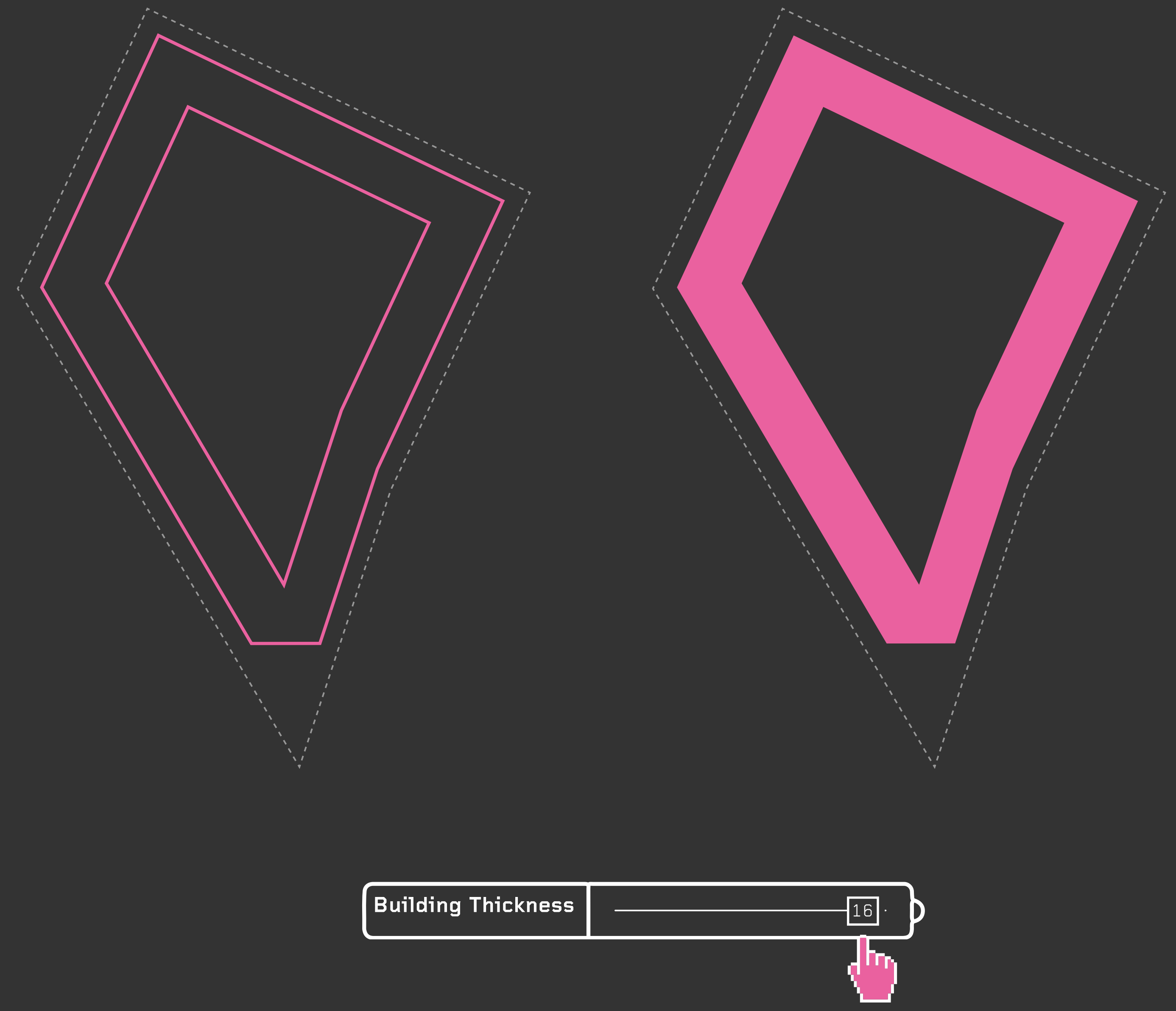


... 3.3.1. 2D_Decoding: Generic Actions:

c. Define Building Thickness

> At the end of the generic actions, the user is asked to create their first building iteration within the generated building plot from the previous action phase. The building footprint curve is offset inwards towards the center of the plot, and with the assistance of the click & slide action agent, the user can adjust the thickness of the building. The offset curve, along with the plot line, defines the boundary edges of the building parcel.

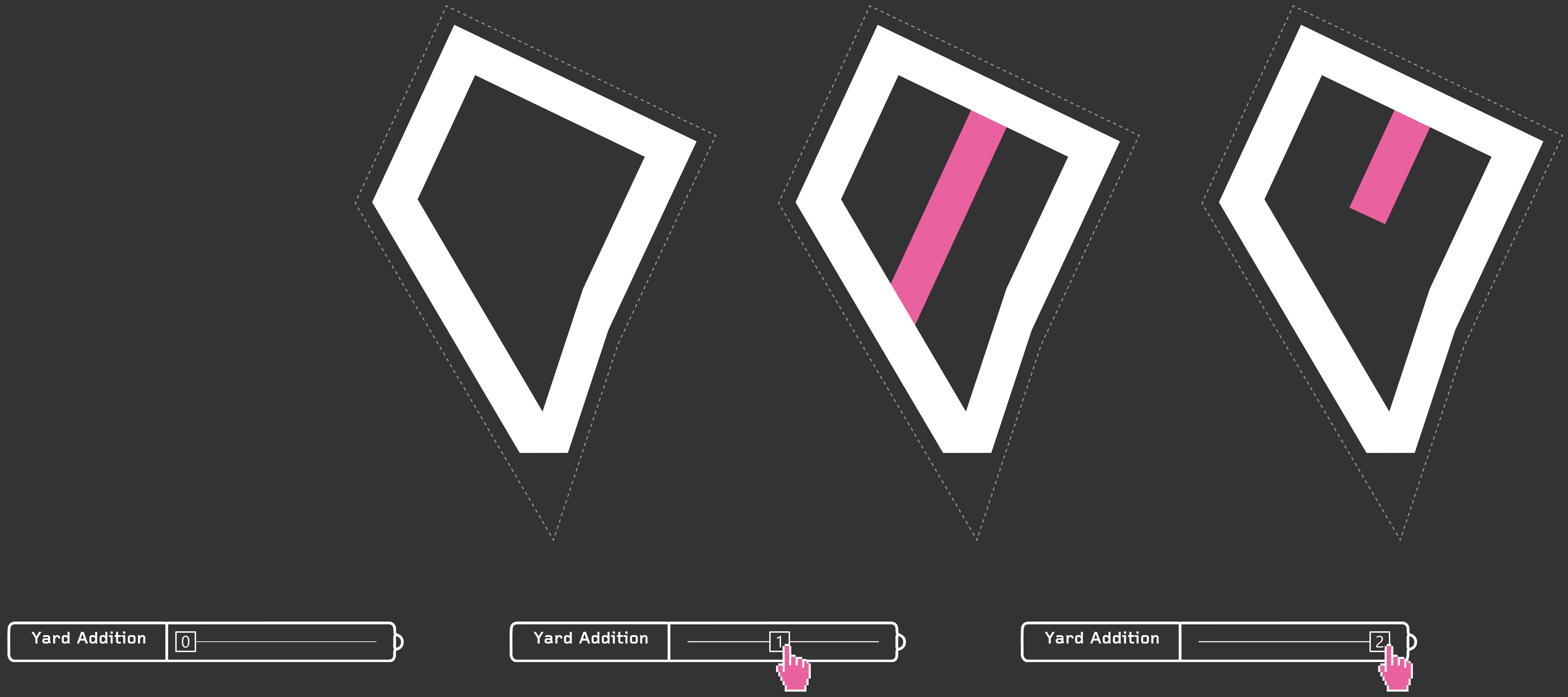
Next, a surface is created using the boundary edges as inputs. This surface defines the area within which the building can be constructed. The resulting surface, representing the building footprint, serves as a reference for designing the building itself.



... 3.3.2. 2D_Decoding: Block Actions
a. Yard Addition

> Block actions start with the yard addition function, which is responsible for enhancing the architectural and spatial characteristics of the project by adding slab-type buildings in the middle of courtyards, both in open and closed city blocks. This step helps to add architectural and spatial richness to the project by creating a variety of different courtyard spaces, each with its own unique characteristics. The use of slab-type buildings in the middle of courtyards can also help create a sense of hierarchy and movement to the project.

The script detects the longest edge face of the block. Upon identifying the longest edge face, the script then adds either a complete slab or a half slab to the courtyard. The designer can navigate between options by using clik & slide agent.

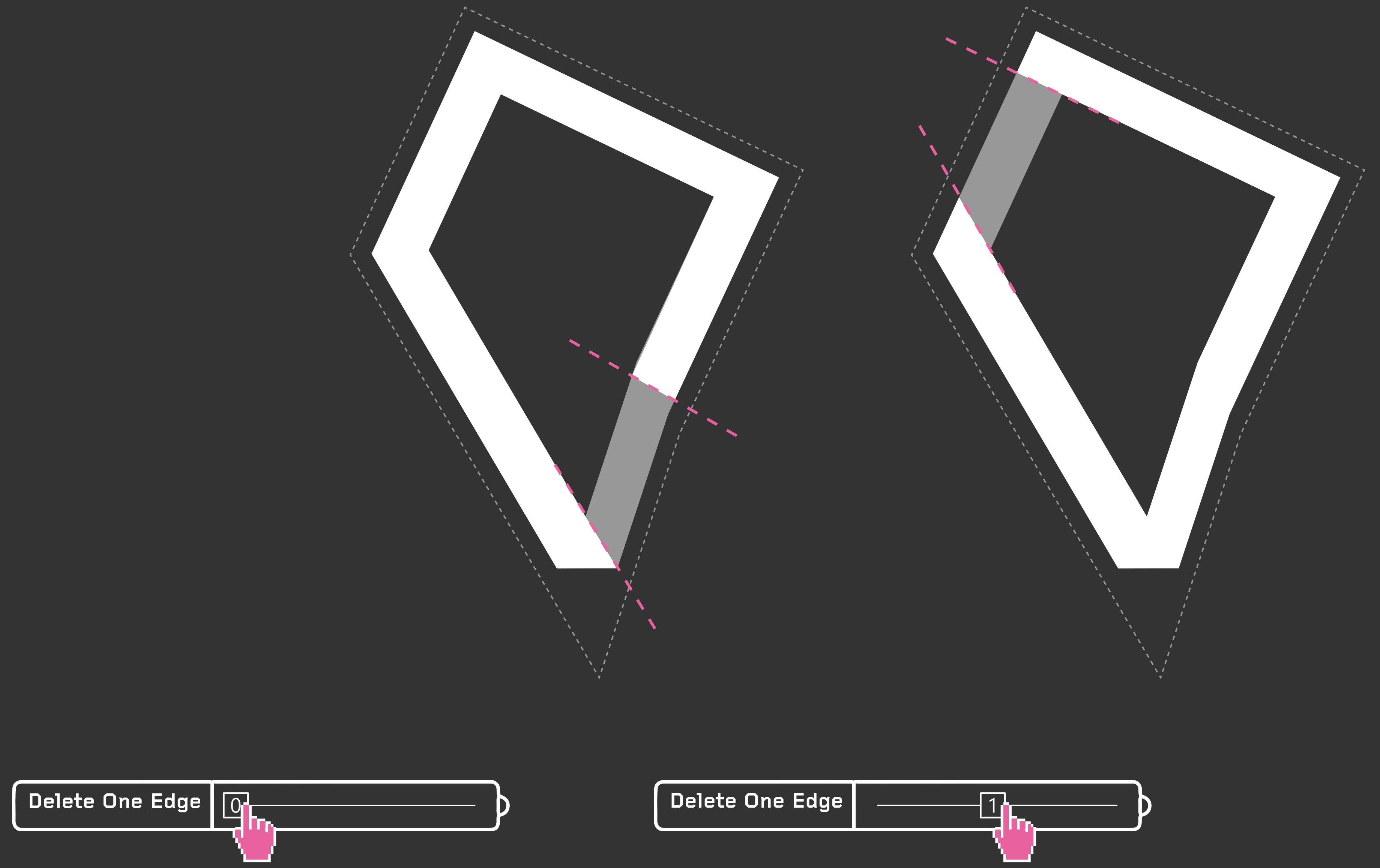


... 3.3.2. 2D_Decoding: Block Actions

b. Delete One Edge

> If the user aims to enrich the city block typology this can be done by further block actions. Creating open city block building typologies can be achieved by deleting certain edges of a closed block building. This process involves identifying the edges to be removed and deleting them from the building model. This is done by identifying the edges to be removed, and then deleting them from the building model. This creates gaps or openings in the building envelope, allowing light and air to penetrate the interior spaces and providing visual connections to the surrounding environment. The process of deleting edges allows users to quickly and easily manipulate the geometry of the building and evaluate the feasibility of different design options. By deleting selected edges, a wide range of open block building typologies can be created, each with its unique character and spatial qualities.

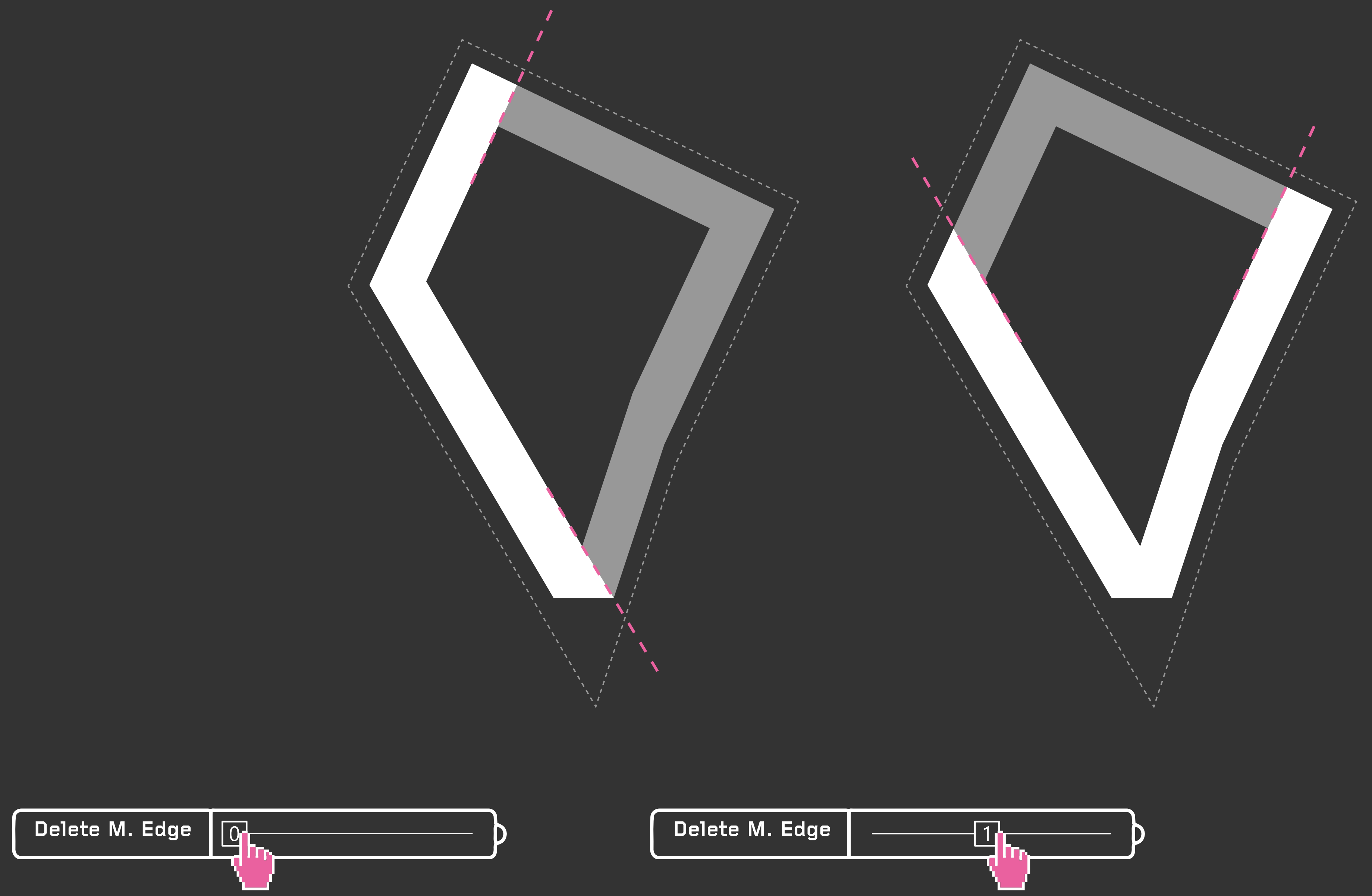
The script assigns a unique number to each edge to translate information into the computational environment. Therefore, users can navigate through different numbers and modify which edge to be deleted by using the click & slide agent.



c. Delete Multiple Edges

> Delete Multiple Edges action utilizes a similar approach as the Delete One Edge action. However, in this case, the script is designed to delete multiple edges simultaneously, resulting in the creation of open city blocks or slab-like buildings.

Similar to the previous action, the script assigns a unique number to each edge to facilitate information translation into the computational environment. This enables users to navigate through different numbers and modify which edges and how many of them should be deleted using the click & slide agent.

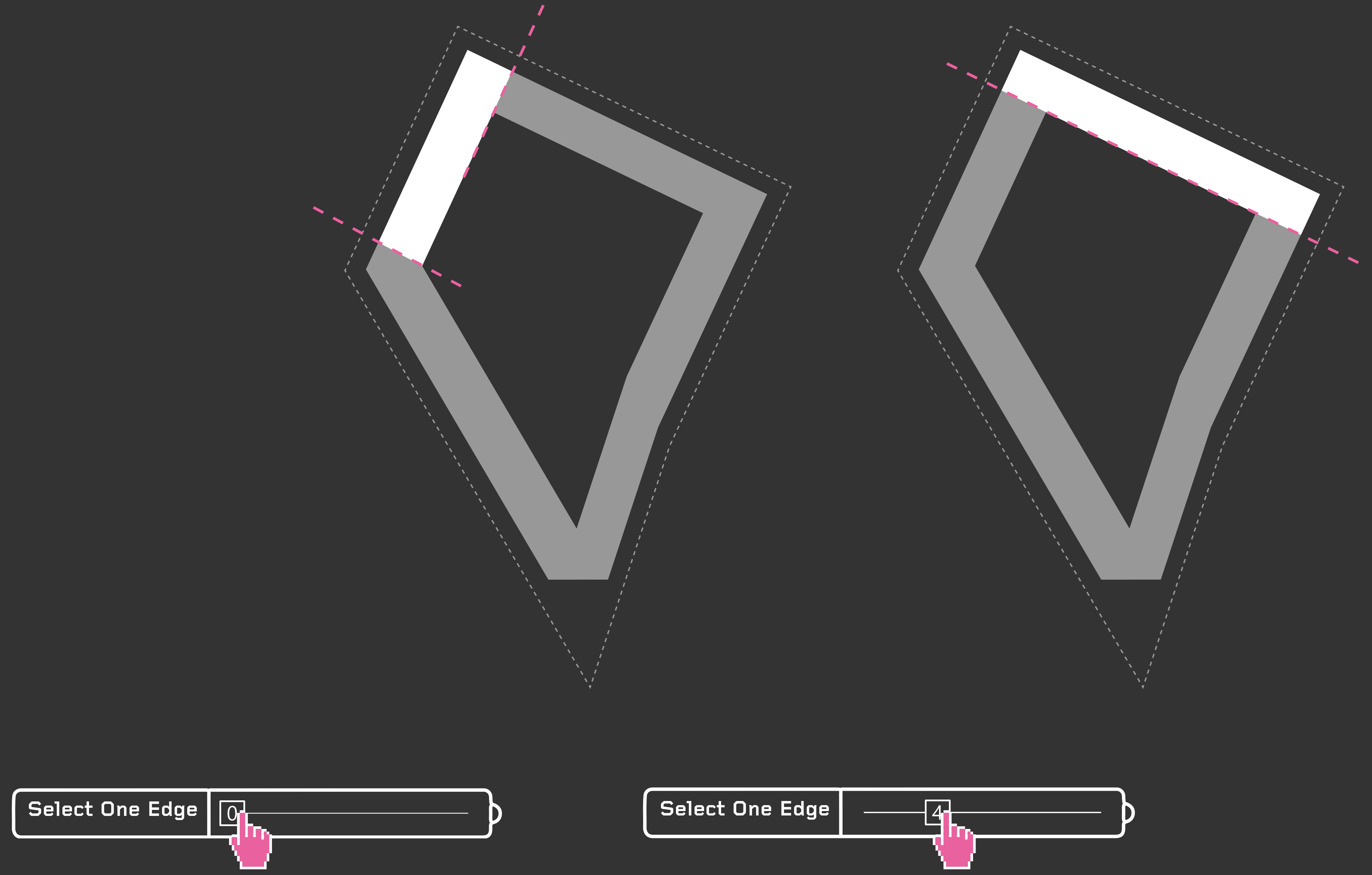


... 3.3.3. 2D_Decoding: Strip Actions

a. Select One Edge

> If the user intends to increase the open spaces within the building plot and restrict the building footprint, they can continue deleting edges until they reach a certain limit, which is always one value less than the total count of edges. To facilitate this process, a "Select One Edge" function has been implemented.

Whether starting from an open city block or a closed city block, the user can utilize this function and, with the assistance of the click and slide agent, determine which edge to retain and consequently create a strip type building.

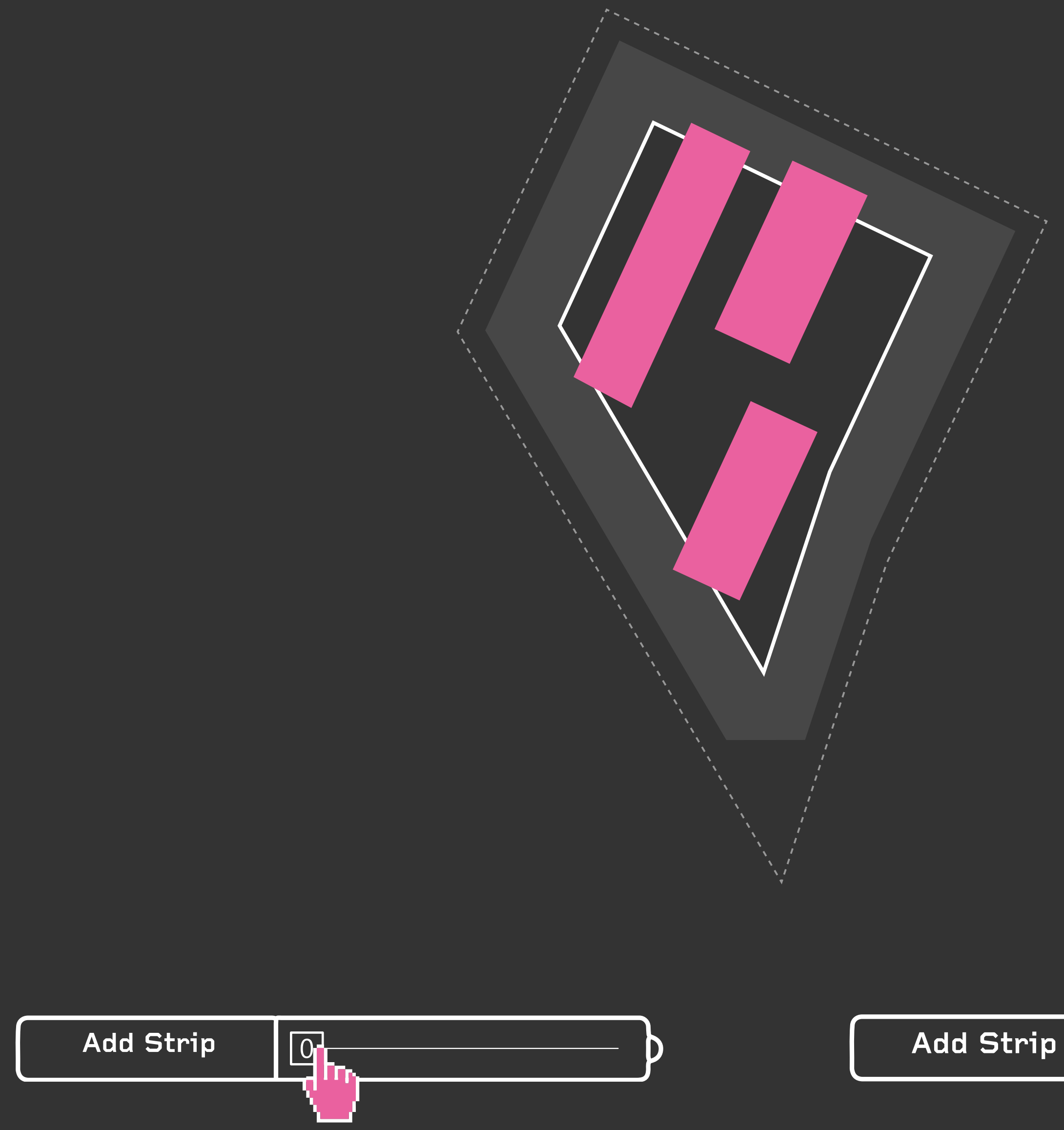


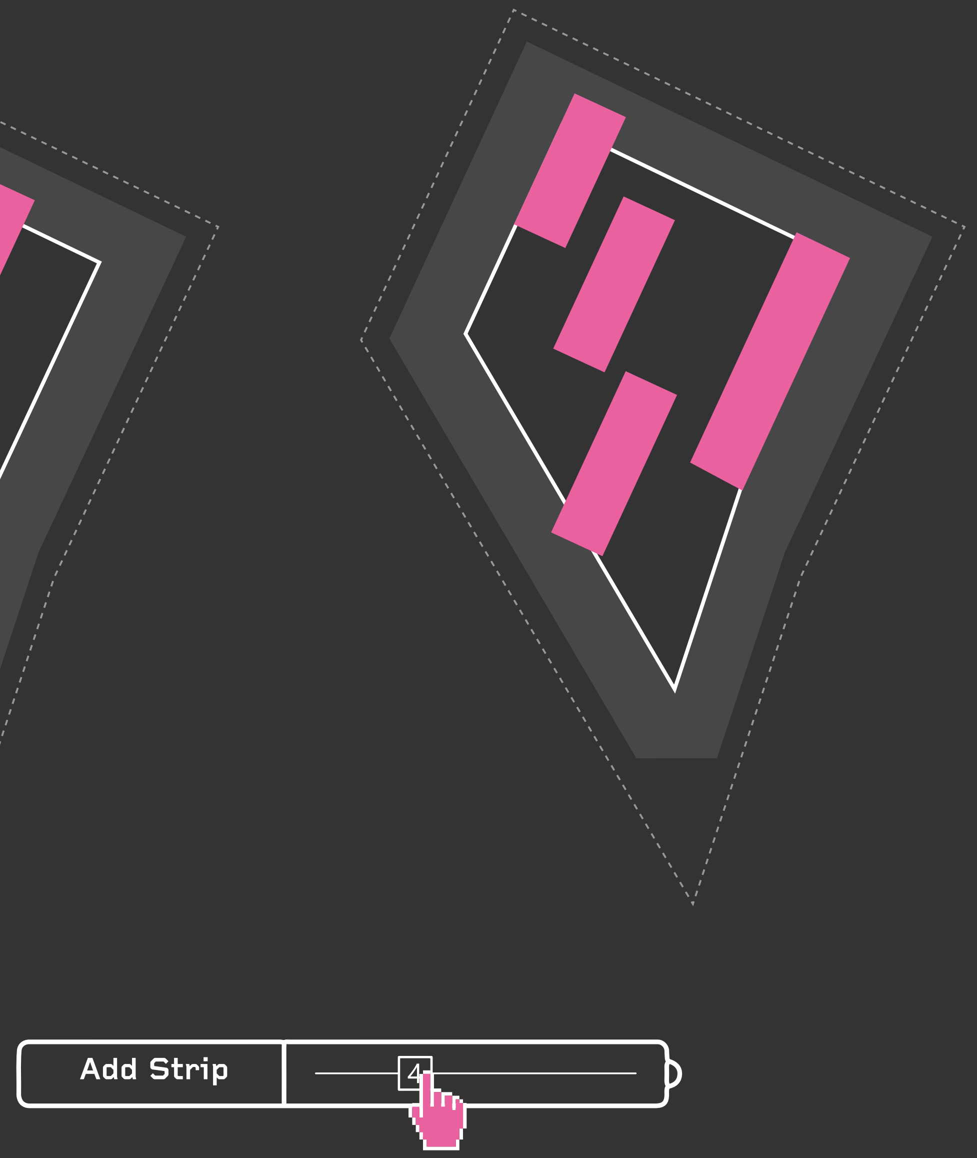
... 3.3.3. 2D_Decoding: Strip Actions

b. Add Cental-Located Strips

> While most of the strip actions are linked to the building surface derived from the closed city block functions, there are certain typologies that cannot be created using the same approach. In order to enhance the spatial richness and the quality of the project, new action functions need to be developed.

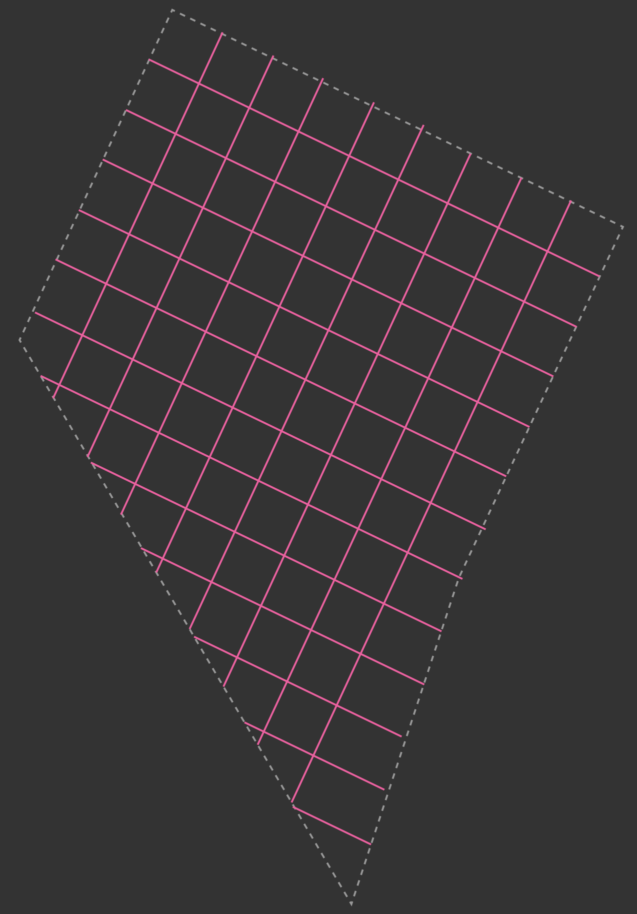
The "Add Central Located" action utilizes the building footprint generated in the previous steps. However, instead of directly using the building footprint surface as a building volume, it uses the footprint as boundaries for building locations and populates points within this boundary. The generation of these points is randomized, and the user can adjust the values using the click and slide agent. The total number of slabs, their locations, and the randomness seed are the parameters that can be manipulated by the user.





... 3.3.4. 2D_Decoding: Point & Cluster Actions

a. Create Grid Layout

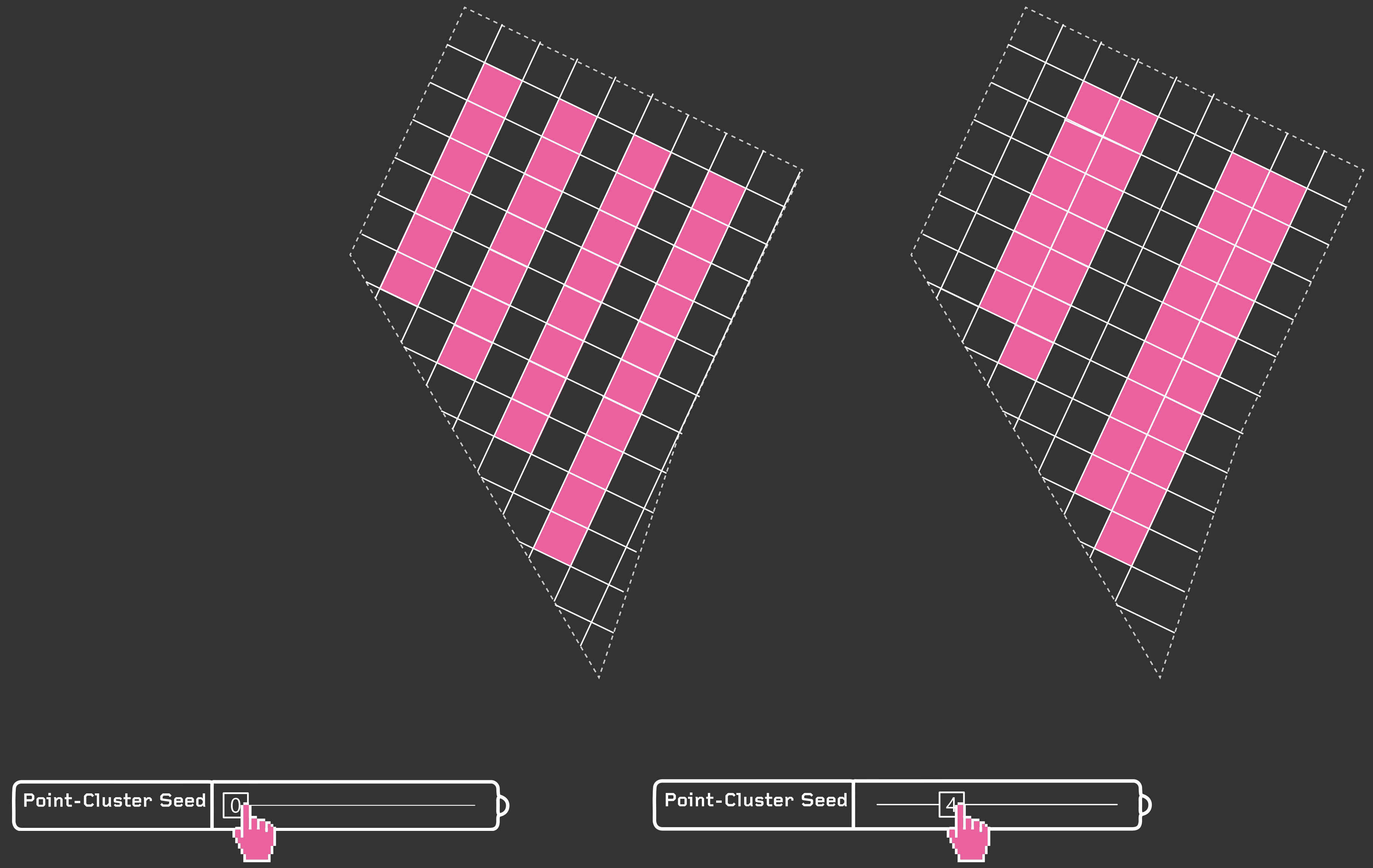


> Create Grid Layout action is the first step in formation of the point-cluster type buildings. This action function detects the edges of the building footprint that were generated in the previous actions and proceeds to perform a series of offset operations. It offsets each edge of the footprint until it reaches the maximum offset count within the boundary. It is recommended for the user to utilize this edge alignment to create point-cluster type buildings that maximize the number of buildings. However, the user has the flexibility to change which edge is selected and aligned with the grids based on their own criteria. Once the edge alignment has been determined, grids are created to serve as the foundation for the point-cluster buildings.

... 3.3.4. 2D_Decoding: Point & Cluster Actions

b. Create Grid Pattern

> In the second phase of the Point-Cluster Actions, various patterns for potential building blocks are made available to the user. This is accomplished by utilizing the grids that have been generated in the previous step as a base. The algorithm begins by sorting the columns into either a horizontal or vertical orientation, starting from the first set of grid curves. Depending on the pattern choice made by the user with click & slide agent, buildings may either be placed with open spaces (such as backyards or gardens) between them by jumping over the next grid set and continuing the creation of buildings in the next set of curves, or various other options as illustrated in the image may be implemented.

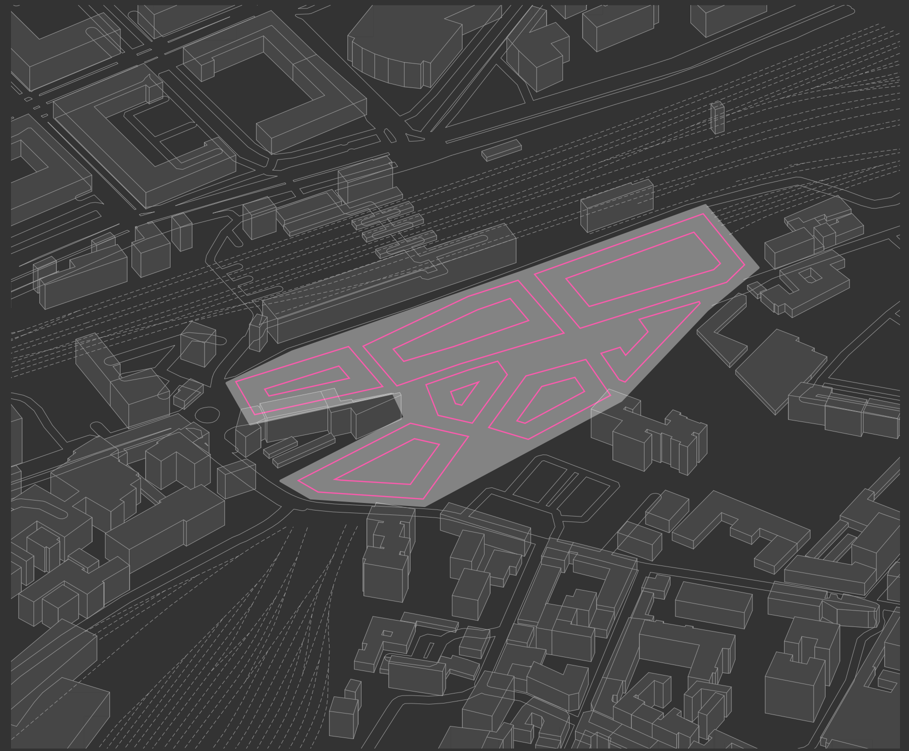


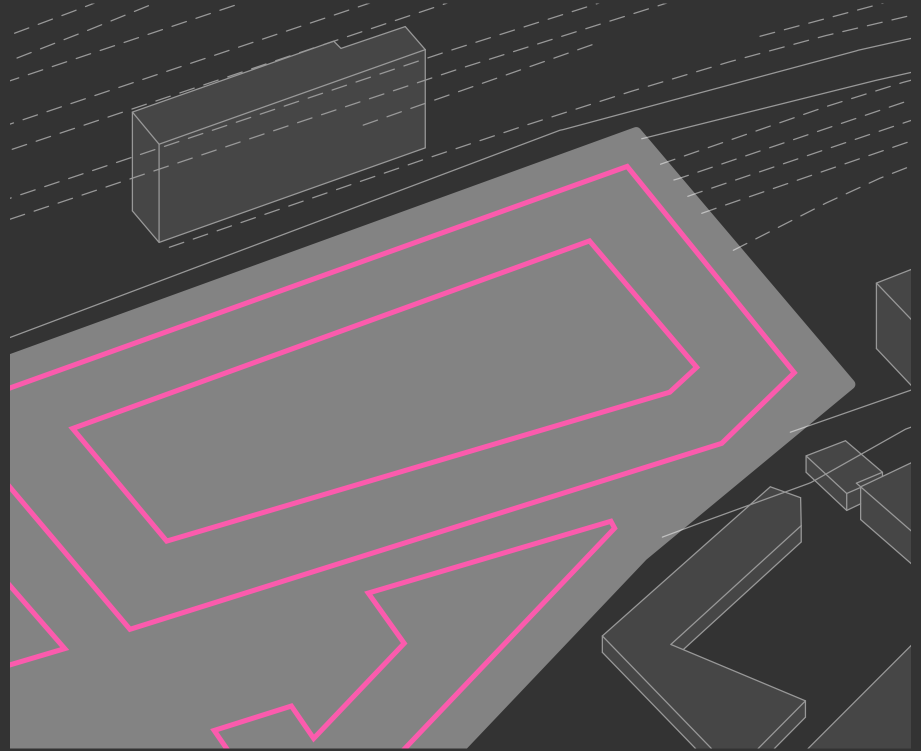
... 3.3.5. 3D_Decoding: Generic Actions

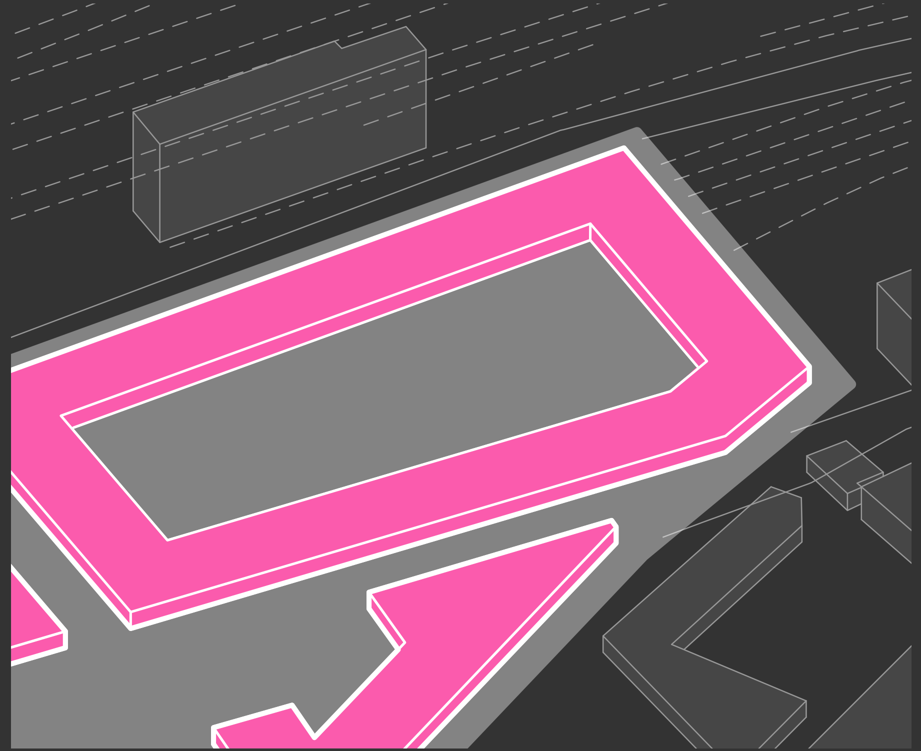
a. Define Floor Height

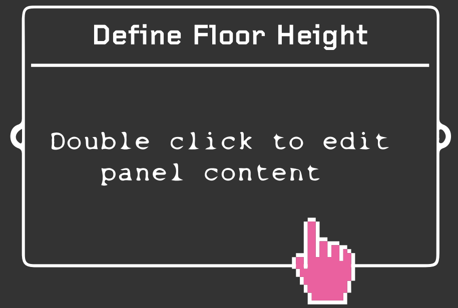
> Once the user has finalized the 2D actions and their preferences (the term "fixed" is used here to indicate that the user has made their decisions, but they still have the opportunity to go back and change parameters later on), the next step is to determine the 3D actions. The first action involves deciding on the floor height using the "Define Floor Height" action. This action utilizes a static panel component, allowing the user to enter an integer or decimal number.

The standard floor height for residential buildings is typically around 2.4 meters, while commercial and public buildings may have higher floor heights ranging from 2.7 to 4.3 meters. However, these measurements can vary depending on the intended use of the building, local building codes, and accessibility requirements. Therefore, the user can decide and create the initial volume iteration for the ground floor of the building based on the design's specific needs.







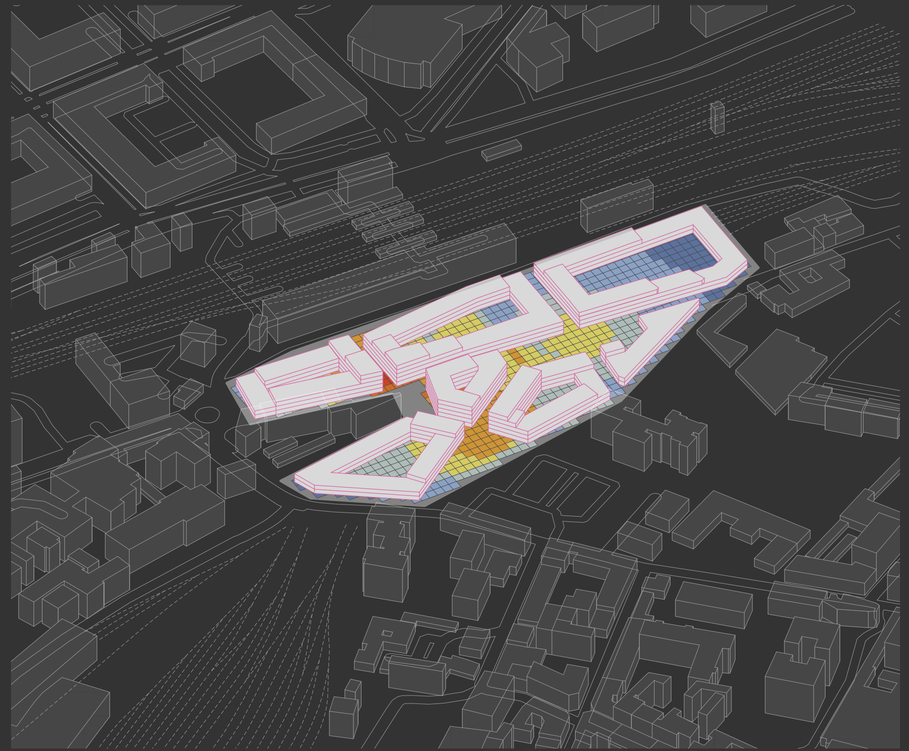


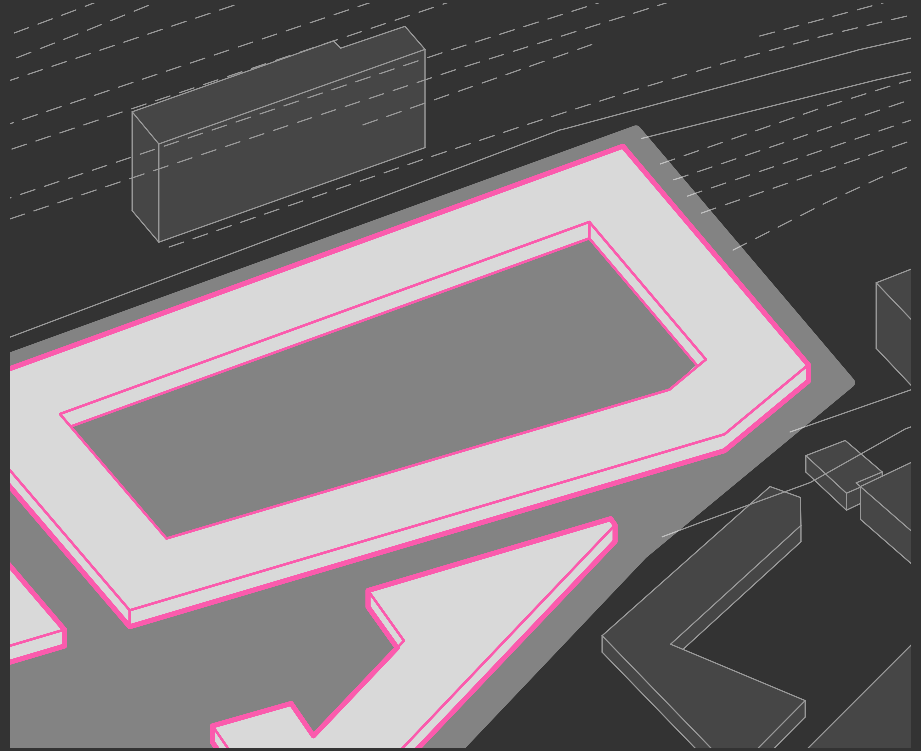
... 3.3.5. 3D_Decoding: Generic Actions

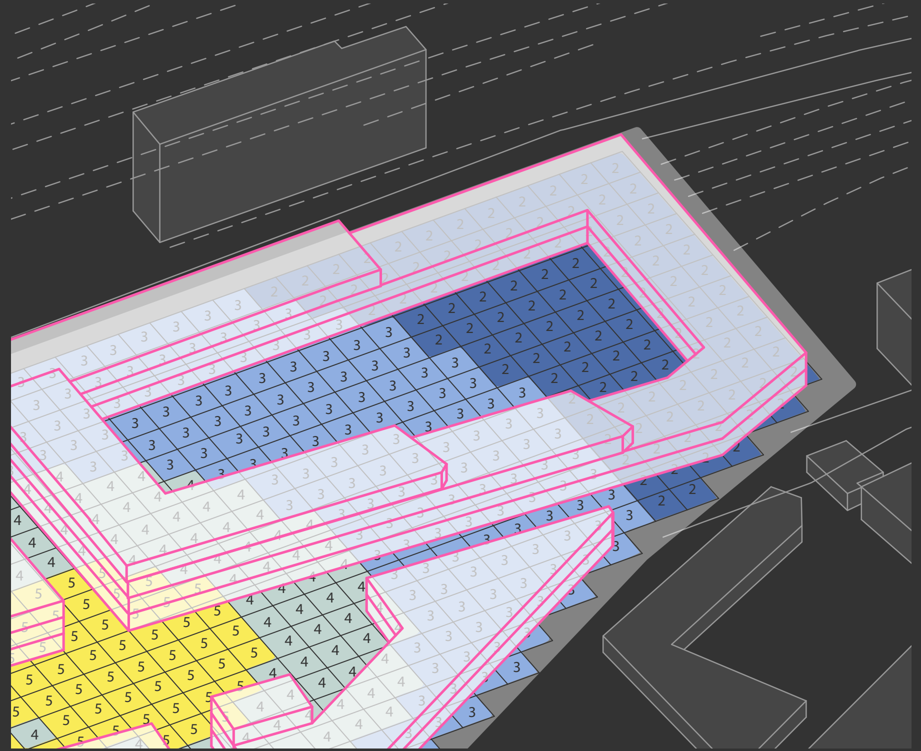
b. Define Height Density

> In the initial phase of Layout Formation, a density heatmap of the site is generated at Page 96 by considering the intersection points of street networks and walkability distances, as well as any custom considerations provided by the user. Each intersecting area is assigned a numerical point, creating a matrix system that covers the entire site area. Building footprints that occupy these matrix points are then added, and the mean value is calculated. This process results in each building being assigned a factor number, which determines the inclination in the Z-axis for each building.

This 3D action aims to incorporate the density of the site not only in the ground level but also in the three-dimensional space. If the user wishes to manually interpret the data and incorporate their own criteria or adhere to local regulations, they can utilize a static panel component. Using the click & slide agent, the user can assign the number of floors for each building plot as an index.







design phase 01

... P-02 EVOLUTIONARY DESIGN SIMULATION

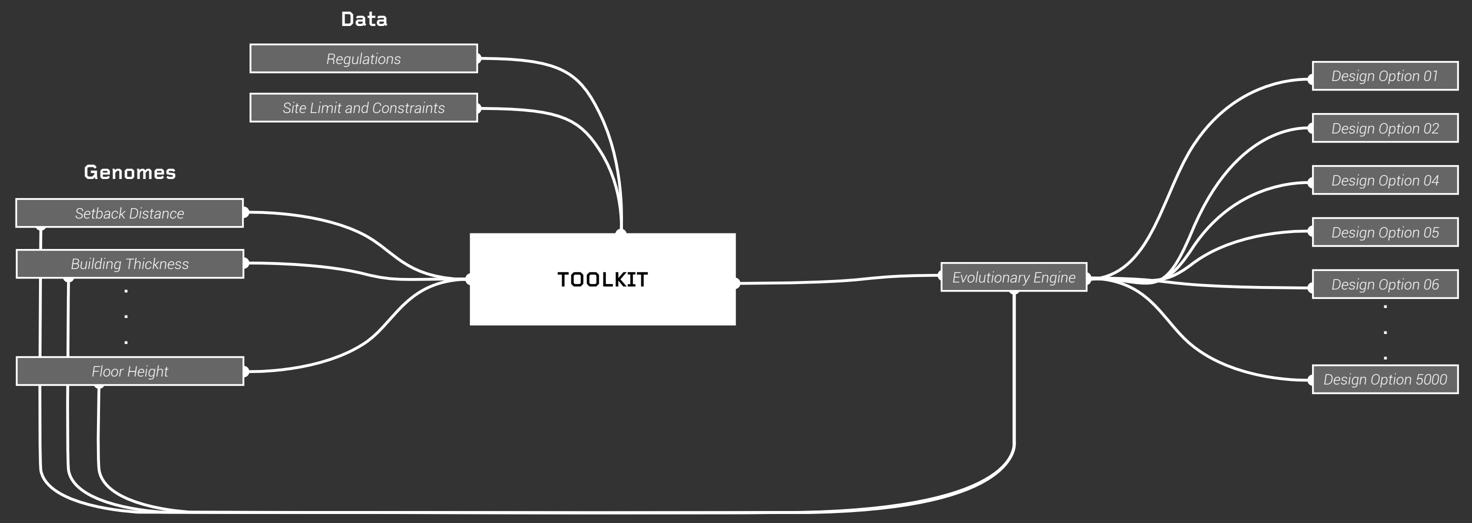
In the following part, the actions and components described in the paper are applied to an existing site in Milano, Italy. The pages provide details about the existing project on the site and the design options generated using the tool. Finally, a comparison is made between the generated results and the existing project, taking into account fitness values.

> To validate the effectiveness of the proposed framework in the paper, we provided two scenarios:

First, we made a comparison between a masterplan project in the Milan region. In this scenario, our objective is to enhance performance in terms of environmental aspects while having as much as mixed typology in the built area, as elaborated in subsequent sections. In this experiment a constant density index (FAR) maintained.

In the second scenario, we created many different extreme typological varieties while still keeping the FAR constant. This allows us to assess the potential usefulness of the framework as a sketching tool during the early design phase in architectural workflows.

EXPERIMENT - 01
Machine-Controlled



EXPERIMENT - 02
User-Controlled

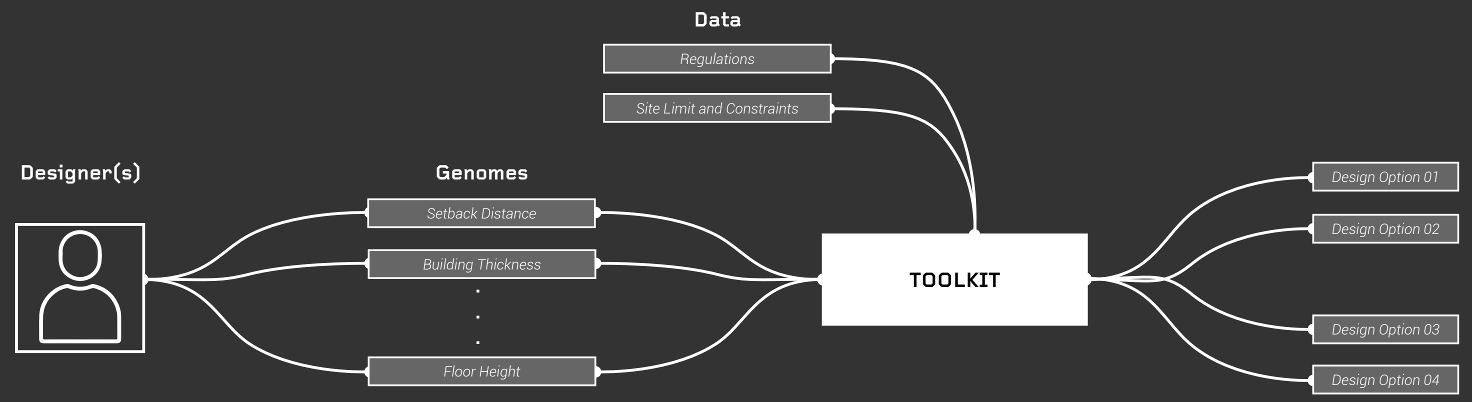
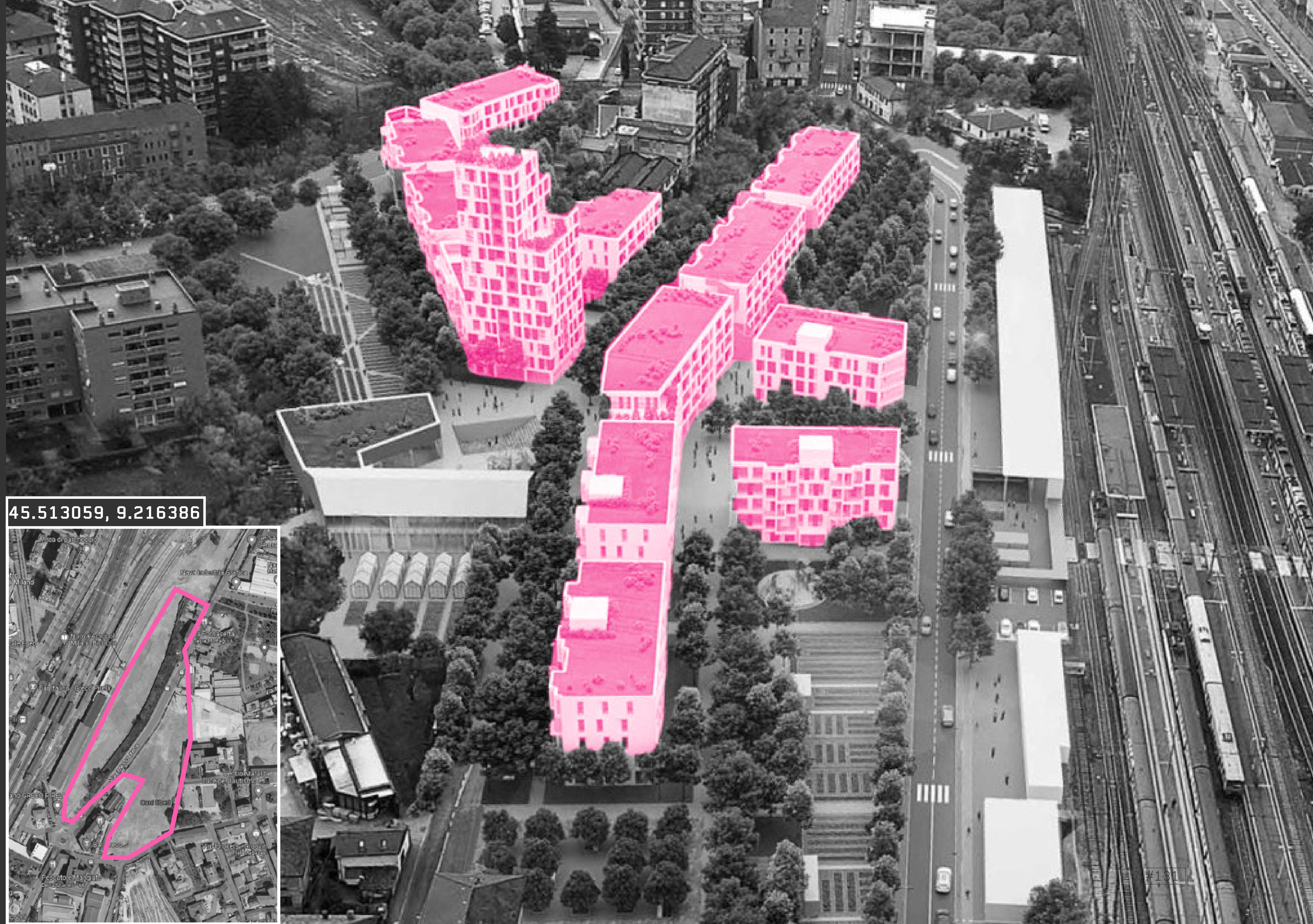


Figure 39: Experiments actors and their framework

Design Phase:1 Part:2
... 3.4. Case Study: Milan, Innesto
... 3.4.1. Project Registration

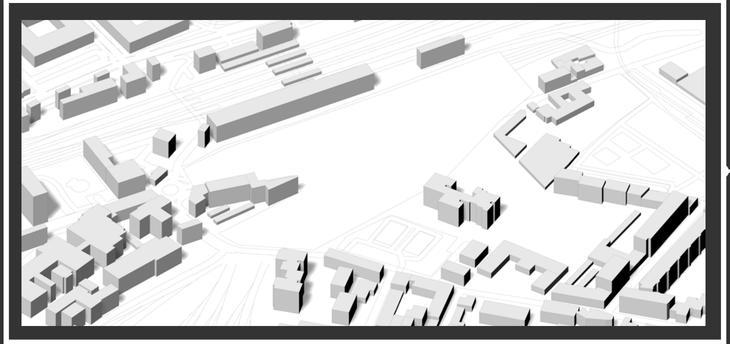
> The Innesto masterplan project is described as the following in the project registration text: The Project is a proposal for a zero carbon "Housing Sociale" scheme in Milan. For this area of approximately 62,000 square meters, made up of three parts, the masterplan project developed by the Milanese architecture studio Barreca & La Varra and by Arup Italia. In this innovative district, around 400 new social housing units will be built (of which around 50 with a garden on the ground floor), 300 student rooms. Milanese suburbs are enriched with a new settlement principle, characterized by a multiplicity of open spaces and relationships - such as arcades, squares, broletti - around residential buildings marked by important services (Circular Economy District, Community Food Hub , Zero Waste Food Store) and green diaphragms (private gardens, educational gardens, avenue of mulberry trees, orchards, community gardens, woodland) with this proposal.



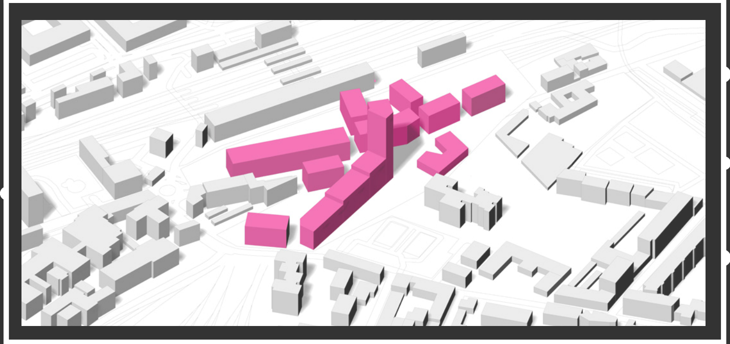
... 3.4.2. Measuring the Project Performance

> In order to create design solutions and compare with Innesto project first a set of numeric data is needed to be collected. Therefore, the Innesto project is mapped using OpenStreetMap database and after mapping the building footprint, a simple volumetric model is created in the Rhinoceros program, utilizing project registration information such as the number of floors, floor heights, and total surface area. Following that, a region of interest is drawn and selected as the simulation area (the area is limited to increase simulation performance, considering the CPU power of the laptop used in the paper). Once the 3D site model is created, a set of analyses is conducted.

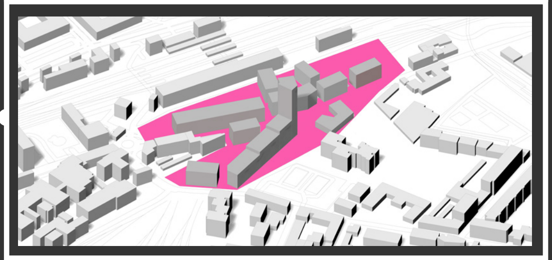
OpenStreetMap Site Mapping



Creating 3D Site Model



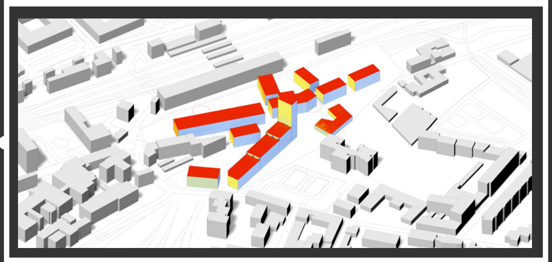
Geometrical Analysis



Geometrical values of the site are measured by using Rhinoceros measurement components:

Gross Site Area: **Gross site area = 35.211 m^2**
Total Building Floor Area: **52.112 m^2**
Floor Area Ratio (FAR): **1.48**

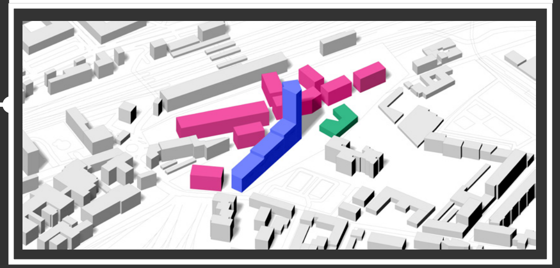
Environmental Analysis



Ladybug v1.4.0 is used to calculate total summer radiation, view percentage and annual ground shadow hour of the buildings. Than the relative values are calculated mathematically:

Annual Summer Radiation per 1m^2: **14.70 kWH/m2**
Annual Average Sunlight Hour on Outdoor: : **4.77 h**
View Percentage of Inhabitants: **%44**

Typological Variety Analysis



Typology variety index is calculated by using the formula of Shannon extropy which quantifies the level of disorder or randomness in a system, providing a measure of the information content or uncertainty present in a data set. Each type of city block is assigned with a unique number and a list is created with those numbers. Formula applied to that list:

Typology Variety Index: **0.16**

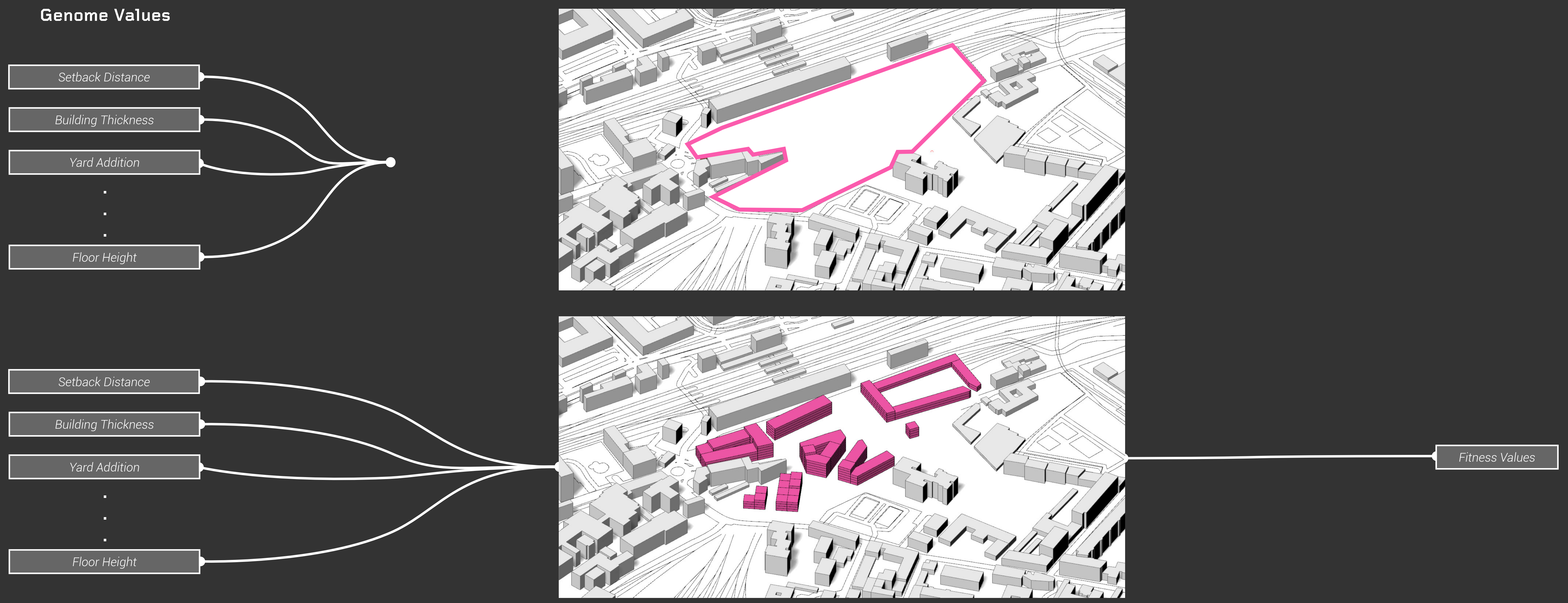
Design Phase:1 Part:2

... 3.5. Design Simulation

... 3.5.1. Genome Values of the Initial Design

> Genetic Algorithm (GA) typically begins with a genetic representation of potential solutions to the problem at hand(Michalewicz,1994). In the initial test conducted for this study, the gene pools and slider values are connected to the functions on an empty canvas. The first set of gene sequences are generated using randomly distributed values for the components, as visually depicted in the illustrations. It is important to note that since this initial generation does not explicitly address the design goals, the resulting designs may lack practicality, attractiveness, or suitability for the given context. Nevertheless, this initial iteration serves as a fundamental starting point for the GA simulation. Once the Individual 0 is created the fitness values are automatically calculated by the framework:

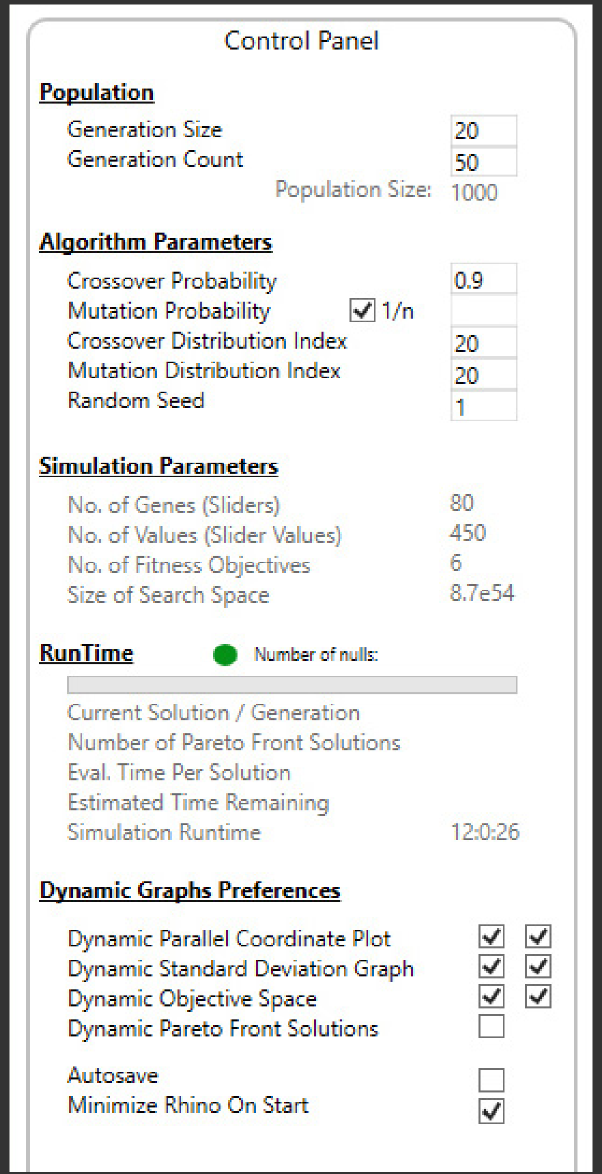
Total Floor Area: **29028**
FAR: **0.68**
Annual Summer Radiation per 1m²: **12.62 kwH/m2**
Annual Average Sunlight Hour on Outdoor: **4.84**
View Percentage of Inhabitants: **42**
Typology variety index: **2.23**

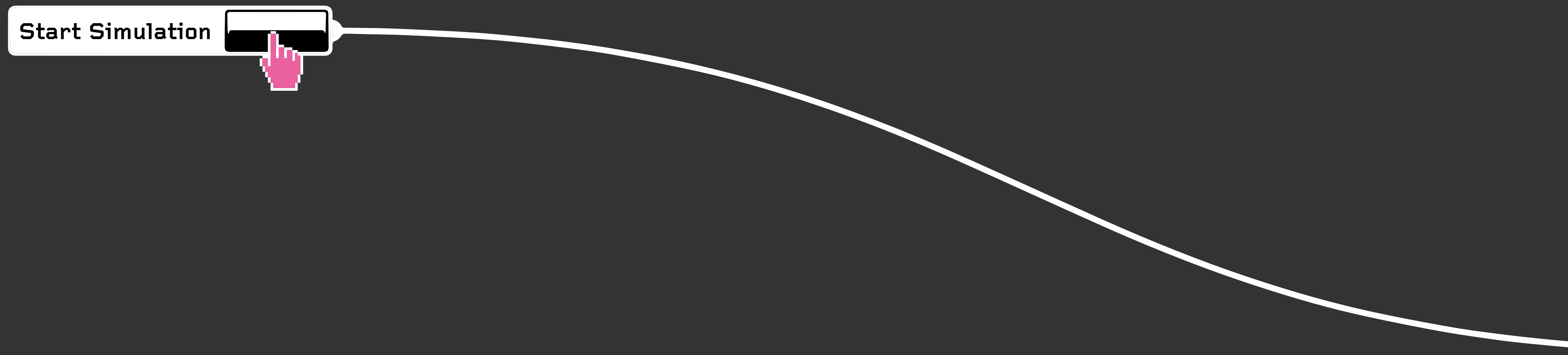


... 3.5.2. Setting Simulation Parameters

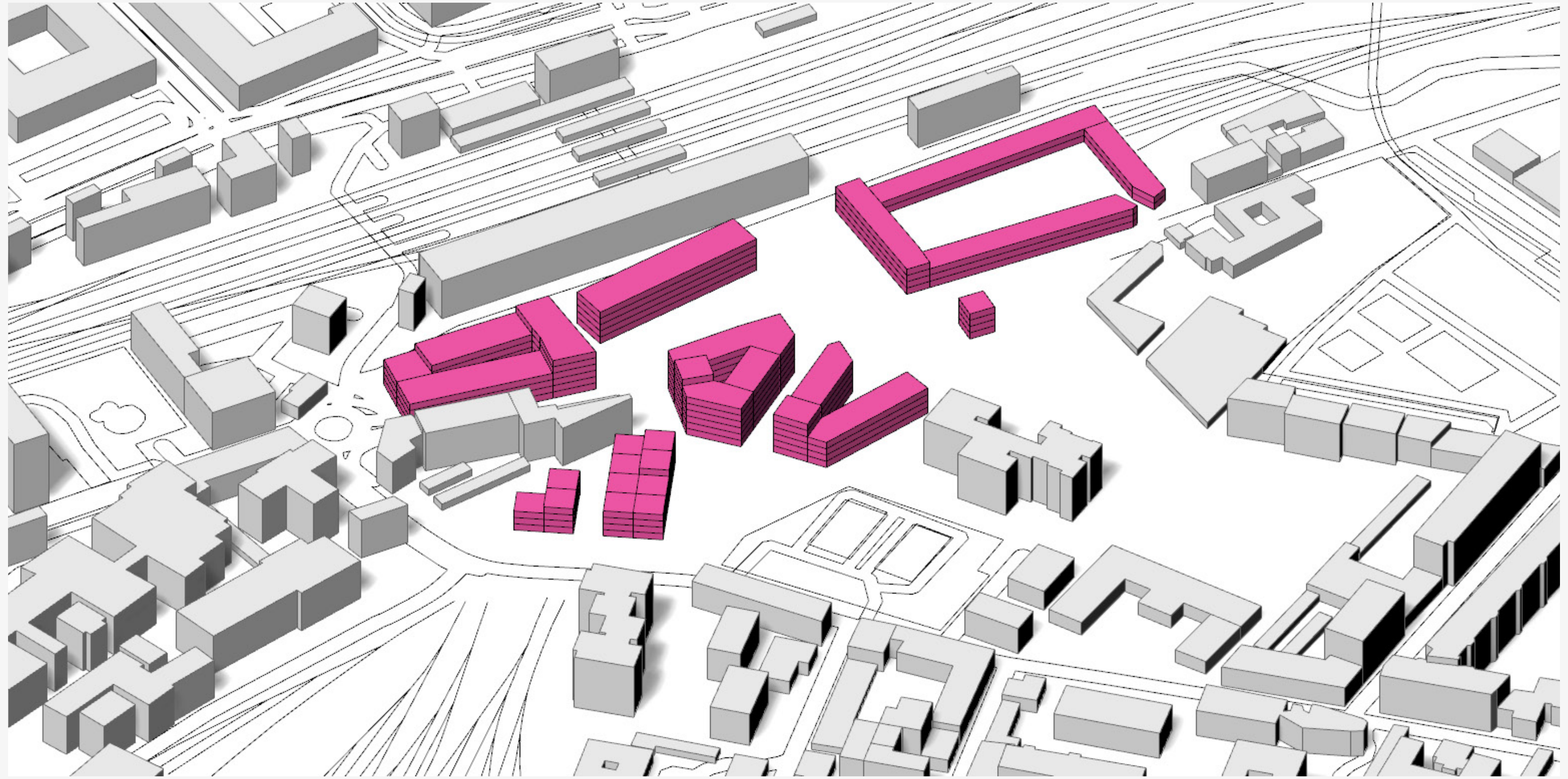
> GA requires an engine to run their simulations. In this study Wallacei has been integrated as the evolutionary engine since it can run evolutionary simulations and gives access to analytics and selections methods within Grasshopper.

Settings in the control panel are kept in default except from the population panel. After making numerous attempts, it is decided to keep population size on 5000 because of the computational power and time concerns.. Once objectives and gene values are properly connected to the wallacei panel, we started simulation and let the engine to fly!

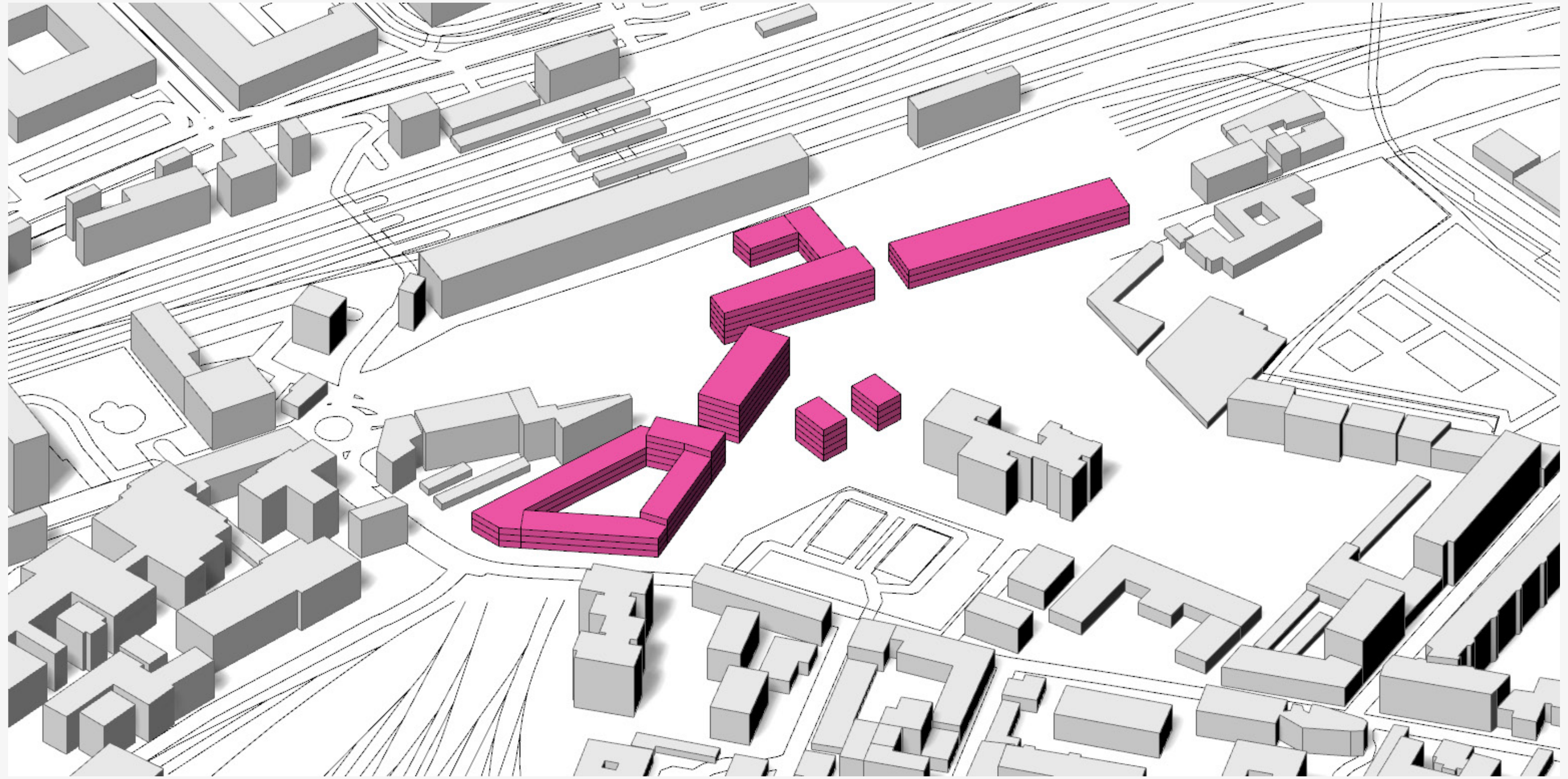




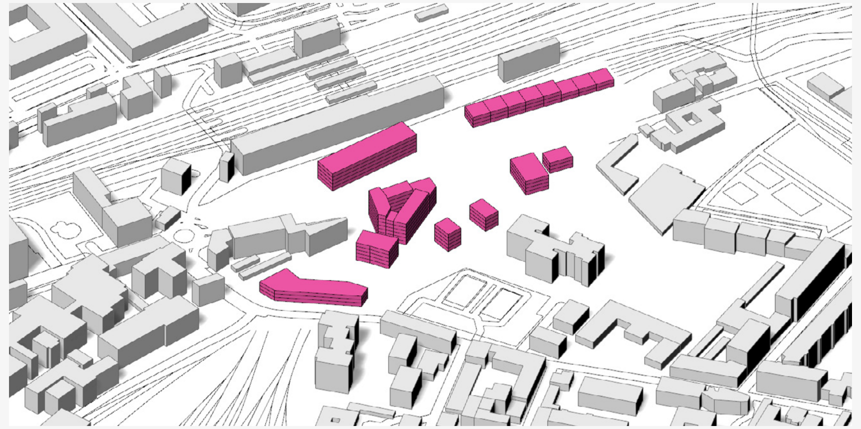
Generation 01 Individual 01



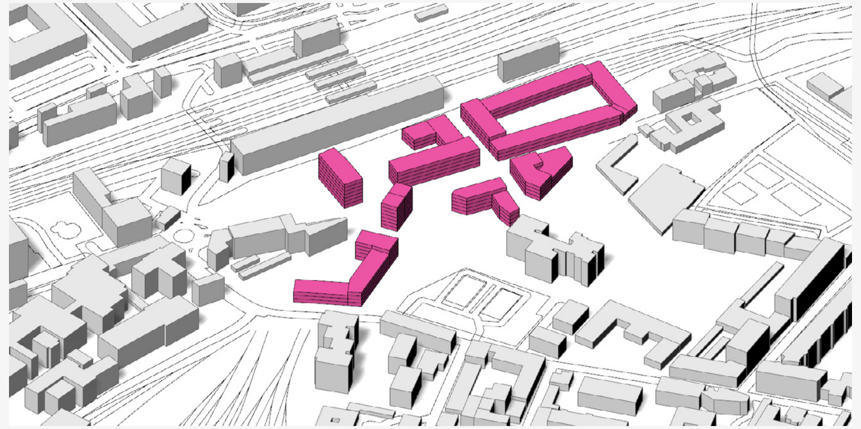
Generation 01 Individual 50



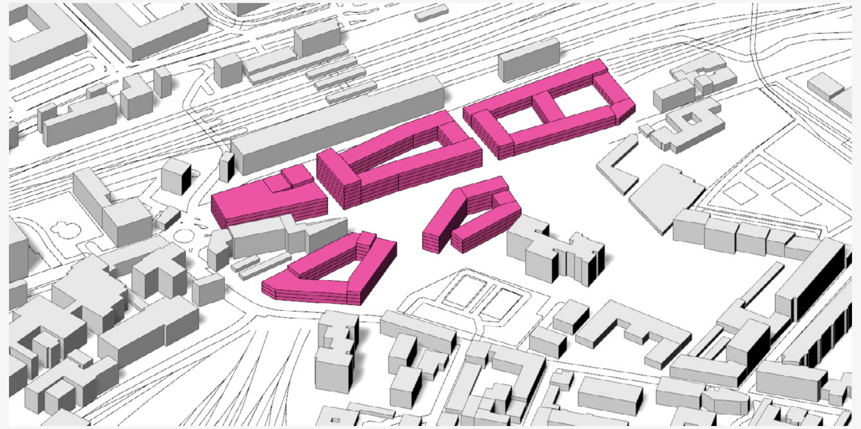
Generation 02 Individual 01



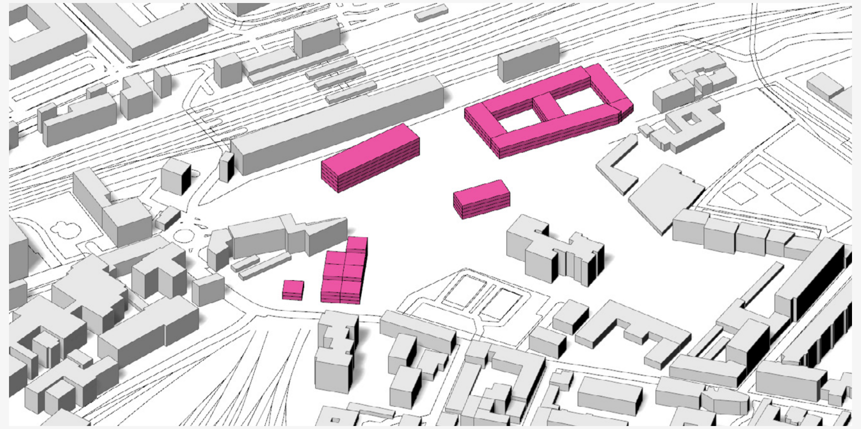
Generation 02 Individual 24



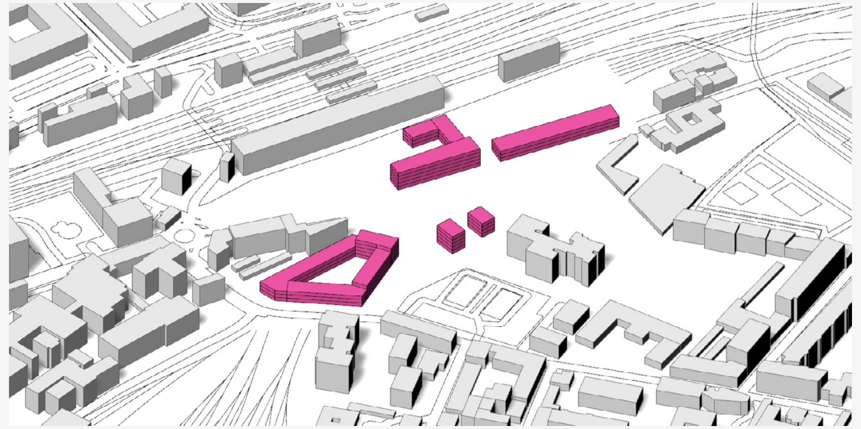
Generation 03 Individual 01



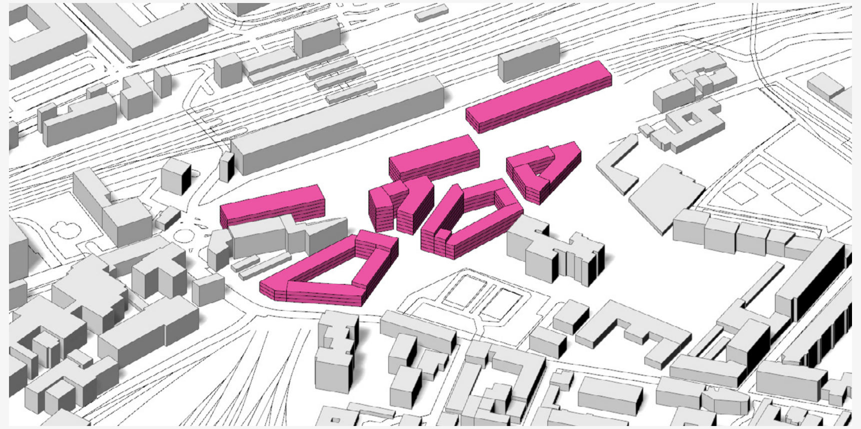
Generation 03 Individual 24



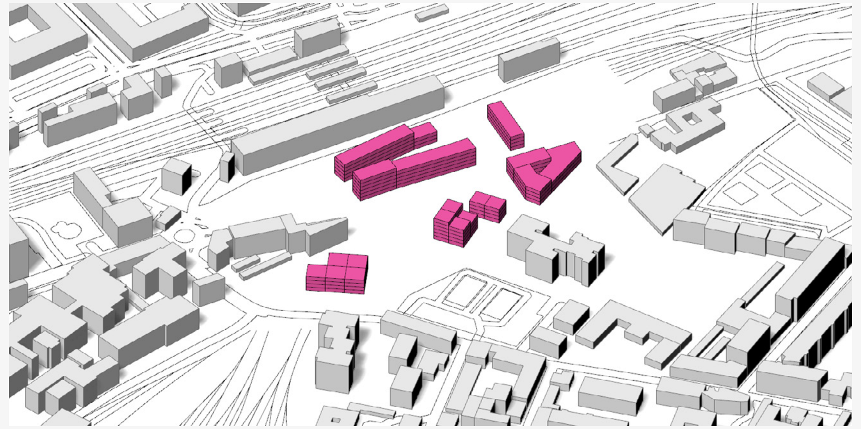
Generation 02 Individual 25



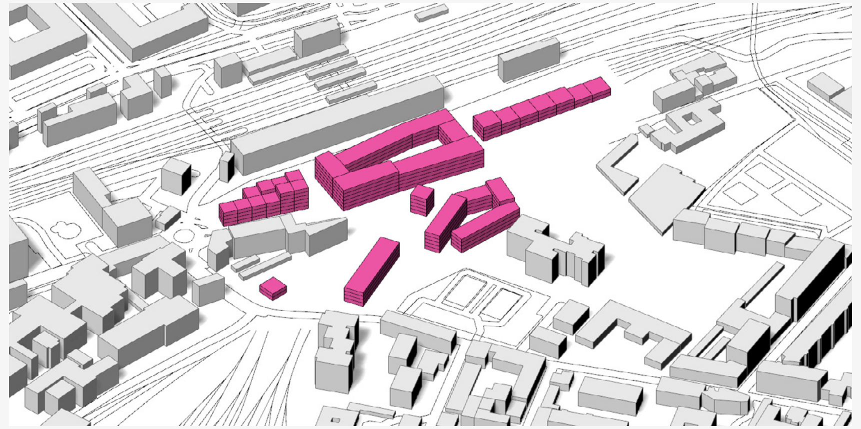
Generation 02 Individual 50

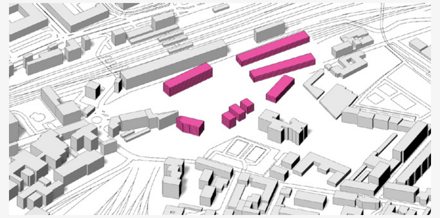
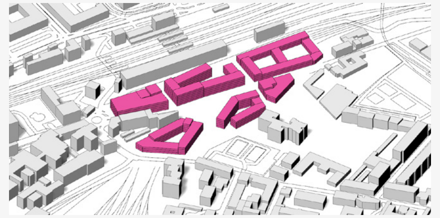
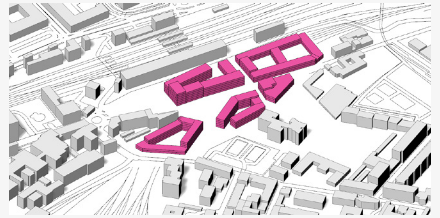
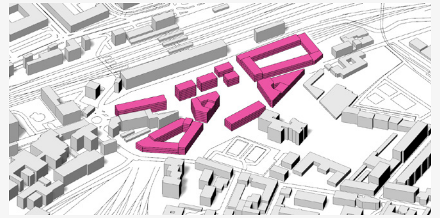
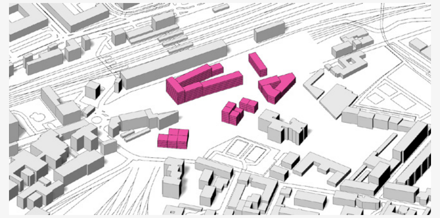
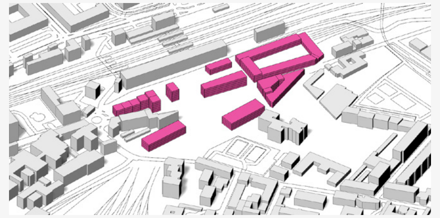
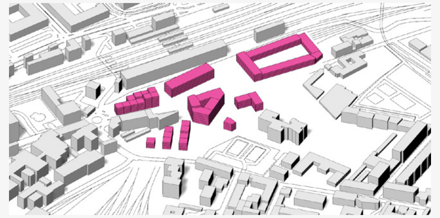
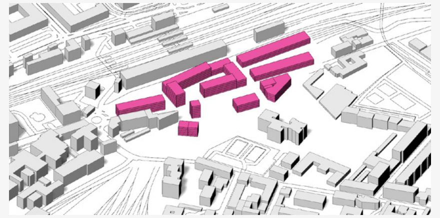


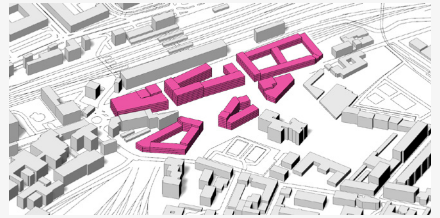
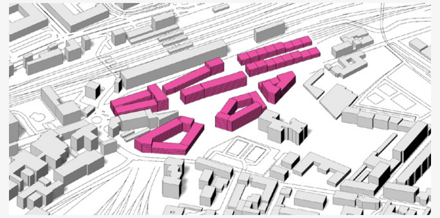
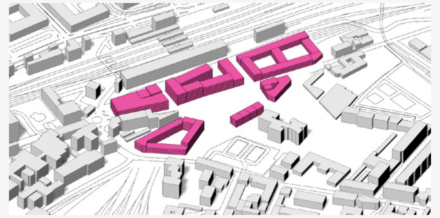
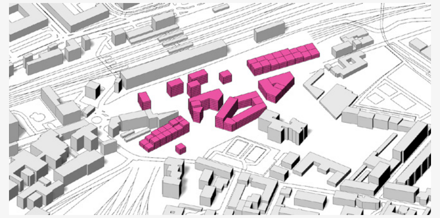
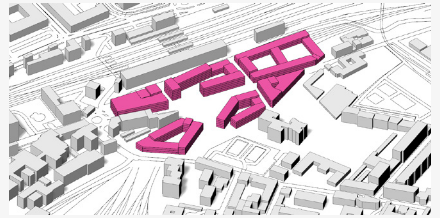
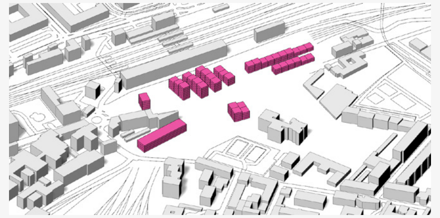
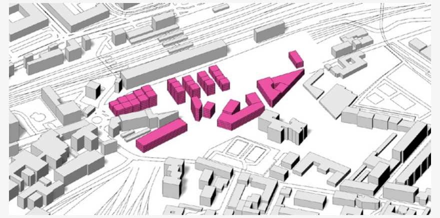
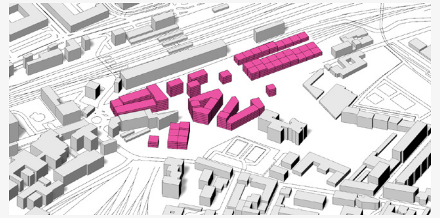
Generation 03 Individual 25

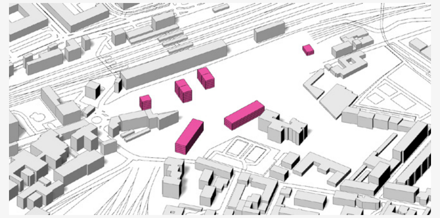
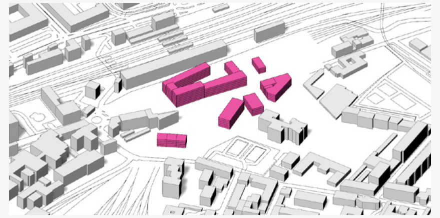
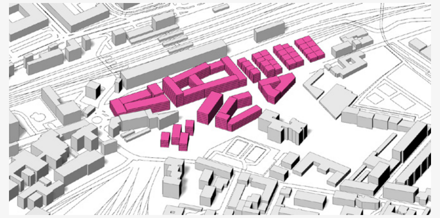
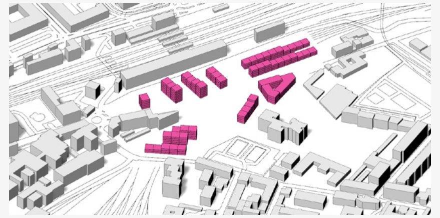
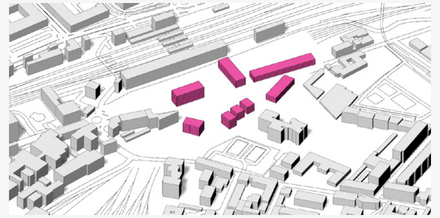
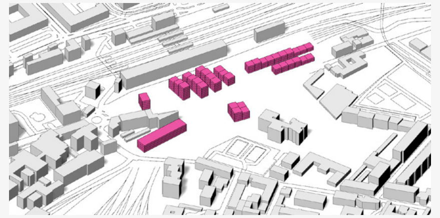
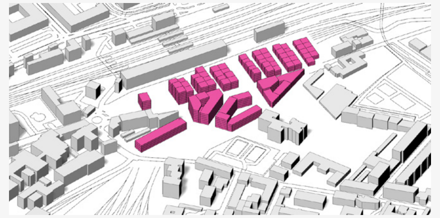
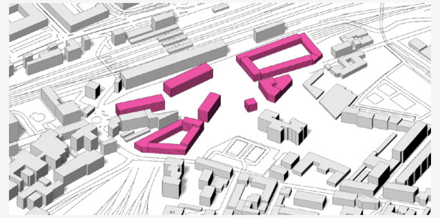


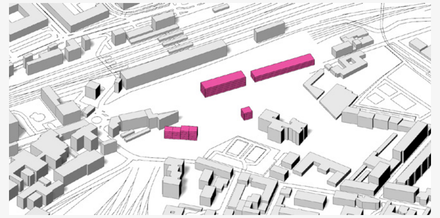
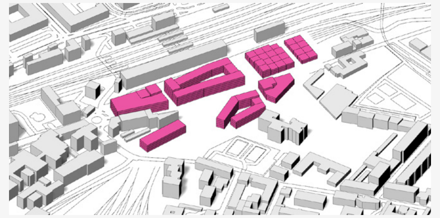
Generation 03 Individual 50

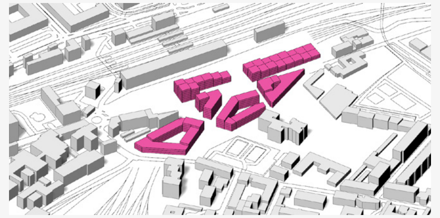
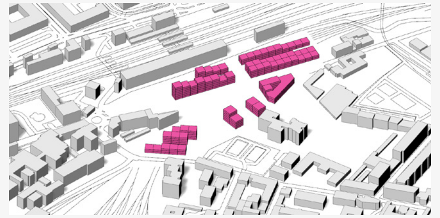
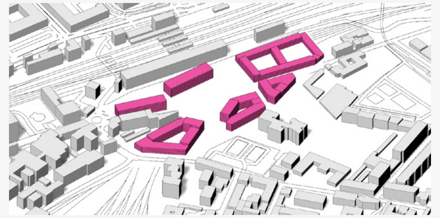
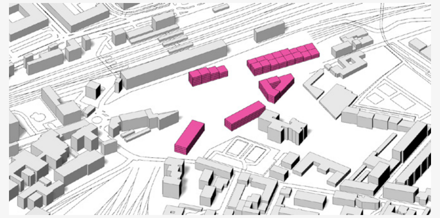
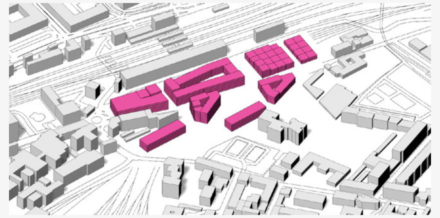


Generation 05 Individual 01

Generation 05 Individual 02

Generation 05 Individual 23

Generation 05 Individual 24

Generation 05 Individual 25

Generation 05 Individual 26

Generation 05 Individual 49

Generation 05 Individual 50


Generation 09 Individual 01

Generation 09 Individual 02

Generation 09 Individual 23

Generation 09 Individual 24

Generation 09 Individual 25

Generation 09 Individual 26

Generation 09 Individual 45

Generation 09 Individual 50


Generation 19 Individual 01

Generation 19 Individual 02

Generation 19 Individual 23

Generation 19 Individual 24

Generation 19 Individual 25

Generation 19 Individual 26

Generation 19 Individual 49

Generation 19 Individual 50


Generation 29 Individual 01

Generation 29 Individual 02

Generation 29 Individual 23

Generation 29 Individual 24

Generation 29 Individual 25

Generation 29 Individual 26

Generation 29 Individual 49

Generation 29 Individual 50


... 3.5.3. Running Simulation

... 3.5.4. Evolutionary Simulation Results

> During the simulation, we encountered several issues, including crashes due to high CPU demands of the algorithm and inappropriate clustering problems in Grasshopper (where some functions were not changing simultaneously while the engine manipulated gene values).

In the end, we conducted 10 simulations, and the final results are highlighted in the right Image. It is observed that generating 5000 design variations took approximately 12 hours. Since no null solutions were generated, the framework can be considered as functioning as expected. However, it is evident from the analytics graphics that to achieve optimized design solutions, we would require larger population sizes.

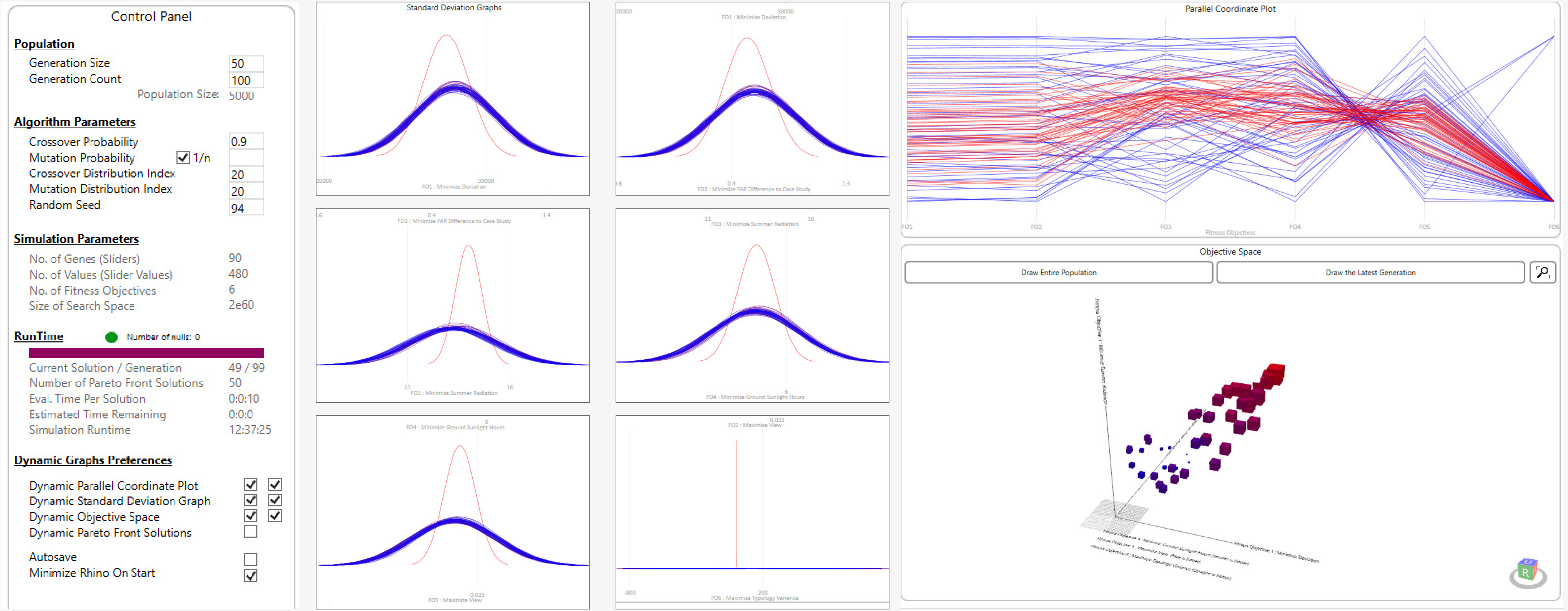
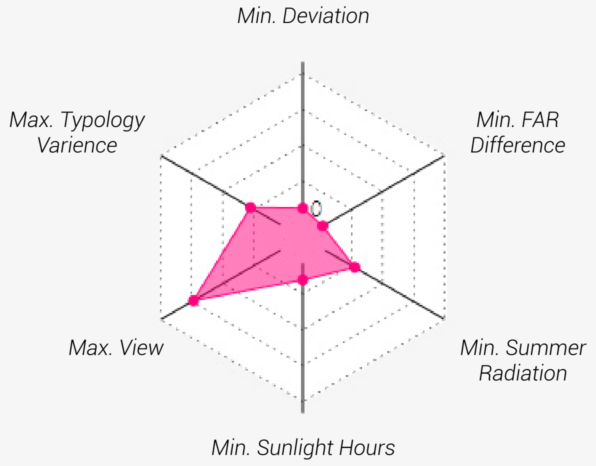
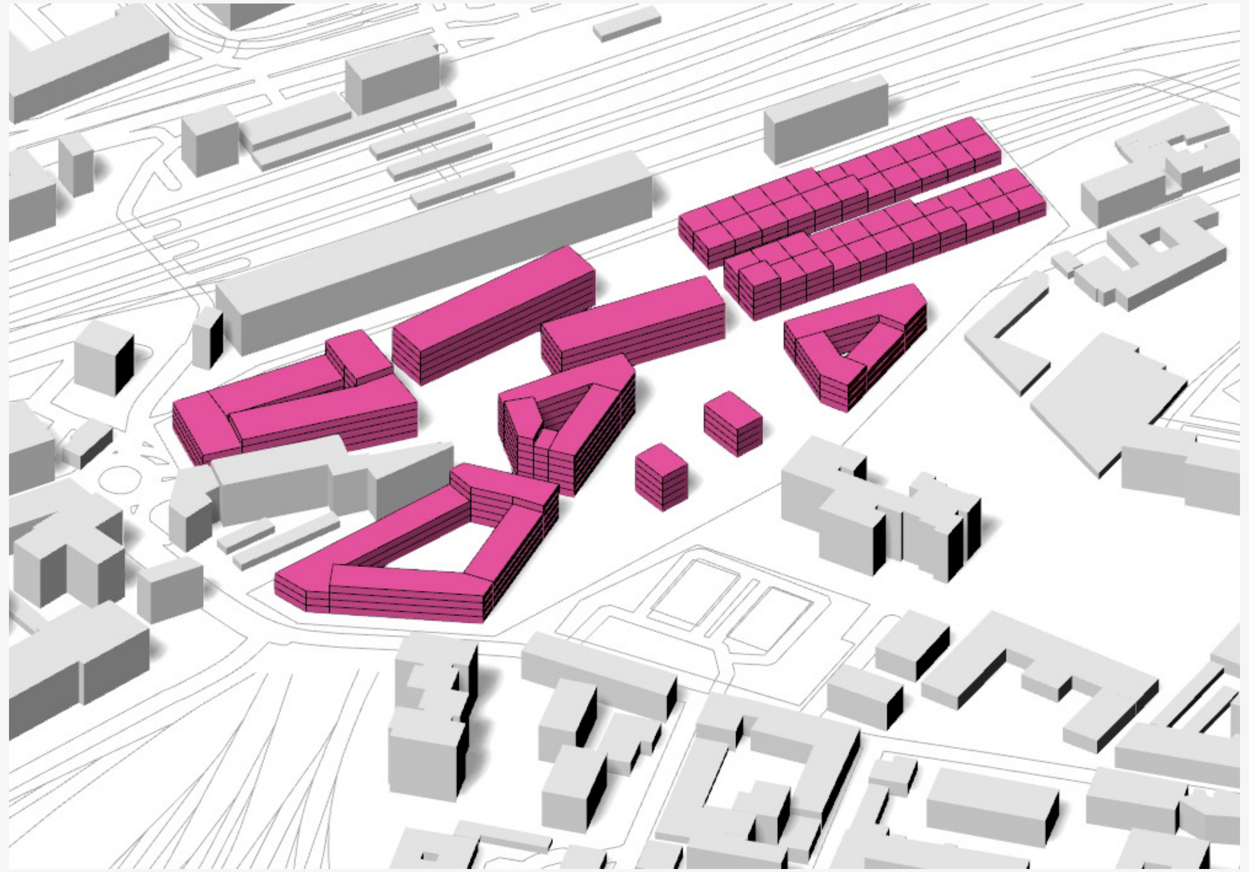


Figure 40: Simulation results and standart deviation of fitness values

... 3.5.5. Comparative Analysis of Selected Phenotypes

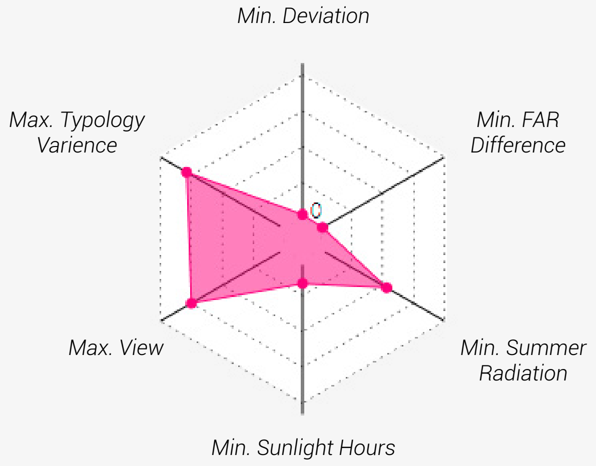
> After the simulation ended, two individuals were selected: G68 I33 and G99I 39. The first selection is based on the test objective, which is the highest number of fitness value that has for typology variance index. Since more than one option shares the same fitness value, they are all sorted based on their other fitness values as well, and the one with the highest performance is selected amongs those individuals.

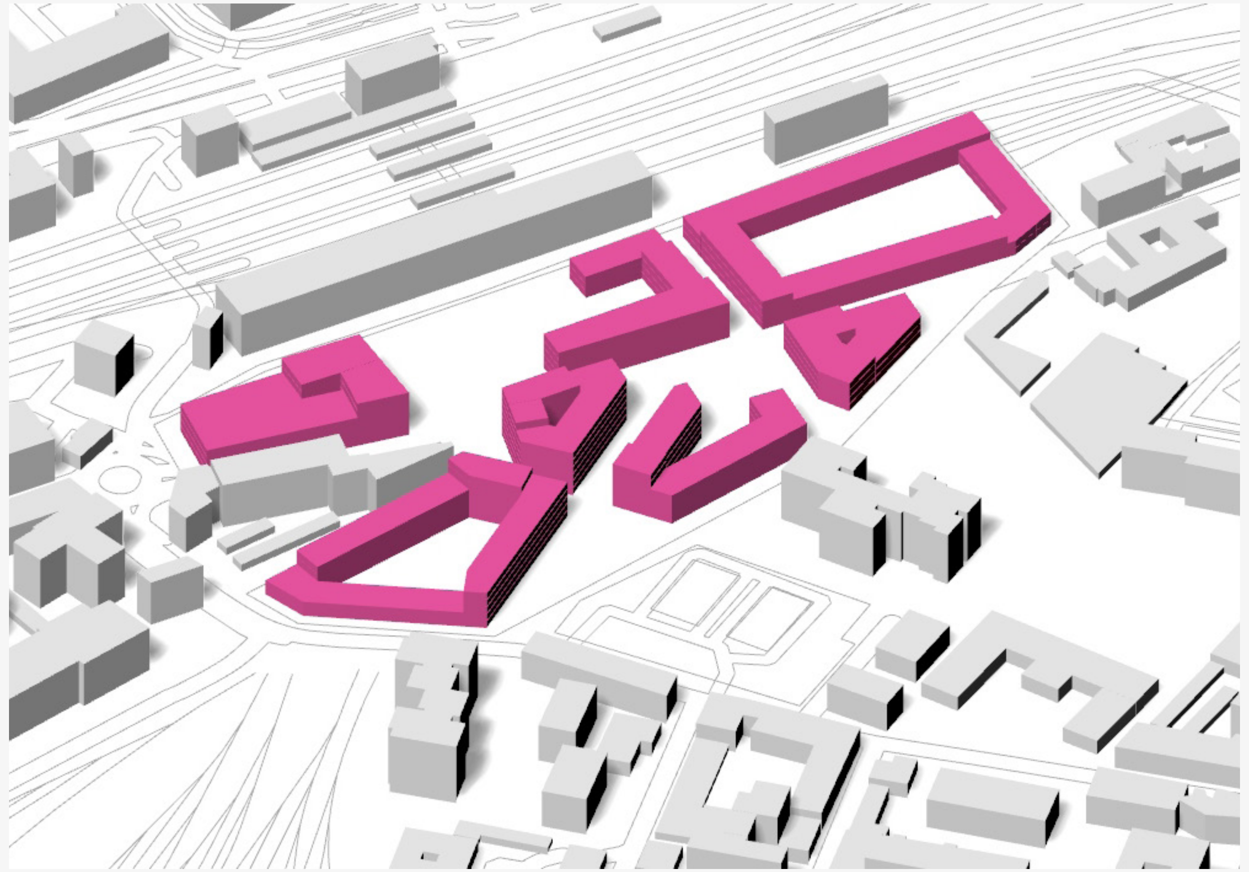




Generation 68 Individual 33

To introduce a contradiction and stimulate discussion, a controversial individual is selected, despite exhibiting a minimum amount of variety. It has demonstrated high performance in the remaining fitness values.





Generation 99 Individual 39

... 3.5.5. Comparative Analysis of Selected Phenotypes

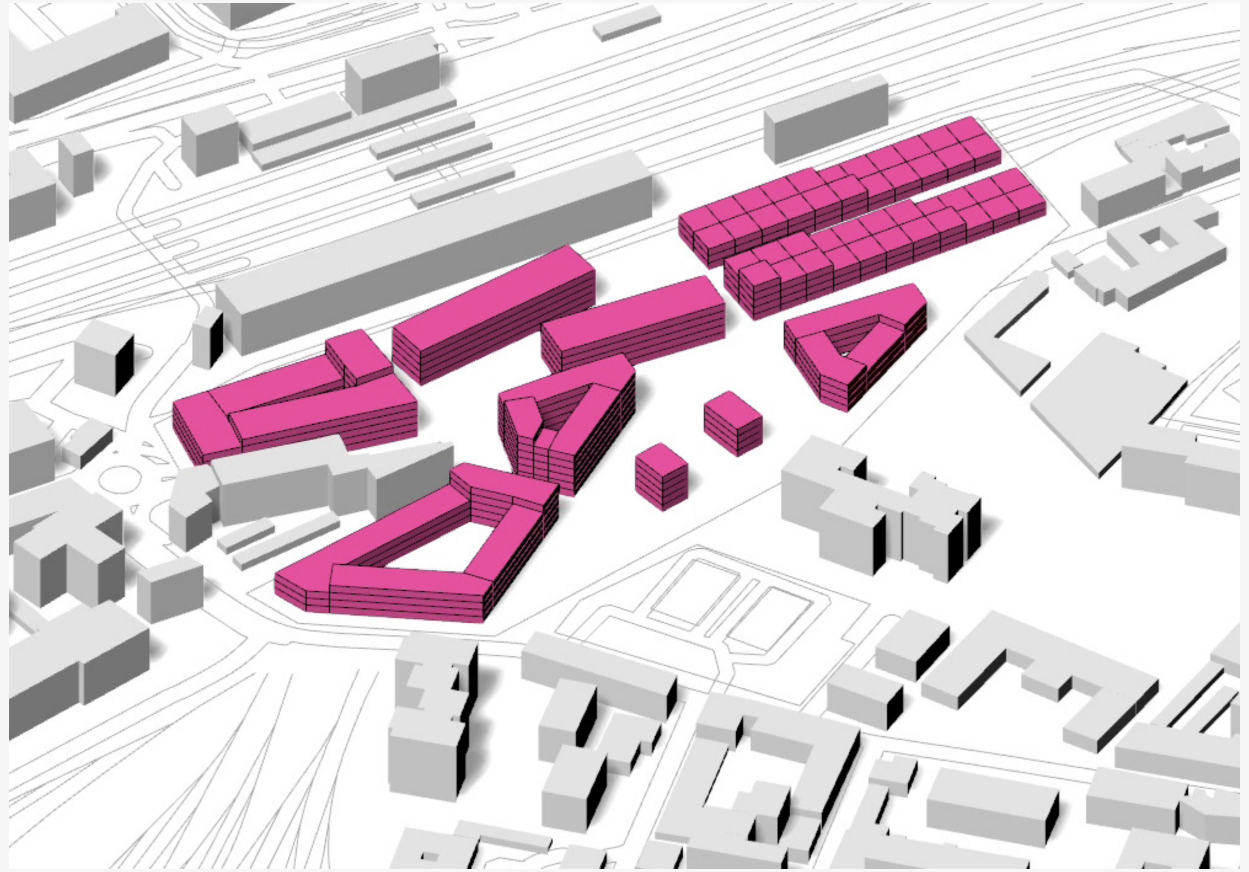
> Since the aim of this simulation is to maximize the typology variety index while keeping the FAR and find optimal design options for comparison with the case study, G68 I39 has been selected as the final result. It is observed that G68I33 outperformed the Innesto project in terms of fitness values, including Summer Radiation, Average Sunlight Hour, and Typology Variety Index, with significant differences.

Annual Summer Radiation
per 1m² Surface:
11.97 kWh/m²/year

Annual Avarage Sunlight
Hour on Outdoor:
3.95 h

View Percentage
of Inhabitants:
%43.47

Typology Variety
Index:
1.66



FAR: **1.47**

Generation 68 Individual 33

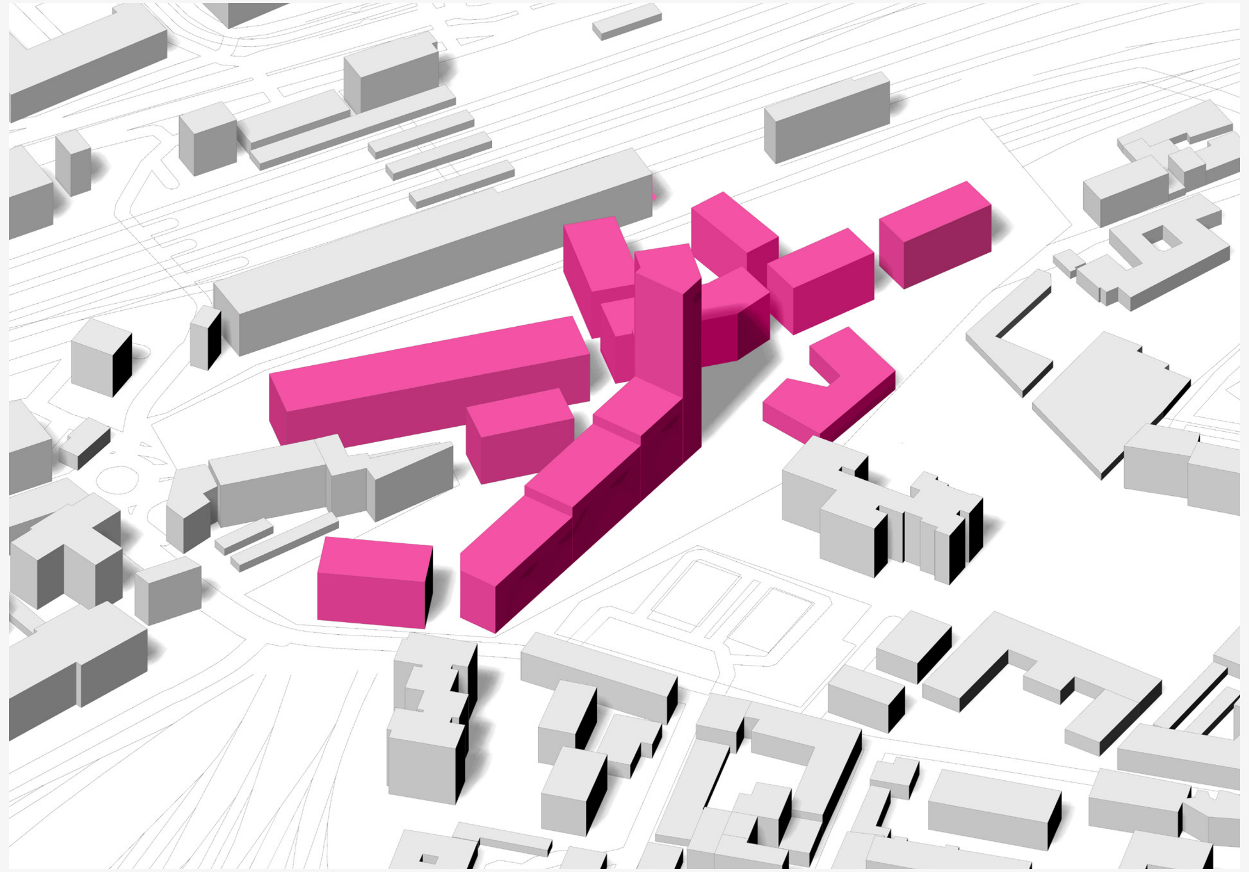
Additionally, it had a slightly lower View Percentage of Inhabitants. Even though the tool passed the test for quantitative values, it is important to note that comparing design solutions should not be solely based on quantitative values, and a comprehensive approach is necessary. This is discussed in the conclusion of the paper, while this experiment is concluded here.

Annual Summer Radiation
per 1m² Surface:
14.70 kWh/m²/year

Annual Avarage Sunlight
Hour on Outdoor:
4.77 h

View Percentage
of Inhabitants:
%44.35

Typology Variety
Index:
0.16



FAR: **1.48**

Case Study: Innesto Project

design phase 01

... P-03 MANUAL EXPLORATION MODE

> In the following part of the paper, the potential and limitations of the tool is tested through case studies involving extreme urban layout scenarios from around the world within the same site boundry. The testing aims to examine the tool's capabilities and constraints in handling challenging urban design situations. By analyzing these case studies, the reader seek to gain a comprehensive understanding of the tool's effectiveness in generating solutions for complex urban environments.

Design Phase:1 Part:3

... 3.6. Extreme Scenarios

... 3.6.1 Scenario:1 Low Rise High Density

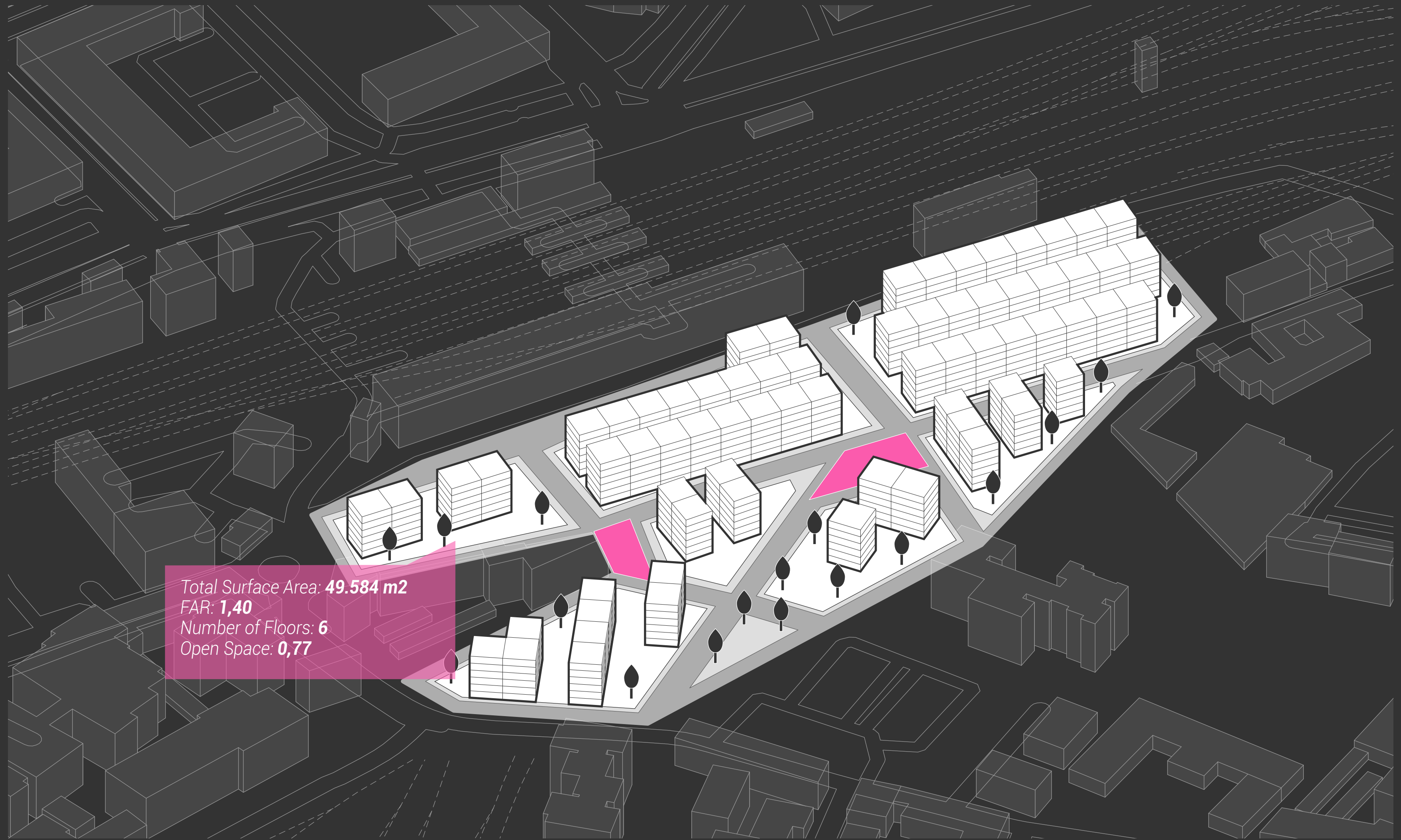
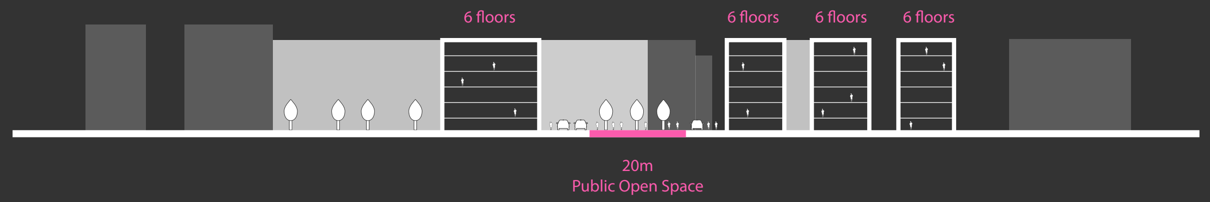


Figure 41: Sluseholmen , Copenhagen, Denmark aeiral view

Design Phase:1 Part:3

... 3.6. Extreme Scenarios

... 3.6.1 Scenario:1 Low Rise High Density



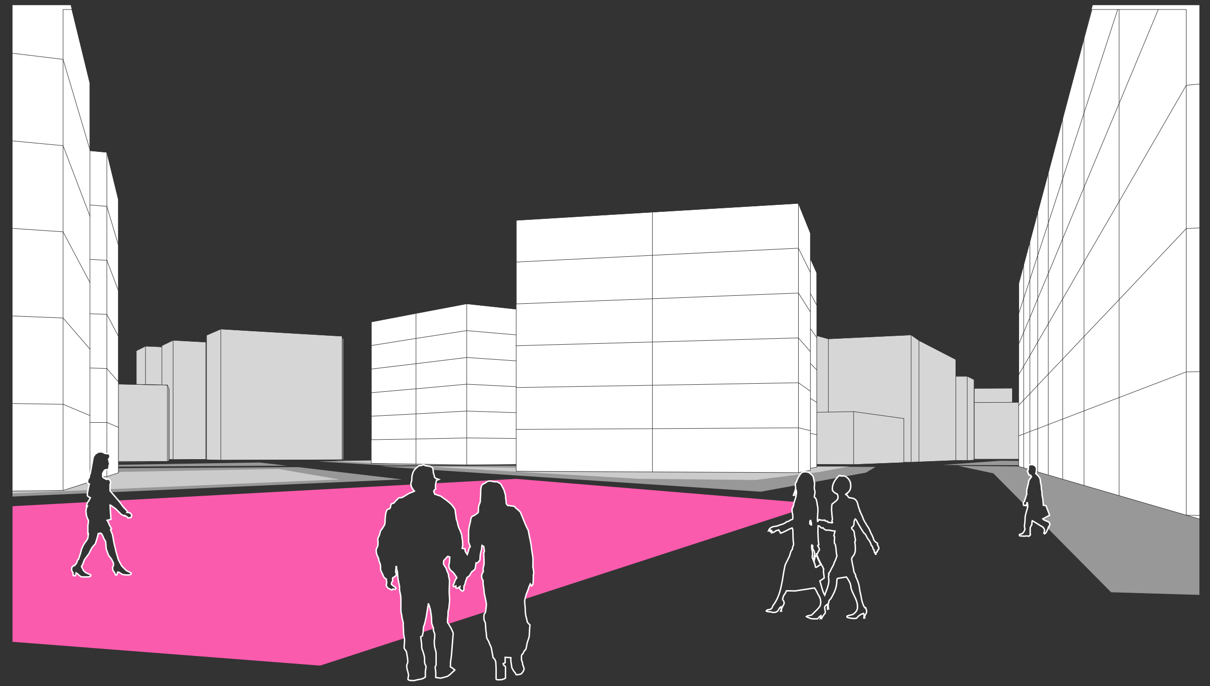


Figure 42: Sluseholmen , Copenhagen, Denmark human-eye view

... 3.6.2. Scenario:2 Courtyard Blocks

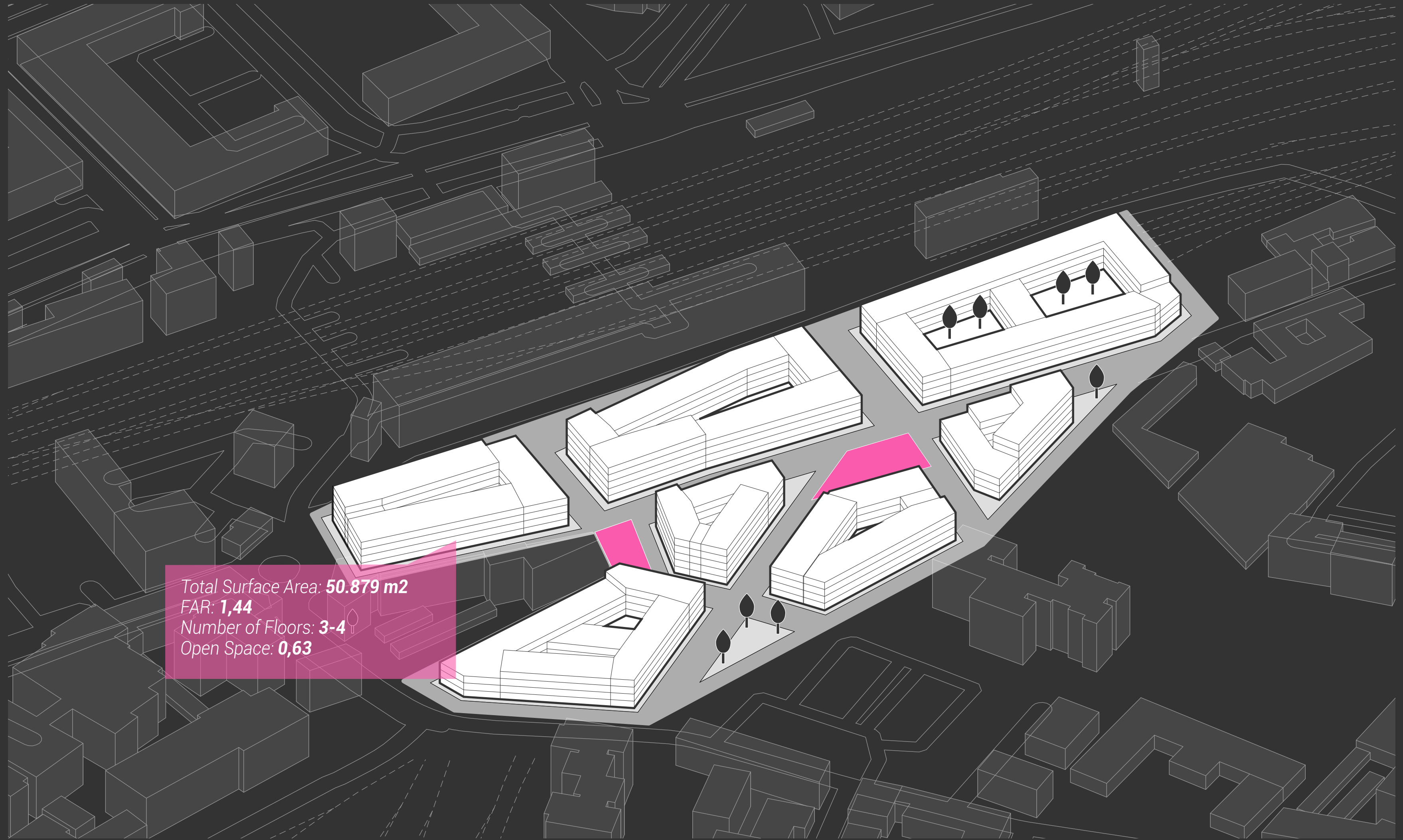
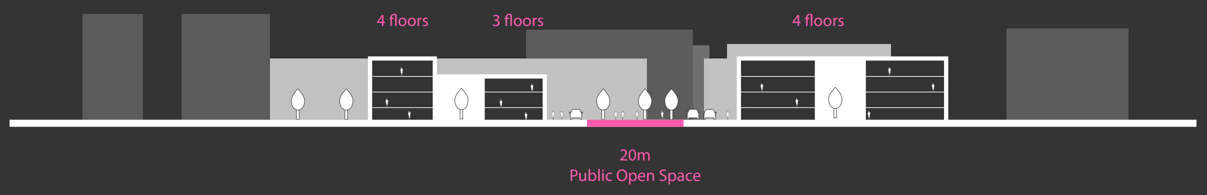



Figure 43: Barcelona superblocks

Design Phase:1 Part:3

... 3.6. Extreme Scenarios

... 3.6.2. Scenario:2 Courtyard Blocks



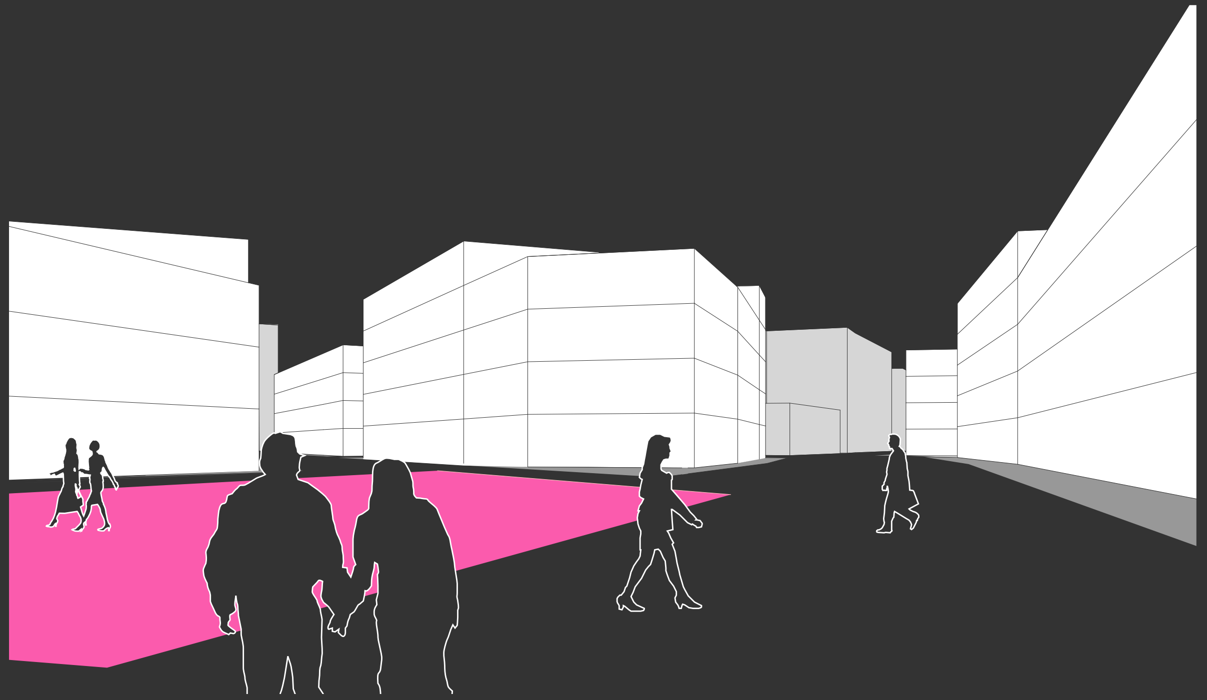


Figure 44: Barcelona superblocks human-eye view

... 3.6.3. Scenario:3 Tower City

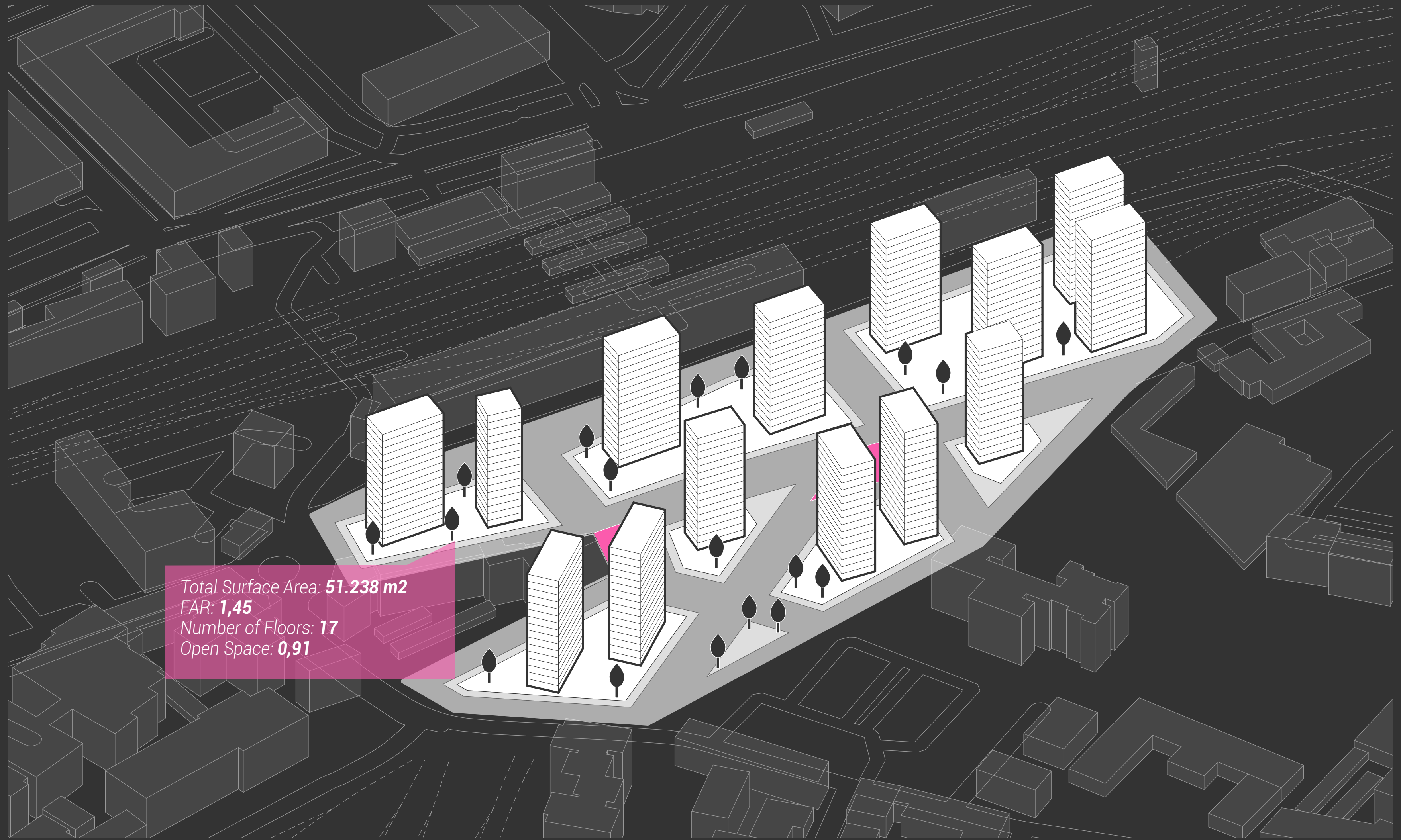


Figure 45: Soyak houses Izmir, Turkey: aerial view

Design Phase:1 Part:3

... 3.6. Extreme Scenarios

... 3.6.3. Scenario:3 Tower City

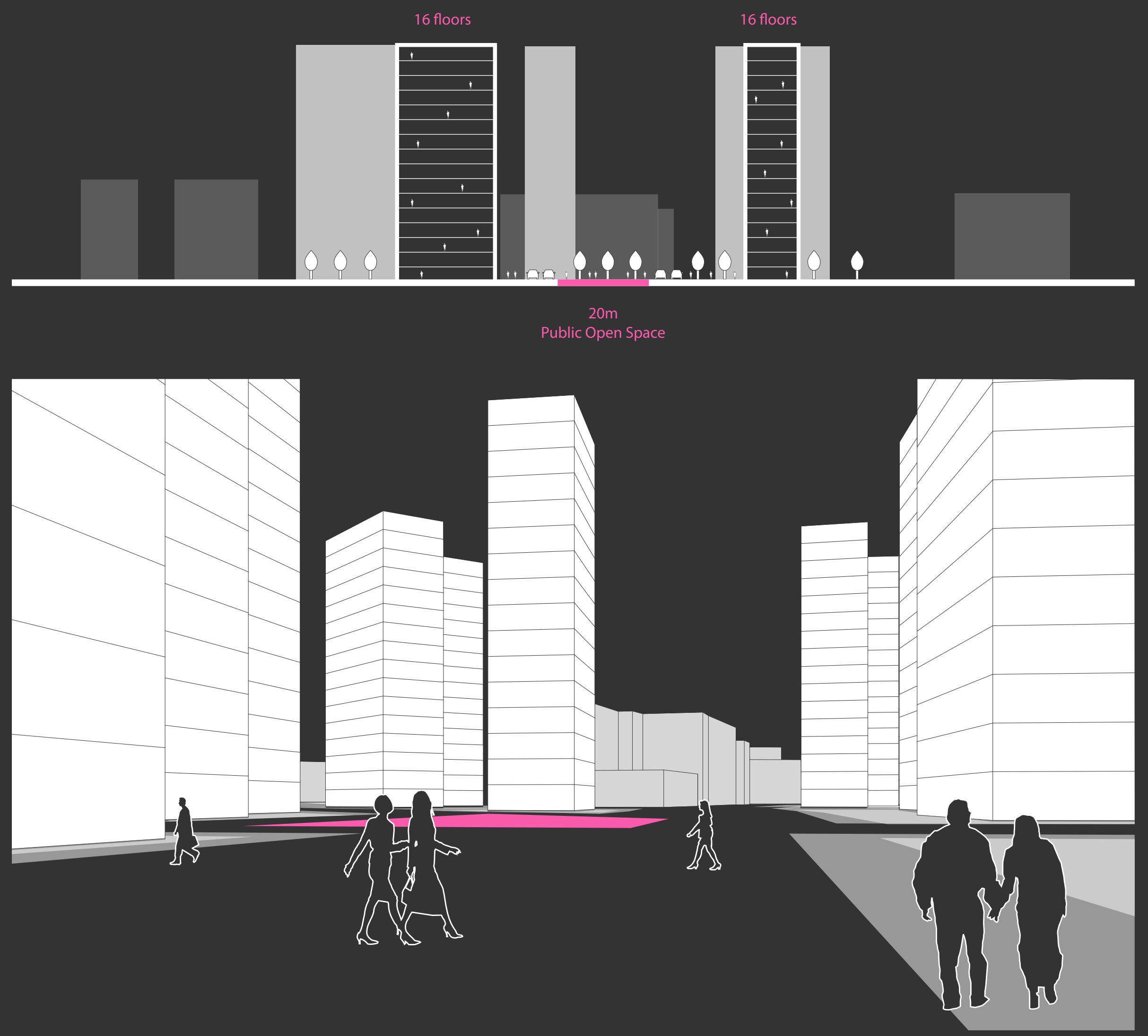


Figure 46: Soyak houses Izmir, Turkey: human-eye view

Design Phase:1 Part:3
... 3.6. Extreme Scenarios
... 3.6.4. Scenario:4 Mixed Typology

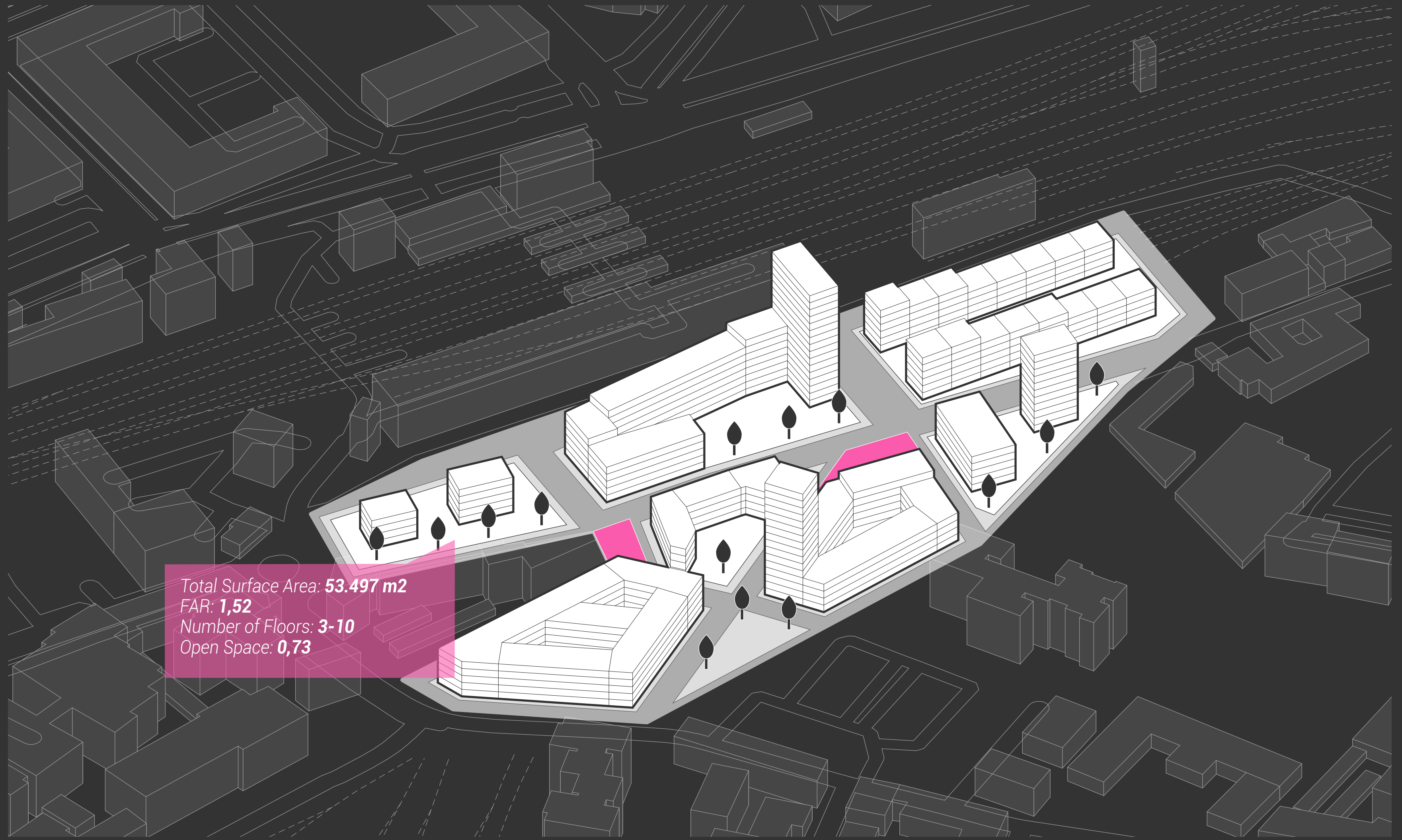
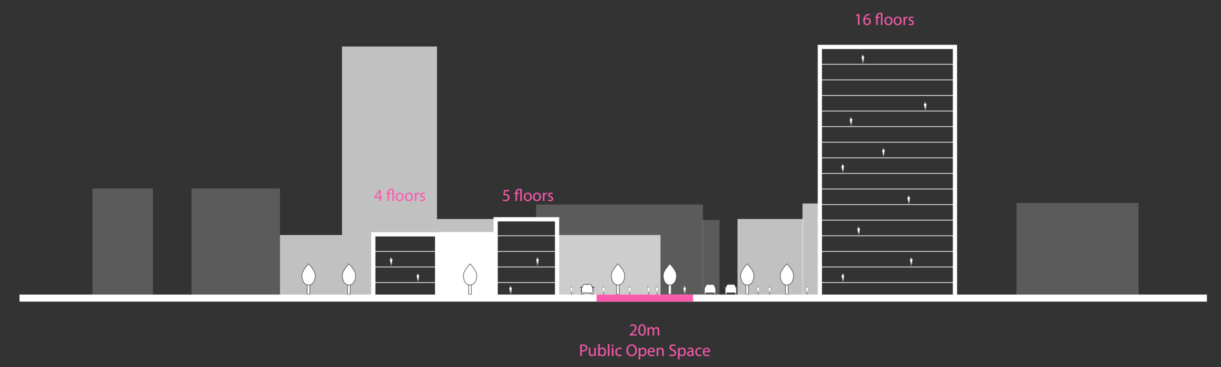


Figure 47: Carlsberg city Copenhagen

... 3.6.4. Scenario:4 Mixed Typology



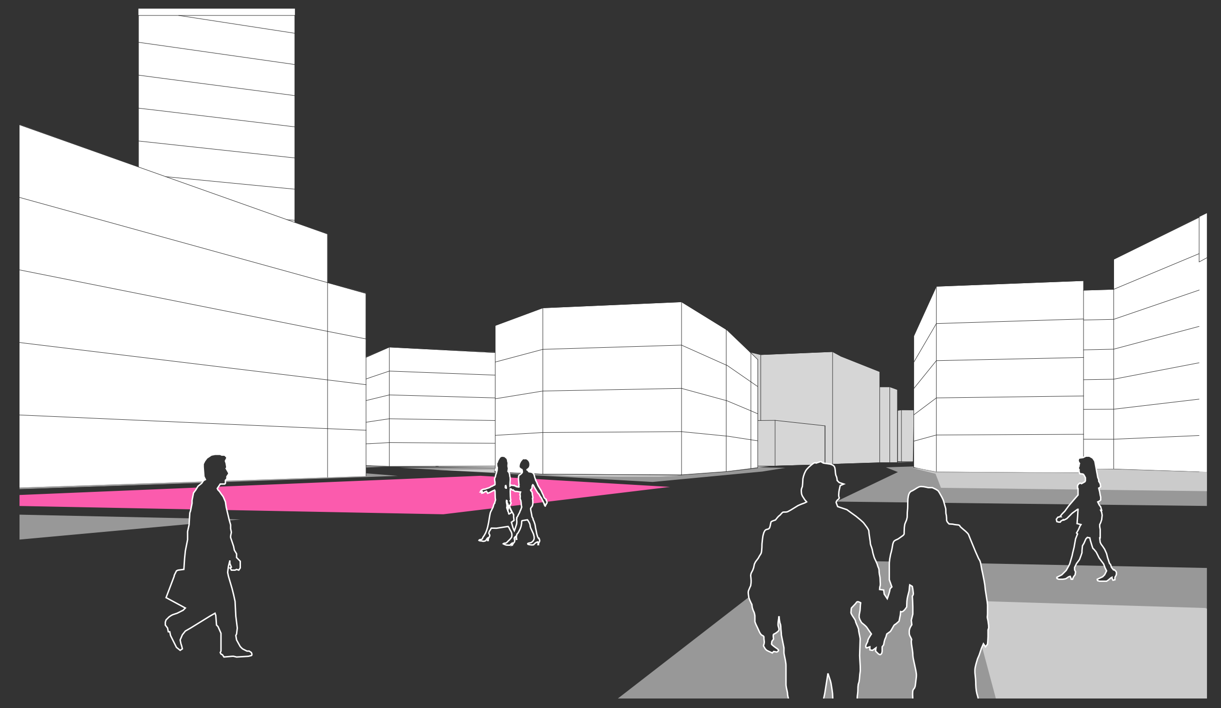


Figure 48: Carlsberg city Copenhagen: human-eye view

... 3.6.5. Scenario:5 Piazza

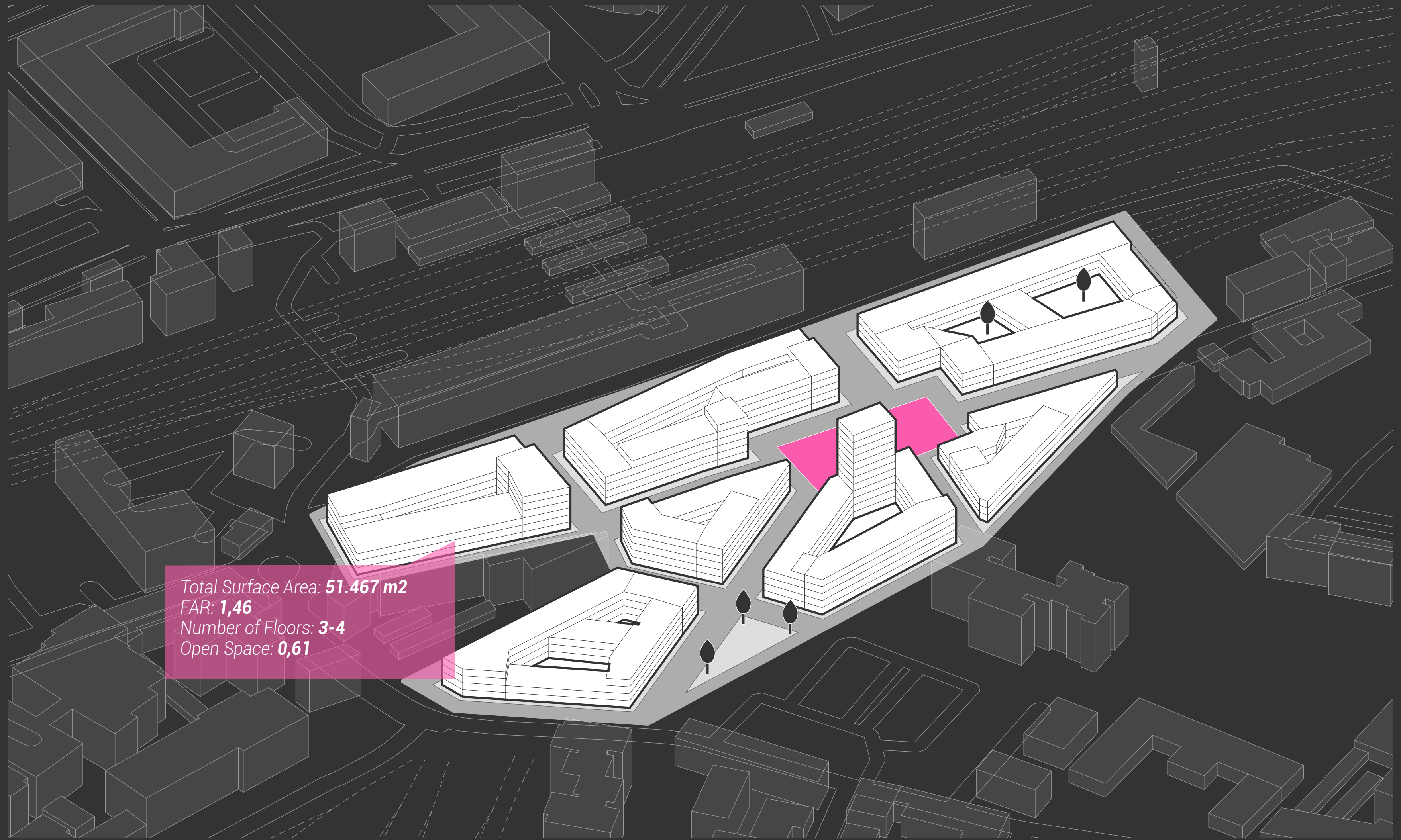
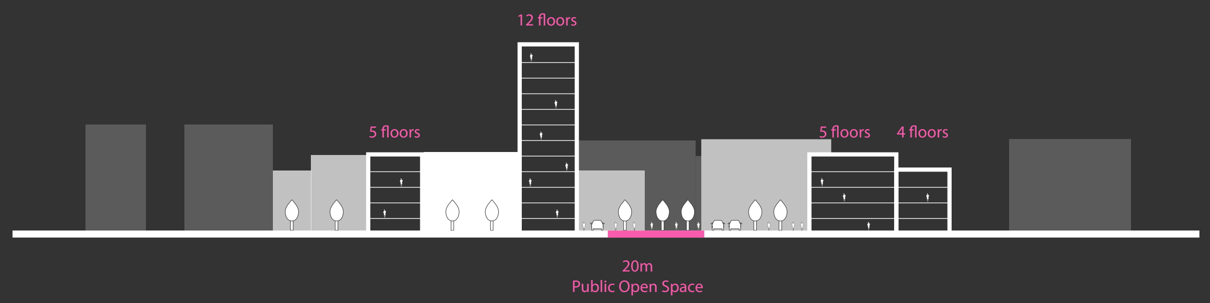


Figure 49: Piazza Castello, Turin, Itay aerial view

... 3.6.5. Scenario:5 Piazza



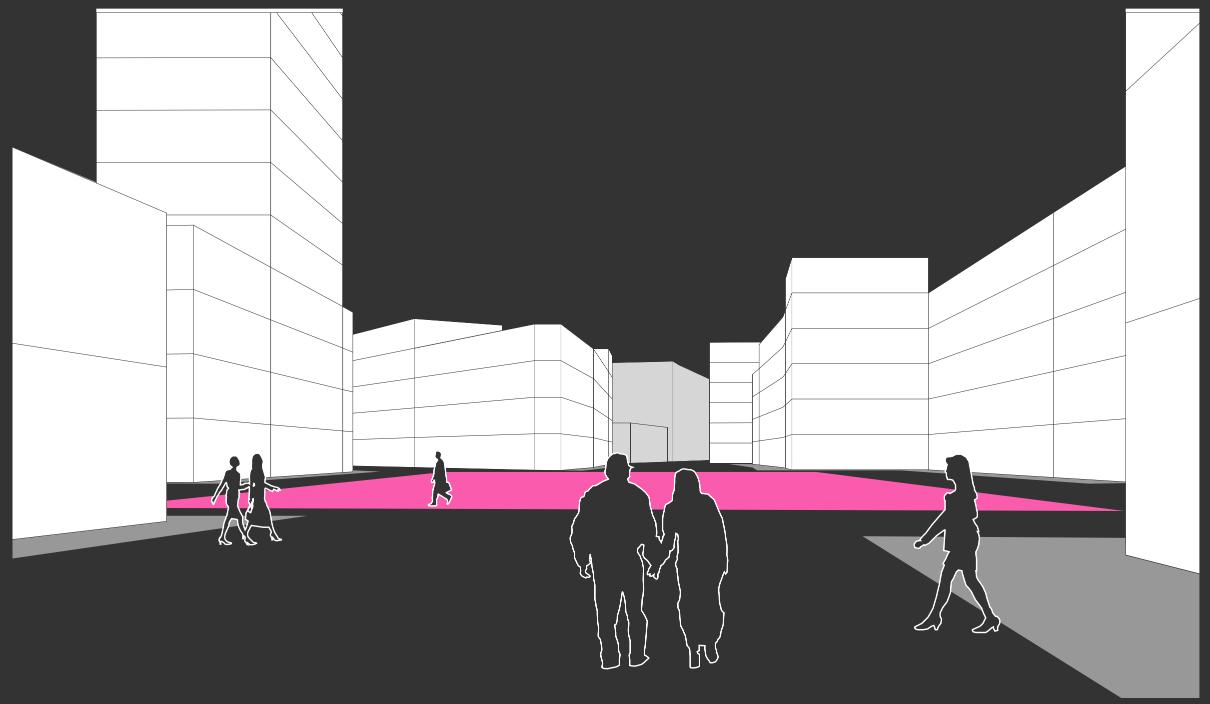


Figure 50: Piazza Castello, Turin, Itay: human-eye view

... 3.6.6. Scenario:6 Central Park

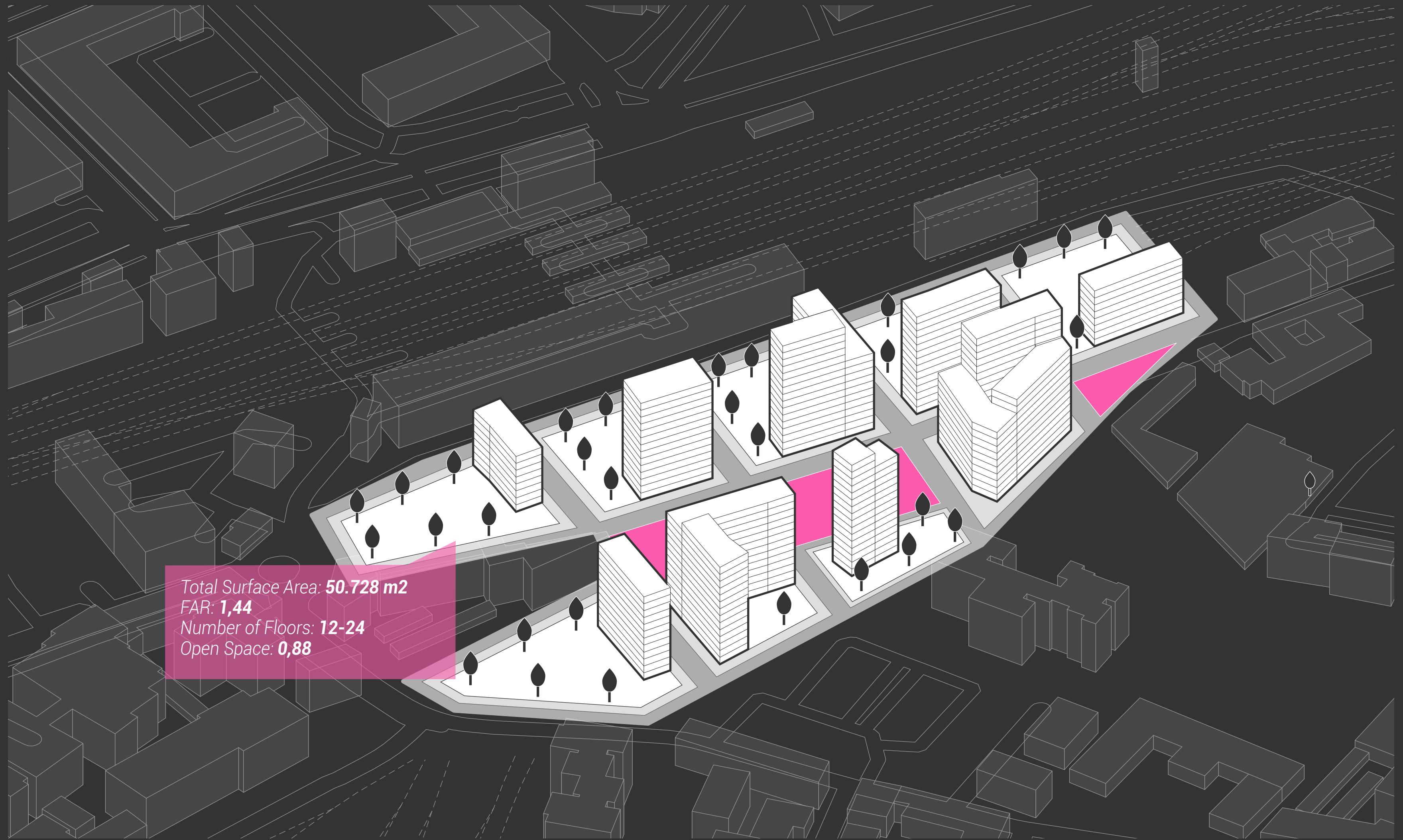
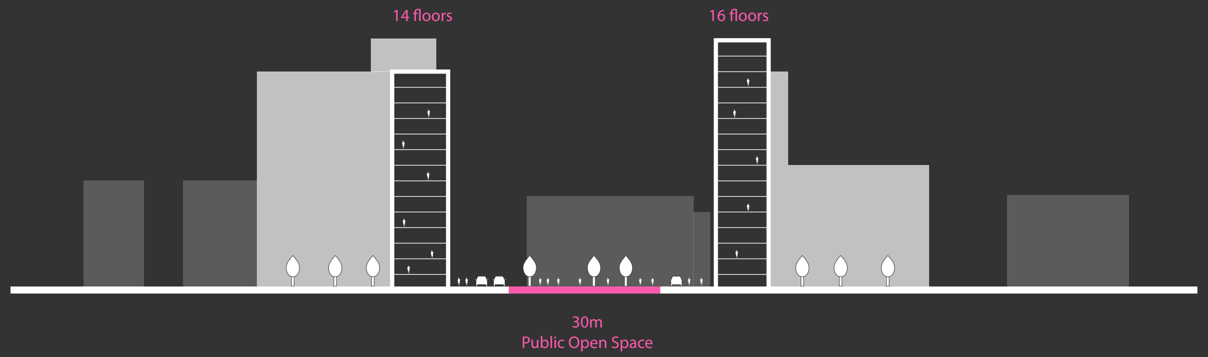


Figure 51: The Central Park, New York, USA

... 3.6.6. Scenario:6 Central Park



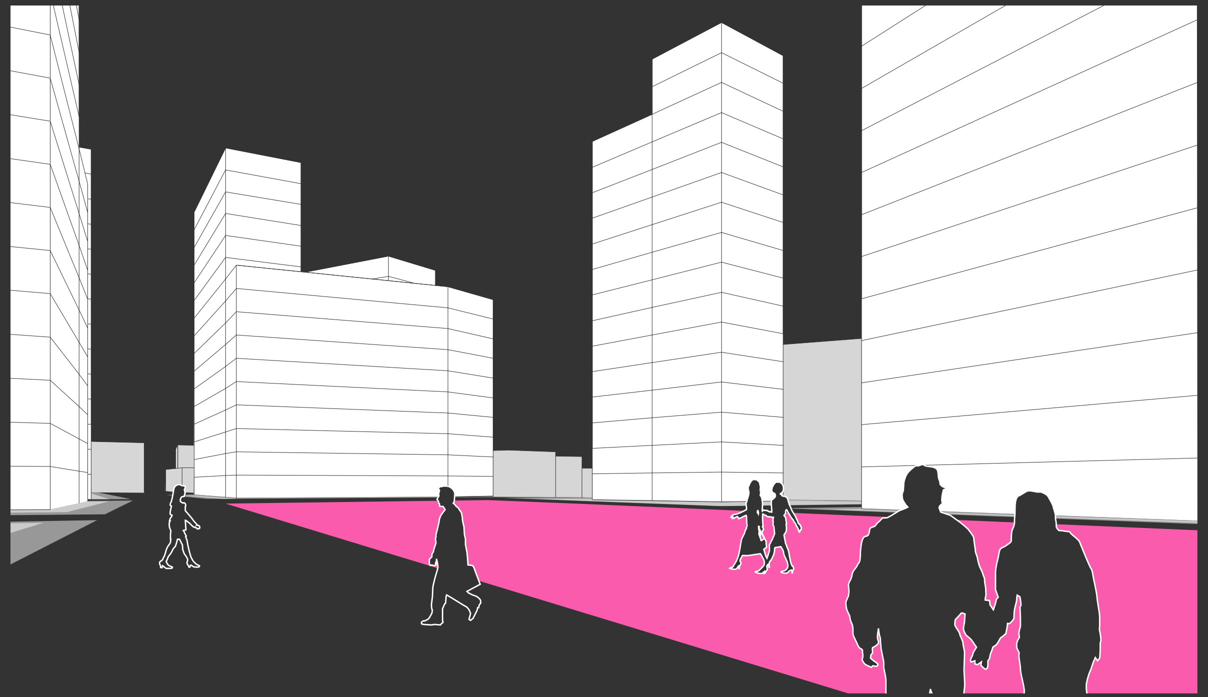


Figure 52: The Central Park, New York, USA: human-eye view

Design Phase:1 Part:3

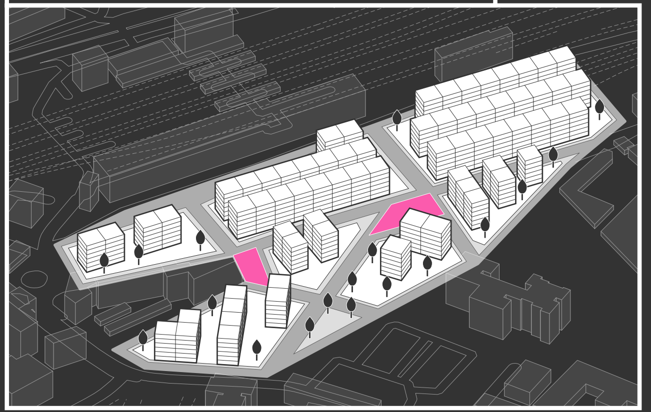
... 3.7. Comparative Analysis of Extreme Scenarios

... 3.7.1. Morphological Criterias

> The images on the right row of the page show-case, each with its own unique characteristics and qualities. Here we compare these options based on three key aspects: Typology Variety Index, Permeability, and Open Space ratios:

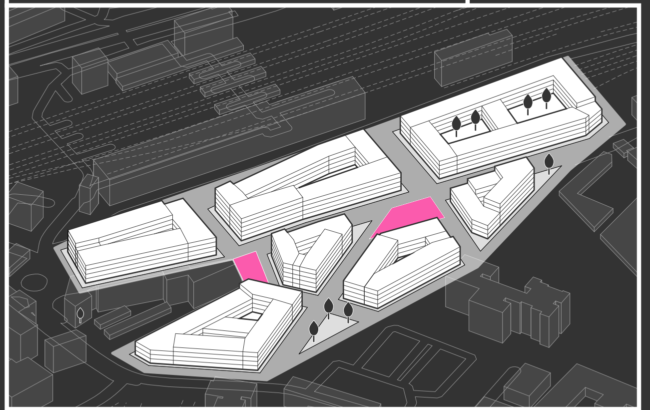
Upon analyzing the results of different master plan configurations, it is evident that each option presents its own set of outcomes. The open space ratios vary across the plans, with some offering higher amounts of open space while others provide less. Permeability levels also differ, impacting the ease of movement within the areas. Furthermore, the presence or absence of typology variety influences the diversity and functionality of the urban space. Taking these factors into account, it is important for designer to carefully consider the trade-offs and priorities when selecting a master plan that best suits the specific needs and goals of the project.

Scenario-1: L. Rise H. Density



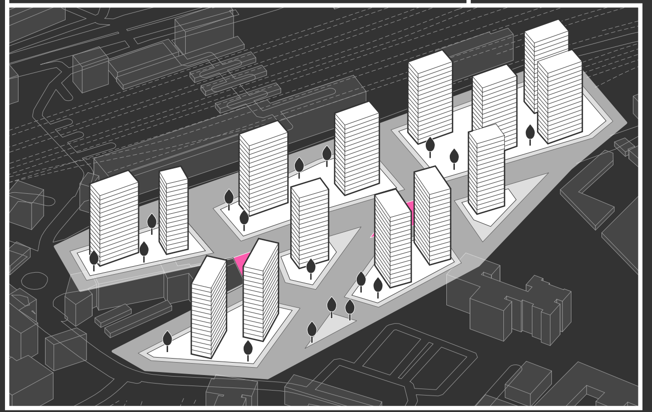
Typology Variety Index: ***0,00***
Permeability: ***Mediocre***
Open Space: ***0.77 (27003 m2)***

Scenario-2: Courtyard Blocks



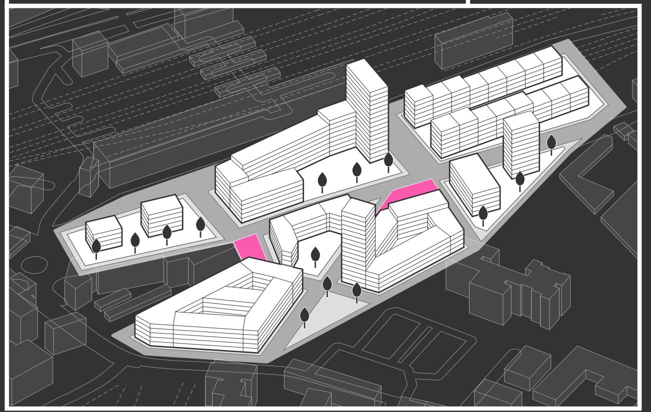
Typology Variety Index: ***0,00***
Permeability: ***Low***
Open Space: ***0.63 (22021 m2)***

Scenario-3: Tower City



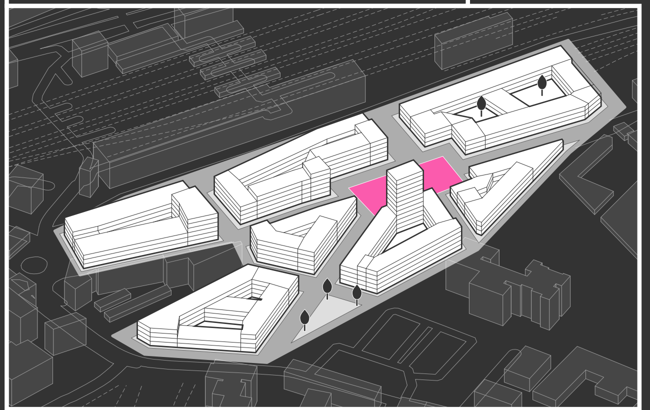
Typology Variety Index: ***0,00***
Permeability: ***High***
Open Space: ***0.91 (32101 m2)***

Scenario-4: Mixed Typology



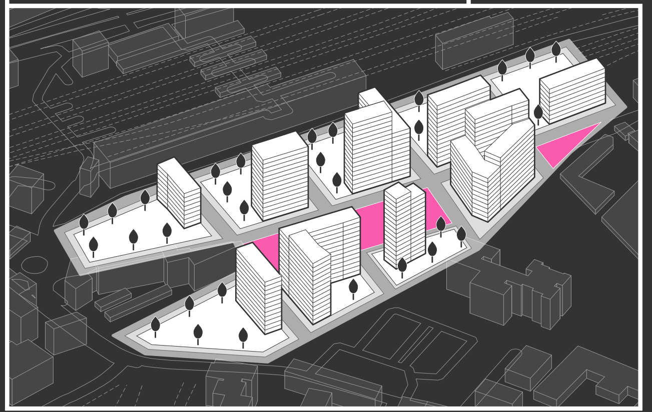
Typology Variety Index: ***1,62***
Permeability: ***Mediocre***
Open Space: ***0,73 (25830 m2)***

Scenario-5: Piazza



Typology Variety Index: ***0,59***
Permeability: ***Low***
Open Space: ***0,61 (21636 m2)***

Scenario-6: Central Park



Typology Variety Index: ***1,55***
Permeability: ***High***
Open Space: ***0,88 (31041 m2)***

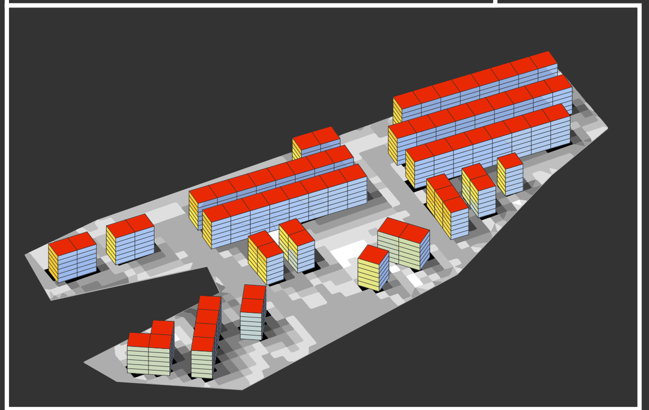
Design Phase:1 Part:3

... 3.7. Comparative Analysis of Extreme Scenarios

... 3.7.2. Solar Criterias

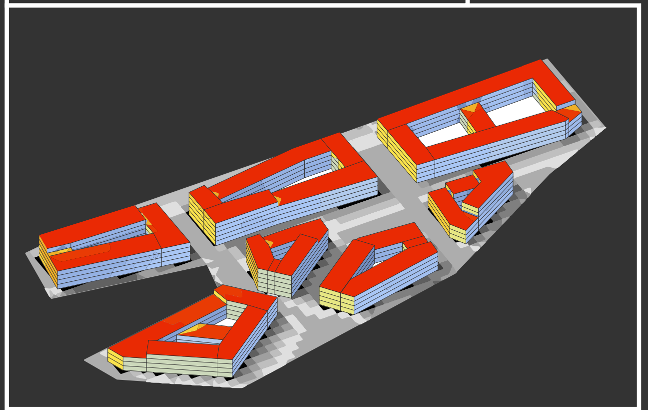
> The radiation ratios vary across the plans, indicating differences in solar exposure within the respective areas. Similarly, the ground shadow hours differ, reflecting variations in the amount of sunlight reaching the ground. These factors play a crucial role in determining the overall environmental quality and livability of the urban space. In light of these results, it is crucial to thoroughly assess the trade-offs and priorities associated with solar exposure and ground shadow hours when selecting a master plan.

Scenario-1: L. Rise H. Density



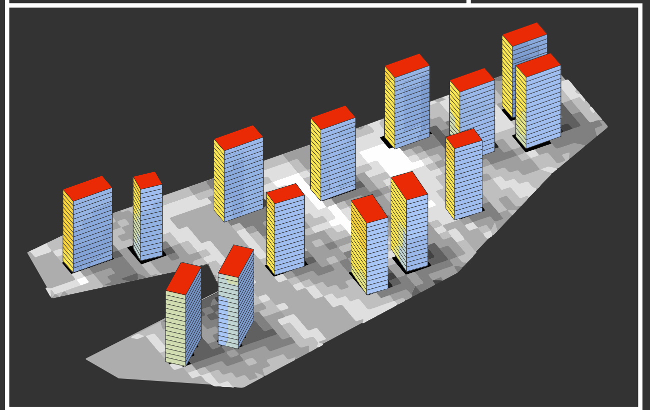
Annual Summer Radiation per $1m^2$: **12,62 kwH/m2**
Annual Average Sunlight Hour: **4,84**

Scenario-2: Courtyard Blocks



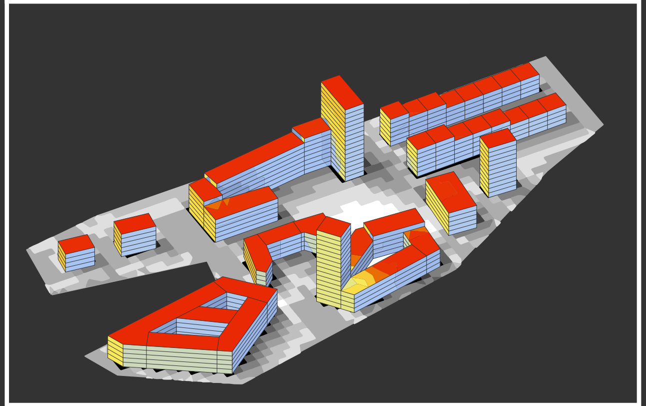
Annual Summer Radiation per $1m^2$: **12,78 kwH/m2**
Annual Average Sunlight Hour: **3,99**

Scenario-3: Tower City



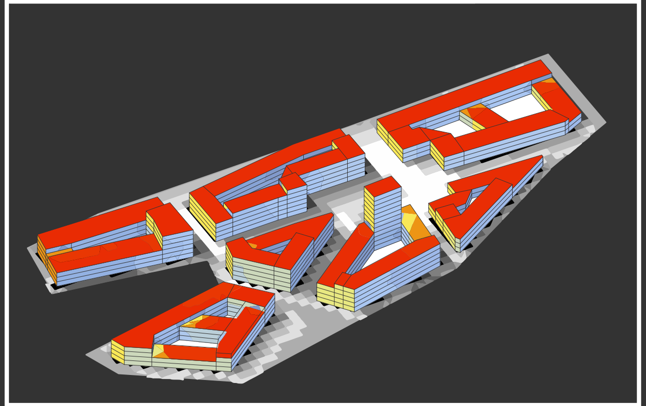
Annual Summer Radiation per $1m^2$: **12,95 kwH/m2**
Annual Average Sunlight Hour: **5,39**

Scenario-4: Mixed Typology



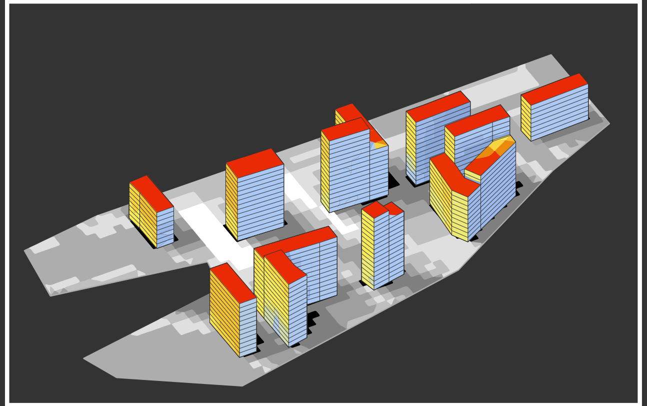
Annual Summer Radiation per $1m^2$: **11,76 kwH/m2**
Annual Average Sunlight Hour: **4,48**

Scenario-5: Piazza



Annual Summer Radiation per $1m^2$: **12,74 kwH/m2**
Annual Average Sunlight Hour: **3,90 h**

Scenario-6: Central Park



Annual Summer Radiation per $1m^2$: **11,38 kwH/m2**
Annual Average Sunlight Hour: **5,52 h**

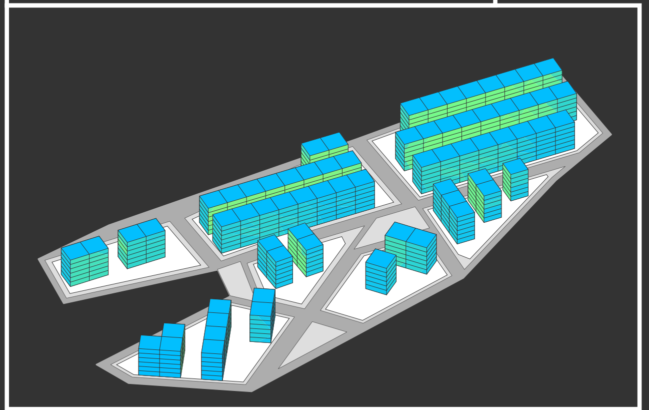
Design Phase:1 Part:3

... 3.7. Comparative Analysis of Extreme Scenarios

... 3.7.3. Visibility Criterias

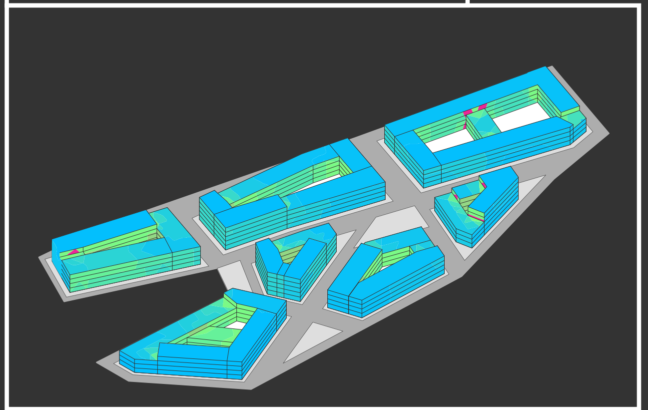
> After examining the results of the view analysis conducted on the facades of buildings,other than the central park which performed poorest, it is evident that options present less varying percentages of views towards the outside, compared to the other analysis that is done. Since the analysis has many settings and contexts. It is possible that in other sites and conditions, the results could have changed vastly. These findings have implications for the overall visual experience and connectivity with the surroundings. When evaluating these results, it is crucial to consider the importance of unobstructed views and the potential impact on the quality of the built environment. By carefully considering these factors, one can select a master plan that optimizes the connection between the built environment and its surroundings

Scenario-1: L. Rise H. Density



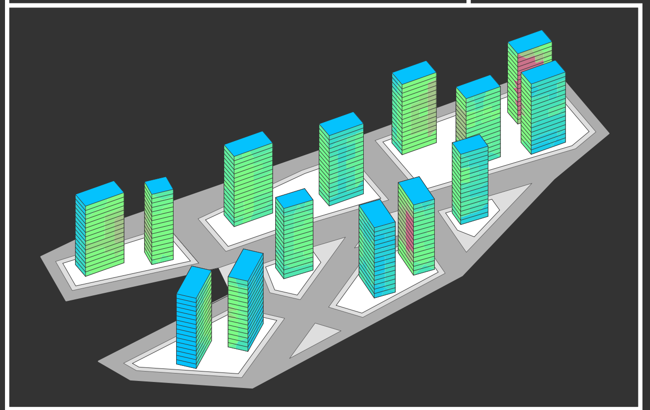
View Percentage of Inhabitants: ***%42,09***
Number of Floors: ***6 Floors***

Scenario-2: Courtyard Blocks



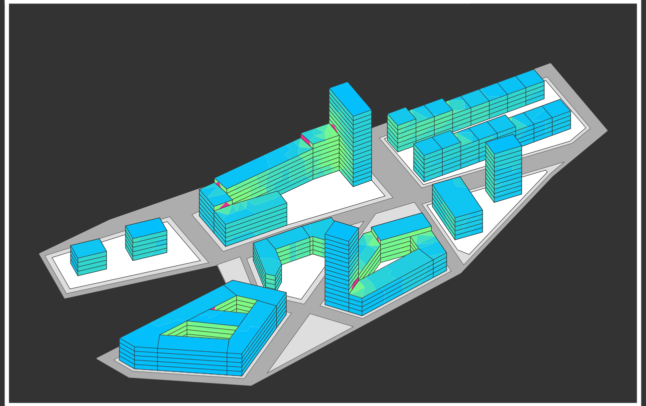
View Percentage of Inhabitants: ***%42,34***
Number of Floors: ***3-4 Floors***

Scenario-3: Tower City



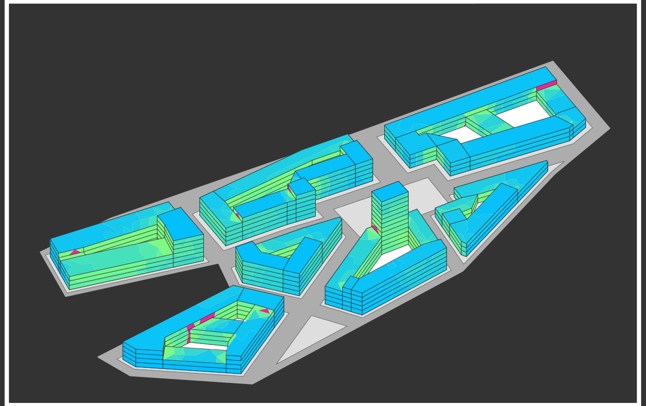
View Percentage of Inhabitants: ***%41,26***
Number of Floors: ***17 Floors***

Scenario-4: Mixed Typology



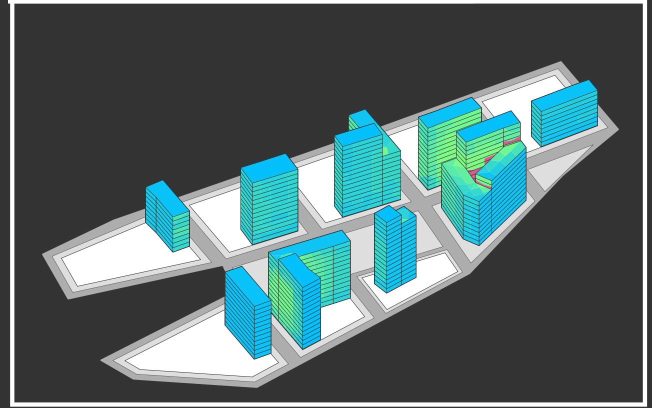
View Percentage of Inhabitants: ***%41,68***
Number of Floors: ***3-10 Floors***

Scenario-5: Piazza



View Percentage of Inhabitants: ***%41,86***
Number of Floors: ***3-4 Floors***

Scenario-6: Central Park



View Percentage of Inhabitants: ***%39,36***
Number of Floors: ***12-24 Floors***

design phase 01
... PAUSE

To explore more about the genetic design algorithms in another scale, we paused the city block creation phase here in this thesis and shifted our focus to building scale in the following chapters. By expanding the analysis to a larger scale, specifically at the apartment level, our aim is to achieve a more comprehensive understanding.

chapter 04:

... BUILT ENVIRONMENT & SPATIAL CONFIGURATION OF AI

> Architectural design presents a multitude of challenges that architects must navigate throughout the design process. Including functionality, aesthetics, sustainability, regulations, budget limitations, technological advancements, collaboration, site-specific considerations, historic preservation, and future-proofing. Architects face the complex task of creating spaces that not only meet the functional requirements of the users but also exhibit visual appeal and align with the client's vision and cultural context. They must also consider sustainable design principles to minimize environmental impact. Therefore, decoding the architectural design process involves understanding the underlying principles and steps involved in architectural design challenges.

In this thesis, our focus is directed towards a specific aspect of architectural design which is the creation of architectural plans. It is important to note that our exploration is limited in scope, as we concentrate on a small portion of the broader considerations involved in architectural design. Instead of conducting an extensive theoretical background study, our objective is to outline common trends and methods for incorporating AI algorithms into the creative workflow developed throughout the paper. By doing so, we aim to have a connected workflow for the paper that encompasses various scales of early design process.

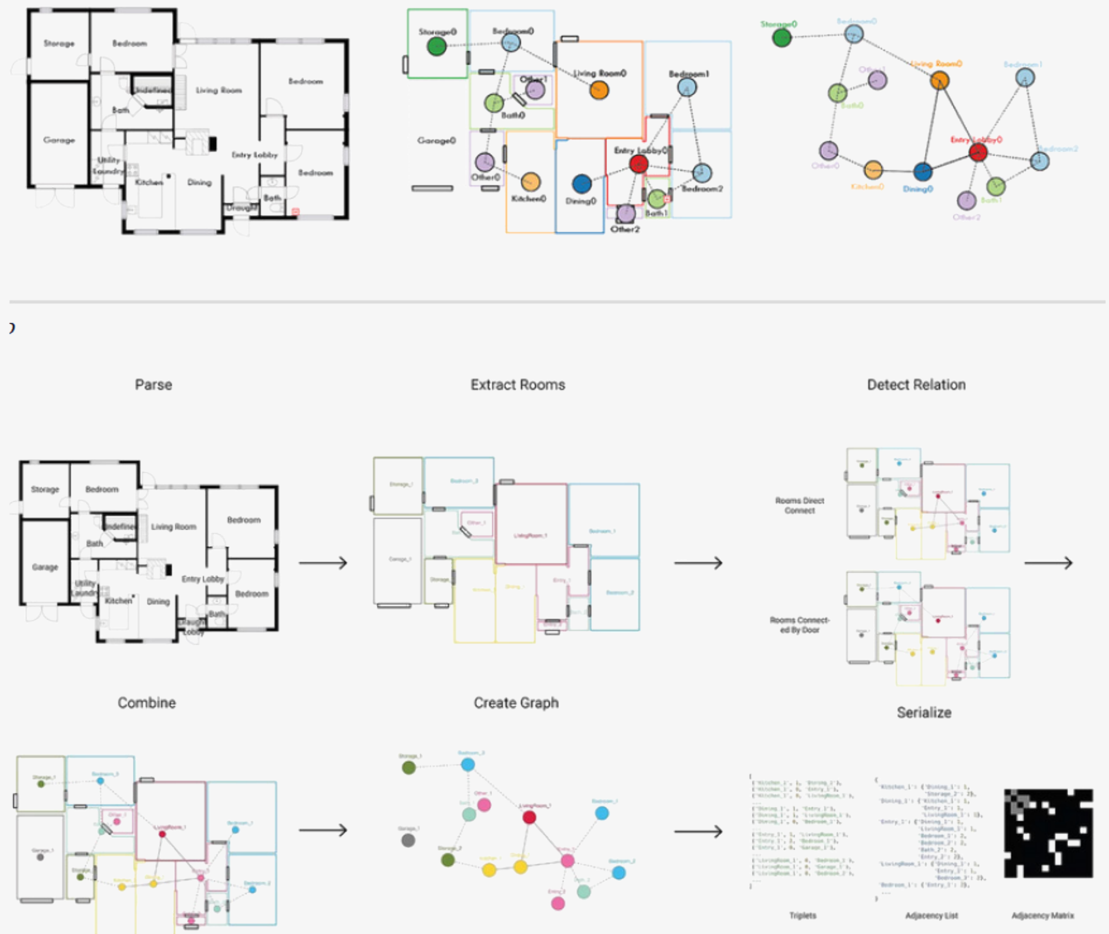


Figure 53: Self-organizing floor plans

As Lovelace stated in the 19th century, “The Analytical Machine has no pretensions whatsoever to originate anything. It can do whatever we know how to order it to perform. It can follow analysis, but it has no power of anticipating any analytical revelation of truths. Its province is to assist us in making available what we are already acquainted with.”(Fuegi, & Francis, 2003). This was a commonly held belief before Turing demonstrated the concept of Thinking Machines, challenging our understanding of machines. Today, we acknowledge that machines have the capability to think and generate art objects, similar to human beings. The mention of human beings in this argument is significant, as it serves to highlight the ongoing debate surrounding the relationship between creativity and AI. Neil Leach raises an important question: “Can we say that a computer is creative, for example, when it has no consciousness and does not even ‘realize’ that it is being creative?”(Leach 2022).

Human creativity involves a complex interplay of cognitive processes, emotional responses, and cultural influences. Architects’ unique perspectives, experiences, and cultural backgrounds shape their creative decisions, resulting in designs that reflect their individuality. AI design, on the other hand, relies on data-driven algorithms and predefined parameters. While AI can generate innovative designs, it may lack the nuanced understanding and contextual sensitivity inherent in human creativity. AI algorithms operate based on learned patterns and statistical analyses, potentially limiting their ability to break free from existing paradigms.

> The question of how the space in which we live can be designed by algorithms instead of humans has been asked many times throughout history. Although the automated generation of spatial layouts is a trendy problem nowadays, there has been extensive research and many applications in this area since the second half of the 20th century. For instance, an early research work, "The Geometry of Environment: An Introduction to Spatial Organization in Design" by Lionel March and Philip Steadman, investigates the logical patterns of entities and the physical and spatial arrangement of buildings. More recently, in 2019, Stanislas Chaillou introduced Generative Adversarial Networks (GANs) as an innovative and effective tool for floor plan creation. GANs, a sublayer of machine learning, have gained significant attention in the field of artificial intelligence.(Chaillou, 2019)

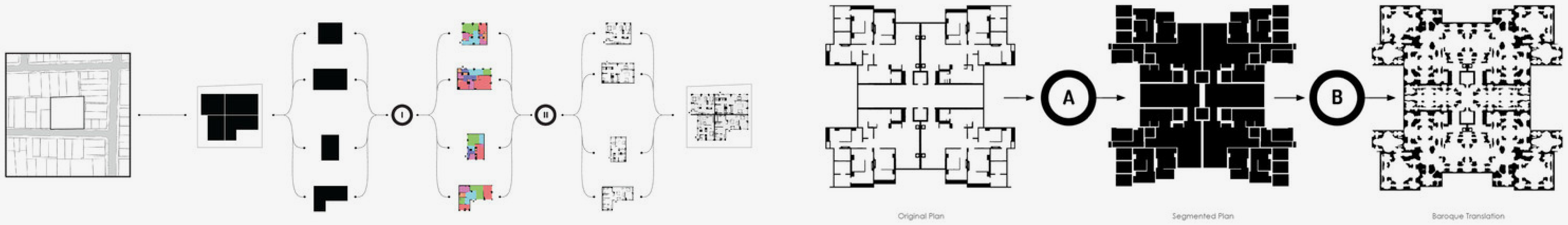
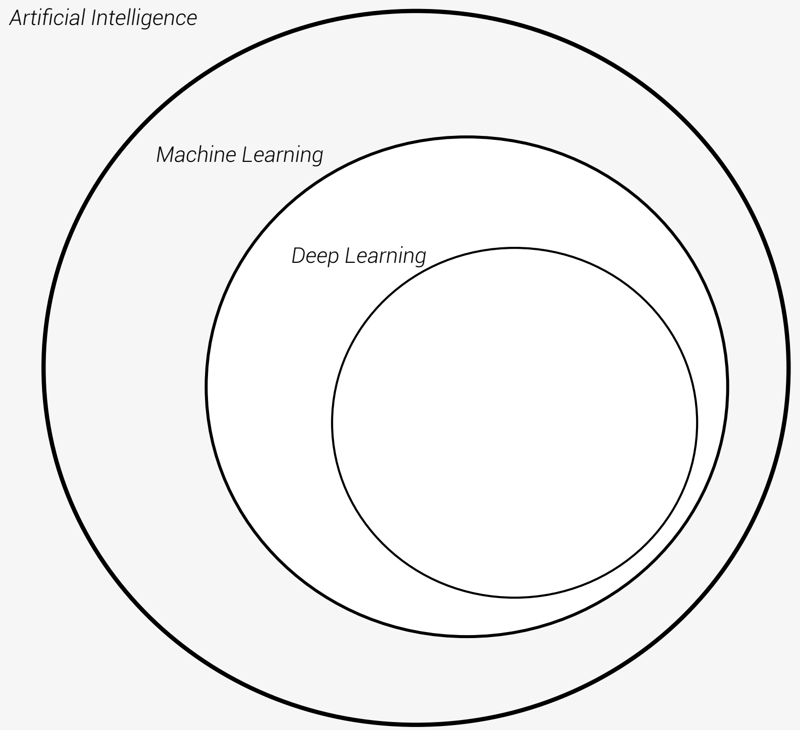


Figure 54: GAN algorithms and floor plan creation

> Machine Learning arises from the search for the surprise factor achieved by computers. It is a subfield of artificial intelligence that focuses on developing algorithms and models capable of learning from data and making predictions or taking actions without being explicitly programmed. Machine Learning involves training a computer system to learn patterns and relationships from examples, enabling it to make predictions or decisions on new, unseen data.

For instance, if you wished to automate the task of tagging your vacation pictures, you could present a machine learning system with many examples of pictures already tagged by humans, and the system would learn statistical rules for associating specific pictures with specific tags (Chollet, 2017).

Machine Learning encompasses various branches or approaches, each with its own characteristics and applications. In order to better understand the techniques of automating the creation of floor plans, it is important to understand various types, such as supervised and unsupervised learning.



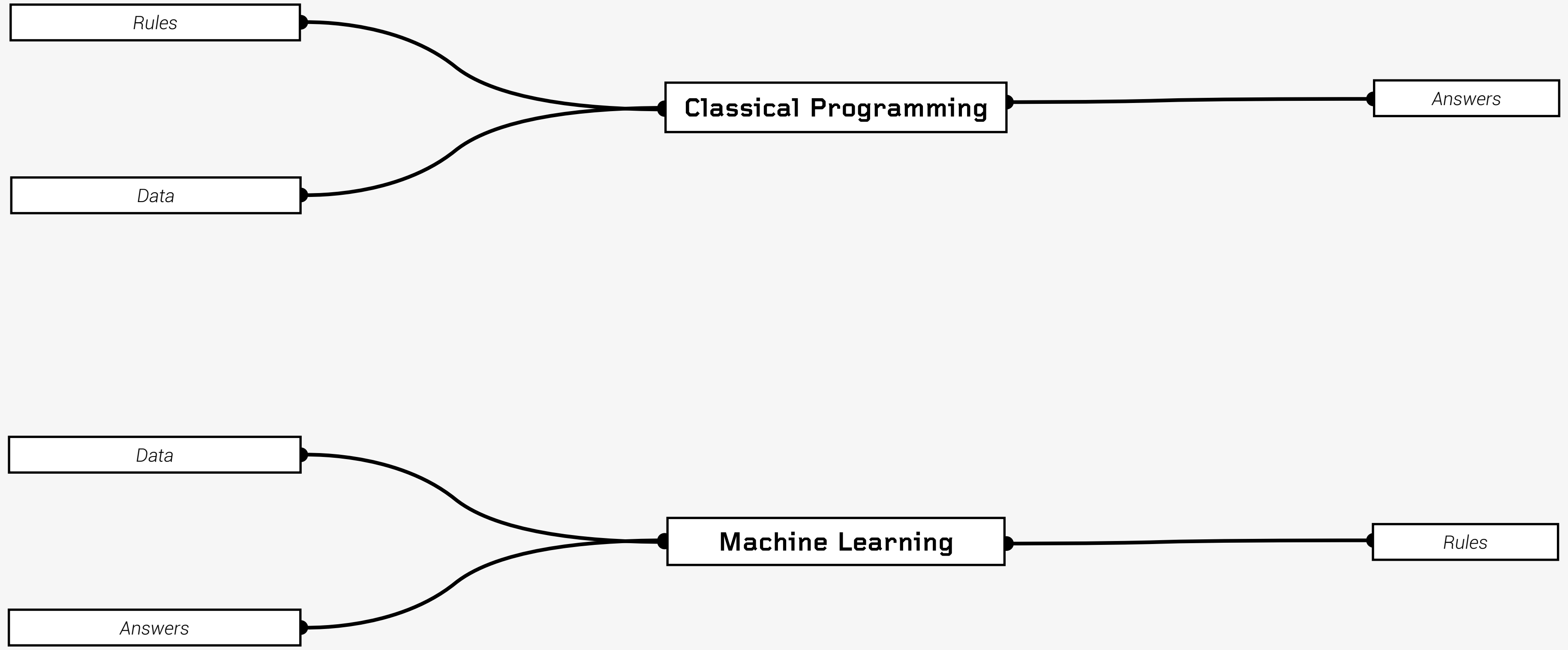


Figure 55: Machince learning classification and comparative analysis

... 4.3.1. Unsupervised Machine Learning

> This branch of machine learning consists of finding interesting transformations of the input data without the help of any targets, for the purposes of data visualization, data compression, or data denoising, or to better understand the correlations present in the data at hand. Unsupervised learning is the bread and butter of data analytics, and it's often a necessary step in better understanding a dataset before attempting to solve a supervised-learning problem (Chollet, 2017).

Clustering and dimensionality reduction are two common techniques used in unsupervised learning. Clustering aims to group similar data points together based on their inherent similarities or proximity, helping to identify natural groupings or clusters within the data. On the other hand, dimensionality reduction techniques reduce the complexity of the data by transforming it into a lower-dimensional space while preserving its essential characteristics. This can be particularly useful for visualizing and analyzing high-dimensional data. As shown in the example on the right image, apples and watermelons are introduced as input data to the unsupervised model, and it understands their characteristics and outputs them based on their relationships.

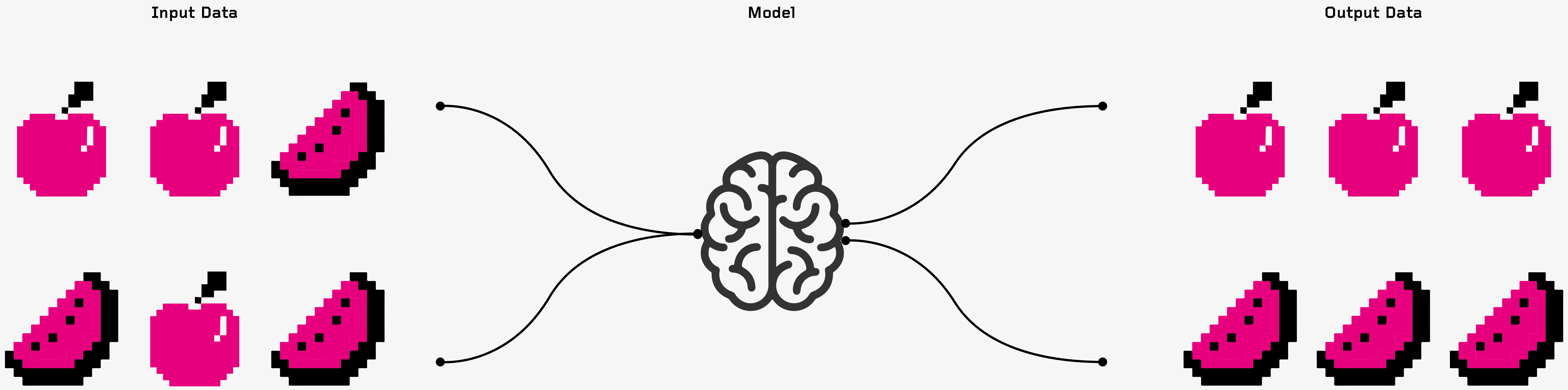
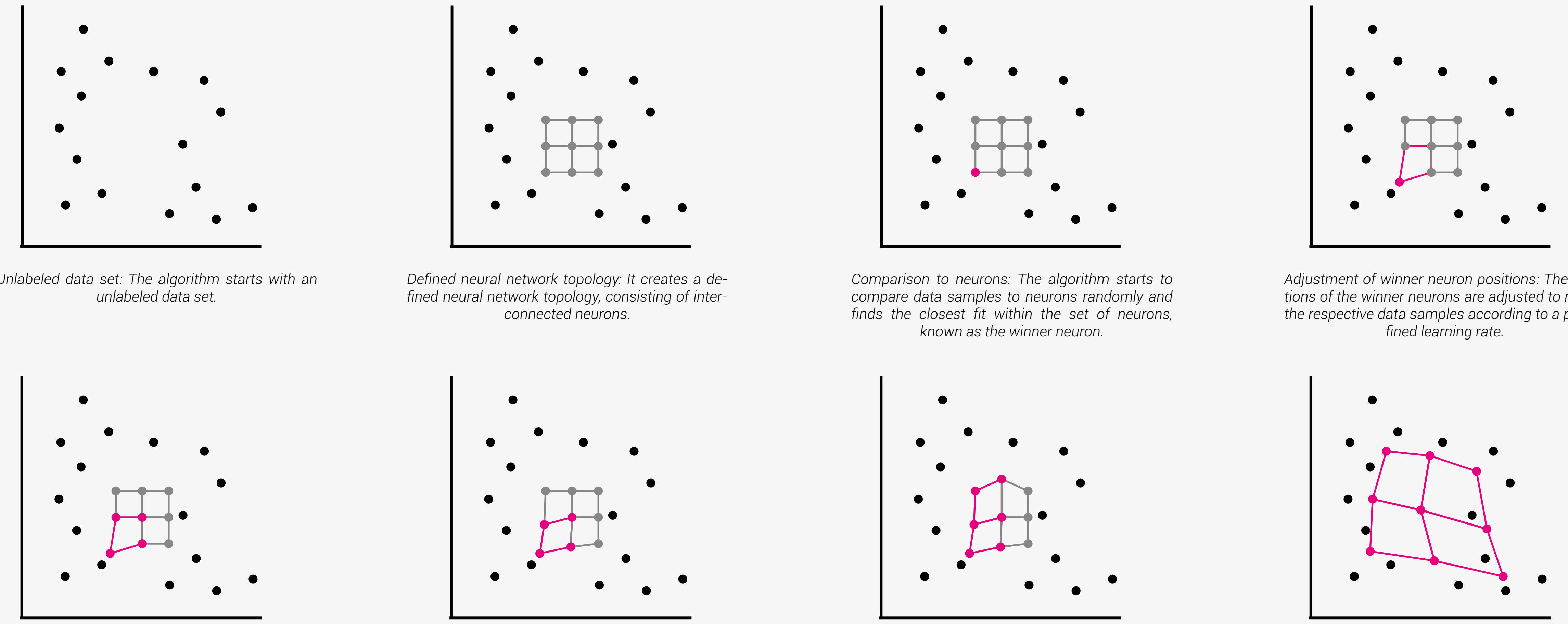


Figure 56: Unsupervised machine learning principle

... 4.3.2. SOM (Self-Organizing Maps)

> Self-Organizing Maps (SOM) is an artificial neural network algorithm used for unsupervised learning. It is inspired by the self-organization observed in the brain's visual cortex. The main objective of SOM is to represent complex and high-dimensional data in a lower-dimensional space while preserving the topological relationships between data points. This is achieved by following steps:

To illustrate the operation, the algorithm begins with an unlabeled dataset containing information to be explored. It creates a neural network dataset, which is a grid-like structure of interconnected neurons. Randomly selecting a data sample, the algorithm compares it to the codebook vectors of the neurons to find the best-matching unit (BMU), which closely represents the sample's characteristics. The BMU's position is adjusted to align more closely with the data sample. This position update information is then spread out to neighboring neurons, which also adjust their positions to a lesser extent. The adjustment is based on the distance from the BMU, with closer neighbors receiving a larger update. This process is repeated with different data samples, gradually adapting the network topology to fit the distribution of the data. Similar data samples tend to be mapped to nearby neurons, creating clusters or groupings that reveal the relationships and organization within the unlabeled dataset. In this way, SOM enables clustering, visualization, and exploration of complex data.



Unlabeled data set: The algorithm starts with an unlabeled data set.

Defined neural network topology: It creates a defined neural network topology, consisting of interconnected neurons.

Comparison to neurons: The algorithm starts to compare data samples to neurons randomly and finds the closest fit within the set of neurons, known as the winner neuron.

Adjustment of winner neuron positions: The positions of the winner neurons are adjusted to match the respective data samples according to a predefined learning rate.

Spread of position update information: The algorithm spreads out the position update information to the winner neuron's topological neighbors.

Adjustment of neighbor neuron positions: Neighbor neuron positions are adjusted based on a predefined neighborhood function.

Repetition for different data samples: The process is repeated for different data samples, iterating through the algorithm again and again.

Network topology fitting: As the algorithm continues to iterate, the network topology gradually fits and approximates the data.

Figure 57: self organization operation steps

... 4.3.3. Supervised Machine Learning

> In this approach, the input data (features) and the corresponding correct output labels are provided to the model during the training phase. The goal is for the model to learn the mapping between the input data and the output labels, so that it can accurately predict the labels for new, unseen data. Supervised learning involves two main components: the input features and the output labels. The input features are the measurable characteristics or properties of the data, while the output labels represent the target variable or the desired prediction. The model learns from the input-output pairs by finding patterns, relationships, or dependencies in the data.

The training process in supervised learning involves feeding the input features into the model and comparing its predictions with the known output labels. The model then adjusts its internal parameters to minimize the discrepancy between the predicted and actual labels. This adjustment is done using optimization algorithms and mathematical techniques such as gradient descent. The iterative process continues until the model achieves a satisfactory level of accuracy. Once the model is trained, it can be used to make predictions on new, unseen data by providing the input features and obtaining the predicted output labels.

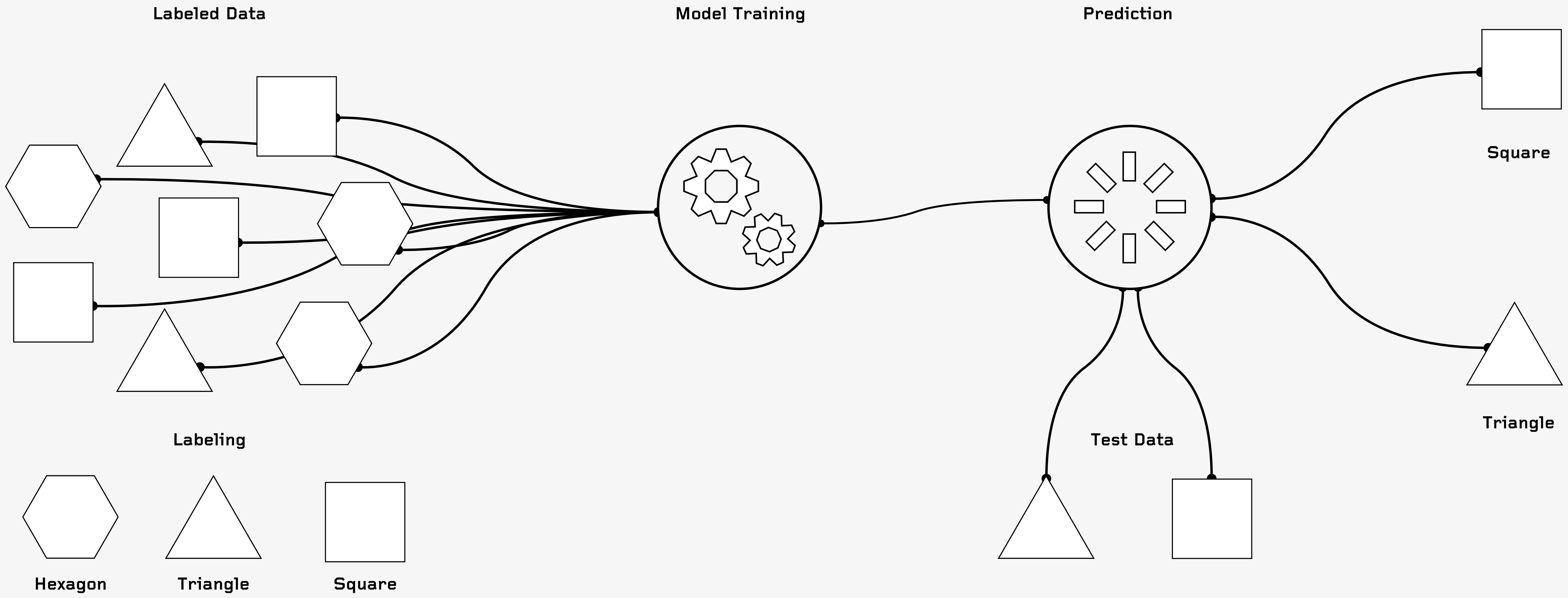


Figure 59: Supervised machine learning algorithm and their working principle

chapter 05

... CREATING FLOOR PLANS

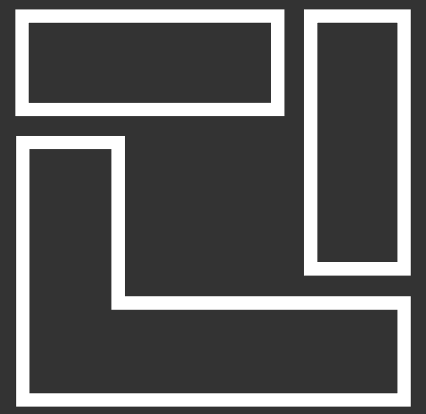
> In the following part of the thesis, one of the design solutions coming from evolutionary design part 1 is taken as a sample and used as a foundation to create architectural plans. The aim of the part is to discuss and investigate the potentials and limitations of the genetic algorithms and their use in architectural plan scale.

design phase 02
... P-01 DECODING THE TOOL

> In the following phase, 2D plan actions are decoded from a designer's point of view to understand how they can be used for creating architectural plan layouts. After introducing the actions, the training of neural networks and the practical implementation of trained data for solving 2D architectural plans are investigated.

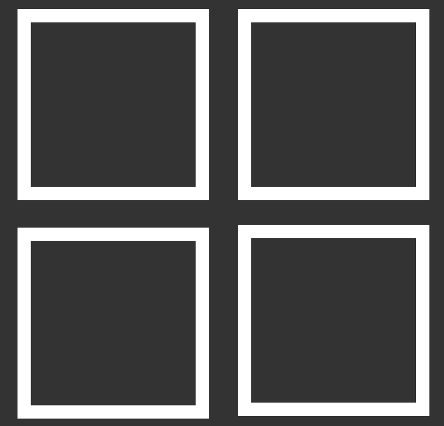
Design Phase:2 Part:1

... 5.1. 2D_Decoding: Generic Plan Actions:



Open Block

Strip



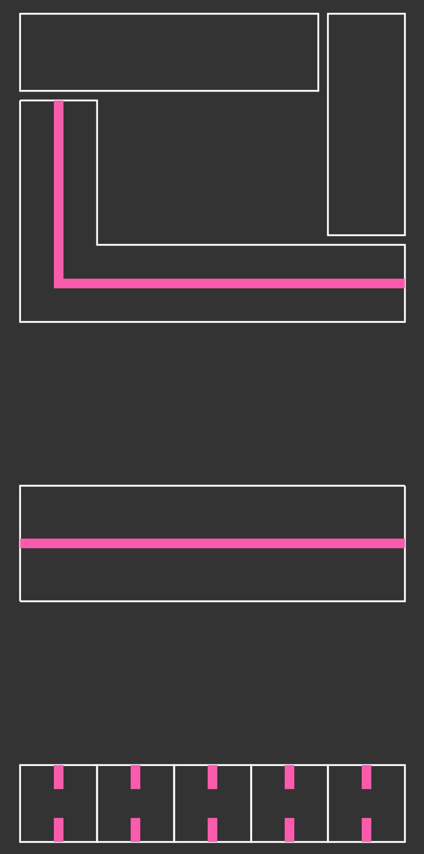
Point & Cluster

> 2D Generic plan actions are classified into three classes, similar to the City Block Formation phase. However, unlike in that phase, all the action sets are categorized under one stage. This decision has been made due to the shared characteristics among the different typologies in the 2D plan formation steps.

... 5.1.1. Circulation Addition

> In order to create a rich spatial organization for an apartment plan layout, several factors need to be taken into consideration. These factors include the function of the building, natural lighting conditions, visual connections with the surrounding area and buildings, privacy, and permeability. When starting the design process, it is important to have the freedom to explore various possibilities and consider these factors.

In this thesis, the first step focuses on determining the corridor locations and their layout for a typical architectural plan. Once the user enters a building, they are going to see generic circulation layout that has been suggesterd by algorithm for their building.This circulation intends to have a generic layout, as shown in the illustration.

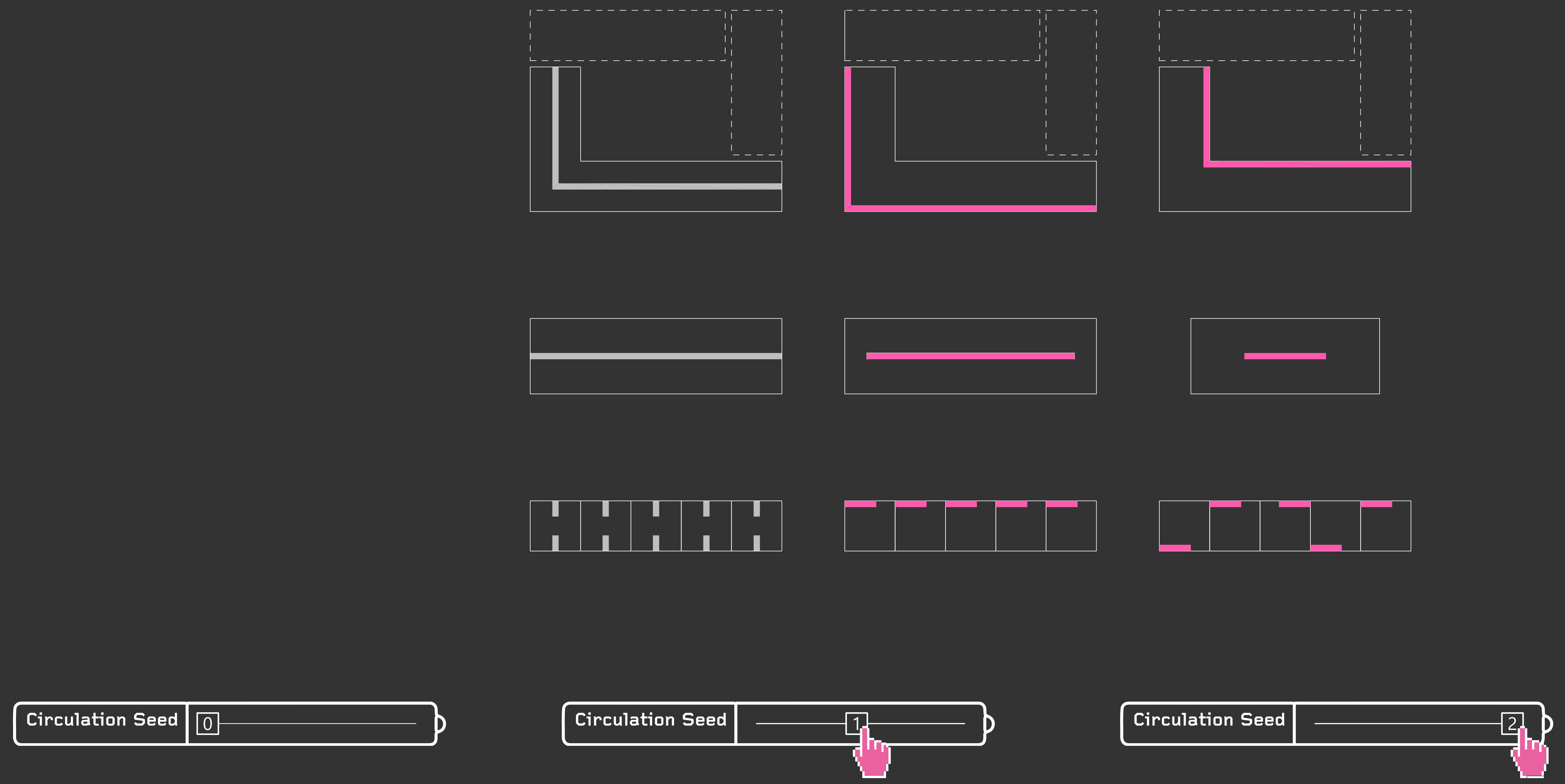


Design Phase:2 Part:1

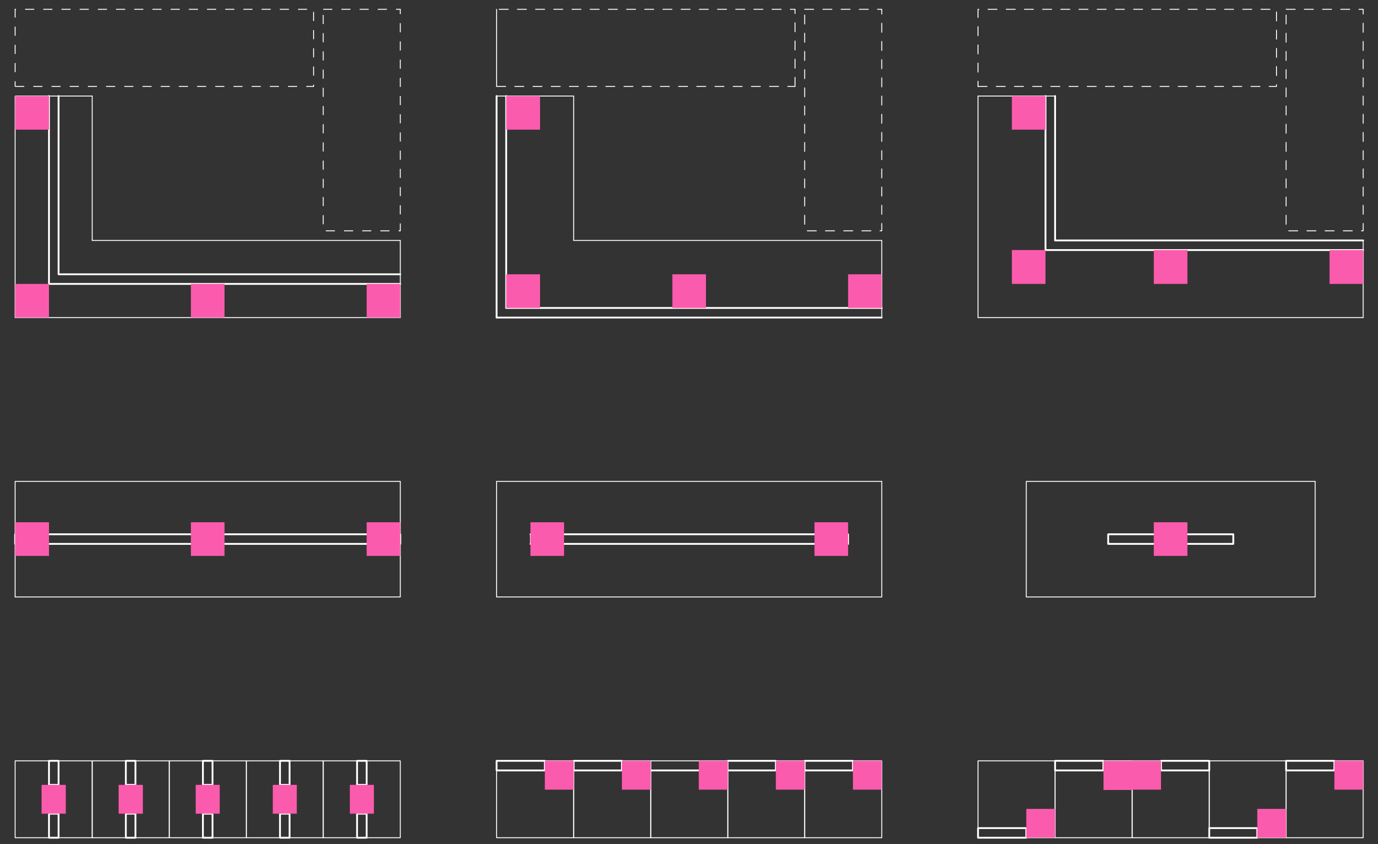
... 5.1. 2D_Decoding: Generic Plan Actions:

... 5.1.2. Circulation Variations

> Depending on the factors the user considers, such as location, spatial organization, size, or similar aspects of the circulation areas, adjustments may need to be made. If the user is not satisfied with the generic circulation layout, they can use the click & slide agent to navigate between pre-defined circulation layouts. Each city block type offers a variety of circulation variation options based on their natural formation. The user can decide on a circulation layout, but they may not have enough information at this stage to evaluate its direct effect on apartment connections or its impact on the inhabitants. Therefore, for the time being, the user can stick with a generic layout and continue exploring other actions. Later, they can come back and observe the significant changes that occur when variations are applied at the end.

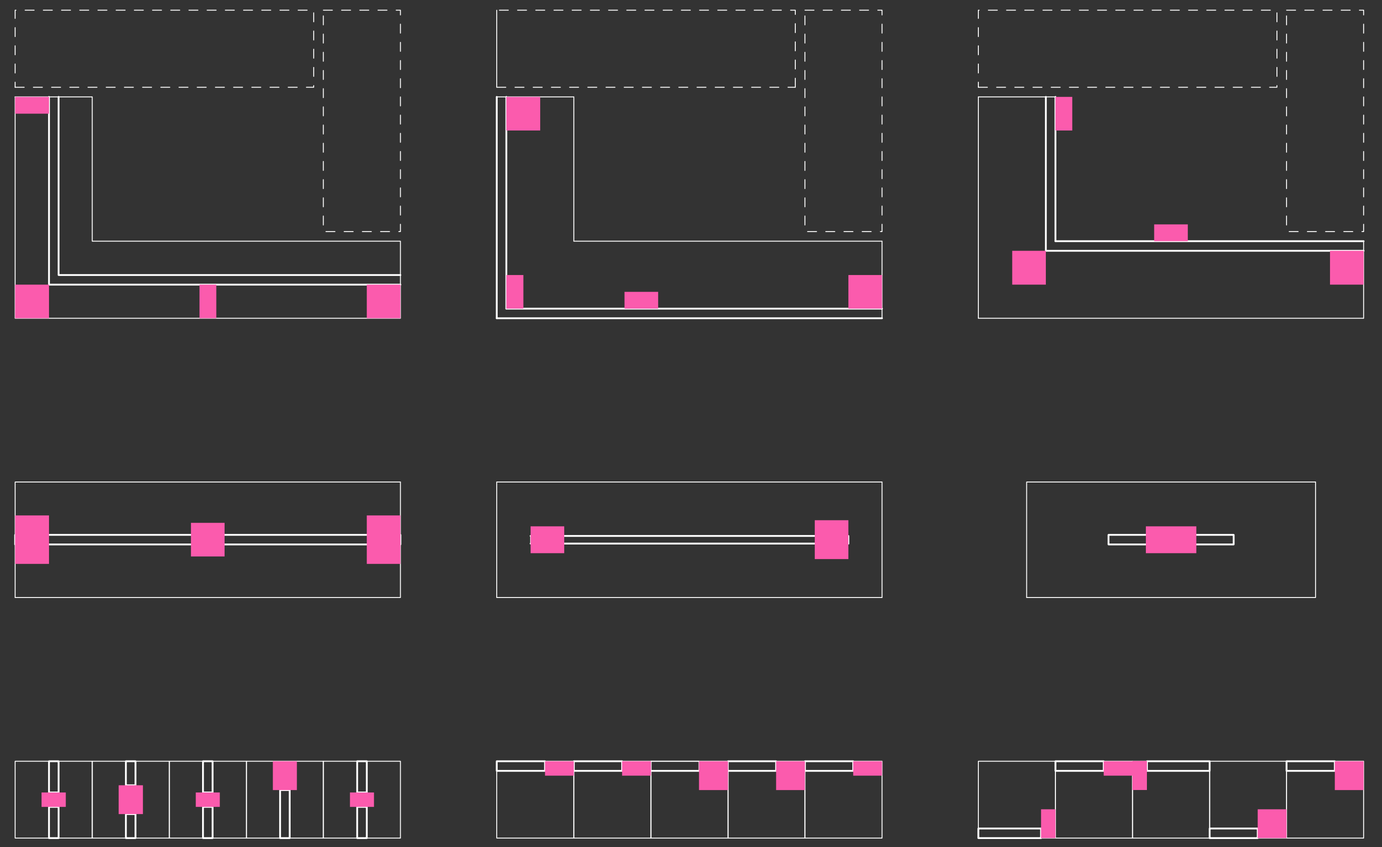


... 5.1.3. Core Addition



> Accessibility and fire regulations are crucial aspects, therefore while locating the cores, we take into consideration the local fire regulations to provide escape access for each apartment in the building. To adjust the regulations, users can utilize the click & slide tool with the panel component.

The core addition function identifies the building footprint and extracts its boundary curve. It then applies an inward offset to create a virtual corridor curve and locates the cores based on that curve ensure that gives access to corridors.



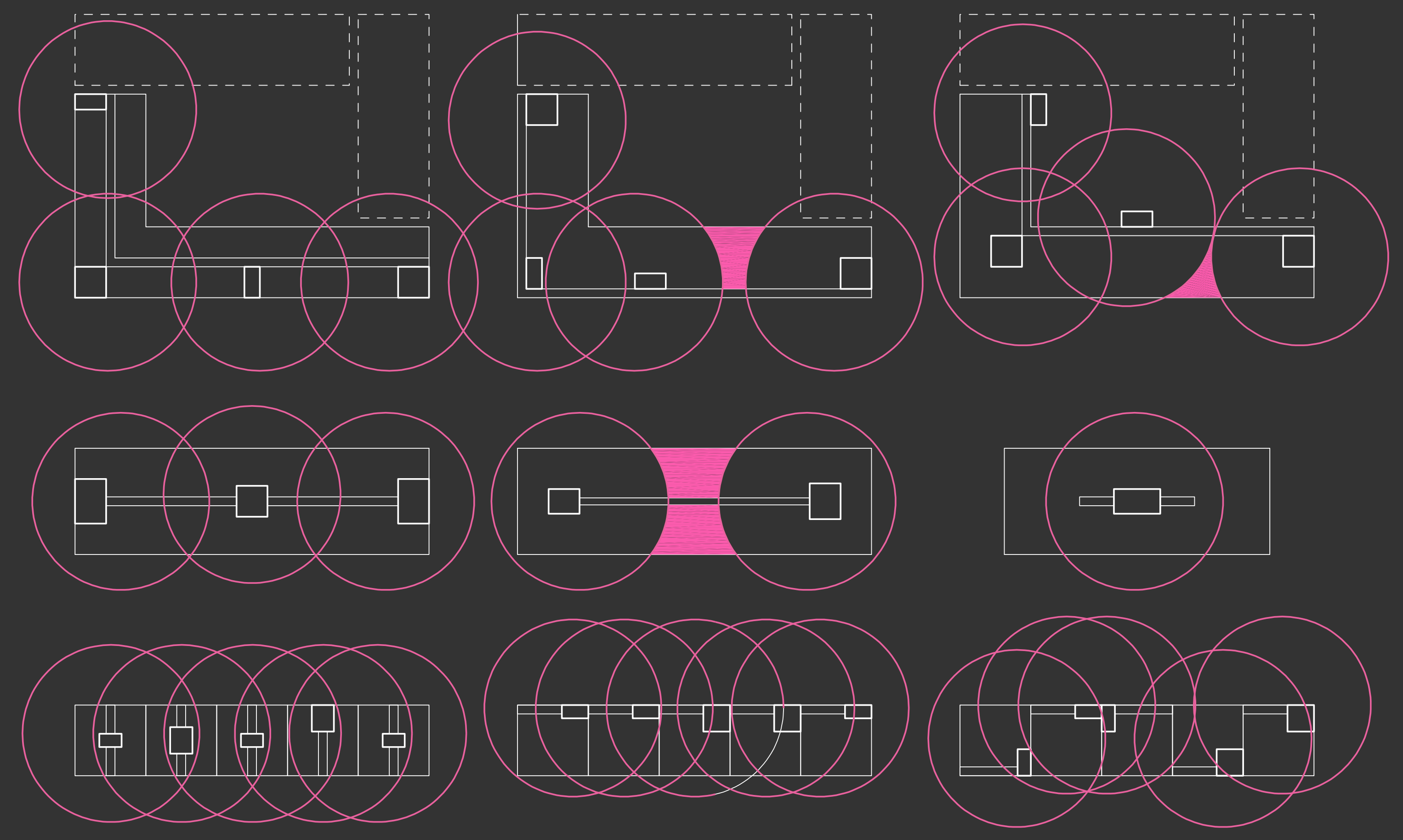
> The initial iteration of the core variation often begins with a generic solution, as shown in the illustration However, user has the flexibility to experiment with different variations of the core elements. For instance, one can choose to locate the cores at the end of the building to establish a direct connection with the building facade. To explore different solutions, the click and slide agent can be used to navigate between options.

The examples illustrated in the image demonstrate alternative configurations that address the aforementioned factors. By considering the specific requirements of the apartment's occupants and incorporating elements such as functionality, natural light, visual connections, privacy, and permeability, a spatially rich organization can be achieved.

Depending on the area of the building and local regulations, there may be varying limitations when defining the size of the cores. Therefore, the user can adjust the width and length of the core area using the click and slide agent to accommodate these restrictions.

> Once the cores are located in the corners, a filtering code is applied to check if the fire regulations are fulfilled. If any of the apartments fail to meet the requirements, the action will display an error message and provide the user with the opportunity to revise the core locations. The algorithm automatically adds a new core and checks again. This process is repeated until the number of cores is sufficient to meet the fire regulations for all the apartments.

If the user prefers to manually interpret the core locations, they can go back one step and utilize the custom core add function with the drawing agent. This allows the user to manually draw a core on the canvas while respecting the limits of the building surface. The newly drawn core will then work in conjunction with the existing ones and undergo the filtering code. If the regulation check is successful, no automatic core will be added.



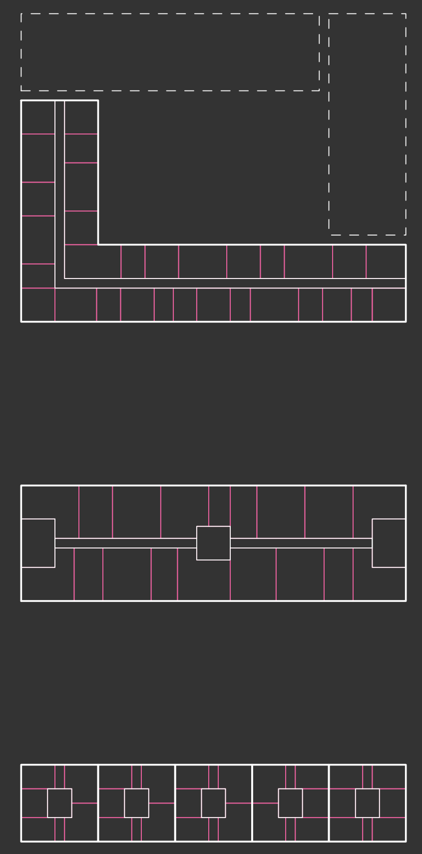
Design Phase:2 Part:1

... 5.1. 2D_Decoding: Generic Plan Actions:

... 5.1.6. Dividing to Apartments

a. Index Based

> Two different methods are introduced for dividing a building footprint into apartment units. Both methods begin after the user decides on the overall core and circulation shape and relationships. Once the 2D polygonal shape is fixed, the index-based method prompts the user to input values into a panel component using the click and slide agent. The input is an integer that corresponds to the width of the apartment. For example, if the user inputs a list of values such as; 3, 5, 6, 10, the algorithm fits the apartments inside the building with the respective widths from the list. After an initial settlement has been made, the user can navigate between different results. For instance, the number of apartments with a 5-meter width might be significantly more than those with a 10-meter width. Then, the user needs to search through the space to find their preferred version.



Design Phase:2 Part:1

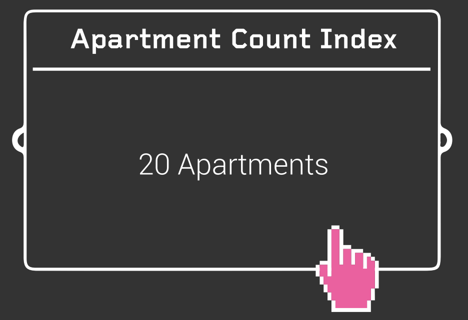
... 5.1. 2D_Decoding: Generic Plan Actions:

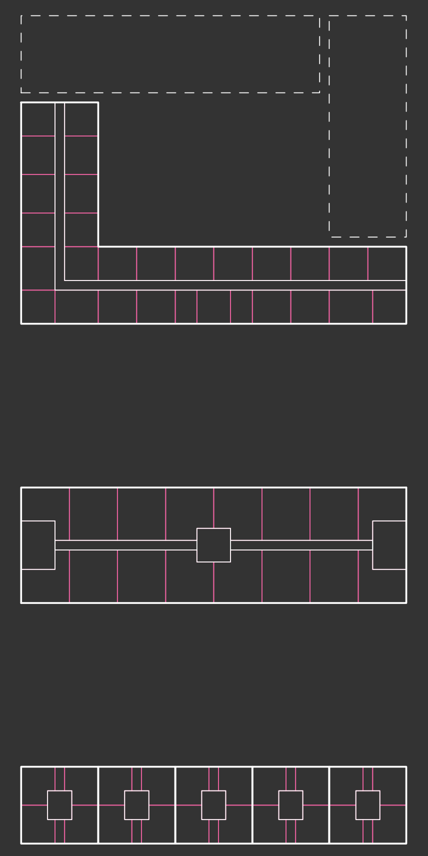
... 5.1.6. Dividing into Apartments

b. Equal Area Based

> In the equal area-based method, the process of dividing the building footprint into apartment units is approached differently. In this method, the user first selects the desired number of apartments they want to accommodate within the building. Once the number is determined, the building footprint is divided into an equal number of units.

The user interface allows the user to input the desired apartment count using a click and slide agent or similar interaction method. For example, if the user selects 20 apartments, the algorithm divides the building footprint into 20 equal-sized units. These units will have similar areas to ensure fairness and equal distribution of space among the apartments.

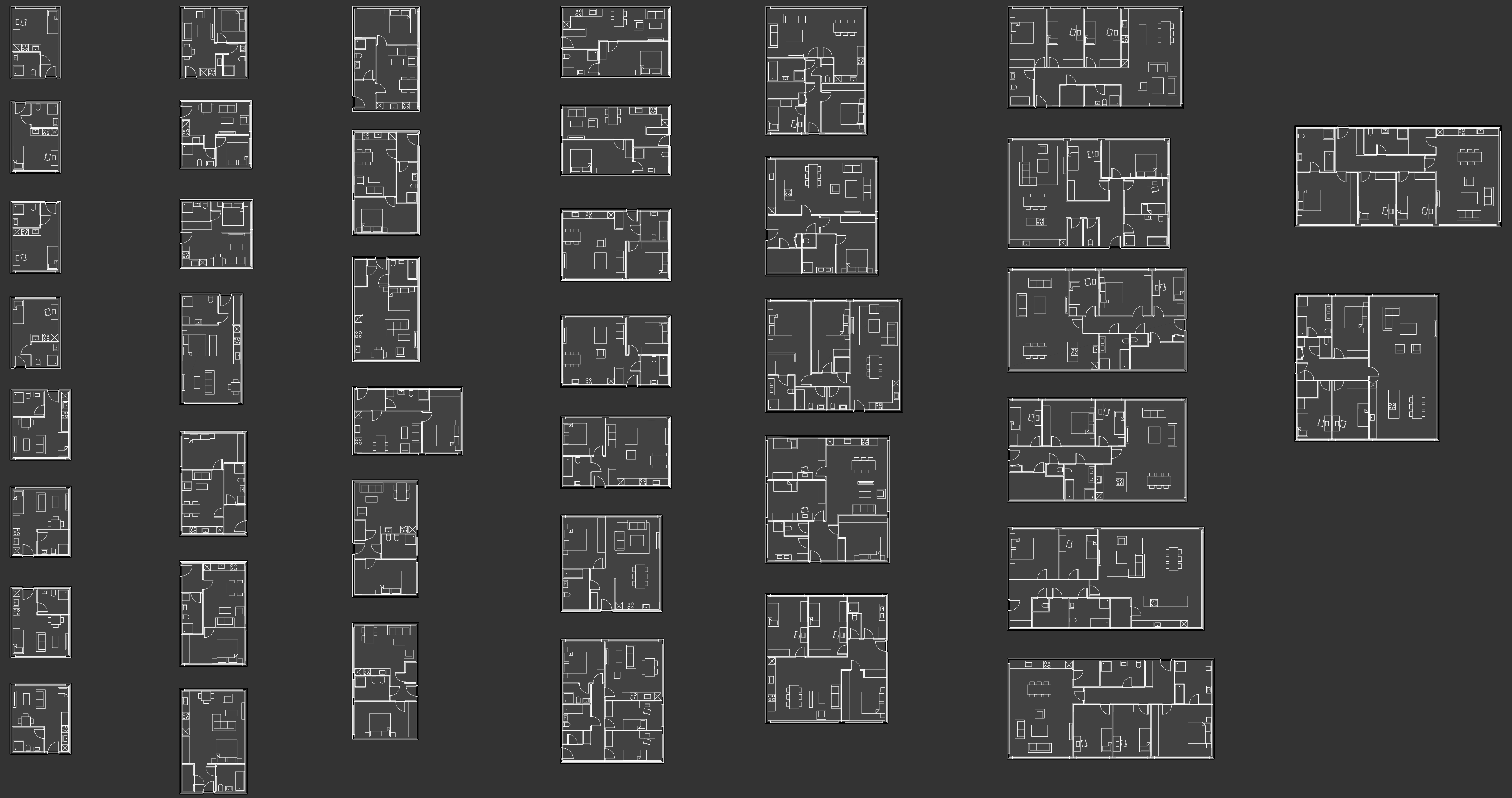




Design Phase:2 Part:1

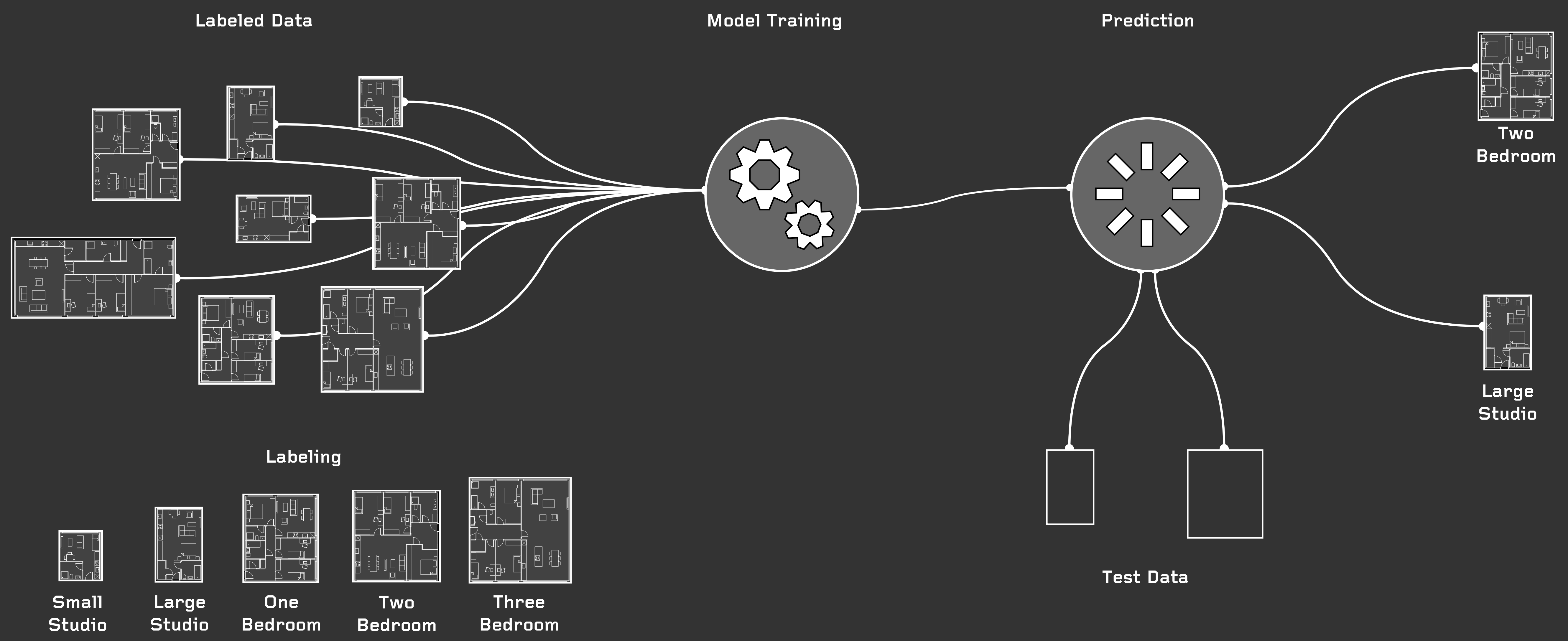
... 5.2. Making Floor Plan Database

... 5.2.1. Database for Training



> In order to fill the divided geometries with an architectural plan, we need a database that consists of architectural plans which can be used as training input for machine learning algorithms. The creation of a database for training a neural network involves several steps. Firstly, a dataset of floor plans is collected, ensuring diversity and representation of the target floor plan types. Key information, such as area, length, width, entrance points, windows, rooms, circulation, and ventilation, is extracted from the floor plans. Next, the extracted data is formatted into a suitable numerical representation for training the neural network. Each floor plan is represented by a set of input parameters that capture essential characteristics for accurate predictions. The dataset is then divided into three subsets: a training set, a validation set, and a test set(Rahmeh, 2022).

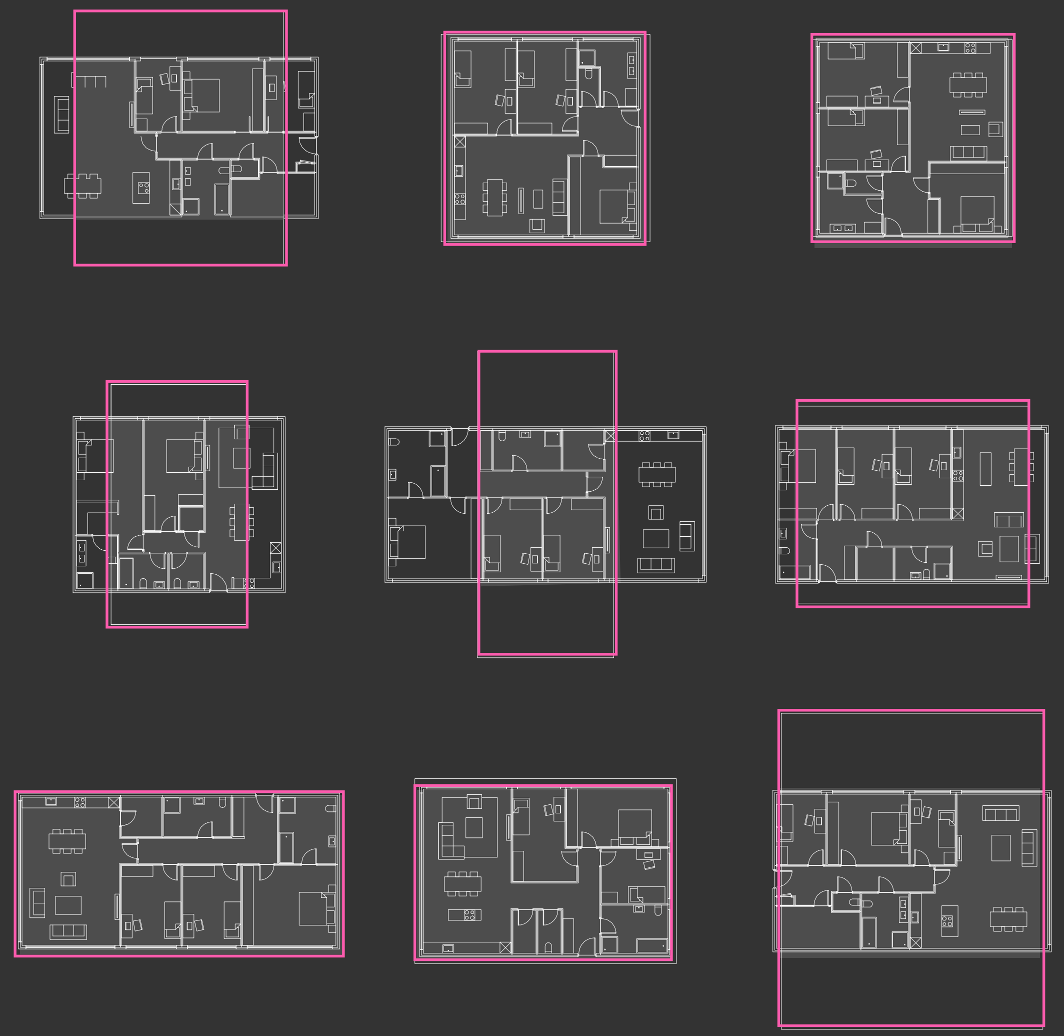
... 5.2.2. Training Neural Networks



> The neural network is designed by determining the number of layers, hidden neurons per layer, and activation functions. The complexity of the relationships to be learned is considered, aiming for an architecture that can effectively capture these relationships. The training process involves training the neural network using the training set. The resilience backpropagation algorithm is applied iteratively to adjust the network's weights and biases, minimizing prediction errors. The network's performance on the validation set is monitored, and adjustments to hyperparameters are made as necessary.

Once the training is complete, the performance of the trained network is evaluated using the test set by considering metrics such as accuracy, precision, and recall. This evaluation provides an assessment of the trained model's effectiveness. Based on the evaluation results, the model is analyzed

... 5.2.2. Training Neural Networks



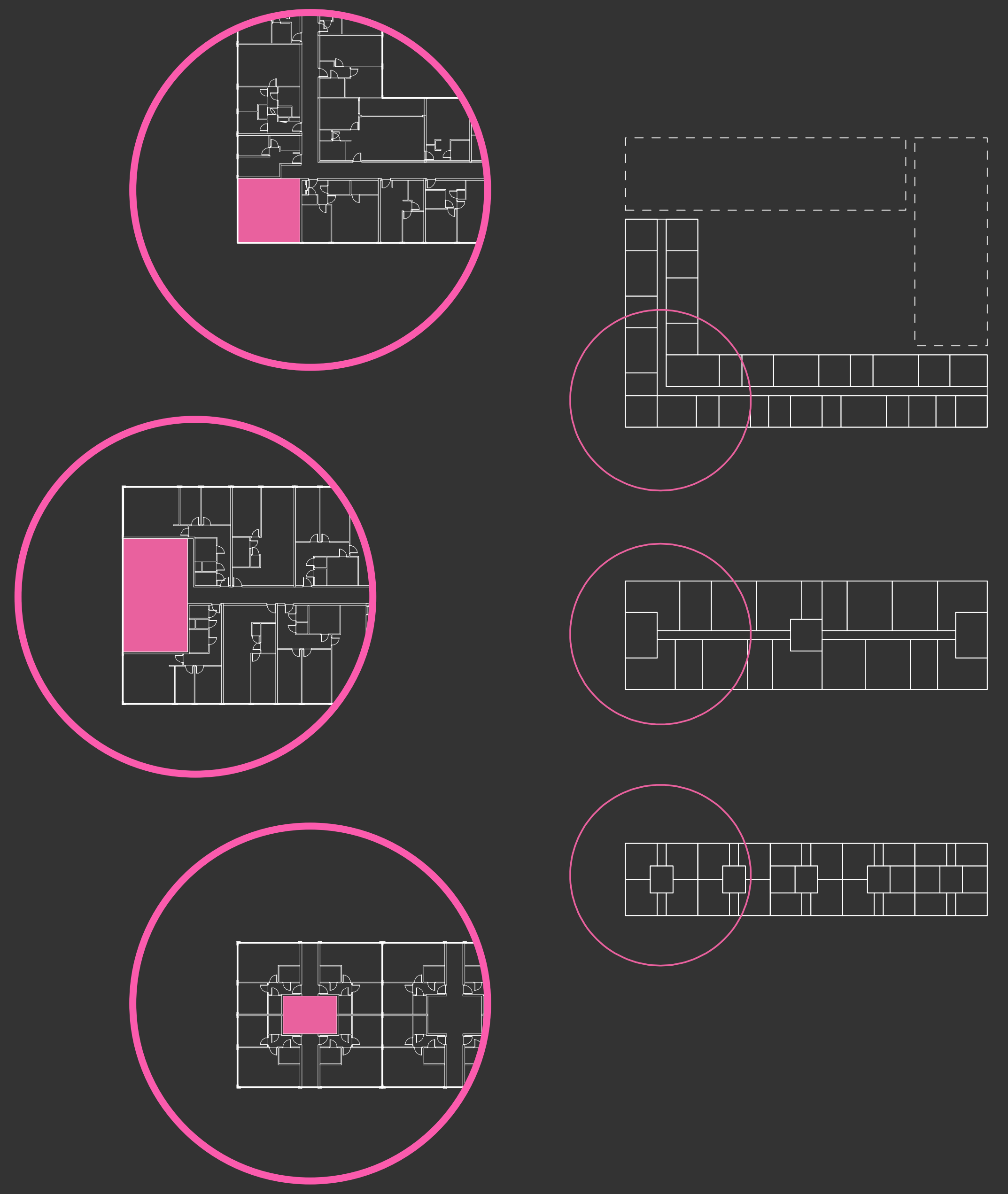
> After conducting an initial experiment with trained data, the following results are generated and illustrated on the right. It is observed that the prediction mechanism worked on a general scale, correctly placing boundaries for specific apartment types. However, due to the limited amount of data during the dataset creation process, errors occurred while placing apartments. For instance, as graphically emphasized on the left, there are empty spaces between the proposed apartment and the given input boundary.

To address the issues encountered in the process, we have two options. We can either go back to the process and repeat the training with a larger dataset, or we can utilize an already trained dataset. Considering the time allocated for the study, we have decided to integrate an already trained network into our workflow. After conducting a brief research, the "Plan Finder" plug-in has been selected for the prediction process and integrated into our framework.

Design Phase:2 Part:1

... 5.2. Making Floor Plan Database

... 5.2.3. Implementing Trained Outputs



> Once the trained data is selected, it is ready to be applied to the layout generation process during the apartment division phase. The boundaries that are defined for the apartments are connected to the probability component, which aims to find the most optimal fit. Designers can use the click & slide agent to explore different fit options by adjusting the building fit seed slider.

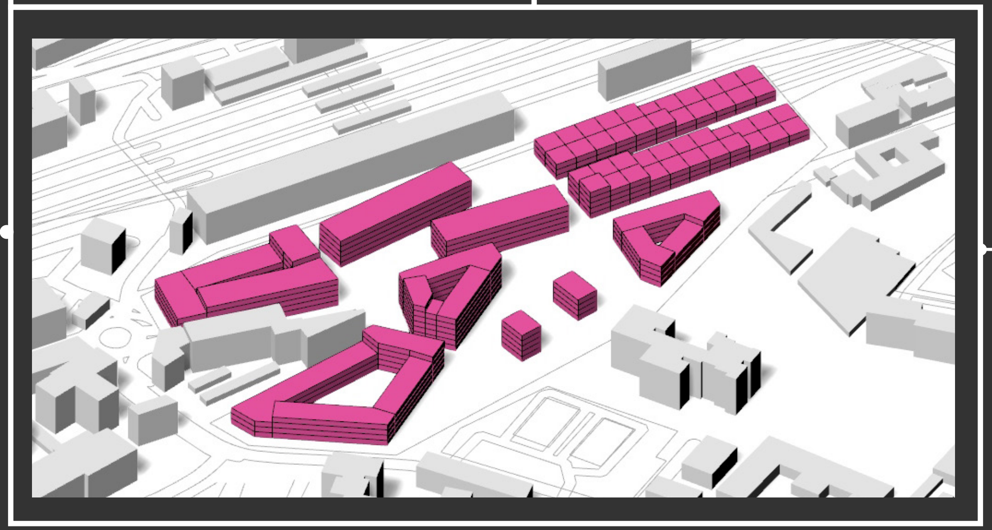
design phase 02

... P-02 EVOLUTIONARY FLOOR PLAN DESIGN

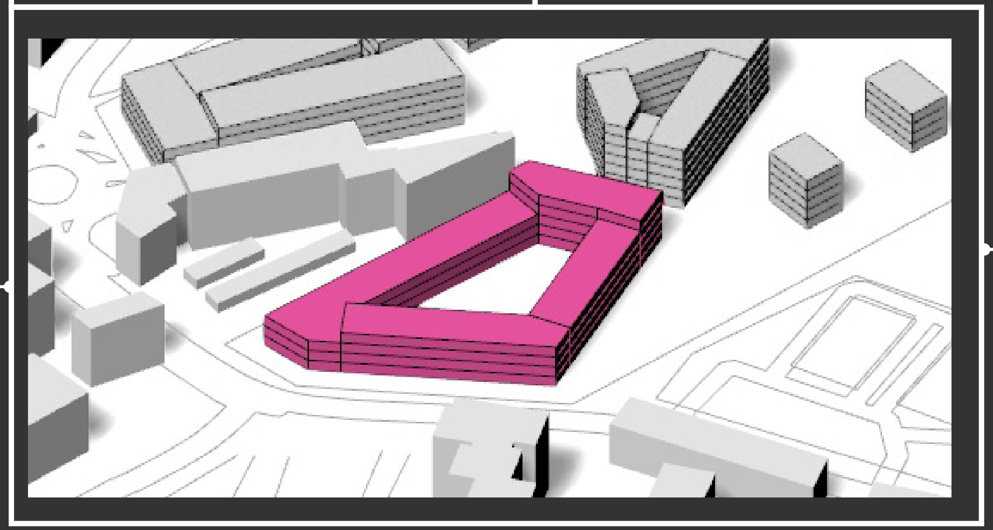
In the following part of the thesis, the actions and components described in the paper are applied to the selected optimization result that are derived from the first genetic design simulation. Finally, a comparison is made between the generated results the fitness values.

... 5.3. Selecting Sample for Simulation

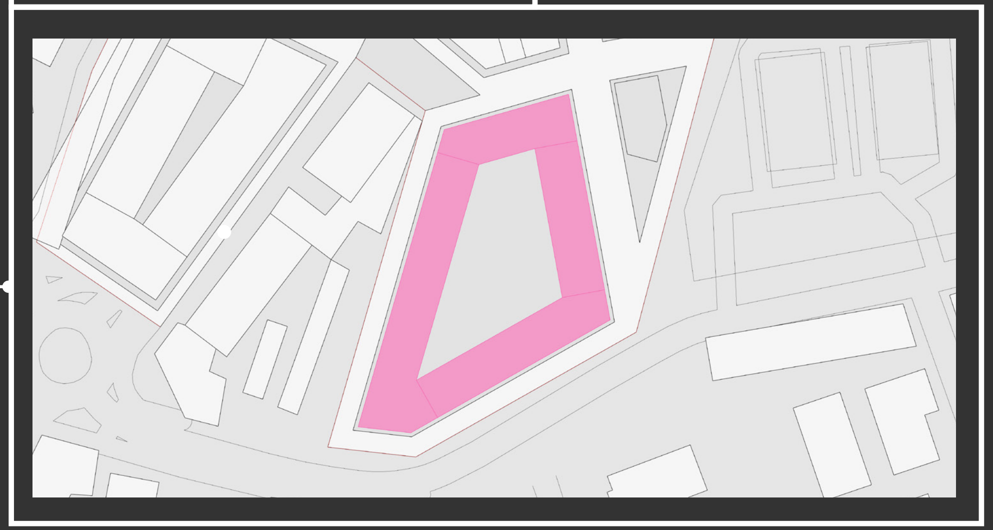
Generation 66 Individual 33



Selected Building for Simulation



2D Building Footprint



> To test the second part of the tool, we first need a test subject to run a simulation. In order to create a joint framework, we have chosen the selected design result that emerged from the first simulation in the urban design part. The illustrated image showcases Generation 68 Individual 33.

Once the subject is selected, we focused on one of the buildings within the sample area. While selecting that building, our aim was to have the most complex structure that allows for more variable geometry. Thus, the illustrated volume has been chosen.

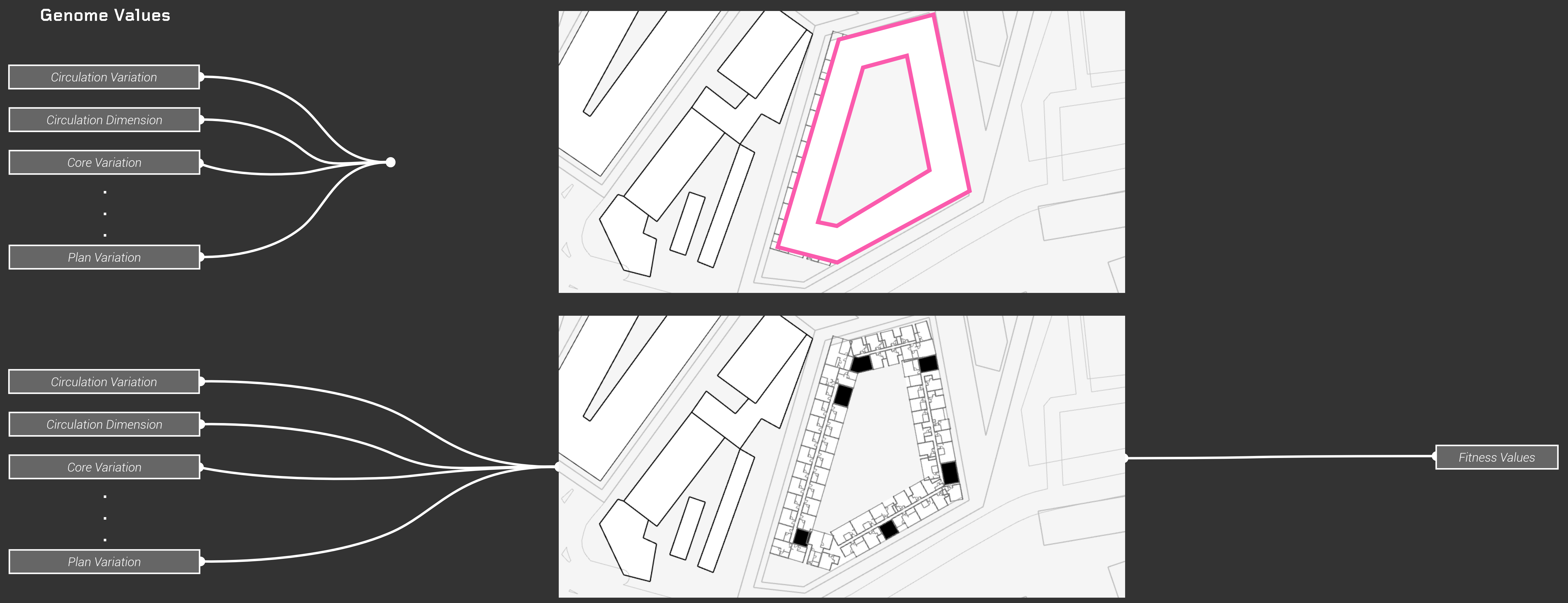
The volume of the selected sample area was divided into three sections based on the density heatmap in the initial simulation. Therefore, we extracted the 2D footprint of those boundaries and defined them as the limits for the plan creation simulation.

... 5.4. Creating Initial Design Solution

> After determining the building footprint, similar to the first simulation, we need to create an initial set of solutions for the given design problem. During the creation of this design solution, we once again randomized the genome sequence and generated an initial design solution, which is represented in the image. The fitness values for the given design solution are as follows:

Plan Fit Index: **20.05**
Total Core Surface Area: **105 m2**
Type of Apartment for Building B & C: **15**
Type of Apartment for Building A: **4**
Proximity of the Core Location for Building B: **3.77**

The Plan Fit Index represents a numerical representation of how well the apartment plans are fitted into the physical space in a 2-dimensional layout. A lower value indicates a better fit for the plan. The objectives of Total Core Surface Area and Type of Apartments for Building A are self-explanatory. Let's now explain the Apartment Variety Index for Buildings B and C. Since the evolutionary engine Wallacei only works in one direction (minimizing fitness objectives), we need to define a variable that, when it has a lower value, provides a higher variety for Buildings B and C. Therefore, we calculated the number of Apartment Types in those buildings and take the reciprocal of it.



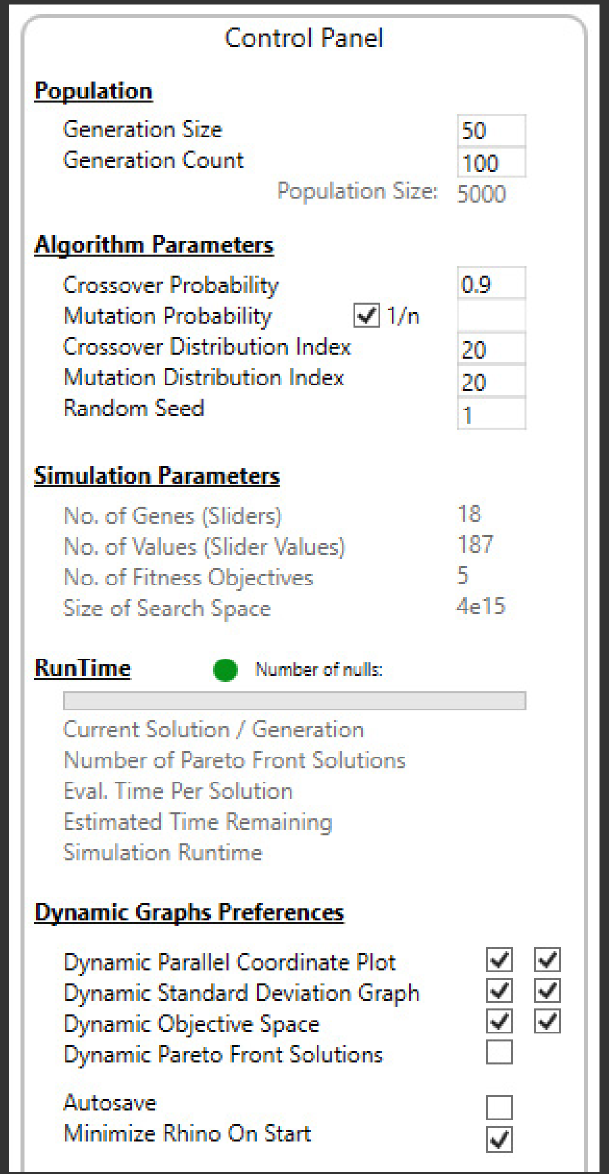
Design Phase:1 Part:2

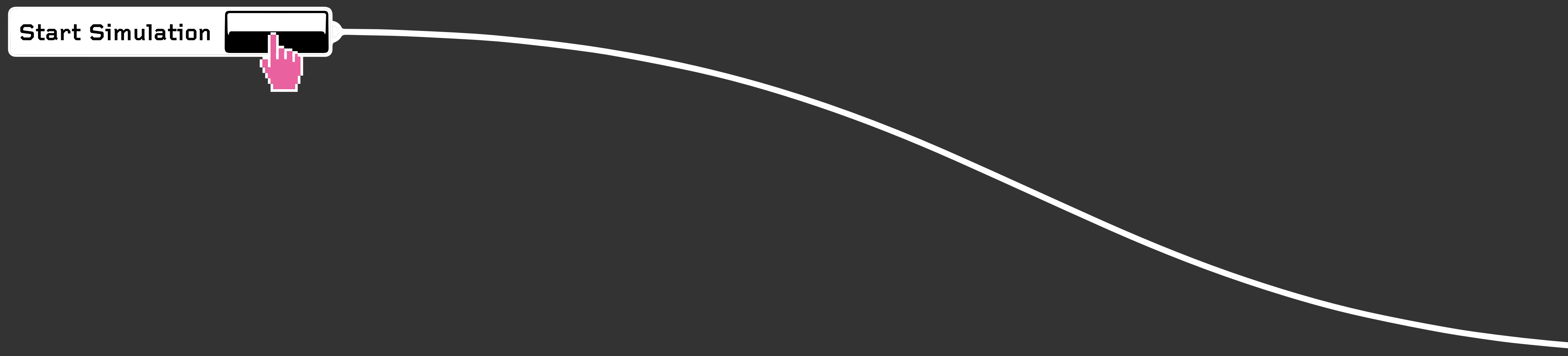
... 5.5. Design Simulation

... 5.5.1. Setting Simulation Parameters

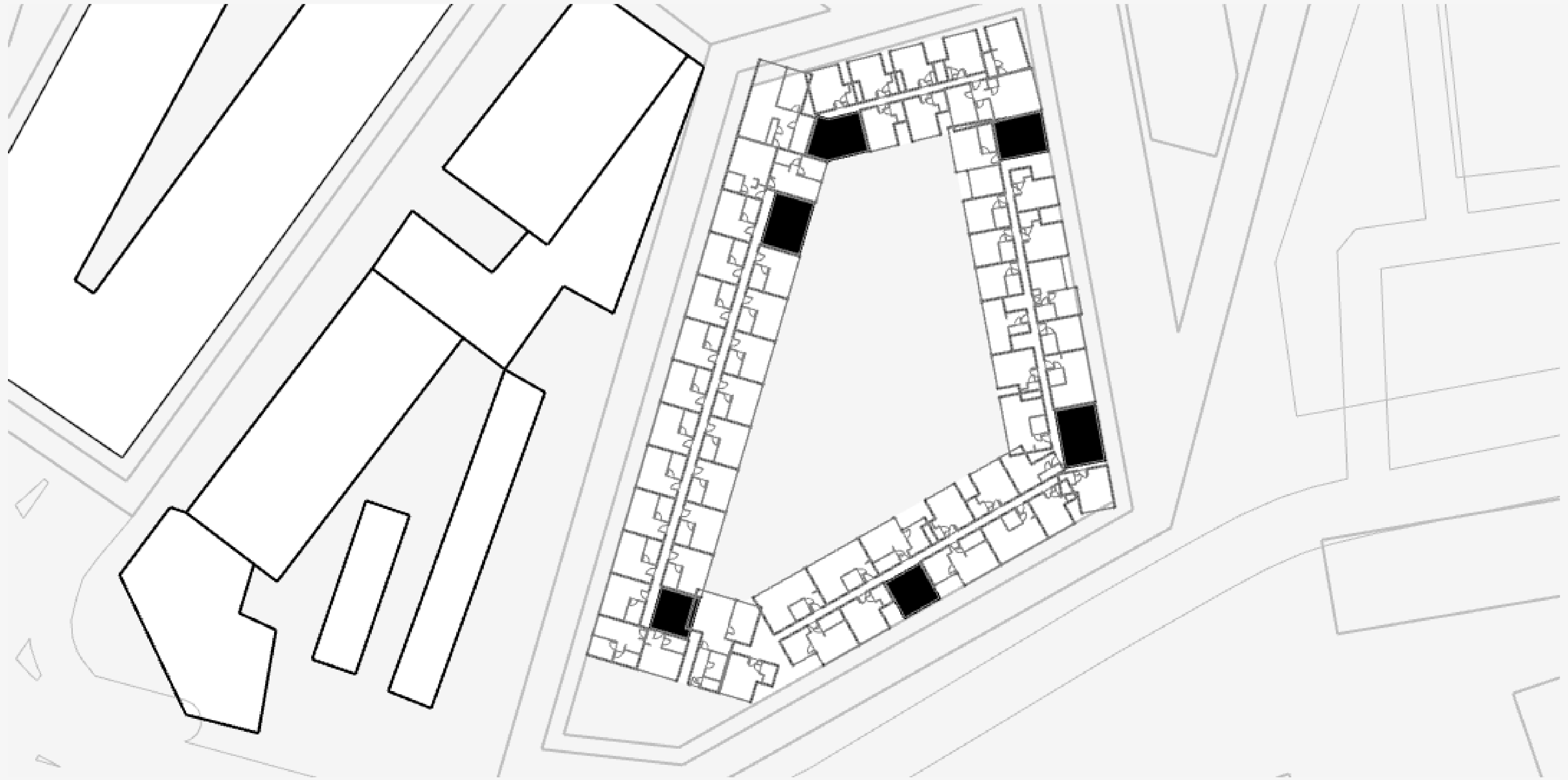
> In this simulation experiment, our primary objective is to maximize the variety of apartment types for Buildings B and C while minimizing the number of types for Building A. Additionally, we aim to minimize the Plan Fit Index, followed by sub-goals of minimizing the core area to optimize the spatial layout of the buildings and reduce construction costs. The expected outcome of this experiment is to test the effectiveness of the tool we are developing for this thesis in generating a variety of solutions for different building typologies.

As in the first experiment, we maintained a generation size of 50 and conducted 100 iterations to utilize time efficiently. With all the parameters set up, we let the simulation run once again to observe the outcomes.

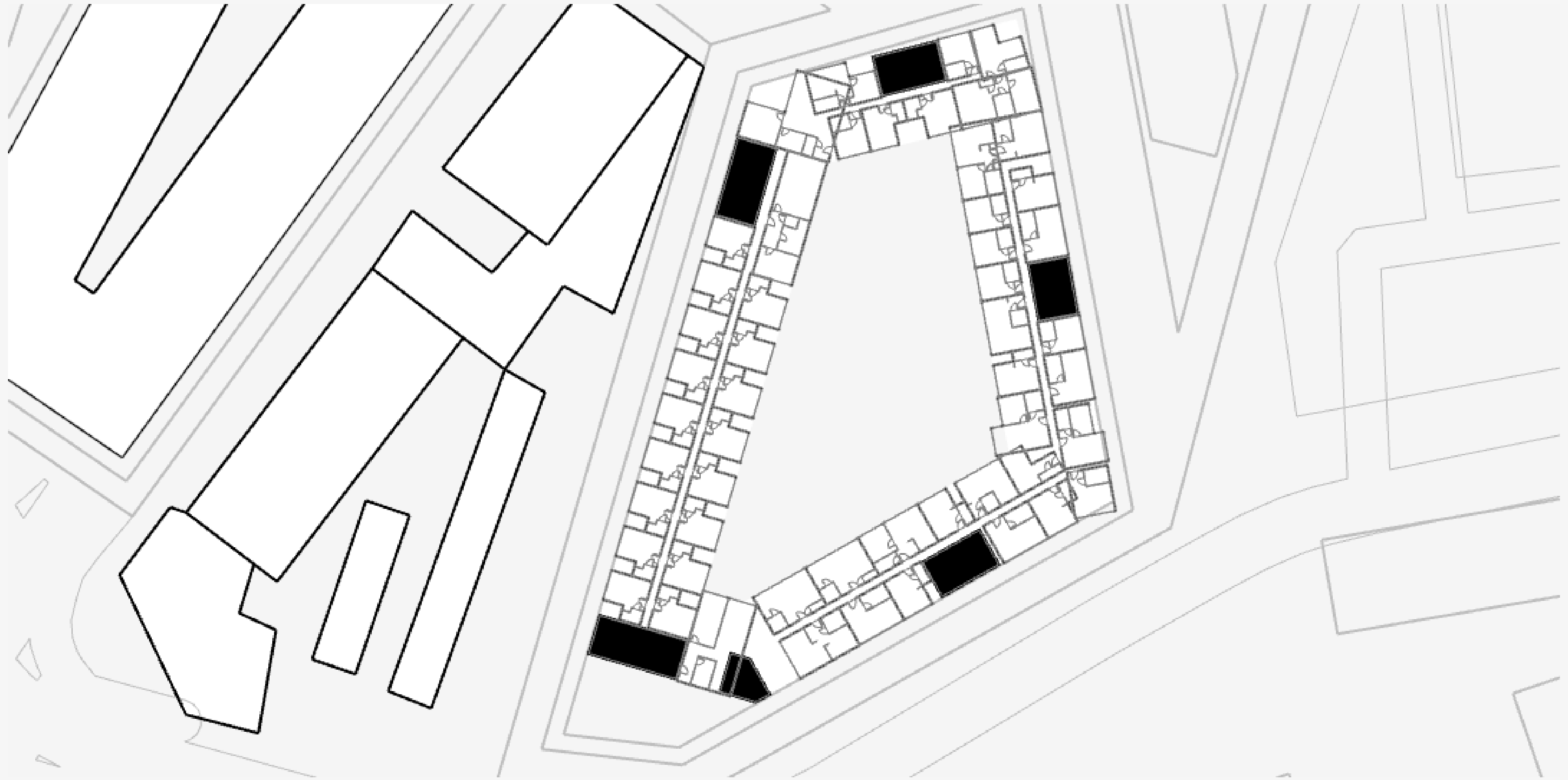




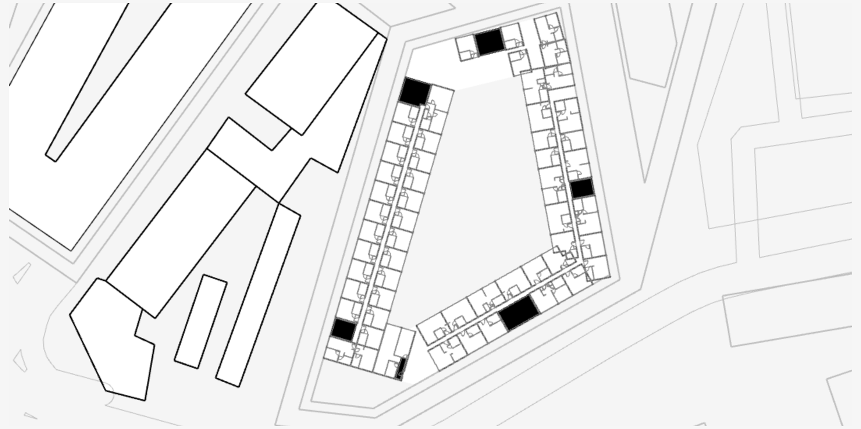
Generation 01 Individual 01



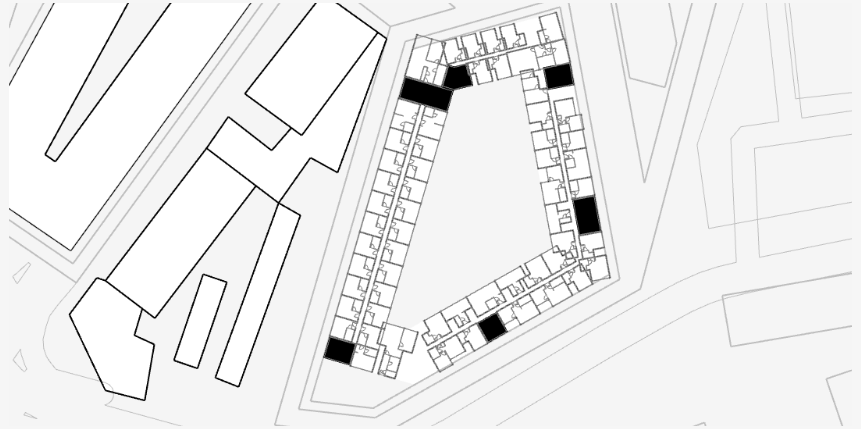
Generation 01 Individual 50



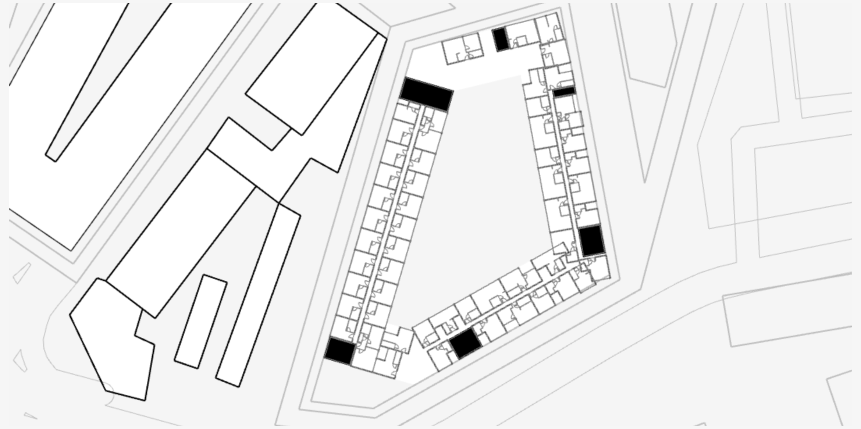
Generation 02 Individual 01



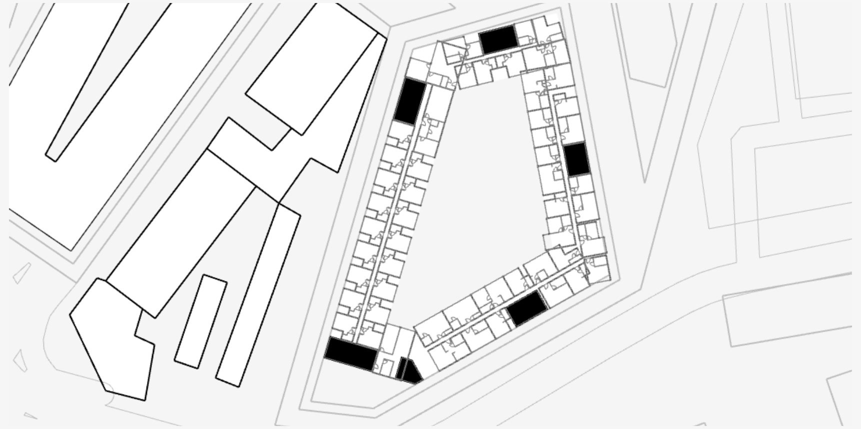
Generation 02 Individual 24



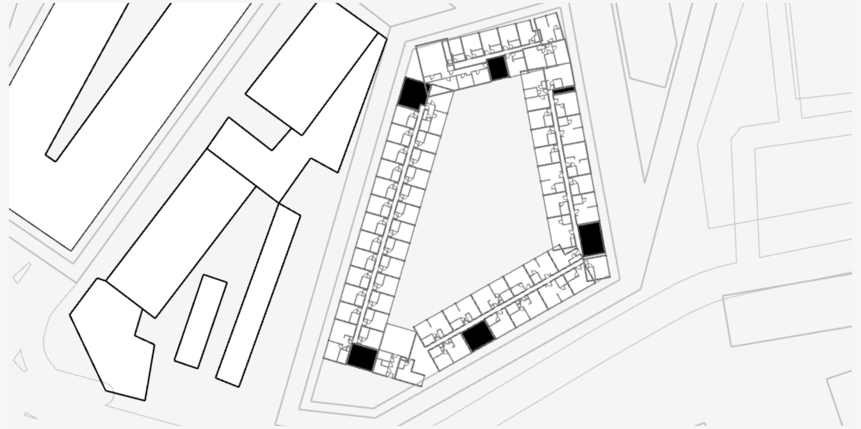
Generation 02 Individual 25



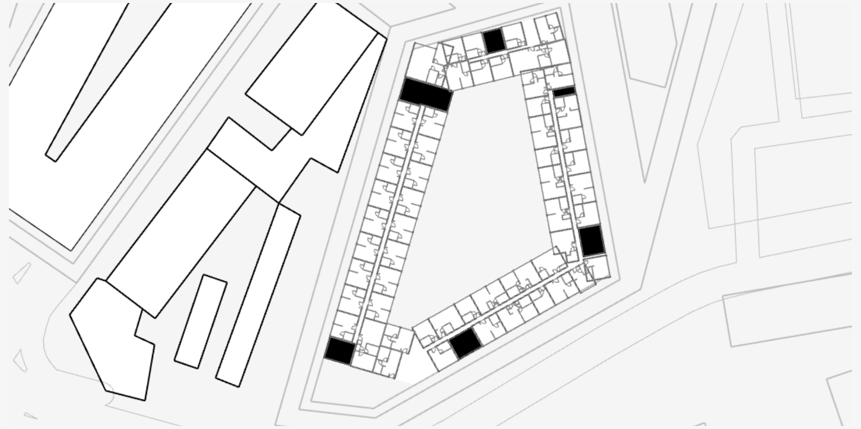
Generation 02 Individual 50



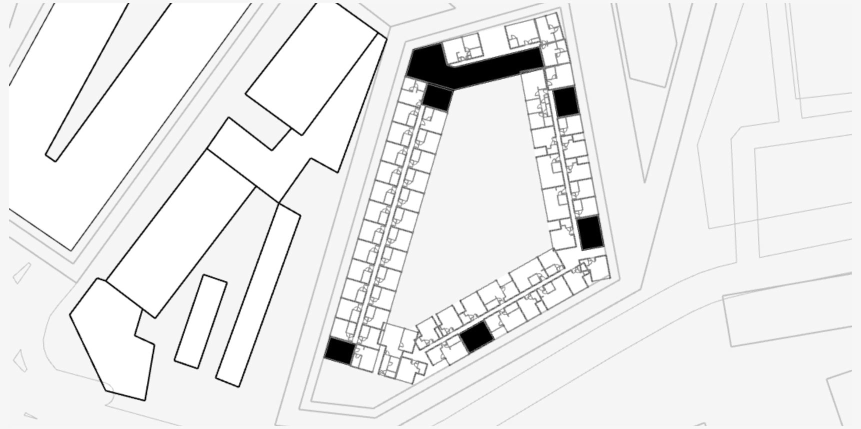
Generation 03 Individual 01



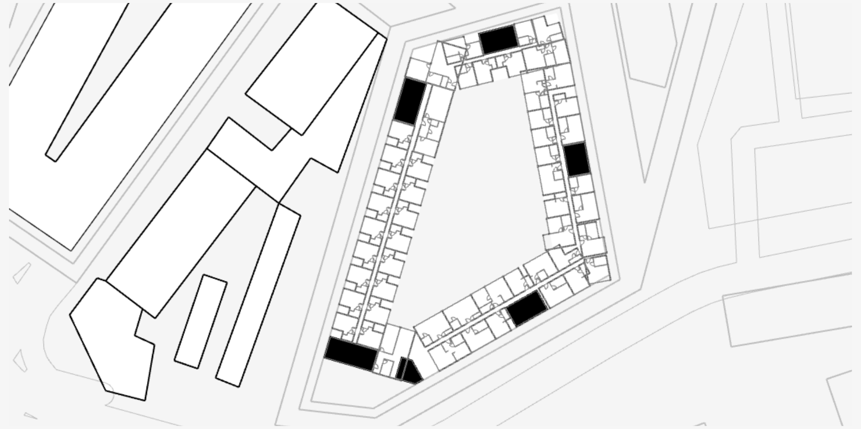
Generation 03 Individual 24



Generation 03 Individual 25



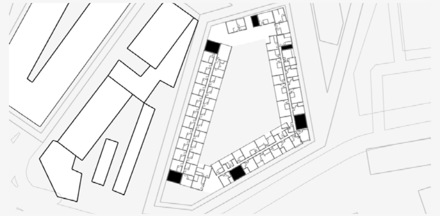
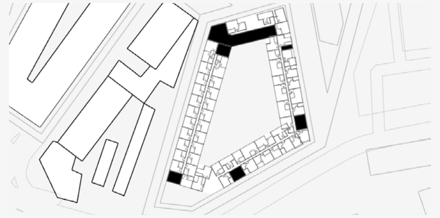
Generation 03 Individual 50

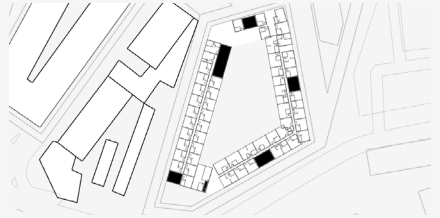


Design Phase:1 Part:2
... 5.5. Design Simulation
... 5.5.2. Running Simulation

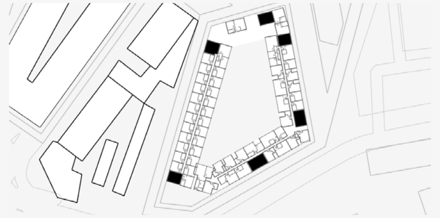
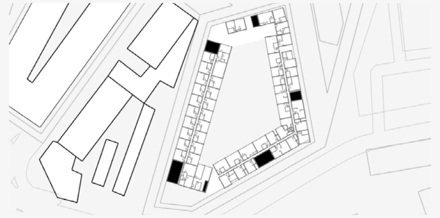
Generation 05 Individual 01

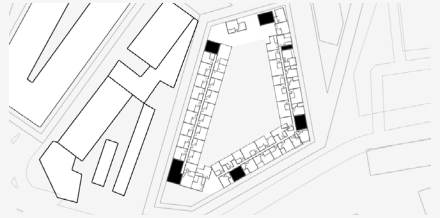
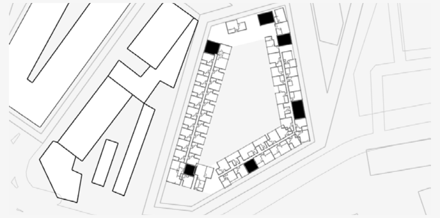
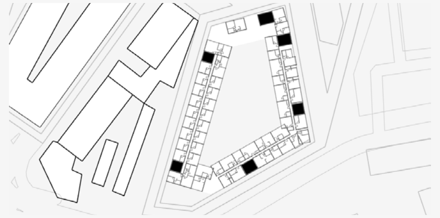
Generation 05 Individual 02

Generation 05 Individual 23

Generation 05 Individual 24

Generation 05 Individual 25

Generation 05 Individual 26

Generation 05 Individual 49

Generation 05 Individual 50

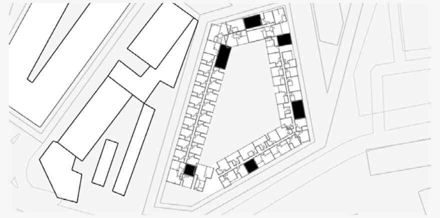

Generation 09 Individual 01

Generation 09 Individual 02

Generation 09 Individual 23

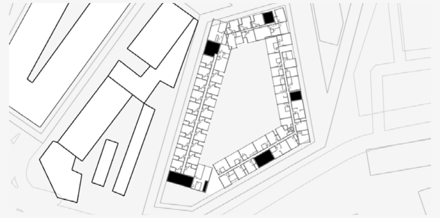
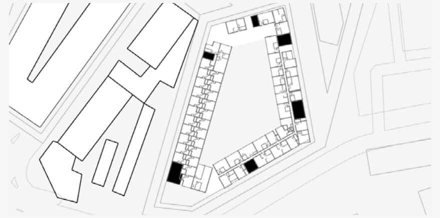
Generation 09 Individual 24

Generation 09 Individual 25

Generation 09 Individual 26

Generation 09 Individual 49

Generation 09 Individual 50

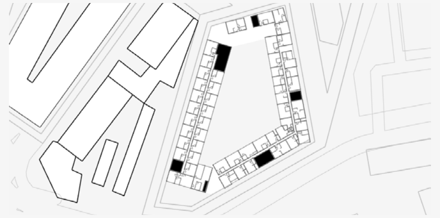
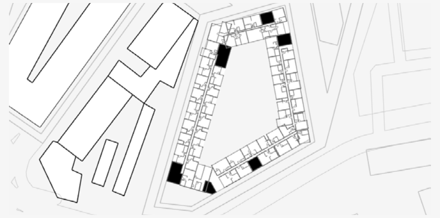
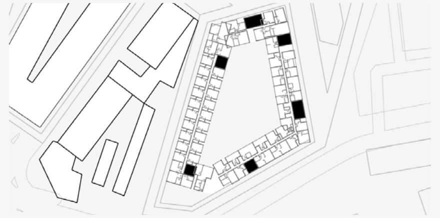
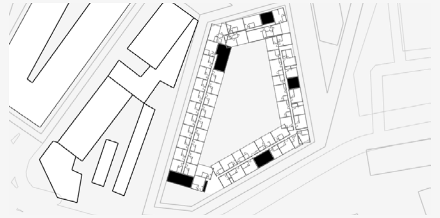
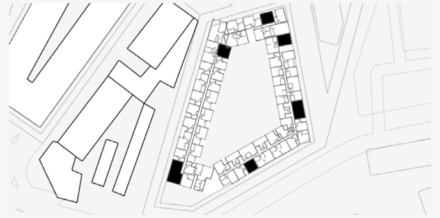
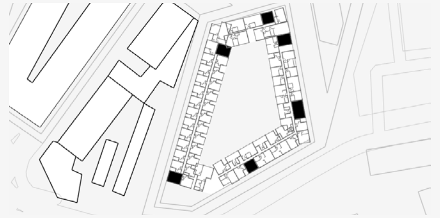
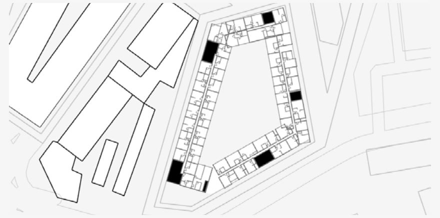
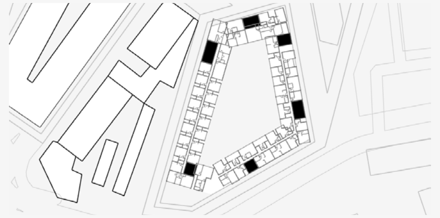

Generation 19 Individual 01

Generation 19 Individual 02

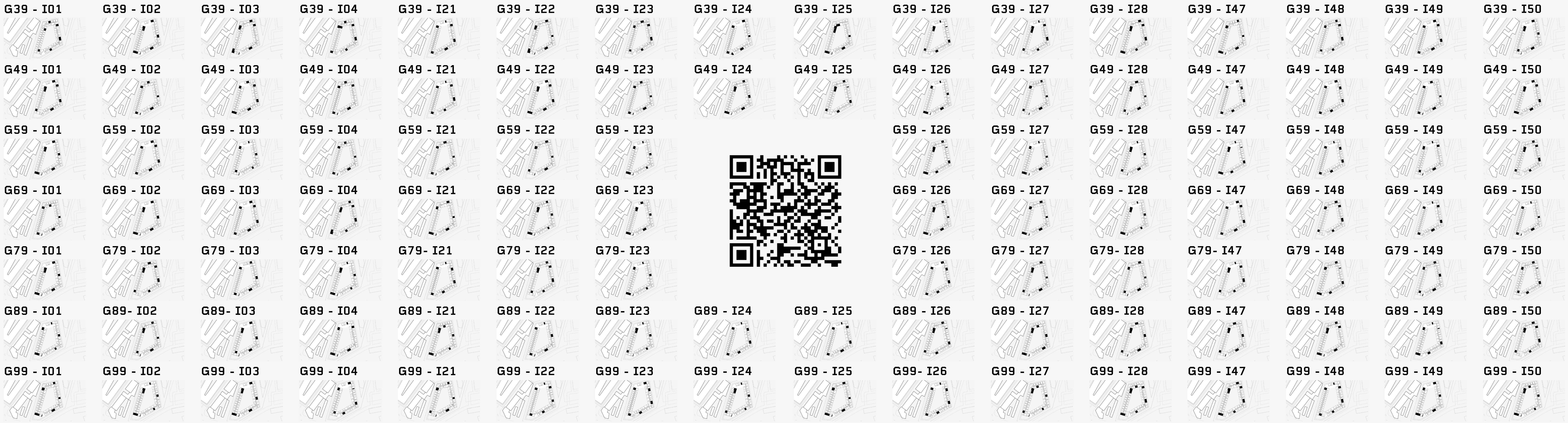
Generation 19 Individual 23

Generation 19 Individual 24

Generation 19 Individual 25

Generation 19 Individual 26

Generation 19 Individual 49

Generation 19 Individual 50


Generation 29 Individual 01

Generation 29 Individual 02

Generation 29 Individual 23

Generation 29 Individual 24

Generation 29 Individual 25

Generation 29 Individual 26

Generation 29 Individual 49

Generation 29 Individual 50


... 5.5.2. Running Simulation



... 5.5.3. Evolutionary Simulation Results

> In this simulation, we did not encounter the same issues as in the previous one, which were caused by CPU power limitations. Since the operations primarily involve a two-dimensional space and require less computational power, we experienced smoother performance. After conducting four simulations, we determined the number of gene values and their corresponding values as illustrated in the image. To address the occasional failure of the plan fitting tool to generate results within the specified boundaries, we implemented a mechanism to create null solutions, ensuring that the optimization process is not hindered.

Therefore, in the end, we conducted a total of five simulations, and the final results are highlighted in the right image. It is noteworthy that generating 5000 design variations took approximately 5 hours.

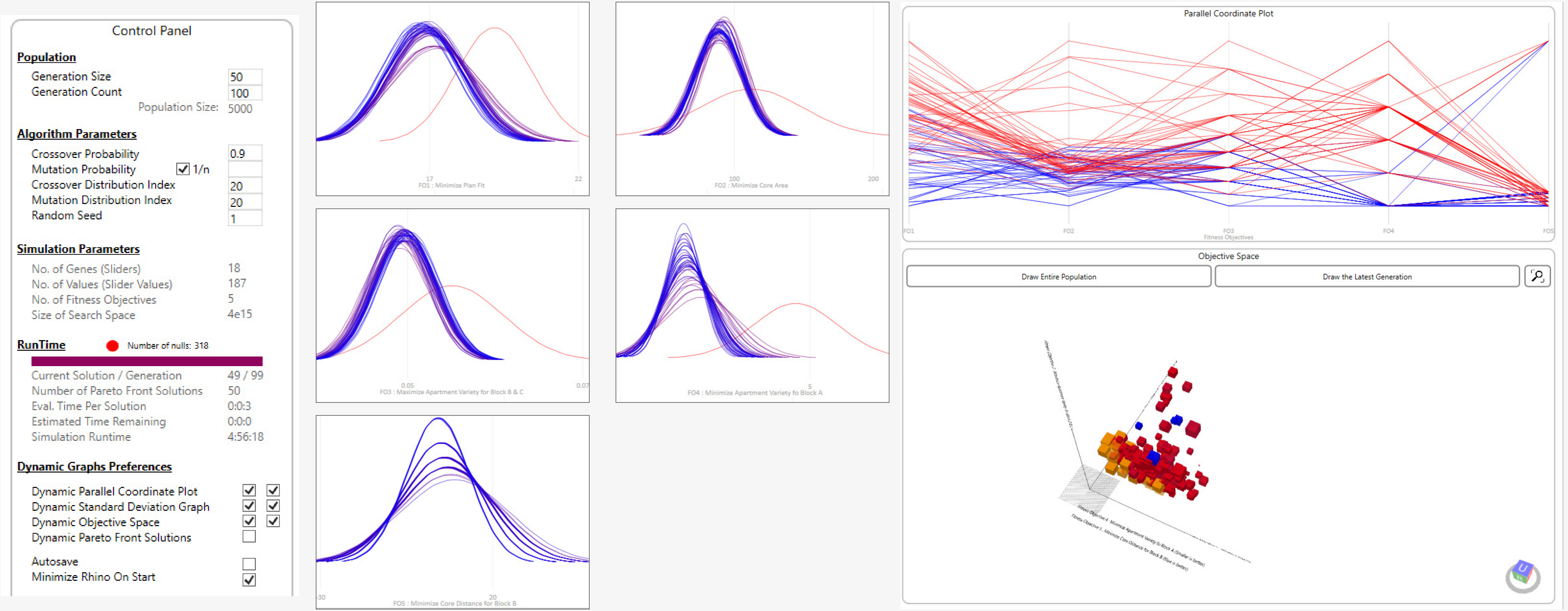
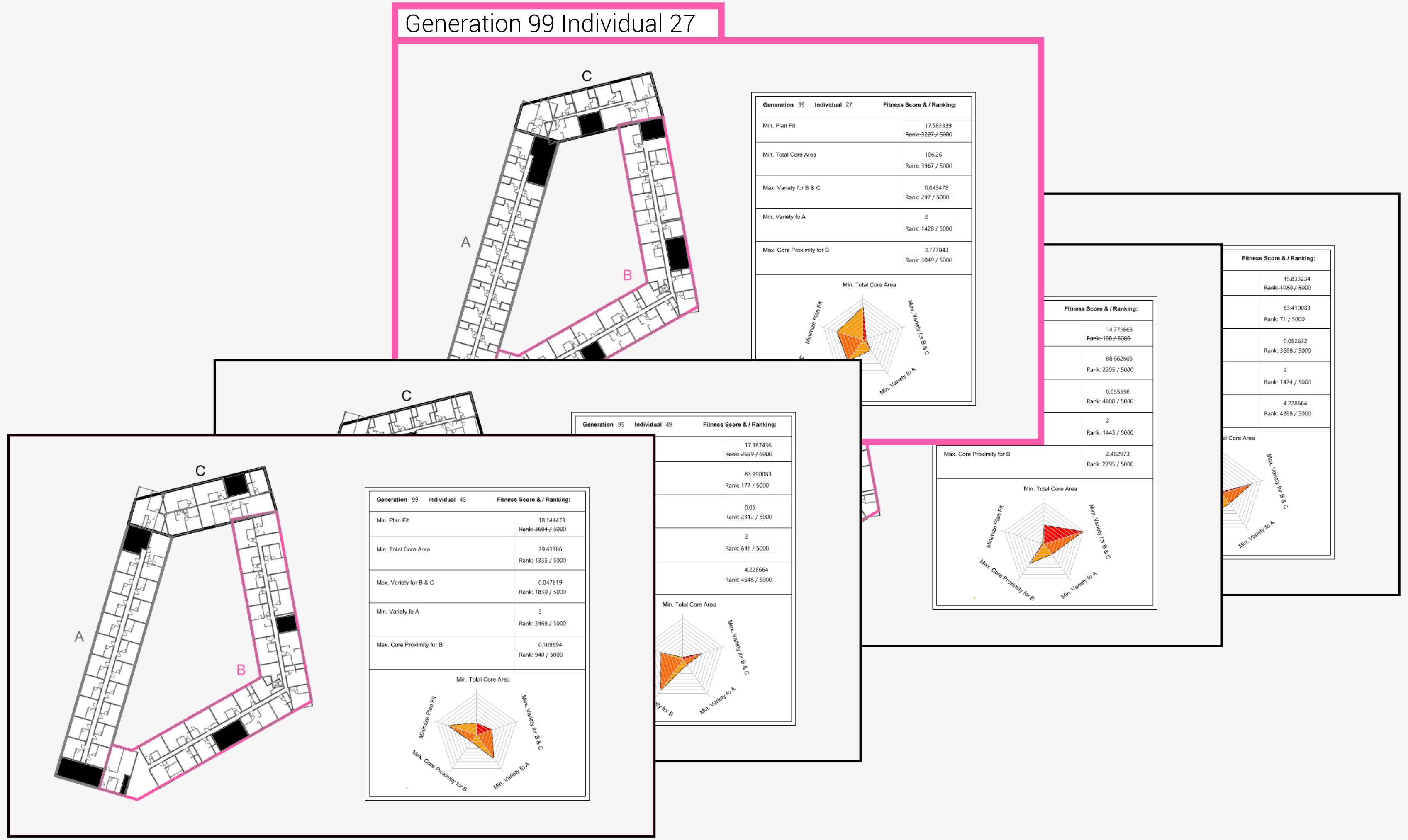


Figure 58: Simulation -2 results and standart deviation of fitness values

... 5.5.4. Sorting & Selecting Phenotypes

> To identify the most desirable design solutions among the 5000 different individuals, we employed a set of sorting methods. Initially, we compiled a list of all the solutions belonging to the Pareto Front in the final generation of the simulation. We evaluated their fitness values and spatial configurations. However, as we were not satisfied with the generated results, we proceeded by listing the fittest solutions in the last generation based on their individual fitness values. Among them, Generation 99 Individual 27 was selected for further examination and analysis.

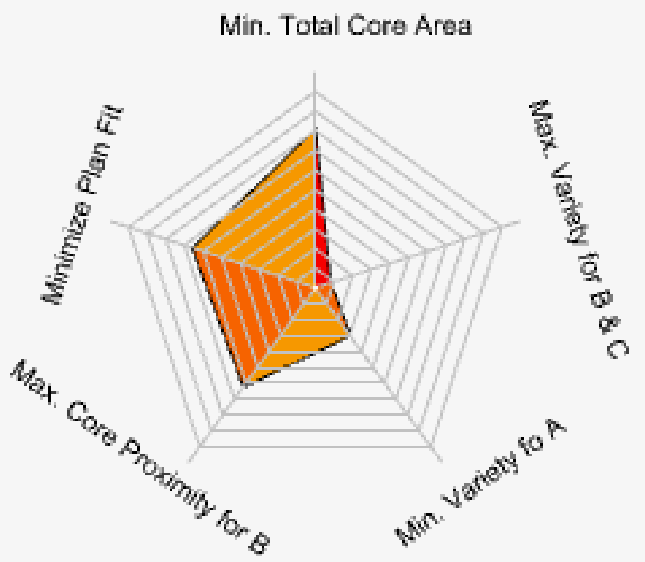


Fittest Ranking Results Based on Single Fitness Values

Non Dominant Solutions (Pareto Front) in the Final Generation of the Simulation

... 5.5.5. Analysis of Selected Individual

> The final design result, which has been selected, is depicted at the center. The callout images represent the different fitness values and their effects on the final design. It is important to note that since the main objective of the design is not to find the most suitable apartment layout for the selected building, there may be unpleasant layouts and configurations within the designed spaces. This is primarily due to the goal of the experiment, and secondarily due to the lack of sufficient training data that has been identified. The primary goal of the experiment is to achieve the highest amount of diversity in Building Blocks B and C, while minimizing diversity in Block A, as indicated by the fitness diamond chart. For a more suitable apartment layout, the reader can scan the provided QR code to explore other design options.



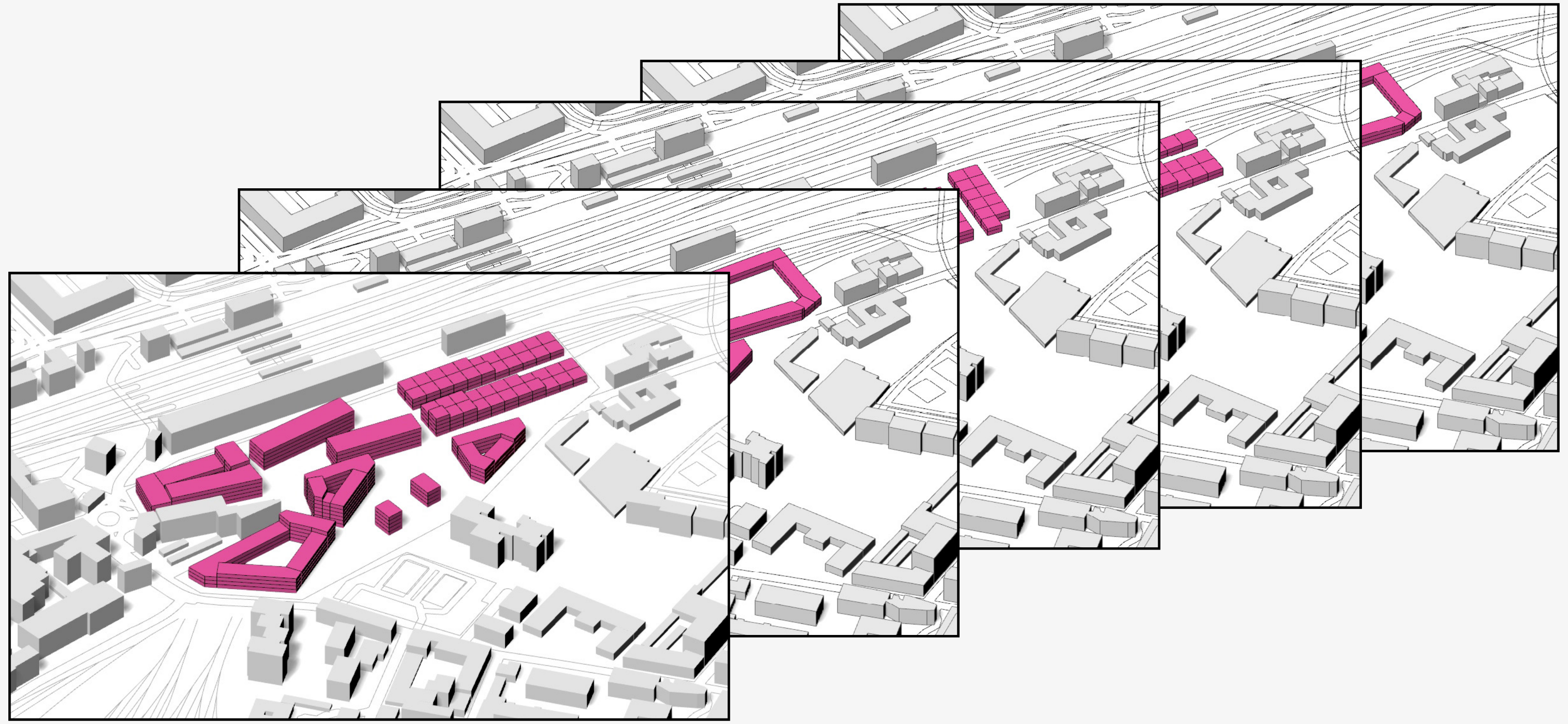
chapter 06
... EVALUATION & DISCUSSION

> The thesis has aimed to develop a comprehensive creative framework that harnesses the power of evolutionary algorithms and parametric design techniques to address the challenges faced in sustainable architecture and urban design. In the first design part, the thesis introduced a creative parametric design tool that creates urban design options for the early design phases. In order to validate the tool two experiments are conducted. We can divide those experiments as machine controlled and user controlled:

In the first experiment, as designers, we have not included the design process rather than defining the goals. The whole exploration process is investigated by the machine by the principles of genetic design. After the simulation ended, population is sorted by the machine based on their fitness values. In the second experiment, as designers we have conducted the whole exploration process rather than using a genetic engine. The aim of the second experiment was to see how extreme scenarios and spatial configurations could be generated and how effectively the tool could assist the designer through the early-design exploration phase.

The results of the two experiments demonstrate that the computational creative framework that has been designed for this thesis is capable of generating well-defined design options that gives high performance in terms of environmental and livable urban aspecs in within a short amount of time, that can compete with those created by an educated group of designers that are working in urban design and architecture industry. The first experiment has successfully demonstrated the capabilities of the computational framework in generating well-defined design options that prioritize environmental aspects and optimize the balance between architectural and planning components. And the second experiments showcased the potential of the framework to create diverse and innovative design solutions.

“Only if virtual evolution can be used to explore a space rich enough so that all the possibilities cannot be considered in advance by the designer, only if what results shocks or at least surprises, can genetic algorithms be considered useful visualization tools.”(DeLanda, 2002). Therefore in our first experiment, we deliberately created an extensive search space, which rendered it impossible to predict the outcomes generated throughout the simulation. However, it is crucial to acknowledge the limitations encountered during the research. The primary concern lies in the integration of genetic design engines. For instance, in our initial experiment, the total search space for design options amounted to $2x10^{60}$. Unfortunately, due to insufficient computer power, we were only able to explore 5000 individuals within this vast search space. The value of $2x10^{60}$ is incredibly large, and dividing it by 5000 would yield an extremely small fraction, rendering it impractical to represent effectively as a percentage.



The results indicate that increasing the iteration numbers has the potential to yield improved results for each fitness value. This limitation underscores the necessity for more powerful computing resources to fully harness the potential of genetic algorithms in architectural design.The future research might better take into consideration of search space and generated individual ratio. Furthermore, the experiment emphasizes the possibility of achieving similar fitness values with a greater variety of urban typologies, which raises questions about our current design approach and highlights the potential of genetic algorithms to generate more diverse and innovative solutions.

Moving on to the second design part of the thesis, the focus shifted to the architectural plan creation level. The development of a plan creator tool involved addressing generic geometrical issues in apartment design, such as spatial configuration, entrance placement, core locations based on regulations, and ensuring direct access to fire escapes for all inhabitants. However, we encountered several challenges with the plan creator tool especially for the genetic optimization process. Specifically, it failed to properly resolve the relationships between buildings that are sharing common relations, such as sharing the same blind facade or same entrances. This inadequacy can be attributed to the script's limited integration of the complex social and contextual relations inherent in the design process. For predicting interior layout of the apartments we used machine learning algorithms. The experiment shows us machine learning algorithms are limited by the information contained in the training data. If certain important features or patterns are missing from the data, the model may not be able to learn them. Additionally, if the training data does not capture all possible scenarios or edge cases, the model may struggle to make accurate predictions in those cases.

In conclusion, this thesis contributes to the field of sustainability design for architecture by presenting a creative computational framework that generates well-defined design options and highlights the potential of genetic algorithms for diversifying and innovating solutions The tool can be utilized by architects and urban designers during the early stages of a project when exploring different design options and generating initial layouts. It can help in analyzing spatial relationships, optimizing site utilization, and considering various constraints and parameters. By leveraging AI algorithms and automation, the tool can rapidly generate and evaluate multiple design alternatives, allowing designers to iterate and refine their ideas more efficiently. Furthermore, the tool can be beneficial for urban planning authorities, developers, and other stakeholders involved in the design and decision-making process. It provides a means to visualize and assess the impact of different design scenarios, facilitating communication and collaboration among multidisciplinary teams.

> Based on the experiments conducted in the thesis, several observations have emerged that can serve as a foundation for future research and advancements in the field:

While we were generating individuals, most of the fitness values that were parameterized were quantitative aspects such as solar radiation, building floor area, etc. However, when comparing two design options, these parameters are not enough since a livable area consists of many layers and social aspects. We also tried to integrate social parameters such as typology variety and parallelly diversity of outdoor spaces in order to increase the overall livable quality of generated spaces. It is important to consider the broader impact of the proposed framework on creating vibrant public spaces, fostering community engagement, and enhancing the overall livability of urban areas. Incorporating social, cultural, and economic considerations into the framework can further support the creation of sustainable and people-centric urban environments. Future research efforts can explore ways to integrate these factors more effectively into the computational framework, enabling the generation of design solutions that holistically address the needs and aspirations of the community.

To unlock the full potential of AI in urban design and architecture, it is crucial to address the barriers between AI systems and datasets. Currently, access to high-quality, diverse, and comprehensive datasets is often limited. Future works should emphasize efforts to collect and curate datasets specifically tailored for urban design purposes. Additionally, there is a need for standardized data formats and interoperability protocols that facilitate seamless integration of AI systems with different datasets. By removing these barriers, designers and researchers can tap into a wider range of data sources, enabling more accurate predictions, simulations, and optimizations. This, in turn, can lead to more informed and sustainable design decisions.

The advent of advanced language models, like ChatGPT and similar prediction models, opens up new possibilities for creative applications in urban design and architecture. These models can understand and generate human-like text, making them valuable tools for generating design ideas, simulating user experiences, and supporting decision-making processes. In future works, researchers can explore the integration of ChatGPT and similar models in the early stages of the design process. Architects and urban designers can collaborate with these AI models to generate design options, evaluate their feasibility, and explore alternative solutions. This approach can enhance the creative process and facilitate more efficient and sustainable design outcomes.

Figure 60: Emerging tools and programs in AI industry

> Bibliography

...1. Negroponte, N. (1972). *The architecture machine: Toward a more human environment.* M.I.T. Press.

...2. Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Longman Publishing Co., Inc.

...3. Pareto, V. (2020). *Manual of Political Economy: A Critical and Variorum Edition*. Oxford University Press. (Original work published 1906).

...4. Yolanda, B. P. M., & Haupt, P. A. (2009). *Space, density and urban form (Doctoral thesis).* Retrieved July 10, 2023, from https://repository.tudelft.nl/islandora/object/uuid%3A0e8cdd4d-80d0-4c4c-97dc-dbb9e5eee7c2.

...5. A+t Architecture. (2017). *50 urban blocks.* Avisa.

...6. Sim, D., & Gehl, J. (2019). *Soft City Building Density for Everyday Life.* Island press.

...7. Burry, M. (1997). *Narrowing the Gap Between CAAD and Computer Programming: A Re-Examination of the Relationship Between Architects as Computer-Based Designers and Software Engineers, Authors of the CAAD Environment, in*: CAADRIA '97 Taiwan 17-19 April 1997, pp. 491-498.

...8. Boeykens, S., & Neuckermans, H. (2013). *Visual Programming in Architecture: Should Architects Be Trained As Programmers? T. Tidafi and T. Dorta (Eds) Joining Languages Cultures and Visions:* CAADFutures 2009 CD-Rom.

...9. Tedeschi, A. and Lombardi, D., (2018). *The algorithms aided design (AAD). In Informed Architecture (pp. 33-38).* Springer, Cham.

...10. Sadeghipour Roudsari, M., & Pak, M. (2013). *Ladybug: A parametric environmental plugin for grasshopper to help designers create an environmentally-conscious design. Proceedings of BS 2013:* 13th Conference of the International Building Performance Simulation Association, 3128–3135.

...11. Tedeschi, A., (2014). *AAD, Algorithms-aided design: parametric strategies using Grasshopper.* Le Penseur Publisher.

...12. Crawley, D.B., Hand, J.W. and Lawrie, L.K., (1999), *September. Improving the weather information available to simulation programs.* In Proceedings of Building Simulation'99 (Vol. 2, pp. 529-536).

...13. Michalewicz, Z. (1994). *Genetic Algorithms + Data Structures = Evolution Programs.* https://doi.org/10.1007/978-3-662-07418-3

...14. Osintseva, I., Koenig, R., Berst, A., Bielik, M., & Schneider, S. (2020). *Automated Parametric Building Volume Generation: A Case Study for Urban Blocks. In Proceedings of the Symposium on Simulation for Architecture and Urban Design (SimAUD 2020), May 25-27,* Online. Society for Modeling & Simulation International (SCS).

...15. Fuegi, J., & Francis, J. (2003). *Lovelace & Babbage and the Creation of the 1843 'Notes'. IEEE Annals of the History of Computing, 25, 16–26.* doi:10.1145/2810201

...16. Leach, N. (2022). *AI, art and creativity. In Architecture in the age of artificial intelligence: An introduction to AI for architects (pp. 61–62).* Bloomsbury Visual Arts.

...17. Chaillou, S. (2019). *AI + Architecture | Towards a New Approach.* Harvard University, 188.

...18. Rahmeh, S. (2022, May 23). *Architectura Intelligence. DigitalFUTURES.* https://digitalfutures.international/workshop/architectural-intelligence/

...19. DeLanda, M. (2002). *Deleuze and the use of the genetic algorithm in architecture.* Architectural Design, 72, 9–12.

> Software

... Rhinoceros
v.7.21/(Main software used to conduct the design)/Computer Software/(2021)/Retrieved from www.rhino3d.com

... Grasshopper
v.1.00/(Main plugin visual programming)/Plugin of Rhinoceros/(2007)/Retrieved from www.grasshopper3d.com

... Testfit
(Used for background research)/Cloud software for generative design/(2017)/Retrieved from https://testfit.io/

... Architechtures
(Used for background research)/Cloud software for generative design/(2020)/Retrieved from https://architechtures.com/en/

... Hypar
(Used for background research)/Cloud software for generative design/(2019)/Retrieved from https://hypar.io/

> Plug-in

... Human,
(Used for referencing and visualizing geometry)/Plug-in of Grasshopper for managing scripts/(2017)/Retrieved from https://www.food4rhino.com/en/app/human

... Sasquatch,
(Used for managing geometry)/Plug-in of Grasshopper for managing geometry/(2019)/Retrieved from https://www.food4rhino.com/en/app/sasquatch-utilities

... Pufferfish,
(Used for managing geometry)/Plug-in of Grasshopper for managing geometry/(2021)/Retrieved from https://www.food4rhino.com/en/app/pufferfish

... WallaceiSolver,
(Used for multi-objective optimization)/Plug-in of Grasshopper for multi-objective optimization/(2018)/Retrieved from https://www.food4rhino.com/en/app/wallacei

... PlanFinder
/(Used for automated floor plan generation and optimization)/Plug-in of Grasshopper for floor plan generation/(year)/Retrieved from https://www.food4rhino.com/en/app/planfinder

... DecodingSpaces
/(Used for generation of plots in urban design)/Plug-in of Grasshopper for generative design/(2019)/Retrieved from https://toolbox.decodingspaces.net/

... Heteroptera,
(Used for data manipulation)/Plug-in of Grasshopper for dynamic relaxation and form-finding/(2023)/Retrieved from https://www.food4rhino.com/en/app/heteroptera

... Ladybug
(Used for sustainable analysis)/Plug-in of Grasshopper for sustainability analysis/(2011)/Retrieved from https://www.ladybug.tools/

... Mosquito
(Used for script management in Grasshopper)/Plug-in of Grasshopper for media management/(year)/Retrieved from https://www.food4rhino.com/en/app/mosquito-media-4-grasshopper

... Metahopper
(Used for advanced data management and workflow automation in Grasshopper)/Plug-in of Grasshopper for advanced data management and workflow automation/(2019)/Retrieved from https://www.food4rhino.com/en/app/metahopper

... Minimum Bounding Box
(Used for geometry manipulation and analysis)/Plug-in of Grasshopper for minimum bounding box/(2021)/Retrieved from https://www.food4rhino.com/en/app/minimum-bounding-box

... Elefront
(Used for managing and automating Rhino and Grasshopper workflows)/Plug-in of Grasshopper for managing and automating workflows/(2023)/Retrieved from https://www.food4rhino.com/en/app/elefront

... Digiwo
(Used for background research)/Plug-in of Grasshopper for generative design/(2020)/Retrieved from https://toolbox.decodingspaces.net/digiwo/

> Database

... OpenStreetMap
(Database for context information)/Website/(2004)/Retrieved from www.openstreetmap.org

... Ladybug.tools/epwmap/
(Database for weather data)/Website/(2011)/Retrieved from https://www.ladybug.tools/epwmap/