Technische Universität München
TUM School of Computation, Information and Technology

Procedural Generation of Virtual Environments as Context for Mixed Reality Experiences in the Automotive Design

Despoina Salpisti

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung einer

Doktorin der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Prof. Dr.-Ing. Matthias Nießner

Prüfer*innen der Dissertation:

1. Prof. Gudrun Johanna Klinker, Ph.D.
2. Prof. Dr. Klaus Bengler

Die Dissertation wurde am 30.11.2022 bei der Technischen Universität München eingereicht und durch das TUM School of Computation, Information and Technology am 08.05.2023 angenommen.

To my Family

Abstract

The use of Mixed Reality experiences in the early stages of evaluating and presenting aesthetic automotive design offers many advantages over mood boards and PowerPoint presentations. With the help of this method, designers can live the experience their design can facilitate and adapt it accordingly. Using this prototyping method requires an approach for implementing the right experience and its context. The latter plays a crucial role in ensuring that designers have the same experience as their end users, and consequently for the success of the experience as a prototype. However, the implementation of the experience context, which includes the virtual environment, and its conditions is still accompanied by many technical challenges.

To gather information about the right experience context, expert interviews with designers and developers were conducted. The results show different use cases for different design departments. By observing the collaboration between them, we also learned about the challenges that make the current process of defining and creating the environment inefficient. These challenges include the large amount of resources required to create the right context and the lack of a common language to describe and interpret the virtual environment. Based on these findings, we have outlined a design for a customized process that has the potential to allow designers to create their own context of experience.

This new process uses a framework for procedural generation of virtual environments. The output geometry is the result of a series of procedural generators that we created using Open Street Map information. These generators model the streets, buildings and vegetation automatically and within an efficient time frame. Furthermore, the quality of the output is influenced by the quality of the exchangeable assets, which are selected based on the geographical area. Moreover, the street geometry provides realistic street conditions as it is based on real map data, while characteristic architectural styles aim to represent the world area defined by the geographic coordinates. The only input the system requires is the selection of an area on the map. This input can be transmitted to the system via the user interface, which we also created. This user interface contains all the options we gathered as requirements in the interview studies, and it should show the designers the overall vision of the implementation, i.e., the ability to develop a complete experience. Using options to select specific weather conditions, times of day, avatars, vehicle and camera perspectives, designers can create their own experience demos and get the final experience.

This new process of defining and creating the experience context, the user interface, and the output generated by the automated framework were the focus of the evaluation studies conducted with designers and developers. The results showed the suitability of

the generated output, a positive user experience with the user interface and the proposed process, and finally the enthusiasm of the study participants about the concept and its vision.

Kurzfassung

Der Einsatz von Mixed-Reality-Erlebnissen in der frühen Phase der Bewertung und Präsentation von ästhetischem Automobildesign bietet viele Vorteile gegenüber Moodboards und PowerPoint-Präsentationen. Mithilfe dieser Methode können Designer die Erfahrungen, die ihr Design ermöglichen kann, live erleben und entsprechend anpassen. Der Einsatz einer solchen Prototyping-Methode erfordert einen Ansatz zur Umsetzung des richtigen Erlebnisses und dessen Kontext. Letzteres spielt eine entscheidende Rolle, wenn es darum geht, dass die Designer die gleichen Erfahrungen machen wie ihre Endnutzer und somit für den Erfolg des Prototyps. Die Umsetzung des Erlebniskontextes, zu dem auch die virtuelle Umgebung gehört, und seiner Bedingungen ist jedoch nach wie vor mit vielen technischen Herausforderungen verbunden.

Um Informationen über den richtigen Erfahrungskontext zu sammeln, wurden Experteninterviews mit Designern und Entwicklern durchgeführt. Die Ergebnisse zeigen unterschiedliche Anwendungsfälle für verschiedene Designabteilungen. Durch die Beobachtung der Zusammenarbeit zwischen ihnen erfuhren wir auch, welche Herausforderungen den derzeitigen Prozess der Definition und Erstellung der Umgebung ineffizient machen. Zu diesen Herausforderungen gehören der hohe Ressourcenaufwand für die Erstellung des richtigen Kontextes und das Fehlen einer gemeinsamen Sprache zur Beschreibung und Interpretation der virtuellen Umgebung. Auf der Grundlage dieser Erkenntnisse haben wir einen maßgeschneiderten Prozess skizziert, der es den Designern ermöglicht, ihren eigenen Erfahrungskontext zu schaffen.

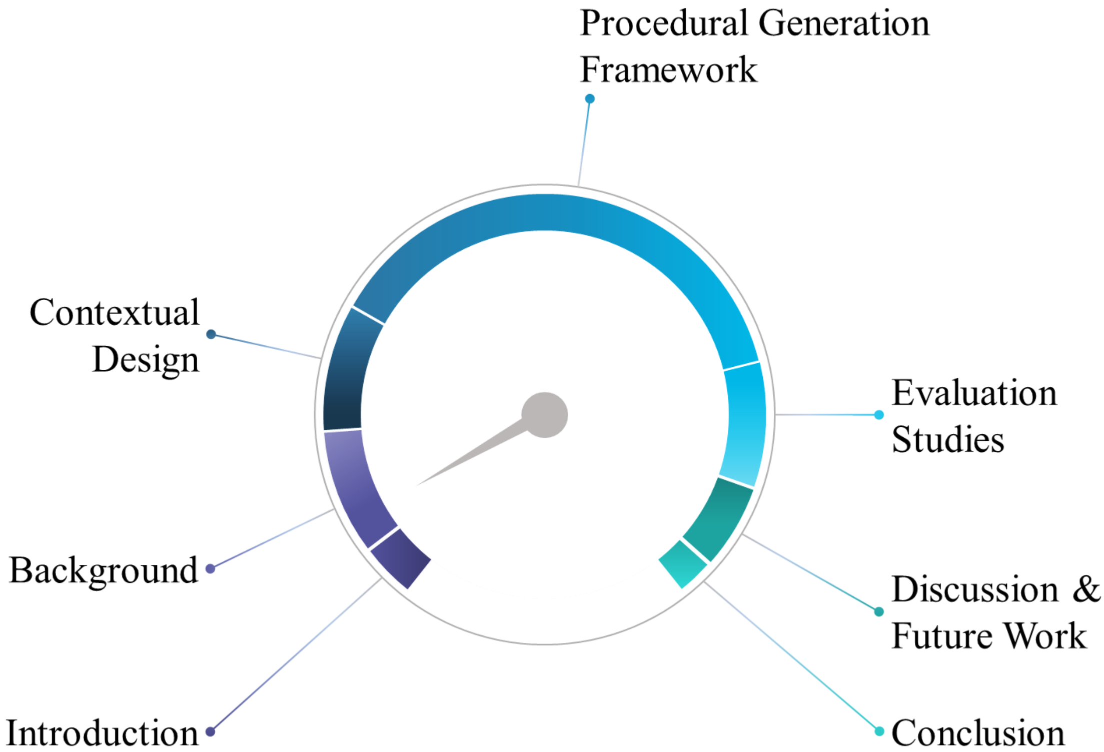
Dieses neue Verfahren nutzt ein Framework für die prozedurale Erzeugung virtueller Umgebungen. Die Ausgangsgeometrie ist das Ergebnis einer Reihe von prozeduralen Generatoren, die wir anhand von Open Street Map-Informationen erstellt haben. Diese Generatoren modellieren die Straßen, Gebäude und die Vegetation automatisch und innerhalb eines effizienten Zeitrahmens. Darüber hinaus wird die Qualität der Ausgabe durch die Qualität der austauschbaren Assets beeinflusst, sowie von charakteristischen Architekturstilen, die auf der Grundlage des geografischen Ortes ausgewählt und angewandt werden. Außerdem bietet die Straßengeometrie realistische Straßenverhältnisse, da sie auf realen Kartendaten basiert. Die einzige Eingabe, die das System benötigt, ist die Auswahl eines Bereichs auf der Karte. Diese Eingabe kann über die ebenfalls von uns erstellte Benutzeroberfläche an das System übermittelt werden. Diese Benutzerschnittstelle enthält alle Optionen, die wir in den Interviewstudien als Anforderungen gesammelt haben und zeigt den Designern somit die Gesamtvision der Implementierung um ein vollständiges Erlebnis zu entwickeln. Mithilfe von Optionen zur Auswahl bestimmter Wetterbedingungen, Tageszeiten, Avatare, Fahrzeug- und

Kameraperspektiven können Designer ihre eigenen Erlebnisdemos erstellen und das endgültige Erlebnis erhalten.

Dieser neue Prozess der Definition und Erstellung des Erlebniskontextes, der Benutzeroberfläche und der durch das automatisierte Framework erzeugten Ausgabe stand im Mittelpunkt der mit Designern und Entwicklern durchgeführten Evaluierungsstudien. Die Ergebnisse zeigten die Eignung des generierten Outputs, eine positive Benutzererfahrung mit der Benutzeroberfläche und dem vorgeschlagenen Prozess und schließlich die Begeisterung der Studienteilnehmer über das Konzept und seine Vision.

Contents

I. Introduction and Theoretical Background



1 Introduction

1.1 Introduction

„Η μεν εμπειρία τέχνην εποίησεν, η δ' απειρία τύχην."[1]

What Aristoteles [1] wanted to express with Polus' sentence is that an experienced person possesses skills they can count on, while an inexperienced person can only rely on their luck. It is an undeniable fact, that when we experience something with our own senses, rather than hearing about it or imagining the experience, we can gain more accurate information about its events and the factors influencing it. This information is crucial for automotive designers when it comes to creating a specific experience for their target users. Experiencing their design under the same conditions as the target users helps them evaluate the experience from their perspective. Based on the evaluation results, they can adjust the design features to facilitate an experience that is both optimal for the target users and conveys the intended emotions.

User Experience describes how a user interacts with a product and how this interaction is perceived. If this product is a car, the user experience is expected to become more complex. This is due to the large number of design elements in a vehicle that can potentially enrich the user experience through a correspondingly large number of interactions. The nature of the experience is subjective rendering the automotive designer's task of understanding, satisfying, and challenging the target user, extremely difficult. When developing a new automotive design, many design aspects should be considered, as well as how they influence the user's experience. The goal is to design a car that offers the experience the designer has in mind and consequently elicits certain emotions [2].

Until today, the evaluation and presentation of the aesthetic automotive design is conducted with the help of sketches, PowerPoint presentations, videos that were shot at the location of the target market, as well as clay and foam models. Only recently, a new trend in automotive prototyping has been introduced which has its roots in experience prototyping. This new direction offers designers the opportunity to experience their own design at early stages of the development process. With the right level of prototyping fidelity, this experience can be used as the basis for important design decisions. The presentation and evaluation method of the aesthetic vehicle design should correspond to

[1] „Experience produces art, but inexperience chance" [3].

the designer's goal, which is to facilitate a certain experience and to evoke corresponding emotions in the user. Accordingly, designers should be enabled to test this experience under the same conditions as the end user. These conditions are characterized by the functional [4] and aesthetic context [5] of the surrounding environment. With the right functional context, the designer can experience a particular design feature. For example, if the design scenario dictates the driver's interaction with the vehicle's central display, a highway would be more suitable than a curvy mountain road. This is because the driver needs to be more concentrated when driving on a curvy mountain road than on a highway, and therefore cannot share much of their attention with the central display. With the right aesthetic context, the designer can experience the vehicle's design in the environment of the end user. In this case, the environment should have the "look and feel" of the target market location and represent the vehicle's character. In terms of the aesthetic context, the environment should not distract the designer's attention too much from the vehicle's features by providing the right amount of detail.

To use this new method of evaluating and presenting the aesthetic automotive design, design scenarios have to be transformed into Mixed Reality (MR) experiences that serve as prototypes and help to make design decisions. These design stories consist of a sequence of actions, performed by one or more characters (e.g., the driver and co-drivers) under certain conditions (e.g., time and place) [6, 7]. They are inspired by common experiences of the target users (e.g., driving on a multi-level highway in Shanghai) and they describe the user's interaction with certain product features in certain situations.

According to the interview studies with designers and developers, as well as the observations of their collaboration, there are still many problems to overcome in the development of these design experiences. One of the most important factors in these experiences and simultaneously one that poses many technical challenges, is the 3D virtual environment, which serves as the context of the experience. The continuously changing requirements of the designers for the environment, the inefficient terms that are used to describe it to the developers and the not-always-available extensive resources which are required to model the environment with the required quality and features lead to a false experience context and consequently to lower-quality design decisions. On the one hand, automotive designers face the challenge of being dependent on developers and the available resources when they need to experience their design scenarios in aesthetic design presentations and evaluations. On the other hand, the environments required for these prototyping experiences are often needed within a limited amount of time. This can be a major issue as the development of these environments with the required graphics quality requires a lot of resources. A specific set of modelled environments is also not a solution as the use cases may vary too much. An additional challenge is the communication of the requirements for the environment to the developers, both in terms of accuracy and time.

The challenges of the current design process and the requirements for the virtual environment that were uncovered in this work highlighted the need for a change in the design process by leveraging procedural generation to offer designers the automated development of their desired environment based on the selection of an area in the world. Procedural generation has reduced the modelling time for virtual environments during the development of many open world games while offering the advantage of being able to create a model without the need for programming skills.

One group of people in need of such a method to improve the efficiency of their work are automotive designers. The automotive industry evolves at high speed. This results in the need to constantly create new design scenarios and consequently to evaluate and present them as corresponding design experiences [8]. Using procedural generation, we provided designers with the opportunity to customize and create their own experience context. By using real map data, these environments were able to provide designers with real conditions for the functional aspect of the context and the ability to describe the environment based on a real map.

A series of procedural generators was created to model the terrain, streets, buildings and vegetation of any area in the world as a first step of enabling designers to model their own experience context. To this end, Open Street Map (OSM) data was chosen as input to the generator, not to model the exact representation of reality, but to model an environment that provides the designers with the appropriate context to depict their design stories. The focus of the procedural generation was on representative architectural styles, minimal user effort in generation and increasing the variety of all geometries except streets.

A series of qualitative and quantitative surveys was conducted to investigate the reliability of the procedural framework, the suitability of its output and the usability of the process it dictates. The results of these evaluation studies confirmed the suitability of the procedural framework for creating the design experience context and its technical acceptance by the designers.

1.2 Motivation & Goals

The goal of this thesis is to help automotive designers develop MR experiences that enable them to better adapt their design to their target users after they have experienced it themselves under the target conditions of use. This main objective was divided into the following sub-objectives:

Current Work Models. To be able to help automotive designers, it is important to obtain information about the current process used to develop these MR experiences, including the communication and cooperation flow involved. This will allow a deeper understanding of the current situation, a better analysis of the sources of problems and the optimal transformation of the current process into one that is better adjusted to the needs of the designers.

Collection of Requirements. To develop a framework that supports automotive designers in the development of MR experiences, the aspects of these experiences and the terms required to describe them need to be known in advance. The experience aspects provide a deeper understanding of the theoretical requirements for optimal development, while the terms indicate the parameters that allow designers to control the framework and achieve the desired output.

Collection of Use Cases. When gathering the requirements for the framework, the relevant use cases should be clustered, and the associated MR experiences should be differentiated according to the target design and corresponding experience focus. In the case of a vehicle, the target design can vary from interior to UI and the focus of the evaluation and presentation changes accordingly. In terms of the experience, this means that a variable experience context is required that can best visualize each design scenario.

Suggested Process. Adapting the current process of developing design experiences is important for incorporating the new framework. Consequently, a new process needs to be defined that designers can and will follow in order to achieve the desired result.

Procedural Generation Framework. A framework that is able to automatically generate the required virtual environment with a minimum amount of resources, without compromising the quality of the produced output, is crucial for the application of the experience prototyping method to evaluate automotive aesthetic design. Furthermore, a framework that allows automotive designers to generate their own experience context independently of developers can lead to both a more efficient process, as the person defining an appropriate environment can create it directly, and a more creative process for designers, as they have the ability to test different environments for their design scenarios in a shorter amount of time.

1.3 Thesis Structure

This thesis is divided into three main parts that include the chapters visualized in the following Figure:

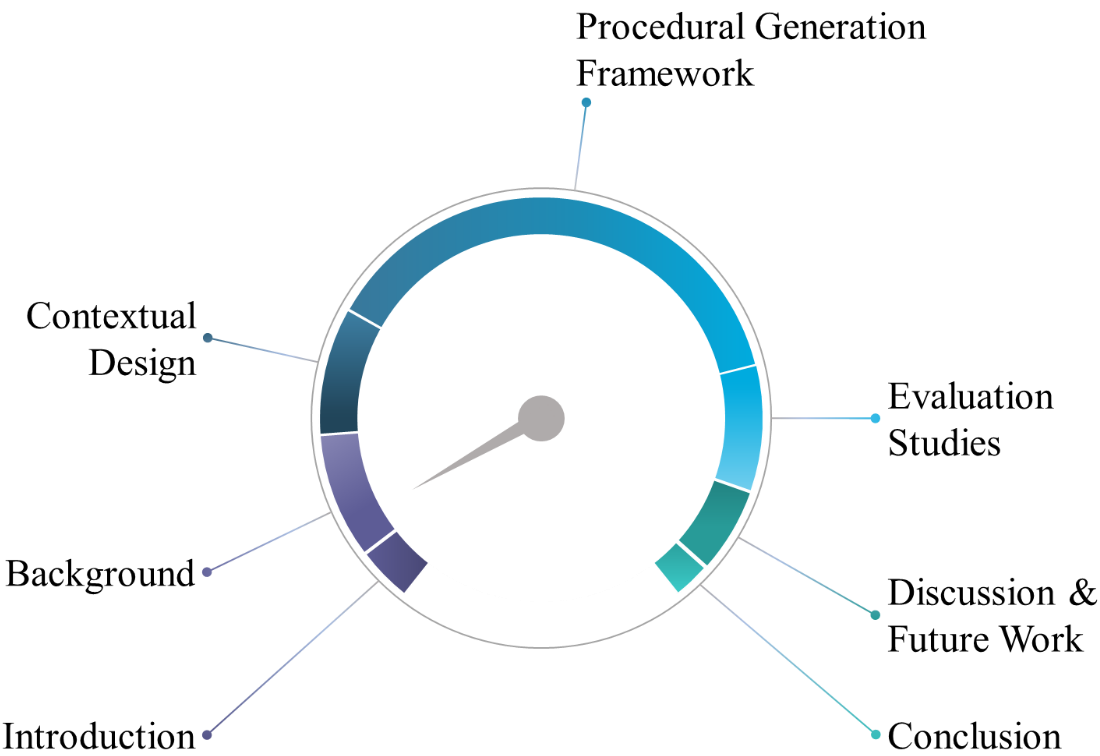


Figure 1: The structure of the thesis.

The first part comprises the *Introduction* and the *Theoretical Background* to offer an orientation within the topic and the different disciplines involved in this thesis. The *Theoretical Background* provides information about the *Aesthetic Automotive Design* and the current *prototyping methods* used to evaluate it. Furthermore, the *designer's perspective* is analyzed and differentiated from that of the user, highlighting the challenges of the designer's task. After a journey through the user's senses, the *sensory modalities* they can perceive through the features of the product design, as well as the influence of each sense and the corresponding sensory modality on the resulting experience, research on the *emotional aspect of user experience* is summarized. Moreover, a number of experience models as observed by *different disciplines* that are relevant to the desired type of prototyping experience are presented, each with its own unique way of describing and unfolding its layers. The *Theoretical Background* ends with the presentation of representative *state-of-the-art simulations* that have been used for automotive evaluation, as well as *procedural algorithms and tools* designed for the generation of street, building and vegetation geometries.

The second part of this thesis deals with the methodology that was applied to develop a new process of creating MR experiences that can be used as prototypes for the aesthetic automotive design evaluations and presentations. This part follows the structure of *Contextual Design*, starting with the studies conducted with designers and developers to gain a deeper understanding of their daily lives, work models and challenges. The results of these studies are presented and used as *requirements* for the *design and implementation of a new framework* that procedurally generates virtual environments based on a selected world map and its data. This framework consists of several procedural generators, each responsible for the generation of a different element in the environment. The procedural generators are described with their respective *flowchart and implementation challenges*, including the *required input data and its provided variables* which in combination with the variables calculated during the procedural framework, facilitate the generation of its output. The second part also includes a chapter on the experiments conducted to evaluate the suitability of the generated virtual environments as context for prototyping design experiences, as well as the usability and acceptance of the complete framework by designers as a new process of developing these experiences.

Finally, the third part of the thesis closes the circle by *comparing the results of the experiments* with the collected requirements. In addition, *possible optimizations* and the *addition of further environmental conditions* are examined. The thesis ends with a *conclusion* that summarizes the most important parts and contributions.

1.4 Prior Publications

Despoina Salpisti, Matthias de Clerk, Sebastian Hinz, Frank Henkies and Gudrun Klinker. 2022. Procedural Street Generation to Support Automotive Designers in the Creation of Context for Mixed Reality Design Experiences. In Proceedings of the European Transport Conference 2022, Milan, Italy.

Despoina Salpisti, Matthias de Clerk, Sebastian Hinz, Frank Henkies and Gudrun Klinker. 2022. Requirements for 3D Environments as the Context for Mixed Reality Automotive Design Experiences. In Proceedings of the Driving Simulation Conference 2022, Strasbourg, France.

Despoina Salpisti, Matthias de Clerk, Sebastian Hinz, Frank Henkies and Gudrun Klinker. 2022. A Procedural Building Generator based on Real-World Data Enabling Designers to Create Context for XR Automotive Design Experiences. In Proceedings of the EuroXR Conference 2022, Stuttgart, Germany.

2 Theoretical Background of Automotive Design Experience

2.1 The Design Process

The development of a vehicle is a complicated process, as shown by the large number of published models describing its development [9-12]. Raabe's model seems to encapsulate all stages of development [11] and distinguishes three phases: “Early”, “Concept” and “Serial Production”.

In the *Early Phase*, several designs are conceived and presented that meet the users' needs for innovation, usability and functionality while conforming to the brand’s values and the company’s resources [13]. This phase includes the designers’ task of creating mood boards and design stories, which are used as the basis for the design concept. Various inspiration sources are collected that are related to the design experience, e.g., architecture and fashion. In the end, the designers are sufficiently inspired and correspondingly ready to depict their design concepts on paper. This process results in the first sketches, which are subsequently refined and rendered.

The *Concept Phase* is characterized by the completion of the final vehicle concept after preparing and integrating all technical components and checking their convergence with the design elements. The designers’ tools in this phase include tape-renderings, to highlight the lines of the exterior design. In addition, exterior models are created from clay at a scale of 1:4 to 1:1. Interior models are also built by 3D-printing certain interior elements e.g., the dashboard, and by adding specific electronic components e.g., interfaces.

The *Serial Production* phase begins with the construction of the first physical prototypes with a complete set of functionalities, which allow the prototype to be tested and all technical requirements for series production to be gathered [13]. In this phase, data control models are constructed for both the interior and the exterior of the car. These models serve as references for the *Serial Production*.

Important design decisions are made during the *Early Phase*. The total costs required to bring a vehicle to market can be strongly influenced by these early decisions [14]. The percentage of costs influenced by these decisions is estimated at 80 % [15, 16]. Problem solving at this stage can also increase the costs significantly.

As a natural consequence, the later these problems arise in the development process, the higher the costs required to solve them. Since reducing the cost and time required to bring a vehicle to market is one of the most important goals in the automotive industry [17], problems should be detected and solved as early as possible in the development process. Through this strategy, new information is obtained, that sheds light on unforeseen problems and contributes to an optimized version of the design [18, 19]. In fact, this information is considered to have the greatest impact on improving the design's quality [20].

2.2 Automotive Design Evaluation

The aesthetic part of a vehicle's design plays a very important role in interpreting the meaning of the product to the user. Accordingly, the aesthetic evaluation is crucial for achieving the right user experience and its interpretation. Depending on the part of the vehicle design and the process phase, the evaluation requires a different method and focus. On the one hand, some designers observe the exterior form of the car, its form lines and features (e.g., exterior lights). On the other hand, other designers are interested in the interior elements (e.g., User Interface, middle console, ambient lights), which include the design of both hardware and software. The designers are divided into even more accurate groups depending on the feature they design (e.g., User Interface). However, all designer groups ultimately target the experience offered by this wide variety of design categories as a complete combination of multisensory interactions.

According to a study [21], the most important visual components of the car and the ones most often used for describing its form are features of the exterior, such as headlights, tires and mirrors [21, 22]. On the other hand, the front emblem, the headlights, the radiator grill, the rear lights and the rear bumper give the car an individual character and thus support recognition. However, the element that supports recognition the most has been found to be the three-quarter front view [21]. In addition, the front view of the vehicle bestows the car emotional affordances, e.g., making it appear friendly [21]. Furthermore, how a car is perceived depends mainly on the form principle groups of "Volume and Line", as well as "Plane/Surface", while the most important form principles of a car are balance, directional forces, scale and proportion [21].

Tian et al. [23] developed their own model of automotive design assessment by first dividing the product features into four categories: rear, front, side and top. The structure of their model consisted of three levels: goal, criterion and factor/attribute. All four feature categories are included in the criterion level, while the target level is where the actual evaluation takes place. Finally, the factor/attribute level consists of four elements of automotive style for each of the four feature categories. The whole model of automotive design style evaluation can be seen in the Figure below.

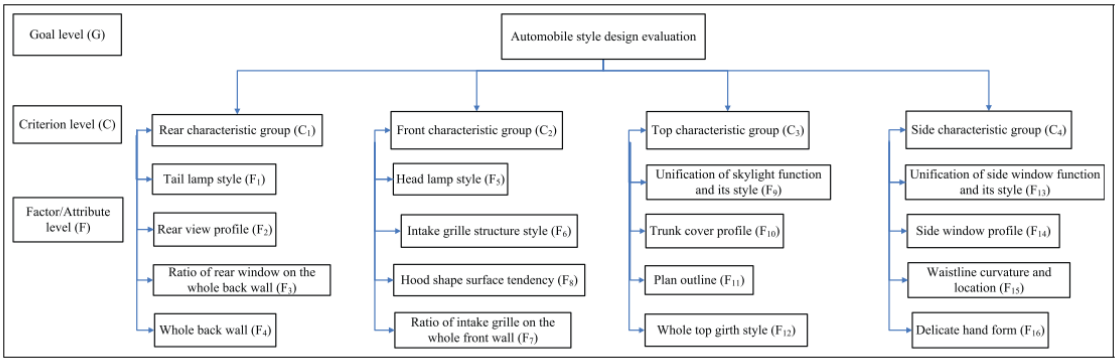


Figure 2: Hierarchy structure of the automobile style evaluation suggested by Tian [23].

Another research team [24] has invented their own method for evaluating automotive design and, in particular, interior interfaces. This is an assessment model based on experience prototyping and includes a customizable physical platform that offers flexible placement of displays and consumer electronics in the driver cabin. Additionally, it offers the possibility to programme the interior interface concepts.

The same type of concept has been adapted by Bordegoni and Caruso [25]. With the help of their Mixed Reality prototype, designers can modify geometries within a vehicle in order to achieve a certain design that can be evaluated in real time.

Focusing on experience, Warell and Young [26] write about a design evaluation process in which the experience offered by the new design is first assessed. Then, automotive experts from the sales department evaluate the new design using a questionnaire created specifically for this purpose.

Loehmann [4] has also experimented with experience prototypes and documented that in order to create believable experiences where users can easily imagine experiencing a certain scenario, prototypes with low fidelity are sufficient. He also notes that usability and design aspects are not the elements that contribute most to the experienceability of a prototype. With his methods for applying low-fidelity and low-resolution prototypes in a real-world environment, he shows that this can and should be done in the early stages of the design process under the fulfilled and crucial requirement that the fidelity and resolution of the prototype meet the participants' expectations of the prototyping experience.

2.3 A New Type of Experience Prototyping in the Automotive Design

Prototyping is about visualizing an idea, testing its concept, and constructing its first version before replicating it. Prototypes in automotive design vary from sketches to Virtual Reality applications. They help designers select one design among others, test the quality of the design and present the design idea to other colleagues from the same or different disciplines. Depending on the objective of the prototype, its virtuality level may vary depending on its ability to realize all necessary features that need to be tested.

As can be inferred from the development process, the design of a vehicle starts from certain experiences that the designer wants to evoke during the interaction between the user and the vehicle. These experiences are called *design stories* or *scenarios*. Each design story has a specific setting with a specific time and place. It also includes one or more characters e.g., the driver and passengers. Moreover, it has a specific goal, such as the use of certain functions in certain circumstances that should evoke certain emotions in the user.

Recently, a new way of design evaluation and presentation was introduced. With the help of this method, designers can experience the design stories for themselves already in the *Early Phase*. This is enabled by game development tools which are used to create Mixed Reality experiences and it is, at its core, a system for creating experience prototypes. This type of prototyping includes the interaction with a prototypical version of the product under the conditions for which it was designed [27]. One of the great advantages of experience prototyping is that designers can slip into the position of the user and let themselves live the experience they want to facilitate through their design [4].

2.4 Two Perspectives: Designer and User

"Different genres of electronic products could enrich and expand our experience of everyday life rather than closing it down. Industrial design's position at the heart of consumer culture (after all, it is fueled by the capitalist system), could be subverted for more socially beneficial ends by enriching our experiences" [28, 29].

To understand what kind of prototyping experience is required to conduct an aesthetic design evaluation, it is important to first acknowledge two different perspectives. The objective of such an experience prototype is to switch from mood boards and PowerPoint presentations during the *Early Phase* of the design process to an experience that allows designers to perceive their end product and adjust their design accordingly. Using this method, designers can test and control the experience their design can provide earlier in the design process, thus increasing the quality of the design. During the *Early Phase*, designers must make important decisions, such as choosing a design that best reflects the company's values and represents the aesthetic of the user group, as well as what experiences should be defined and enabled by the chosen design. Although the design process of a vehicle involves a lot of prototype testing with the end users, this prototyping experience should accommodate the needs of the designers. This is where the distinction between the two perspectives plays an important role in designing the prototyping experience. The designers' perspective during the experience of observing, perceiving, interacting and evaluating the design is different than the perspective of a user.

It is safe to state that designers have many difficult and versatile tasks. After defining their goal, designers form ideas that fit their users and sociocultural background and that can be realized within the existing time and budget constraints [30, 31]. They have to make decisions while taking into account the user's requirements, the brand identity, as well as the degree of novelty. The ability to process information about the conditions of the design process (e.g., time frame and budget) and the user's needs and goals is, therefore, an important tool in the designer's daily life [30, 32]. Before processing information about the user, the designer has to acquire it first. Obtaining such information through interaction with materials, users and other colleagues is, therefore, a skill crucial for the designer, as it can influence the efficiency of their decisions [30, 33, 34]. Bordegoni and Caruso [25], describe this flow of information as "feed-forward" and "feed-back". They also acknowledge the importance of good communication between designers and engineers. These skills are of great importance for designers when it comes to combining aesthetic and technical design and communicating their ideas to the rest of the disciplinary team [35, 36]. Since designing a product is part of a development process, the designers also need management skills that render them flexible in integrating practical considerations into design ideas [30, 37].

Buchenau and Suri [27] tested a method that makes it easier for designers to collect information about users by letting them watch the users experience prototypes of the designed product. The results of this study confirm how helpful this prototyping method is for designers involving the collection of valuable information through an active way of discovery.

Studies [30, 38] with designers observe an influence on the designer's decision through changes in the social environment and experiences with new interactions. These are emotional changes, which have great potential to lead the designer towards more intuitive and emotional design methods. Ho [30] concludes that the best approach is for the designer to aim to create a rich experience for the user while communicating their own principles. This helps designers create a relationship with users [30].

The role of the designer is characterized by considerable complexity. On the one hand [29], designers need to challenge and engage users. They are the creators of products which convey certain ideas and evoke specific emotions. If users want to explore the message of the product, they should let themselves be challenged by it. In this approach, designers only consider their own visions and principles and require users to be open to new experiences. Such an autonomous role was also mentioned by Tan [30, 39]. He described designers as storytellers, because they use their design as a vessel for communicating ideas, which are meant to be comprehended through emotions.

On the other hand, designers should consider the users for whom they are designing. Forlizzi and Battarbee [35] believe that designers should identify the needs, expectations, use cases and interpretations of the audience they are designing for. Studies [30, 35, 40, 41] have confirmed the success of involving the user in the design process by investigating the impact of emotions and experiences of users on design decisions. Conversations with users have been used as means to link activities with products [35, 42]. Moreover, ethnographic research has been conducted to gain information about a group of people based on their use of products and services and their interactions inside their environment [35, 43]. The term “ecology” has been used in this context to define the sum of information that describes a particular group of people based on cultural patterns [35, 44].

Forlizzi and Battarbee’s holistic approach [35] dictates that the designer should work with others in a multidisciplinary team, so that they can experience their design through many different lenses and be inspired through all these different perspectives. This concept, which has also been mentioned by many other researchers, has been called design empathy [35, 45-47].

Norman and Ortony [2] made an attempt to link the emotions intended by the designers with the emotions evoked in the users, and for this purpose, divided the product aspects into utility and appearance. The result of this experiment included the two categories of emotional responses from users with the criterion of how the emotion was generated. After interacting with a product, the user may feel certain emotions that were not necessarily intended by the designer. This is what Norman and Ortony call [2] “emotion by accident”. The designer’s attempt to consider these kinds of emotions during the design process and prevent them from arising is called “emotion-prevention”. There is also the reverse case, where the user gets to experience emotions that were intended by the designer and are, therefore, called “emotions by design”. Similarly, the process used by the designers in order to achieve this type of intended emotions is called “emotion-promotion”. Finally, there is also the case of “emotion-indifference” which accordingly represents a design process in which no emotions, accidental or intended, are considered. When a designer focuses more on utility than appearance, emotions by accident are more likely to happen whereas a designer’s focus on appearance leads to a higher likelihood of emotions by design.

Norman describes his model of design aspects (visceral, behavioral and reflective design) by explaining that a good designer should take into account the emotions triggered by all three different design aspects. He adds that the goal is not always to evoke positive emotions in the user. In reality, many products happen to elicit a combination of emotions by design and emotions by accident and it is a designer's job to anticipate the conflicts created by these two types of emotions and find solutions to them during the design process [48].

When investigating the role of the automotive designer, Liem et al. [21] found that some designers base their decisions on their intuition. Warell gives the reason for this [49] by describing the major difficulties encountered by the automotive designer. These include the difficulty in specifying and describing the experiential aspects of the design [49, 50], which leads designers to create scenarios and use mood boards. Another difficulty that affects the outcome is that the designer's plan at the beginning of the design process is not always fully realized until the final product [26]. Finally, it is extremely difficult to predict the way users will perceive the product in their experience, as much as designers may want to [26]. Despite all these difficulties, it is particularly important that designers try to process the principles and values of a culture and include them in the form of a symbolic meaning conveyed by their design.

Warell [49] justifies the solution that designers use to combat these difficulties, namely visualizing their design, and he highlights the technological gap that exists in achieving product experiences already during the design process. The gap widens even more when considering the great range of multi-faceted experiences a car can offer, which should be filled with a combination of the most positive emotions, such as enjoyment, pleasure, comfort and meaningfulness. The aspects of automotive design are diverse and range from aesthetics to self-identification, from performance to reliability and from service to brand recognition. In addition, Warell [49] explains the need to design holistically appealing vehicles by focusing on the product experience and trying to understand the parameters that influence it.

Schifferstein and Desmet [51] have written about the ideal designer tools in order to design coherent and rich experiences. They highlight the importance of having a palette of basic blocks that can be considered as design elements that are able to evoke basic sensory experiences, as well as being able to choose and combine these elements into a design that can appeal equally to all user's senses, thus creating a holistic experience. Now, imagine an automotive designer's space filled with experience blocks which the designer can use to create, communicate, present and evaluate their new and untested design concept in a context that fits the design stories, from which the concept was inspired. Using such tools, automotive designers can discover potential problems or opportunities to optimize their design by experiencing for themselves the potential interactions with the design and the emotions resulting from that experience. This is the purpose of experience prototyping [27].

Norman and Ortony [2] can help us look at the user's side. While interacting with a product, a user receives emotions, some of which may be the result of utilitarian product features. Again, this is the case of emotion by accident and these emotions are usually negative. They are the result of poor functionality or malfunction of the products. However, it is not impossible that positive feelings also arise by accident, which are related to the utilitarian aspects of the product features. In addition, users have certain expectations regarding the way a product should function. This parameter may lead to negative or positive emotions by accident depending on whether the product meets the user's expectations or not.

When it comes to the point that Norman and Ortony [2] describe as the user's side of emotions by design, they use the example of a gift. When someone wants to buy a gift, they choose a product that does not necessarily have new and authentic features, but as a whole can trigger warm and positive feelings in the recipient and show how loved and appreciated this person is. In other words, a product does not have to be something extraordinary and special for the user to feel positive emotions, which leads to the conclusion that emotion-promotion can also be achieved with normal means and a special overall product quality.

The designer's side is about understanding the user and creating certain emotions for them. The user's side is about receiving and interacting. Users interact with a product and receive affective reactions which may or may not be the ones intended by the designer. These responses depend on many personal parameters, such as memories, expectations, needs and cultural values, resulting in a subjective product experience and making it difficult to predict the way a user will perceive a product. However, hope exists in the fact that some responses are easier to predict than others. Designers have a better chance of controlling emotions through the visceral and behavioral aspects of design than through the reflective ones even if this control is indirect. In conclusion, the designer has the ability to control emotional responses to some degree, by enabling a certain interaction and consequently a certain type of experience, but the actual end result is absolutely dependent on the user.

For this reason, many authors suggest involving users in the design process. In this way, users can communicate their preferences and perspectives, their values and needs, which will eventually allow designers to understand the user's way of perceiving and consequently collect information that can help them foresee the user's experience with their product [27]. Bordegoni et al. [52] also suggest involving users in the design process, with designers and users working together in a loop series of feedback and design integration which has to happen on the early stages of design assessment.

2.5 How Does a User Perceive a Product?

"To the Greeks, experience was the outcome of accumulation of practical acts, sufferings and perception gradually built up into …skill…There was nothing merely personal or subjective about it." [53, 54].

“Gibson (1979) formulates this notion as: ‘Perceiving is an achievement of the individual, not an appearance in the theatre of his consciousness’.” [55, 56].

At the center of every experience stands the person who has it. That person is the one who interprets the experience, gives it a meaning and narrates it to others. No one is a better expert on their own experience than the person who had it. Even if two people were in the same room where certain events took place, they would both come out with a different story to tell.

This space where certain events take place can be anything. It is the environment in which the experience is born, continues, and ends. It could be a room, a house, a car, physical or virtual. Even the description of this environment takes on its own special subjective form in the mind of the person having the experience in it. The factors that lead to this subjective shaping of the environment and events of the experience vary significantly and most of them have their roots in the characteristics of the person interpreting the experience, however conscious that interpretation may be. The object in our type of experience is a product, that is also integrated into the environment in one way or another and whose description by the user is also influenced by the same factors. Therefore, the perceptual skills used to experience the product under certain environmental conditions are very valuable information in the study of prototyping experiences. When more than one sense is involved, the designed stimuli should result in a harmonious and coherent experience. This is a great challenge for designers as we will see later in this Chapter.

Overbeeke et al. [36] have divided the user’s product perception skills into three categories: cognitive, perceptual-motor and emotional. Through this categorization, we can observe the user’s active involvement not only through their activities (perceptual-motor skills) but also through their thinking and feeling, thus incorporating many of the subjective factors mentioned above into the experience and making it a personal one. The user’s range of human perceptors is used every time a person encounters a new product. They interact with it by testing its functions, interacting in certain ways and using their senses. This same method is used to explore not only products, but also our entire environment. This information is, then, interpreted through cognitive processes into a specific type of product, while at the same time trying to revive memories of similar ones. After perceiving the product, recognizing its functions and creating memories of it, the user will have certain expectations when encountering a similar product. These expectations must be fulfilled for a good evaluation. In addition, Hekkert and Schifferstein [57] explain that these human skills, which are trained during interaction with products, can be developed into expert knowledge about the environment, as well as skills that lead to a more efficient and safe interaction. We immediately notice the impact that the environment has on these skills.

The user's product perception skills are linked to product characteristics. Hekkert and Schifferstein [57] highlight the structural/formal, the material and the compositional properties, as well as features of embedded technologies and labels them as different types of product characteristics. On the other hand, the properties of the product have also been divided into the following categories: content, presentational style, functionality, interactional style [58-60]. Hassenzahl [60] has also commented on the topic by investigating which product attributes lead to certain types of emotions. He distinguishes between pragmatic and hedonic attributes, which lead to different interaction types and, consequently, to a different set of required user skills for the interaction.

2.5.1 The Visual Experience

It is not always possible to perceive a product without illusions. Sometimes the environment influences the perceived information leading the user to false judgements. In their effort to achieve the intended experience, designers should be aware of all effects that can be caused by a number of factors (e.g., environment, combination of stimuli) and influence their users' experience. Objects of visual perception include the shape, material and illumination of the product.

The perception of shape is influenced by the direction of illumination. A change in the tilt and slant of the light source will lead to a different perceived curvature of the product [56, 61]. In addition, contrast and thus the level of ambient light, affects the perception of depth [56, 62, 63]. Further studies have been conducted leading to many conflicting opinions about the influence of specular highlights on the curvature of an object [56, 64-67]. Another change in the perception of shape worth mentioning can be caused by the effect of chromostereopsis, especially when a red object is viewed in front of a blue background [56]. Hue has no effect on the perceived shape [56, 68].

The perceived material also changes depending on certain parameters. One of these parameters is the object's luminance and color contrast with the environment. If the relationship between the luminance and color of the object and the environment stays the same, then the user perceives the same albedo even if the light intensity changes [56]. These two effects are called luminance and color constancy. Another parameter is the illumination of the environment, which affects the perception of the material's albedo. Studies have shown that the albedo of a surface is most accurately perceived in ambient illumination [56, 69].

As humans visually perceive the world around them, they process the visual information to reconstruct it in their memory. Gestalt Theory includes the organization principles used in this reconstructing process [56, 70]. The most important principle of Gestalt Theory is the Law of Prägnanz, according to which when we see a shape, we try to put it into the simplest and most regular shape possible that is simultaneously similar to what we see. As important as the laws of Gestalt Theory are for the functioning of the visual system, they can also lead to altered perceptions through illusions, e.g., the face/vase illusion.

Research has also shown that stereoscopic vision is good for three-dimensional perception, that three-dimensional textures are good for high accuracy in determining the source of light and that an occluding contour is necessary for recognizing shape from shading [56].

2.5.2 The Tactual Experience

The role of vision in the product experience is undoubtedly of great significance. Nevertheless, the tactual aspect of the experience is necessary to explore the product and its functions. Touching objects allows the quick and accurate recognition of three-dimensional objects and intensifies familiarity. Touch is also crucial for creating affection and intimacy [71-73].

A tactual experience cannot exist without physical interaction. This interaction includes touching and being touched, using both hands and the whole body. In this way users can physically engage with the environment and become physically aware of themselves and others [73]. Sonneveld and Schifferstein [73] add that seeing one’s own body is not sufficient to achieve self-awareness in a product experience, unless it is combined with the ability to touch it and feel touched.

The parameters that influence tactual perception lead to the categorization of tactual experiences. One of them is the type of touch, active or passive, which distinguishes between touching a product and being touched by it [73, 74]. Naturally, both types of touch occur at the same time. However, during an active touch the user focuses on the object whereas in passive touch their attention is drawn to the sensations in their body. In a third case, the user is made aware of both attention targets. Different body parts vary in their suitability to experience active or passive touch.

Sonneveld and Schifferstein [73, 75] also mention the case of touching objects through other objects. This is another category of tactual experience, dividing it into direct and indirect touch. For example, when a person rides a car, they touch the road through the car tires.

Other parameters that influence the tactual experience are related to the product itself. Sonneveld and Schifferstein [73] mention “transparency”, which refers to the object’s ability to allow users to “feel through the object, to incorporate it, and to direct their attention to something else in their environment”. Transparency can be reduced by “tactual noise”, which refers to the annoying sensations created through tactual interaction.

Movement plays a very important role in the perception of tactual information [73]. Products have certain tactual properties, each of which has its own suitable movement with which to be perceived. Research has shown the need for a systematic tactual scanning in order to achieve a consistent tactual perception. The term “dynamic touch” refers to the exploratory process of swinging an object in order to get an initial understanding of how to use it.

These so-called tactual properties of the product are related to the material, the surface with a certain texture and patterns, the geometrical properties and the moving parts [73]. Each of these parameters can be explored through a specific type of tactual interaction. The material properties of an object, such as hardness and softness, can be perceived by exerting pressure [73, 76], while stiffness and flexibility can be detected by bending and wrenching [73, 77]. Releasing the pressure and observing the reaction of the material provides information about the object’s elasticity and springiness.

The temperature of an object can be perceived quickly by touch but only if it is an extreme temperature [73]. If this is not the case, a person can only perceive the difference in temperature between their body and the object after touching it for a long time. Temperature flow is an effect that helps us recognize cold and warm objects. Touching a colder object causes warmth to be transferred from the body to the object. In addition to the temperature difference, the material of the object must have a low temperature resistance, such as glass and metal. Otherwise, materials whose temperature is lower than the body temperature still give a warm sensation when touched. After a certain time of touching the object, the temperature of the object is perceived as neutral.

Exploring the texture of an object requires stroking its surface, especially for fine textures [73, 78]. Another alternative, suitable for coarser patterns, would be static touch [73, 79]. There are several categorizations of the dimensions, in which textures can be perceived. According to Hollins et al. [73, 80], these dimensions are: rough/smooth, soft/hard, sticky/slippery and bumpy/flat, which depends on the other three dimensions. On the other hand, Picard et al. [73, 81] observed four different dimensions: Soft/harsh, thin/thick, relief/no relief, and hard/mellow.

In order to explore the geometrical properties of an object, the user needs to grasp it, hold it, manipulate it and follow the contours with their fingers. In the case of larger objects, dynamic touch is required, whereas in the case of curved surfaces, the user needs to consider the direction of their scanning movement. Research on this subject distinguishes [73, 82] three categories of shape: abrupt surface discontinuities, e.g., edges, continuous three-dimensional surface contours, e.g., curved vs. flat and orientation of surfaces, e.g., horizontal.

Similar to visual perception, there are parameters that can lead to incorrect tactual perception. Previous tactual experiences can influence future ones. For example, having had a prolonged tactual interaction with a concave surface leads the user to perceive a flat surface as convex [73, 83]. If touching the object precedes seeing it, the object appears smaller than expected.

Weight perception is influenced by previous weight perceptions ("After prolonged holding of two objects of different weight in each hand, the weight of two objects of the same weight is estimated as different" [73, 84]) and temperature. Other influencing parameters include the surface texture of the object ("The smoother the texture, the heavier the object is perceived." [73, 85]).

Being touched by an object causes the user to feel certain sensations that depend on the type of interaction and movement. There are three different types of skin sensations depending on the sensors involved [73, 86, 87]: touch sensations, e.g., superficial and deep pressure, warm and cold sensations, pain sensations.

Depending on the location, quality, intensity and duration, skin sensations can be classified into the following categories [73, 88]:

- Light touch, referring to being touched without simultaneous skin deformation,
- Pressure, referring to maintained touch with skin deformation,
- Vibration, referring to a rapid adaptation of the skin by rhythmically stimulated sensors,
- Cold and warmth,
- Pain,
- Itch and tickle and
- Physical pleasure.

The ability to detect being touched, the area of touched skin, the duration and the intensity of touch depend on the tactual sensitivity, which varies depending on the part of the body [73]. It also depends on the spatial aspects of the afferent neurons and the size of the reception areas in the somatosensory cortex [73]. In addition, skin sensitivity can be trained, and it decreases with age [73].

Users often attach a certain personality to a product by describing it with human characteristics [73]. They also tend to express the tactual properties of a product directly as traits of its personality (e.g., cold → cold personality). Similarly, a product can be assumed to have its own intentions [73]. This effect is caused by the behavior a product seems to show during a physical interaction with the user. A product may appear to respond through its own movement to the movement of the user or reject instead of not responding when it does not move [73, 89, 90]. Users interpret this product behavior in terms of the following intentions [73]: Wanting to be touched and explored, to cooperate, to play and to take care of or love or hurt somebody.

Products may sometimes give the impression of behaving affectively towards the user [73], to which the user may respond with equivalent feelings. These emotions can be divided into physical pleasure (e.g., “lust, pain and disgust”), affection (e.g., “love and hate”), vulnerability (e.g., “trust and fear of getting hurt”), energy (e.g., “tension and relaxation”), action tendency (e.g., “approach and avoidance”) and the tactual characteristics reflected in self-experience (e.g., cold sensation → cold personality).

A product can give two types of information: information describing itself and its functions and information about the physical environment in which it exists [73]. A designer should pay closer attention to what tactual feedback their product should give because this information may help the user achieve their goal, but it also has the potential of misleading them [73]. The help that this information can offer is perceived by the user as its integrity [73]. In addition, depending on the user’s expertise the tactual feedback will be experienced and interpreted into honest or not honest information [73].

Familiarity is another important subject related to tactual experience. Becoming attached to an object is a result of frequent touching between user and product [73]. Familiarity and the feeling of ownership come together, and they become more intense over time, as people accumulate memories with the object [73]. Furthermore, touching objects helps people recognize them in an affective way [73]. Feeling familiarity with an object and recognizing it become easier after the object has received some imperfections over time [73].

Tactual interaction is not the same for every product. Learning how to interact with a product might be challenging for the user [73]. A user needs to develop “tactual knowledge” in order to be able to interact with certain products [73]. While acquiring new skills for interacting with some products, the user may find a specific space in which to express their own style of interaction [73]. On the other hand, there are also products that do not allow such freedom [73]. A person may feel overwhelmed by an object until they have learnt the skills necessary to interact with it, attaching a corresponding type of personality and intention to the object [73].

2.5.3 The Auditory Experience

Products also provide auditory feedback that consumers use in functional or experiential ways [91]. However, the type of use is not always conscious to the user. Furthermore, similar to how people use visual feedback in order to determine how an object feels, they also use auditory feedback to determine the shape or other physical properties of objects [91-98].

In our daily lives, we are surrounded by a wide variety of sounds, depending on the location and the culture [91]. Some of these sounds belong to the group of industrial sounds, which are generated by industrial products, cars and user interfaces [91]. Industrial sounds include consequential and intentional sounds, the former coming from mechanical moving parts [91, 99] and the latter being carefully planned sounds that are intentionally added to a product [91].

Van Egmond [91] created a framework for auditory perception that takes into account the aspects and parameters that influence it. Sound passes through the outer and middle ear in order to reach the inner ear, where it is processed [91]. In addition, sound can be measured by the following parameters: sharpness, harmonicity, loudness, roughness, and tonalness [91, 100-104]. An increase in the value of loudness, sharpness or roughness leads to a decrease in sensory pleasantness [91]. On the other hand, an increase in tonalness results in an increase in sensory pleasantness [91]. Another parameter that influences auditory perception is memory. The user processes sounds according to their memories and the meaning they attach to them [91].

A sound can trigger a certain emotion, which is not always an evaluation of that sound [91]. Van Egmond [91] distinguishes between basic emotions [91, 105] as the result of a simple appraisal and cognitive emotions [91, 106] as a result of a more complex appraisal. A designer should know what kind of emotions they want to evoke through their sounds in order to know which auditory features they should modulate and in what way [91]. Another difference between basic and cognitive emotions is that cognitive emotions take longer to process because more memory information needs to be retrieved. According to one study [91, 107], the time it takes a person to evaluate a sound in certain emotions is a clue to whether this emotion is basic or cognitive.

2.5.4 The Olfactory Experience

As mentioned earlier, smell and taste do not play such an important role in product experience, except in some specific cases, where they play the dominant role: food, beverages, perfumes, healthcare products and hygiene products [108]. Since automotive designers stimulate the sense of smell during the automotive experience through scent dispensers inside the car, it is necessary to study the olfactory experience and the factors that influence it.

We taste as a result of the stimulation of chemoreceptors on the tongue and other areas of the oral cavity [108]. The perceived taste is described using the following terms: bitter, sour, sweet, salty and umami [108]. A taste experience can be measured by these qualities, as well as the intensity of taste and its affective dimension [108].

We smell as a result of the stimulation of receptors in the nose [108]. Smell also has its own vocabulary to describe the experience (e.g., “grassy”) [108]. Olfactory information warns us of or motivates us to take a certain action, and it can influence communication [108].

Research shows [108, 109] that the accuracy of memories evoked by all senses is relatively the same. However, the perception of olfactory information can lead to more intense emotional reactions. Moreover, after a certain period of exposure, the user no longer perceives the odor with the same intensity and consequently cannot recognize it as easily [108, 110, 111]. As expected, intense odors can maintain their intensity longer than weak odors [108, 112]. The intensity of an odor perceived by the user can be influenced by the cognitive and emotional reactions to it [108, 113]. Preferences for odors vary according to age, culture and context [108, 114-118].

2.5.5 Multisensory Product Experience

"It is only through experience that the concepts become embodied knowledge, and thus recognizable in experience." [73].

In the last chapter, we saw how users can perceive each type of sensory information, which parameters influence this perception and which senses can inform the user about the same product properties but not always with the same accuracy. Schifferstein and Spence [119] refer to this last effect as overlap of sensory modalities ("For example, people can both see and feel the shape of an object, its size, and the roughness of its surfaces" [119, 120]). This sensory overlap leads people to have certain expectations about the information collected with certain senses after collecting information with another sense [119, 121, 122].

Sensory overlap can occur when a pair of sensory stimuli is more likely to occur together due to previous experiences [119]. Also influenced by previous experience is the fact that users perceive certain similarities between colors, sounds, tastes, smells and other sensory stimuli [119]. This is because certain dimensions of sensory experience are common to many, if not all sensory modalities, e.g., intensity, spatial location and duration [119]. Schifferstein and Spence [119] highlight that this phenomenon should not be confused with "synesthesia", since synesthetes experience a "certain and reliable" stimulus of a certain sense after perceiving another sensory modality [119, 123]. They also emphasize that designers should pay attention to these associations between different sensory modalities and use them in their design by investigating the tendencies of these associations [119]. Although this cross-modal effect usually goes unnoticed by the user, it often influences the sensory experience [119].

Schifferstein and Spence [119] suggest providing a large number of sensory modalities to create a richer experience [119, 124-126]. This also applies to virtual environments, where offering a variety of stimuli can lead to a higher level of presence. Designers could use this information and try to enrich the user's experience by stimulating the perception of different sensory modalities, while keeping in mind that the way the user perceives the product through their senses may not always be the way the designer planned. This is one of the many reasons in favor of a virtual simulation where the designer can experience their own product as they create it and decide which sensory channel is best suited to convey a certain message [119, 127]. Furthermore, the designer could use this opportunity to check whether the messages conveyed result in an overall coherence [119, 128]. According to Schifferstein and Desmet [51], designers can achieve a coherent and rich multisensory experience using four different tools:

	Designer competence	*Material expertise*
Intuitive	Sensory sensitizing: Training to become sensitive to impressions products can evoke through all the senses	Sensory sampling: Building a collection of sensory qualities
Structured	Sensory communication: Using terminologies that describe sensory characteristics	Sensory building blocks: Using systems that describe the structural properties of sensory information

Figure 3: Designer tools for a multisensory design process [51].

The potential to achieve an ideal experience for the user with the help of these different tools depends on the designers themselves. However, this design space can be enhanced in order to support the application of these tools in the design of the product experience already during the design development process. This support could consist of, e.g., a tool for designing multisensory experiences developed for designers.

Schifferstein and Desmet [51] also mention the requirements for multisensory design. These include knowing how different combinations of sensory stimuli are perceived and how changing certain design features influences the sensory stimuli and consequently the messages they convey, as well as discussing the effects of these changes with the rest of the design team. All these requirements could also be supported by a design space where designer teams can meet, experience new designs, make changes, and see the consequences of these changes directly during a design experience.

This design space should also contain sources of inspiration, ranging from materials and sounds to smells and images. They should be kept there for designers to use during the development process [51]. In addition, designers should have basic building blocks as tools to create a multisensory experience, e.g., parameters like hue, saturation and brightness to create colors and sound waves to synthesize a sound [51]. As the communication of ideas plays a crucial role in the design process, this design space should have a common language, a language that every team member understands regardless of their expertise or background. This could be achieved by sharing the same sensory experience.

Being part of the experience in this design space could mean a lot to the designer. It could inspire them to experience the sensory modalities they are designing, as well as help them recognize which sensory modalities are less stimulated than others, so that they try to create a richer and novel experience by designing new functionalities that address those particular modalities [51, 119].

On the one hand, having the right tools to create the experience they want to facilitate through their design can help designers create a successful product. On the other hand, the context of use of the product also plays an important role in the cognitive process of product perception. Therefore, the "situational, cognitive, social and cultural" background of the user, on the basis of which they will judge the product, must be taken into account [108]. The user perceives the product through their senses and is affected by the mental images they have created for their senses even before they use the product [119].

During a multisensory experience, the user may need to shift their attention from one sense to another. Studies [119] show that shifting attention from or to the sense of touch takes more time than shifting attention from or to vision and audition. Furthermore, when perceiving sensory inputs, the user tries to organize them based on specific criteria [119]. The most important of these is the perceived location and time [119].

Similar to how humans explore their environment with their senses, they do the same when it comes to exploring a product. Every product experience is multisensory [129]. Fenko et al. [129] have highlighted the importance of knowing which sense is dominant in the experience of a particular product in order to achieve a more financially efficient design. The study results of cognitive psychology are perhaps not so surprising. While experiencing a product, consumers mainly use their sight. The effect of visual dominance is sometimes so strong that in a number of experiments the information gained through vision [129, 130] influenced the consumers' sense of touch while the conflict between the visual and tactual information went unnoticed. Other studies [129] have also shown dominance of visual over tactual information. This effect was observed in users who performed specific tasks, such as determining size [129, 131], length [129, 132], curvature [129, 133], depth [129, 134] and spatial location [129, 135] of the product. In cases where there was a conflict between visual and auditory information, the sensory information captured by vision was again dominant. Visual dominance was also found when users tried to determine the position of auditory signals [129, 136].

Fenko et al. [129] have created a hierarchy of senses based on the results of their experiments. From the most useful to the least useful sense for describing a product, this hierarchy included the following senses: sight, touch, audition and olfaction. In addition, they report that experiencing a product by sight and touch makes it most easily identifiable and evokes the "clearest memories".

Schifferstein and Desmet [129, 137] studied the effects of the different sensory information collected when interacting with everyday products by blocking one sense and observing the results. When the blocked sense was vision, a significant increase in task difficulty was noticed. However, the study also concluded that users' experiences became more intense as was the use of other senses. Blocking the sense of touch led also to a significant increase in task difficulty, especially when the task required "subtle coordinated movements". Another consequence of preventing users from touching the products was that users felt emotionally alienated from the products. After blocking audition, communication problems occurred, while blocking olfaction resulted in a decreased appetite for foods. Schifferstein and Desmet [129, 137] assumed from that last result that blocking audition and olfaction deteriorates the emotional experience of the user.

The same group of researchers investigated the relationship between sensory dominance in product experience and time. The following statements were some of their research findings and conclusions [129]:

- Through vision, the user perceives the greatest amount of information in the shortest amount of time. According to Schifferstein and Spence [119], this could also be because this information is available to the user much earlier than information obtained through other senses and because visual information can be processed much faster.
- The particularly important role that vision plays in the perception of a product can be limited mainly to the functional interaction between user and product and to the conscious experience of this interaction. The same applies, albeit in a lesser extent, to the sense of touch.
- Audition and olfaction still play an important role and have a significant influence on the emotional perception of product experiences.

Nevertheless, the importance of the role of sensory modalities depends on the nature of the product. Accordingly, other experiments conducted with different product groups showed lower results for vision than for other senses [129]. Schifferstein [129, 138] defined the three most likely parameters for determining the role of senses in the perception of products, which are the product itself, the frequency of its use and the importance attached to the interaction with it. That last parameter suggests that the role of senses also depends on the personal values of the user. Many researchers have shown that the more ambiguous the interpretation of a sensory information, the less importance is attributed to it when forming a judgment about a multisensory product experience [119].

Furthermore, a correlation between the role of the senses and the stage of ownership has been found [119]. According to this, the role of vision is most important during purchase, while thereafter the role of the other senses increases greatly. During ownership, the appearance of the product may deteriorate due to use and changes in fashion trends, leading to a decrease in the importance of vision [119, 139]. During the buying process, designers compete for the user's attention through the sensory modalities their products can offer. Since vision is the sense that collects the most information in the shortest time, the product that attracts the user's attention has a great advantage. After purchasing the product, the user's admiration of the product's appearance starts to fade. At this stage, the appearance is no longer as important for the user as the use and operation of the product, which leads to other modalities playing a greater role. At a certain point, the user can become attached to the product. This is when the feeling of ownership is strongest. At this stage, the emotional aspect of product experience plays a particularly important role. Audition and olfaction were mentioned as senses that can enhance the emotional aspect of the product experience. It is in this phase that these senses gain particular importance. As we have already seen, touch is a sense that helps the user to see the product as their own. In other words, it reinforces familiarity with a product and a sense of ownership, which are important for the final phase of the relationship between user and product.

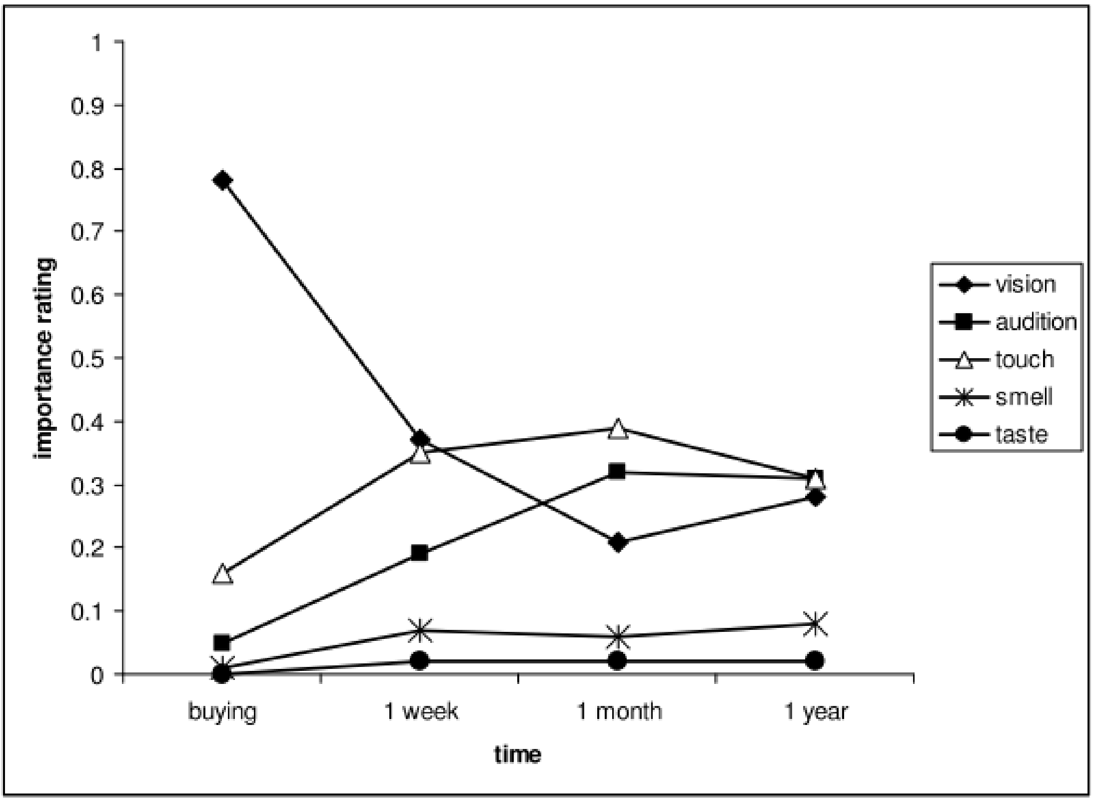


Figure 4: A graph by Fenko et al. [129] showing the importance of each sensory modality over time.

Schifferstein and Spence [119] saw a connection between changes in the importance of the sensory modalities and the interaction. During an interaction between user and product, each sense can dominate, depending on the characteristics of the product. Every physical product has the possibility of being seen or touched, which explains why vision and touch have a greater chance of dominating. Usually, a product only produces sounds when interacting with it, which gives audition not as much chance to dominate. Much less than usual, people are interested in smelling a product, unless the odor of a product is important for that particular case, such as food. When the product belongs to such categories, smell has a much bigger chance to dominate.

Schifferstein and Spence [119] conclude that a designer's ability to interpret the experiential influence of different design variables and modulate them accordingly, makes all the difference.

All the parameters that we now know from the previous chapters and how they can affect the perception of the product and consequently the whole experience have to be taken into account by the designers. It now becomes clear how complicated the evaluation of such experiences is for designers and thus how important it is to have a design space that allows them to experience their own design under the right conditions.

2.6 Emotions in the Design Process

"There is no such thing as a neutral interface. Any design will elicit emotions from users, or convey emotions from its designer, whether or not the designer intends this or is even conscious of it." [29, 140].

"…when two coffee makers basically make the same pot of coffee, we take the one that gives us a pleasant, desirable, or inspired feeling" [29, 141].

Designers strive to create an experience that can elicit certain emotions in their users, which renders emotions a crucial aspect of the target prototyping experience. It is, therefore, important to understand how emotions influence the experience and how designers use them to create better products.

It is undeniable that a user is only satisfied with a product if the product meets all his needs, depending on the use case, environment, occasion, and other factors that define the user's expectations. According to Jordan [142], there are three levels of requirements that the product must fulfill in order to satisfy the user. The product should fulfill the task for which it was designed, trigger emotions through interaction that fit the context of the task, and finally, the feeling of ownership should be associated with positive feelings that correspond to the way the user wants to be seen by others.

This shows the importance of the emotions that arise while interacting with a product. As Desmet and Hekkert [29] wrote, the same statement has been made by many authors, who have added that there is no such thing as an emotionally neutral product. They have also mentioned that design decisions have an impact on the experience that can lead to negative outcomes if not sufficiently taken into account. In this case, the key word for a better design is "engaging", which together with "authentic" and "easy to use" completes the principles that can lead to better results, when considered in the design process [29]. Overbeeke and Hekkert [29, 143] have also shown that the perception of a product as engaging contributes to its perception as easy-to-use. Therefore, the designer should focus on the experience of the user created through the interaction with the product and consider the user as always looking for the emotional aspect in their experience with products.

So, emotions are very important for product design. But how should emotions be considered during the design process? It is a common technique to involve users in the design process by giving them the opportunity to express their feelings and opinions about the new product [29]. In addition, the intentions and goals of the designer should also be considered while they are trying to challenge and inspire the user through their design [29]. Simultaneously, approaching emotional design through research and theory can also shed light on the trends of user preferences and the relationship between product and emotional responses [29].

Hassenzahl [144] also highlights emotions as a consequence of product characteristics by affirming that emotions are at the center of a product or any other type of experience. An emotional experience is not possible without our body and its nature is defined by culture [144]. Emotions help a person to identify themselves, to analyze the behavior of others and their own [144]. Jordan [49, 145] considers emotions as the consequence of the interaction with the product.

In car design, emotions have the same amount of impact. Research on this topic has shown that emotional experiences derived from interaction with car models involve mixed emotions [146]. This is an expected consequence given the complexity of the car as a product due to its various design aspects. Each of these aspects has the potential to trigger a different type of emotion in the user, while the emotion evoked by one aspect can influence the emotion of another.

Warell [49], identifies a specific group of emotions in the product experience that are the result of “visual”, “hedonic” or “aesthetic” design. Related to the last type of emotions, the aesthetic design of the car and the emotions it elicits in the user, are among the top ten purchase criteria for mid-sized sedans in Japan, Germany, and the USA [49]. The appearance of products plays a decisive role not only in automotive design, but in all products. It has a great influence on aesthetic appeal, emotional experience, brand expression and identification [49].

Emotions have been highlighted as an important aspect of product experience in many experience models [35, 48, 142, 147-149]. According to Norman [48], there is almost always a conflict between the different levels of emotion except in the case of a successful design that is able to achieve great results for each level of emotion.

When we interact with products, we are in a safe situation in most cases. Humans have learned to use emotions to analyze their environment and everything in it [149, 150]. Thus, we always look for emotions that confirm the beneficial nature of a product [149, 151]. According to the appraisal theory, a person first evaluates the situation and based on this often unconscious and automatic cognitive process, decides which emotion is most appropriate [149, 152].

2.7 Automotive Design Experience through 5 Perspectives

From the previous chapters, it can be concluded that emotions are at the heart of the product experience, which is highly subjective and depends on a lot of the user's idiosyncratic features. Focusing on the experience of the user during the design process and processing the information that emerges from studying it, can lead to better designs that fit the user and elicit the intended emotions. To increase the quality of the design from the start of the development process, the Early Phase is suitable to conduct presentations and evaluations of the new design among designers. This helps designers to crystallize the final design and the experience it should facilitate. And what better method to evaluate their design than experience it themselves under the same conditions as the user [27].

To understand what kind of experience designers require for their presentations and evaluations, it is important to acquire an overview of the different aspects that influence and form the target experience. For a spherical overview, five different perspectives are relevant: the Automotive, the Design, the Mixed Reality, the Game and the Prototyping Experience.

2.7.1 The Automotive Experience

The automotive experience is as versatile as the product itself. Design stories are crucial to the automotive experience because they describe the content and are thus an indicator of automotive trends.

Aesthetic quality and the experience it can facilitate are particularly important for automotive design [153]. Aesthetics and brand are two of the main reasons for users to buy a vehicle, as they increase the desirability and appeal of the product [49]. Warell [49] described the visual experience between user and vehicle using his VPE framework while highlighting its subjectiveness. According to this experience framework, the automotive aesthetic experience consists of two dimensions, the actual aesthetic experience, which he calls "Presentation" and the assignment of meaning which is called "Representation". Each of these two dimensions is divided into three "modes" which are derived from the sensory, cognitive, and emotional aspects of every design experience. With his model, Warell [49] highlights the unique aspects of the automotive experience, going one level deeper than the general design experience.

One of the modes in the dimension of "Presentation" is called "Impression" and describes the part of the experience in which the user perceives the vehicle and its characteristic elements and thus forms their first impression. The mode of "Appreciation" deals with the user's approval of the visual aesthetic forms and structures and the recognition of the aesthetic principles used in creating the approved design elements. This dimension ends with the mode of "Emotions" which are evoked during the previous mode and are the result of the user's judgement.

The dimension of "Representation" consists of the three modes: "Recognition", "Comprehension" and "Association". In the first mode, the user identifies the vehicle by recognizing the design elements through the comparison of their similarity to design elements they have already been exposed to. In the second mode, the user understands the meaning and importance of certain design characteristics, such as the properties of specific design elements. Finally, the third mode signifies that the user has created meaning by associating the design concepts with a brand based on social and cultural ideas.

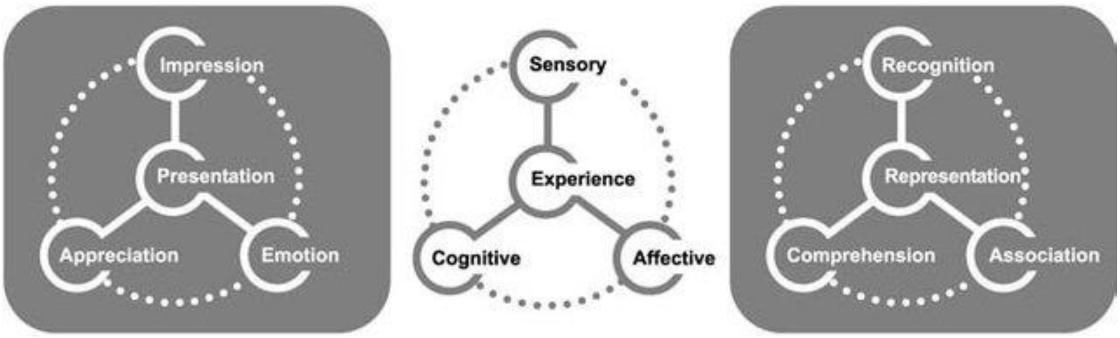


Figure 5: Warell's VPE Framework [49].

The automotive design experience can be influenced by factors related to both the user and the product, as well as external factors [26, 49]. Repeated exposure to a new design is one of them [154]. The aesthetic evaluation of new designs, which occurs only once and determines whether the design remains in the design process or gets eliminated from it, is not sufficient. Multiple exposures to a design that was at first negatively reviewed may lead to aesthetic evaluations that eventually lead to positive results. At first glance, a new design might appear too novel, but this impression wears off as one is more often exposed to it. Coughlan and Mashman [154] highlight the equal importance of "initial pleasingness" and "freshness durability". In conclusion, a design may be rejected after being exposed to it multiple times and the initial positive feelings lose intensity. However, if the opposite effect occurs, then the user's initial enjoyment may last longer during the life span of the product.

Many researchers have already recognized the importance of offering the designers and the rest of their multidisciplinary team, as well as the users, the chance to experience the design, especially the very complicated and multi-faceted design of a vehicle [25].

Kansei Engineering is a popular method that was invented to support designers in general through consumer feelings and tendencies and in the case of automotive design, it was used to structure this intuitive information [21, 155]. Today, many automotive design prototypes begin as sketches and digital renderings, that provide a starting place for discussion and a creative basis for alternative ideas [4]. With each revision, this originally small idea transforms into something more detailed until it eventually becomes very complex. This progress of the design, as well as any changes and improvements should always be communicated to all participants of the design process.

However, this is not an easy task. Automotive designers use stories and storyboards to narrate experiences to other team members. These members could be managers with limited time who do not yet know the details of the project. Other members would be engineers whose language does not include stories and scenarios. Interactive prototyping was applied to solve these problems. However, an interaction and a story behind it are not always sufficient to be understood by all members with various backgrounds [4]. On the other hand, experience is something that everybody understands and perceives for themselves. Of course, this presupposes that the experience prototype is of high maturity [4].

Aesthetics is a very significant part of the automotive design [153]. They are one of the main reasons why a consumer will buy a car. Like aesthetics, brand is also part of a vehicle's appearance and can have the same effect on a consumer's judgement [153]. These facts make the aesthetic evaluation of design particularly important. Nevertheless, designers have to rely on their training and intuition in order to judge the quality of a vehicle's aesthetics.

Aesthetics not only help the vehicle stand out, but also deliver a message of values that are meant to be associated with the brand [153]. The most important sensory modality used for the perception of aesthetics and brand is vision. The assessment of the aesthetic quality and consequently the overall quality of the vehicle depends on the visual stimuli.

There is a conflict behind the relationship between aesthetics and brand. The principles of aesthetics include novelty, while the principles of brand require a sufficient degree of familiarity for the purpose of brand recognition. This conflict, combined with the fact that aesthetics is a subjective matter, should be considered by designers and judges of the automotive design. In this case, designers must rely on their intuition and experience [153].

2.7.2 The Design Experience

Regardless of how complex the design concept of a vehicle is, the experience it facilitates contains the aspects of any product experience. Therefore, it is important to also look at these experience models and understand their content.

Product experience is subjective, as it depends on the characteristics of the user and the product [149]. According to Desmet and Hekkert [149], there are three aspects of product experience. The first aspect is the aesthetic experience, where the user's senses are pleased and captivated by the interaction with the product. The second aspect involves the interpretation of the experience and the decision of what it means for the user. Finally, the user evaluates the product according to the advantages and disadvantages it can bring to their well-being, based on their emotions.

Norman [48] defines the aspects of the design experience by making an analogy with the three different brain modes that are activated during each aspect. In the visceral mode, the user evaluates the design based on its "look and feel". In the behavioral mode, they evaluate the product's usability, performance, and functions. In the reflective mode, they interpret the product and its meaning and reflect on the experience based on their sociocultural characteristics.

Figure 6: A photo of the Mini Cooper S, as it appears in [48] with the caption: ""It is fair to say that almost no new vehicle in recent memory has provoked more smiles." (Courtesy of BMW AG.)" [48].

McCarthy and Wright [148] additionally consider the spatio-temporal character of the design experience and add the sensory, the emotional and finally the compositional aspect. The "sensual thread" of experience refers to the perception of a situation and the environment in which it occurs. The "emotional thread" contains the subjective emotions that make the same experience different for each user. In the "compositional thread", the experience is divided in parts while defining the relationship between each of the parts and the overall experience. Finally, each experience has a "spatio-temporal aspect" that may differ in interpretation from the real time and space, in which the experience actually takes place.

According to Hassenzahl [60, 144], the user experience is subjective, it changes depending on its setting and it is dynamic in its duration. Hassenzahl emphasizes the crucial difference in the perception of the experience as a user compared to a designer. Hassenzahl and Tractinsky [147] define the user experience as a result of the user's characteristics, e.g., their dispositions and expectations, the product's characteristics, e.g., usability and functionality, as well as the context in which the user interacts with the product. Similarly, Obrist et al. [156] uses aspects of the user experience that we have already seen and places them under the influence of a context. Roto [157] also acknowledges the importance of the context in which the interaction between user and product takes place.

For Forlizzi and Battarbe [35], there are three types of experience. The first type is the continuous evaluation of all people and things that surround the user. The second type of experience is one that the user perceives as complete and often has a particular impact on their life. Finally, the co-experience refers to sharing an experience with others, which can influence its interpretation.

Each of these models of design or user experience reveals different aspects of this so complex and versatile subject. However, there are several important aspects they agree upon. These aspects include the emotions elicited by the whole experience, which can be influenced by the perceived aesthetics and usability of the product, as well as by the interpretation of the experienced meaning. In addition, they all emphasize the subjective nature of the experience, as it depends on the user's inner state and the characteristics of the product, as well as the experience context, which can influence the way the experience is perceived and interpreted. Finally, they all mention the great influence of the aesthetic and behavioral characteristics of the design on the user's emotions and the assignment of meaning.

As a developer, when attempting to aid the designers in their difficult tasks, it is especially important to understand that design experience is subjective. This means that designers have to slip into the position of the user and experience their own design under the same conditions as their target users. The suitability of this prototype depends greatly on its ability to provide the right situations for experiencing the design stories, or in other words, as observed in the experience models, on the right context. In the use case of a car, the product is used in a road environment that facilitates certain driving conditions and provides the context of a prototyping experience for the aesthetic automotive design evaluation.

2.7.3 The Mixed Reality Experience

With the goal of replacing sketches and PowerPoint presentations with prototyping experiences at such an early stage in the design process of a vehicle, an increase in the virtuality of the experience becomes necessary. Important parameters of the target experience, such as the target vehicle design, are not available in the real world. Additionally, the spatio-temporal aspect [148] with a suitable environment and conditions is difficult to achieve and cannot be used with a virtual vehicle. Therefore, Mixed Reality is necessary for the visualization of the target design and the experience context.

Starting from the right end of the virtuality spectrum, we can use Virtual Reality to completely visually replace a physical prototype, but some senses will be left unstimulated. For designers, however, all senses are important. Therefore, we move the virtuality level to the middle of the spectrum, where we can visually replace any type of physical prototype that does not yet exist in the Early Phase with a virtual twin and fill the experiential gaps by using hardware parts.

Virtual Reality (VR) is the computer simulation of a real or metaphorical system, that allows a user to interact with a simulated system in real time [158]. There are four key elements that describe the functionality of VR [159], which are described below:

- Virtual Environment: The user perceives the Virtual Environment as synthetic, interactive and illusory through the corresponding device. The sensory information that the user receives inside this environment mimics that of a physical one [158, 160]. Specific rules and relationships apply to the objects that exist in this virtual world [159].
- Immersion: There is the mental and the physical immersion [159]. The mental immersion results in the feeling of deep engagement and involvement combined with the suspension of all disbeliefs. Being physically immersed results in the feeling of entering a medium with one’s own body, while at least one sense is being synthetically stimulated. The first immersion that a person experiences is the physical when they first enter the virtual world. The mental immersion can then be experienced after the person has suspended any disbeliefs about the originality of the virtual world around them.
  Immersion can be achieved not only through a VR system but also through other media, such as television, radio and books. However, physical immersion is a characteristic feature of the VR system. The extent to which a user perceives something differently than they would in the absence of external influences could be used to assess the suitability of a medium for achieving immersion.
- Sensory feedback: Another characteristic feature of a VR system is to provide direct sensory feedback to the user depending on their physical position. Immediate interactive feedback is possible by using a high-speed computer. In

addition, a tracking system is necessary to base the sensory output of the VR system on the position of the user.
This feature enables the experience of a simulated physical reality. Furthermore, this experience allows the user to try out scenarios that would not be possible or even too dangerous in the real world, as the virtual world can have its own rules.

- Interactivity: A VR system gains authenticity when the user's actions (e.g., changing location and picking up objects) result in reactions from the virtual world.

The aspects of the virtual experience have been grouped into sensory, affective, relational, cognitive and active [161, 162]. The sensory aspect refers to the sensory input that is perceived accordingly through the senses and is generated by sensory hardware and software. The cognitive part of the virtual experience involves the user's mental engagement with tasks and situations that occur in the virtual world. The affective aspect focuses on the user's emotional state and the degree of similarity between the emotions elicited during the virtual experience and those elicited during a similar experience in the real world. The active aspect can be interpreted as the extent to which the user relates to the virtual experience and its context (e.g., avatars, environment and scenario). Finally, the relational aspect includes the social aspects of the experience, also referred to as co-experience.

A more technical approach to virtual experiences is presented by Parés and Parés [163]. According to their model, the virtual experience is the result of the relationship between the user and the Virtual Environment (VE). The Virtual Subjectiveness (VS), consisting of the logical and physical interfaces, the mapping and the behaviors, acts as an intermediary between the user and the Virtual Environment. The logical interface is the "point of view from which the environment may be constructed" [163, 164] and defines e.g., the field of view and the viewing direction. The physical interfaces consisting of sensors and displays deliver the sensory input to the user while the mappings describe how they are linked to the logical interface. The behaviors complete the virtual experience by defining the ways of interaction with which the user may influence the Virtual Environment, thus establishing the relationship between the logical interface and the Virtual Environment.

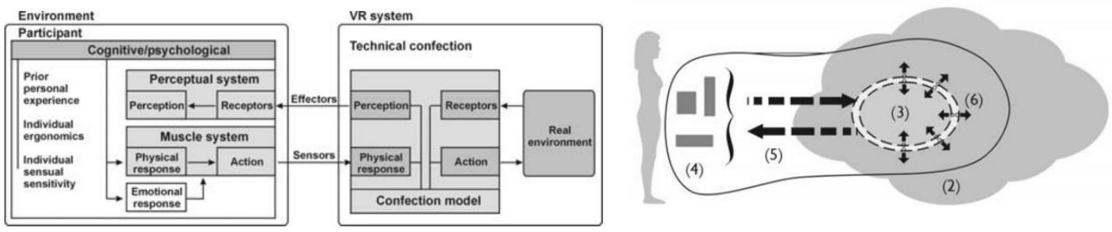


Figure 7: Left: A Virtual Reality system described through the participant's perspective and the technical perspective [163, 165]. Right: The Virtual Subjectiveness as described by Parés and Parés [163] ("The Virtual Subjectiveness composed by the physical interfaces (4), the logical interface (3), the mappings (5) and the behaviors (6)" [163]).

On the other hand, Waterworth and Waterworth [166] offer a psychological approach to the virtual experience. In their attempt to evaluate virtual experiences, they have developed a model consisting of three axes. The first axis describes the user's level of attention to the environment in and around their body. The more we consciously process the current situation, the more presence we feel in the currently present environment. The locus axis refers to the user's degree of attention to the virtual and physical world. Finally, the sensus axis measures the user's level of conscious arousal.

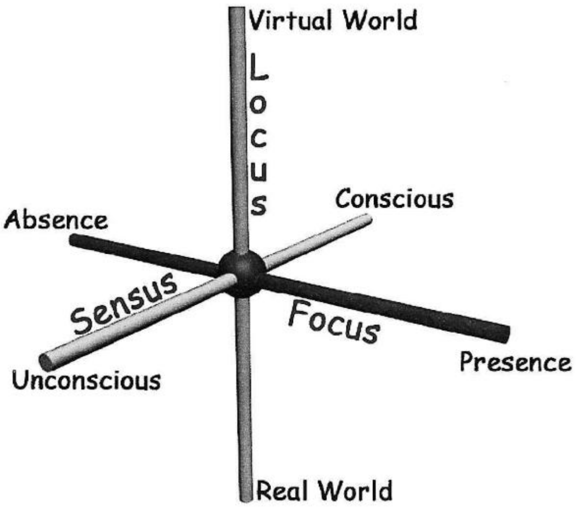


Figure 8: The three axes of Waterworth and Waterworths's model of virtual experience [166].

The experience of being in a mediated environment is essentially what the term "presence" refers to [167]. Presence has been defined as "the extent to which a person fails to perceive or acknowledge the existence of a medium during a technologically mediated experience" [167, 168]. It has been divided into physical presence, which describes the feeling of being physically present in the mediated environment and social presence, which is elicited as the feeling of being together and interacting with a virtual partner. These two types of presence can overlap and create co-presence, which encompasses characteristics of both types. The feeling of presence is influenced by the characteristics of the media and the user. The characteristics of the media include the content, i.e., the objects, characters, and environment. As proposed by Sheridan [169], the amount of sensory information provided to the user, the degree of control the user has over the sensory mechanisms, as well as the user's ability to influence the environment are parameters that determine the degree of presence.

In summary, Mixed Reality experiences are influenced by the hardware and software that help make them possible. Therefore, it is important to consider the system that allows us to build the experience as a combination of technical aspects that lead to psychological aspects and consequently control the similarity of the synthetic experience to a corresponding real experience.

2.7.4 The Game Experience

"The game-world is a "labyrinth devised by men, a labyrinth destined to be deciphered by men" [170, 171].

Game development tools are used to visualize the design stories and facilitate the corresponding experience. Therefore, the target experience is also influenced to a certain extent by the aspects of the game experience.

Many game experience models focus on the relationship between video game and player [172-175]. According to one of these models [172], the following features of video games have specific psychosocial effects:

- the ease of control ("The extent to which a player finds the actions to control the game clear and intuitive"),
- the progress feedback ("The extent to which it is clear to the player how well he or she is doing in the game"),
- the audiovisual appeal ("The extent to which a player appreciates the audiovisual styling of the game"),
- the goals and rules ("The extent to which the overall objective and rules are clear to the player") and
- the challenges ("The extent to which the specific challenges in the game match the players skill level")

The psychosocial effects include:

- the mastery (“A sense of competence and mastery derived from playing the game”),
- curiosity (“A sense of interest and curiosity roused by the game”),
- immersion (“A sense of immersion and cognitive absorption, experienced by the player”),
- autonomy (“A sense of freedom and autonomy to play the game as desired”) and
- meaning (“A sense of connecting with the game, resonating with what is important”).

Another model also considers the video game and the player while adding a third aspect to the game experience, namely the gameplay [173, 174]. The video game aspect consists of:

- the *mechanics*, which define the functioning and progress rules of the game,
- the *interface*, which includes the aesthetic and interaction elements of the game,
- the *narrative*, which is the plot of the game and
- the *consistency*, which is necessary to balance all the previous aspects.

The player has certain motivations that keep them playing the game, a background, shaped by their personal history and expectations of the video game elements. Finally, the gameplay refers to the ambient setting, which is the game environment and its conditions and the platform on which the game is played. In a previous version of the same model [174], gaming experience was a fourth aspect and included immersion and flow.

According to Bernhaupt [175], the game experience is enabled by the video game and the player’s interaction with it. In this model, the video game consists of the gameplay, which refers to the rules and story of the game, and the environment, which is the game’s visual and auditory representation. The interaction between player and game is called Puppetry and it is perceived as the way a player approaches the game until they feel like their actions have an impact on the course of the game. This interaction is influenced by the available actions that the game allows the player to have (Control), the player’s feeling of owning the game after being able to affect what happens with their actions (Ownership) and factors that influence the player’s attempt to gain control of the game (Facilitators, e.g., time and aesthetic values).

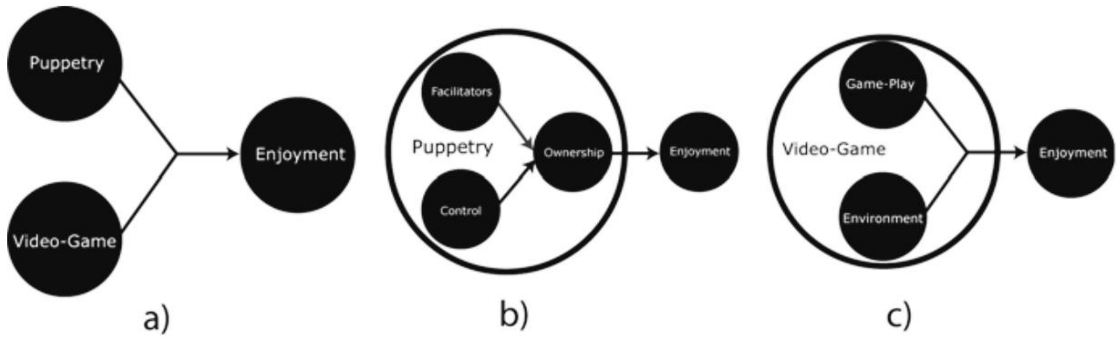


Figure 9: The CEGE model [175].

Calleja's model of game experience attempts to incorporate all experiential dimensions by defining six frames of involvement and connecting them to the player's motivation [171]. The first dimension is affective involvement, which happens when the player's emotional state is affected by the game. It is related to the player's motivation to "escape" from their reality [176] and find excitement. This attempt to escape can be successful through the contribution of virtual environments, which can provide the feeling of excitement to players who like to travel. Virtual environments in games are powerful due to their ability to affect the user emotionally through their aesthetics while engaging them spatially by leading the player to explore them. Depending on the design, games achieve narrative involvement of the player, by letting them create their own narrative or leading them to complete specific tasks before reaching certain milestones in the plot. Tactical involvement refers to the player's satisfaction after setting goals and completing tasks, whether set by the game or by the player. The satisfaction of achieving a high score and optimizing one's skills compared to other players can motivate players in its own unique way. This performative involvement may change focus depending on the way the game rewards the player. Engaging in the social aspect of the game experience through communicating with non-playable characters or writing in the games' group chat results in the shared involvement.

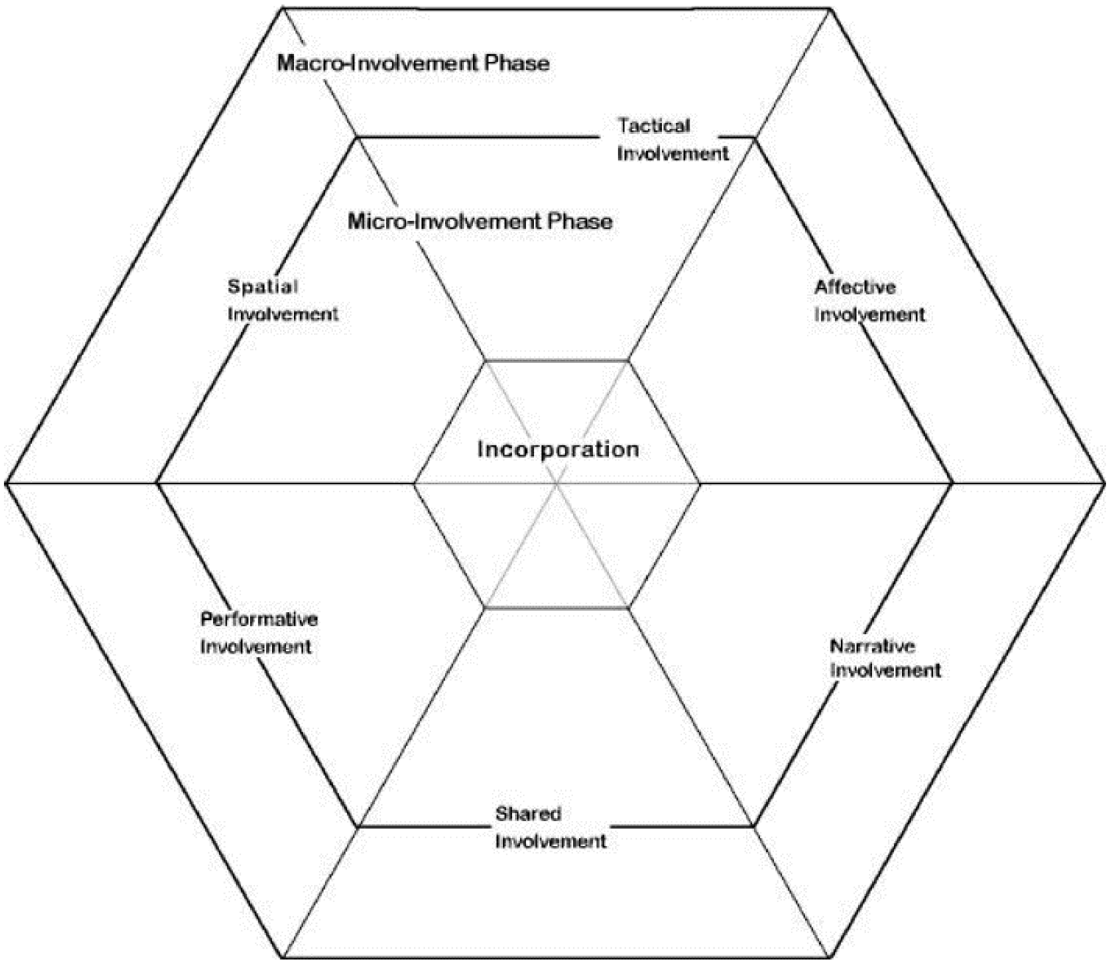


Figure 10: The model of digital game experience [171].

A complete perspective on game experience cannot exclude the autotelic experience and flow [177]. A person can experience flow when they manage to focus their complete attention on achieving their goals. Flow improves the quality of the experience and the overall quality of life. There are several ways to achieve flow, which are related to using the capacity of one's mind, body and memory, as well as leveraging activities, such as thinking about philosophical concepts, communicating and writing.

Ermi and Mäyrä [178] define three dimensions of gameplay experience. The first is the sensory immersion, which refers to the aesthetics and audio feedback of the game. The second is challenge-based immersion, which can be achieved by offering the right amount of challenges while allowing the player to develop their skills. The third and final dimension is activated when the player is absorbed by the story of the game or completely identifies with a certain character.

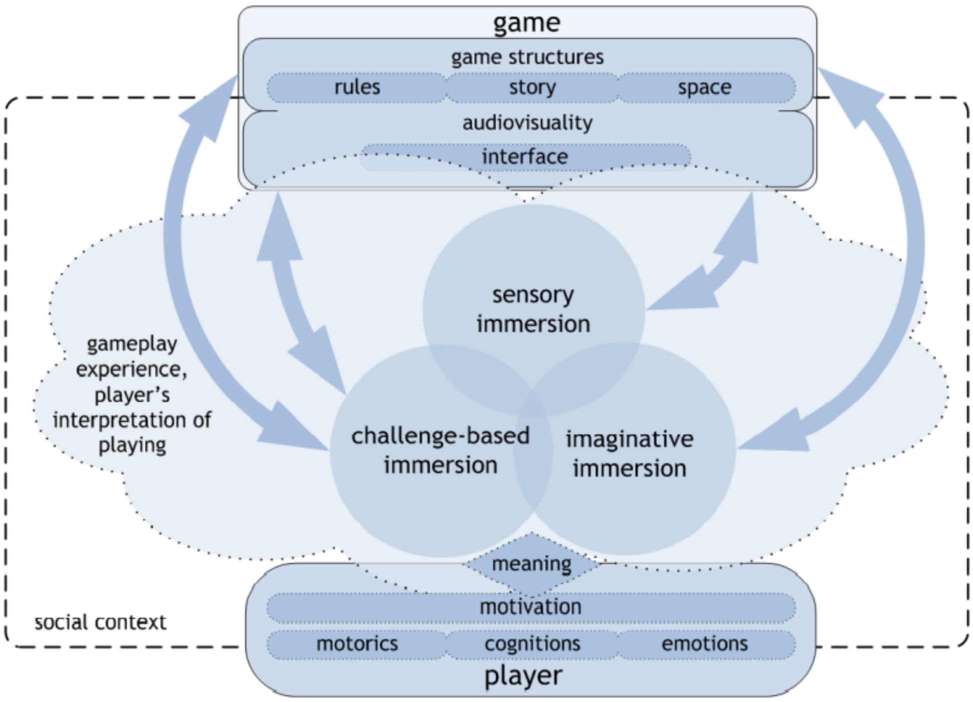


Figure 11: A gameplay experience model by Ermi and Mäyrä [178].

Instead of a game experience model, Sánchez et al. [179] describe the model of playability, which they define as the "degree to which specified users can achieve specified goals with effectiveness, efficiency and specially satisfaction and fun in a playable context of use". According to their model, playability can be described with the help of seven attributes: Effectiveness, Learnability, Immersion, Satisfaction, Motivation, Emotion and Socialization. These attributes also describe User Experience [180], on which this model was based, but are interpreted in the context of games.

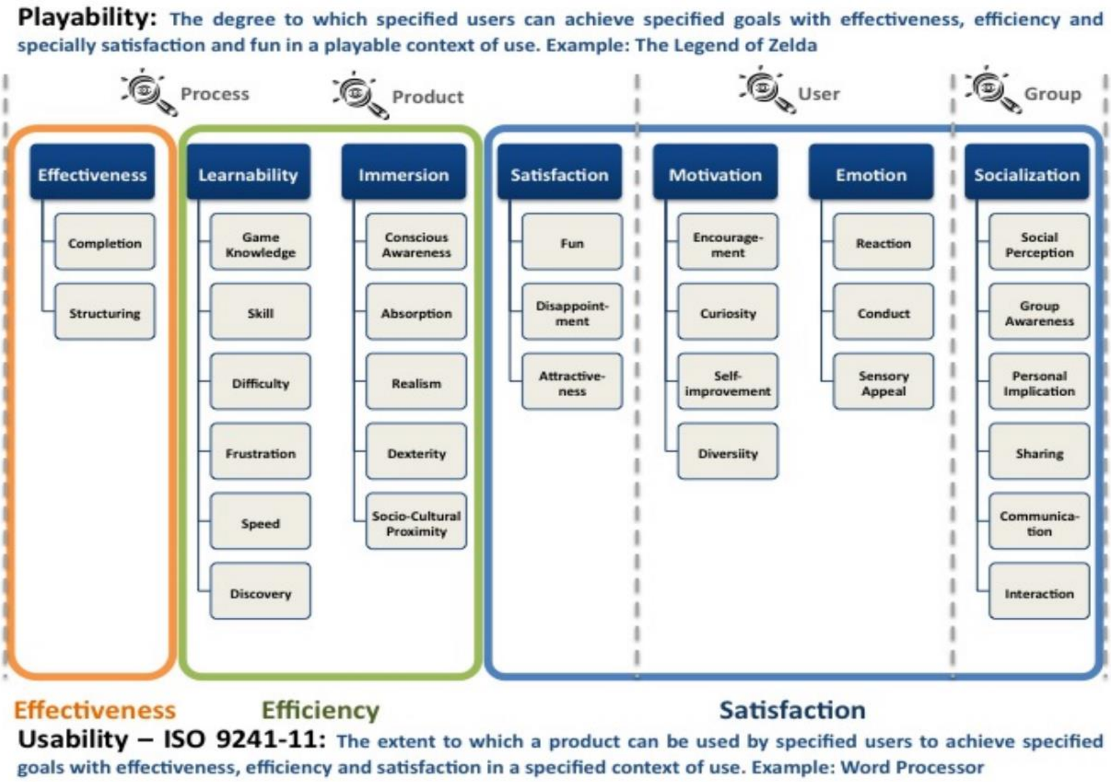


Figure 12: The graph by Sánchez et al. [179] showing the Playability model.

Based on a study in which video game players were asked how they perceive their game experience [181], the participants provided six major categories. As the name of the first category suggests, it contains all Emotional Responses of players, whether positive or negative. The second category is Game Play, which is interpreted as the way the game is played, the degree of the players' immersion and satisfaction. The next category is the social aspect which represents the relationships between players that result from game cooperation or competition. Outcomes of game play is the fourth category, related to anticipated or unanticipated outcomes, ranging from achievement to obsessive playing. The final two categories are the Goals, which describe the individual or collective goals of players and Personal Qualities, which represent the player's skills and abilities.

Poels et al. [182] describe the game experience in nine dimensions, which show up as slightly different experiences during and after the game. The following table provides a clear and efficient overview of the model:

Dimension	In-game experiences	Post-game experiences
ENJOYMENT	*fun, amusement, pleasure, relaxation*	*energised, satisfaction, relaxation*
FLOW	*concentration, absorption, detachment*	*jetlag, lost track of time, alienation*
IMAGINATIVE IMMERSION	*absorbed in the story, empathy, identification*	*returning to the real world*
SENSORY IMMERSION	*presence*	*returning to the real world*
SUSPENSE	*challenge, tension, pressure, hope, anxiety, thrill*	*release, relief, exhausted, euphoria*
COMPETENCE	*pride, euphoria, accomplishment*	*pride, euphoria, accomplishment, satisfaction*
NEGATIVE AFFECT	*frustration, disappointment, irritation, anger*	*regret, guilt, disappointment, anger, revenge*
CONTROL	*autonomy, power, freedom*	*power, status*
SOCIAL PRESENCE	*enjoyment with others, being connected with others, empathy, cooperation*	*accomplishment in a team, bonding*

Figure 13: A digital game experience model by Poels et al. [182].

The aspects of the game experience cannot be fully attributed to the target experience, but since they are connected to the tools used to create these experiences, they can be considered as partly influencing factors. The greatest difference between our target and a game experience is that our player aims at using it not for their entertainment but for the evaluation and presentation of their design. Nevertheless, both players receive certain emotions which are indicators of the success of the experience, only in our case these emotions play the role of informing the designer in which ways the design should be changed or maintained. However, the simulation that designers require to reproduce the experience that their design provides can be considered as a serious game, that carries the various aspects we have mentioned in this chapter to its own unique extent, just like any other game.

2.7.5 The Prototyping Experience

The target experience is meant to be used by automotive designers as an experience prototype. Therefore, the fifth and final perspective to describe the target experience should be that of Prototyping.

Loehmann [4] delves into the world of experience prototyping and provides a spherical overview of all its aspects. According to his definition, “prototyping describes the iterative and economic implementation of different representations of a design idea during different stages of the design process by using any kind of medium with the purpose of understanding, exploring, communicating and evaluating form and function of a future interactive system or parts of this system”. One of the most important uses of prototypes is their contribution to the communication of design ideas, not only between people from the same department, but also from different disciplines and with different skills. Prototyping also reportedly leads to emotionally positive experiences for the people who use it. It helps employees interpret failure as a method of learning and increases satisfaction and motivation by providing them with immediate feedback on their performance [4, 183].

Prototypes have certain parameters by which they can be described. One of these is the prototype’s fidelity, which has been described by Houde and Hill [4, 184] as the degree of visual and functional similarity to the final design. Two main criteria for choosing the degree of the prototype’s fidelity are the phase of the production process [4] and the goal of the evaluation for which they are used [4]. Low-fidelity prototypes have been suggested for early phases of the design process [4, 185], for experience prototyping [27] and to boost creativity in brainstorming new ideas [4, 186]. However, if the complete functionality and interactivity of the prototype is needed to conduct extensive user testing, then high-fidelity prototypes should be preferred [4]. Particularly important in experience prototypes is the part that is visible to the user. Therefore, the front-end of the prototype should be implemented as real as possible, while the back-end can be simulated with simpler methods [4, 187].

Prototypes can also be described by their level of detail, also called as their resolution [4, 184]. Low-resolution prototypes are encouraged in the early stages of the design process to increase creativity for alternative ideas and to stimulate discussion of high-level issues [4, 186].

Depending on the range of functionalities and the level of detail, a prototype can be horizontal or vertical [4, 185]. Horizontal prototypes contain a wide range of functionalities that are not implemented in great detail. On the other hand, vertical prototypes provide only a few functionalities but a high level of details. Therefore, if an evaluation aims only at finding out the right functions of the final design, a horizontal prototype appears to be the right choice. However, if the evaluation aims to optimize the implementation of the functions, then a vertical prototype should be preferred.

Another approach to prototyping is to distinguish between throw-away and evolutionary [4, 188]. The first method follows the creation of several disposable prototypes that are created to serve as “stepping stones towards the final design” [4, 188]. It is recommended for focusing on specific design issues, but it requires much more resources than evolutionary prototyping. The second method starts with a specific prototype and evolves it until it becomes the final design. The result can be a robust product but planning for the prototype’s optimization and guaranteeing its success should be completed from the beginning.

A further distinction has been made by Buchenau and Suri [27] between passive and active prototypes, based on the level at which the user can interact with them. Passive prototypes, such as storyboards and sketches, offer a way to communicate design ideas but no way to interact. Active prototypes, such as experience prototypes, require the active participation of the user, as they “are needed to successfully (re)live or convey an experience with a product, space or system” [4, 27].

Bordegoni et al. [52] have created a framework for mixed prototyping that consists of three axes. The first axis describes the prototype, which can vary from real to virtual. The second axis refers to the user who can also range from real to virtual. The third axis displays the type of interaction the prototype offers and can range from purely visual and consequently indirect interaction to visual and haptic or, as interpreted in the framework, direct interaction. The term haptic refers to active touch [189].

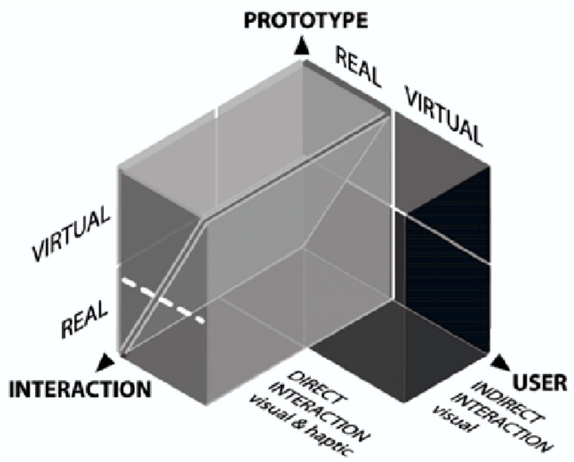


Figure 14: A framework for Mixed Prototyping [52].

Lim et al. [190] offer a different view of the variables that can be used to describe prototypes. They mention the prototype’s appearance as a dimension that can describe “the physical appearance of a design” with variables, such as size and color. The data dimension contains “the information architecture and the data model of a design” with variables, such as data size. The functionality dimension describes the "functions that can be performed by the design". The interactivity dimension refers to "the ways in which people interact with each part of a system" (e.g., feedback). The spatial structure dimension describes the way "each component of a system is combined with others". Lim et al. [190] also define the best prototype, as "one that, in the simplest and most efficient way, makes the possibilities and limitations of a design idea visible and measurable". This economic principle of prototyping should be taken into account, when deciding on the values of the dimensions for a particular prototype [190].

Filtering dimension	Example variables
Appearance	size; color; shape; margin; form; weight; texture; proportion; hardness; transparency; gradation; haptic; sound
Data	data size; data type (e.g. number; string; media); data use; privacy type; hierarchy; organization
Functionality	system function; users' functionality need
Interactivity	input behavior; output behavior; feedback behavior; information behavior
Spatial structure	arrangement of interface or information elements; relationship among interface or information elements—which can be either two- or three-dimensional, intangible or tangible, or mixed

Figure 15: Variables that can be used to describe a prototype by Lim et al. [190].

After studying the target experience through the Automotive, Design, Mixed Reality and Game Development perspective, we turned our focus to its Prototyping nature. In this perspective, we gained new insights that connect the target experience to the development process. Furthermore, we are reminded of the influence that the available resources have on the type and level of detail of the prototype. We can conclude with the fact that a successful experience prototype is not necessarily the one that simulates perfect realism, but consists of a combination of aspects, each developed to a certain necessary level, so that it can support the designer in their task of evaluating and presenting an experience.

2.8 State of the Art in Procedural World Generation

A fair amount of research has used driving simulations to conduct evaluations in the automotive industry. However, there is little research for automotive designers, their requirements for Mixed Reality prototyping experiences, the methods to implement these requirements, as well as to provide the designers with their own automated tools.

In this Chapter, the state-of-the-art research is divided into virtual simulations that have been used for automotive evaluations and procedural generators that have been created for various use cases.

2.8.1 Simulations for Automotive Evaluations

The automotive design involves a highly complex combination of functional and aesthetic features. The need to evaluate and explore the experience facilitated by this combination has led to the use of game development tools for the generation of virtual environments. The associated evaluations and studies have focused on specific road systems in the generation of the environments, which play the role of the context in, e.g., training simulations [191] and in-vehicle user experience evaluations [24]. In some cases, as in this empirical study that focuses on the driving conditions [192], the environment was even generated using procedural generation tools. Virtual environments have also been used in studies that focus on communication between vehicle and pedestrians [193], as well as specific user experiences inside the vehicle in Virtual Reality [194, 195].

Mixed Reality prototyping is widely used in the automotive industry. Augmented and Virtual Reality are used to evaluate interior design elements in physical prototypes. The same medium is also used in the usability evaluation of UI interaction methods and to support maintenance and manufacturing processes. Further use cases for Mixed Reality prototyping in the automotive design process include the collection of feedback from end users on the aesthetic and ergonomic quality already in the early stages of development [25, 52, 196, 197].

The focus of all these simulations, regardless of their virtuality level, lies not on the virtual environments, which were generated to meet the needs of these specific studies. It lies on a specific situation, experience or set of interactions and the user’s reactions, coping techniques and feedback. Modelling such a specific environment from scratch for each use case with a high graphics quality or selecting from a set of specific environments does not cover the variety of the required use cases, which is essential to match the creativity and inspiration sources of the automotive designers. Further challenges, such as the nature of the designers’ trial-and-error process of defining the environment and the large amount of resources needed to create suitable environments for design presentations confirm the conclusion that a new framework is needed to develop environments for design presentations and evaluations. This should allow the procedural generation of the environment based on real-world data and contribute to higher efficiency in the environment development process.

2.8.2 Procedural Generation

Procedural generation is an algorithmic method of creating data, with the created output varying from 3D assets to complete levels. One major advantage is the reduction of development time. It has been used to generate virtual environments for many recently released games (e.g., Horizon Zero Dawn [198] and Tom Clancy's Ghost Recon Wildlands [199]). Another important advantage of this method is the ability it offers to adjust the degree of control over the generated output and the resulting potential to provide this control to people without programming skills.

2.8.2.1 Street Generators

The creation of procedural street generators is aimed at modelling road networks in fantastic cities. The input of these generators requires the user to follow methods, such as creating and editing tensor fields [200], specifying a start sequence for an L-system and values for parameters, such as population density and network pattern [201], providing vector data for roads and rivers [202] and dividing the map into regions and choosing a template for the road generation [203]. Other inputs include the sketch of the roads [204, 205] and a high-level graph that represents control nodes or paths [206].

The roads can be generated in grid and radial patterns by using the hyperstreamlines of the provided tensor field [200]. Another method is to leverage the L-system [201], which is a grammar-based rewriting system that recursively modifies a start string based on a specific set of rules. In [202], the generator models the streets and rivers based on the provided vector data and modifies further changes that the user makes in the world. The template-based system in [203] takes into account information about the population density of the area from corresponding maps and divides the regions provided by the user into approximately equal parts. Starting from a high-level graph [206], a lower-level graph is computed that adapts to the control nodes of the higher level. Secondary roads are then modelled expanding inwards by using an L-system with predefined parameters. In [204, 205], the roads are modelled based on the roads placed on the terrain by the user using vector lines and polygon tools. The Voronoi algorithm, subdivision and L-systems have also been used in an attempt to generate road networks that correspond to the real-world road patterns of two South African cities (Johannesburg and Cape Town) [207]. The shortest path algorithm has been applied to generate optimal roads [208, 209]. This approach uses weights that include terrain features (e.g., river) and information (e.g., population density), as well as anisotropic costs that depend e.g., on position and direction [208].

Real-world data is used as input to generate road networks, mostly for specific use cases, such as urban planning [210] and military training [211]. In these examples, a similar approach was followed that starts with downloading the target area from Open Street Map and the heightfield of this area from a provider (e.g., United States Geological Survey (USGS)). In the first use case [210], the CityEngine software is used to model the roads without texture and street lines, extrude the OSM footprints, add detailed textures and model natural features based on the satellite texture of the terrain. In the other use case [211], the roads have been modelled based on the information provided by OSM and pre-modelled buildings and street elements have been placed by CityGen3D [212, 213].

To be able to cover our use cases and maintain a certain degree of control over the procedural generation process, the output geometry should contain information about its elements. This information stored in points, primitives or vertices can then be used to add further details. For example, the information about the direction of each lane is necessary to add a traffic simulation that can be automatically adjusted to the generated environment and place traffic signs along the streets. State-of-the-art tools for the procedural generation of streets and worlds, such as the "City Map Generator" (Figure 16) or "SceneCity" add-ons for Blender, create geometries (Figure 17) that do not provide this kind of information. In addition to this disadvantage, the insufficient graphics quality offered by other tools, such as the Mapbox's plugin for Unity (Figure 17), combined with the fact that it can only be used with a specific Game Engine, makes the tool inefficient for the requirements of our ideal environment generation process. Other software, such as the Unreal plugin by Cesium (Figure 17) can generate 3D environments based on photogrammetry data. In this case, however, the output quality depends on the resolution of the input data, which has to be purchased from external companies. Depending on the quality of the input data, the output geometry often requires further processing, e.g., to remove unnecessary artefacts or to replace certain objects with higher quality assets. Moreover, the input data is often not available, which means that additional costs are incurred for obtaining it. Finally, another tool that is based on real map data is CityEngine [212, 213] that can generate streets and buildings but requires a lot of calibration within a complex system of parameters to adjust the appearance of the environment.

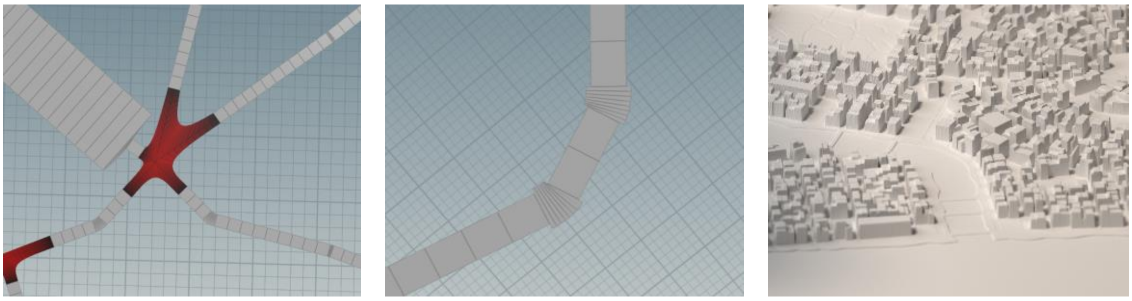


Figure 16: Left and middle: Three modelling errors generated after applying the Street Generator node in Houdini. Right: Environment generated in Blender with the City Map Generator [214].

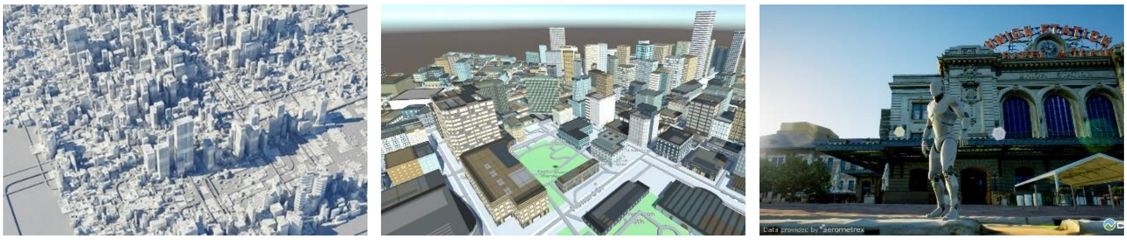


Figure 17: Left: Environment generated in Blender with the SceneCity add-on [215]. Middle: Environment generated in Unity with the Mapbox plugin (up) [216]. Right: Environment generated in Unreal with the Cesium plugin (down) [217].

2.8.2.2 Building Generators

In this section, we review the different approaches used for the procedural modelling of 3D buildings based on their methods, their use cases, and the potential to adjust the level of control of their output.

Best known for its ability to generate plants [218], the Lindenmayer- or L-system can also model an object hierarchy and mesh. Therefore, it has also been widely used in the generation of buildings [219, 220]. In the corresponding cases, the L-system has been applied to shape the building by setting certain parameters and to add façade details by covering it with a texture.

The shape grammars have played an important role in the design of industrial architecture. A characteristic example has been presented by Rau Chaplin et al. [221], with their user-friendly interface that has been developed to adapt an existing exterior and interior design of a building and provide architects with a tool to reduce design time and cost while maintaining high quality. Further methods that also rely on shape grammar have been used to generate walls [222], model cities [213], increase the variety of styles by allowing more building shapes [223], or using vectorial shapes [224].

Also members of the same algorithmic family, the split and control grammars have been used to design the façade by dividing it into compartments and processing them according to a set of rules [225]. The result includes highly detailed buildings, but with a correspondingly high polygon count and a very complex definition of rules to add new architectural styles.

A semantic approach in combination with shape grammar techniques has been applied by Tutenel et al. to design and model a consistent floor plan [226]. A similar approach has been taken to create the building with predefined façade module positions [227], based on the input data from an XML file containing specific parameters and the geometry of a 2D floor plan.

Real data has also been used as input for the procedural generation of buildings. In such cases, floor plans have been designed by training Bayesian networks [228], and buildings with multiple LODs have been modelled by using a custom shape grammar [229]. In other cases, the generation has focused on the external environment. In such a case, Open Street Map data has been used as input for the random generation of buildings along the provided streets using the CityGen3D software [230].

A modern approach to procedural building generation uses the Wave Function Collapse (WFC) algorithm. This algorithm accepts a sample pattern of the desired output and reconstructs and replicates it with different variations [231].

Building façades have also been modelled using image-based procedural generation techniques. The output geometry of this method, which is also based on shape grammar, includes 3D façades without a connection to a building structure, as well as realistic façade layout designs [232]. Large-scale environments have been modelled using model synthesis which requires a small example model [233, 234]. The input example model should contain just the right amount of information to recognize the repeating patterns and create a successful, more complex and natural-looking model [235].

After exploring the existing methods and tools for the procedural generation of buildings, we came to the conclusion that their level of complexity and the diversity of their outputs do not exactly fit the use cases and needs of the automotive designers. Clearly, a new approach is needed, that is adapted to the requirements of automotive designers and developers. One of the factors that makes these use cases special, is that they usually involve a driving simulation. This means that only the lower part of the buildings is visible through the car windows and to the driver, while the buildings further away are visible through the windshield but not in full detail. These conditions shift the focus of the procedural generation on the sections of the buildings that can be seen better by the driver. Due to the high performance required to render the vehicle in as much detail as possible, this fact can be used as a great advantage to reduce implementation time and increase performance by not building the interior of the buildings and the façades that are not visible from the road. Another factor is defined by the requirement for realistic driving conditions and buildings that represent the look and feel of a particular area in the world, as will be shown in the next chapters. This dictates the essential use of real map data and excludes the application of a large portion of the solutions in their original form, which are found in research. Moreover, since technical knowledge is usually not part of the expertise of automotive designers, the generation of buildings should require minimal effort and not increase their workload. Consequently, building generators with many input variables are also not suitable for automotive designers.

2.8.2.3 Generators of Topology, Biomes and Ecosystems

The topology of a virtual environment is usually simulated with a heightmap, a 2D grid that contains elevation values for every point of the terrain [236]. Methods for creating a heightmap include mid-point displacement, erosion simulations and fractal noise [237]. As we have already seen, the heightmap of a world area can be obtained from different providers, each offering different areas, such as USGS [210, 211, 238]. Other methods that provide the input for the generation of terrain include the definition of splines [239], which in combination with deformation features and a diffusion equation model a detailed terrain. Terrain can also be modelled to accommodate existing features, such as rivers [240].

After the main terrain is created, the focus of the modelling shifts to its characteristic features. Common procedural techniques for creating oceans include “setting a fixed water level” or “starting a flood algorithm from points of low elevation” [237]. River networks are generated by starting with a single river and subdividing it recursively [241]. A curved river has also been modelled by starting with a single triangle that contains an entry and exit point and subdividing it recursively [242]. Another method applies a simple physics simulation to let river particles fall from mountain ridges and then process the terrain accordingly [243].

The L-system has also been used for the generation of 3D plants, whose set of rules, in this case, can model a plant, starting from the root, followed by branches, whose size decreases in each step until the leaves are added [244]. A graph-based method has been presented by Lintermann and Deussen [245], where plant components are placed in a graph and connected components are listed in subgraphs. In an intermediate graph, instances of these components are added to form a plant. The plant models of Makowski et al. [246] and their simulated growth are based on a framework for large ecosystems that takes into account the interaction of plants with each other and the resulting branch structures. Other methods for procedural generation of plants include sketch-based techniques [247-249], reconstruction from images [250-253], videos [254], as well as laser-scanned point sets [255-257].

Biomes are “major terrestrial or aquatic life zones, characterized by the vegetation type in terrestrial biomes or the physical environment in aquatic biomes” [258]. They are characterized by certain temperature ranges and amounts of precipitation. These two variables determine the flora and fauna that live in these areas.

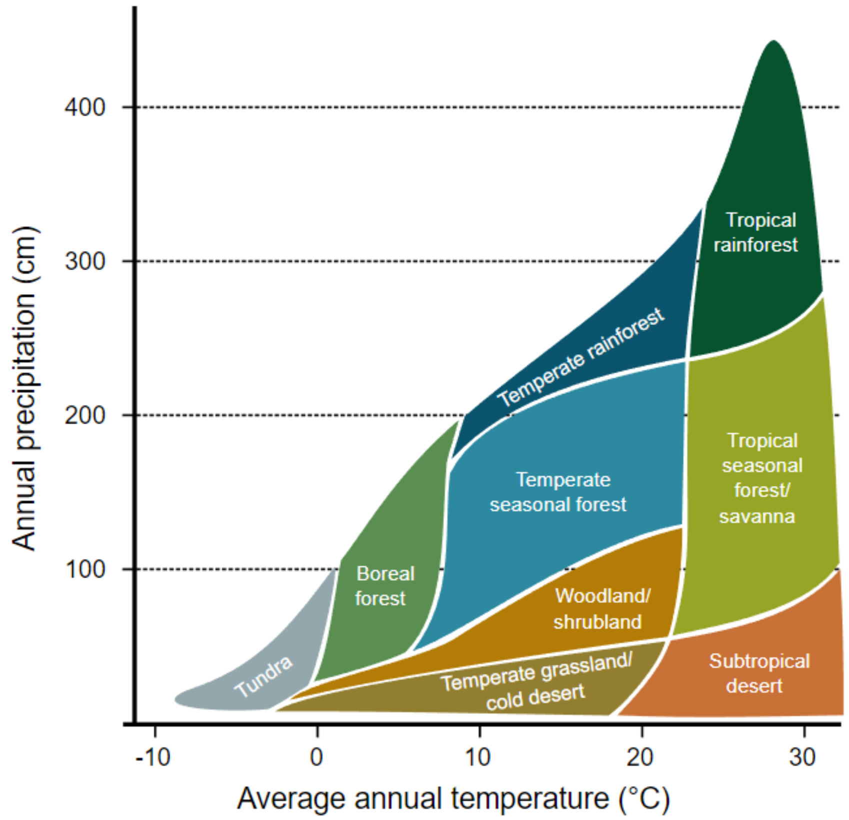


Figure 18: The graph shows the relationship between annual precipitation and average annual temperature by Navarras [246].

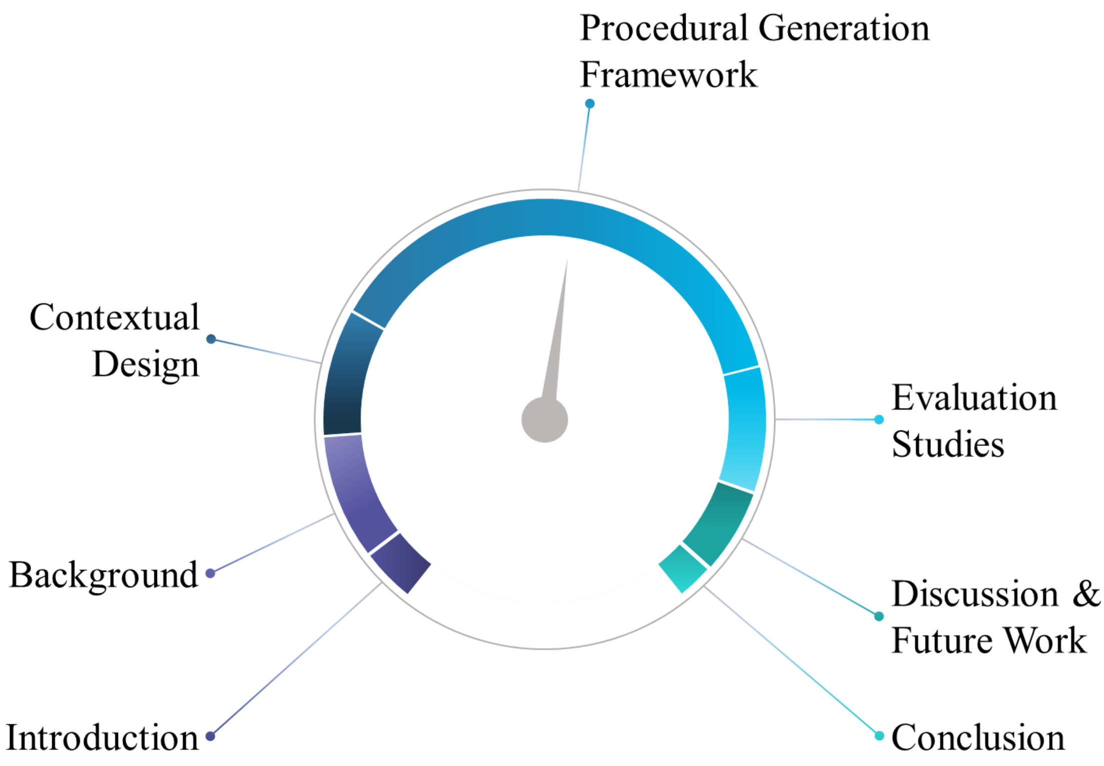
In order to simulate ecosystems and consequently place the right plants in the right density and position, the model of Deussen et al. [259] receives a heightmap and a water map, as well as several pieces of information about the plants with which the terrain is to be populated (e.g., rate of growth, optionally an initial plant distribution). A simulation is then carried out to calculate the distribution of the plants, also taking into account the competition of the plants for soil, sunlight and water. Another method for vegetation distribution has been presented by Hammes [260], who defines specific ecosystems based on elevation data, relative elevation, slope, slope direction and multi-fractal noise. Depending on the ecosystem, a certain number of plants are randomly placed and textures for the terrain are generated, also considering the level of detail.

A different approach was followed by Fischer et al. [261] to create the terrain, determine the biome and place the plant assets. After creating the terrain using a noise function, certain parameters are calculated including the temperature decline, the wind directions, a precipitation distribution and the resulting biome based on the Whittaker diagram [262]. To refine the terrain details, an existing Digital Elevation Model (DEM) is used and blended with the original terrain according to the DEM's weights. Finally, predefined assets are clustered into categories according to size and other parameters, such as organic and inorganic.

Studies in botany and forestry have provided many approaches to the parameters that describe an ecosystem model. Their models focus on the biological processes of plants and biomes and work with developmental attributes [263] and availability of resources [264]. Other models leverage information on the diversity and influence of plant species [265], plant productivity [266], as well as the impact of climate change [267].

The procedural generation of terrain and vegetation for environments that are suitable for the aesthetic automotive design presentation and evaluation should focus on real map data, as will be conclusively explained in the next chapters. The main challenge in this task is the automatic mapping of the biome to the geographic coordinates that are selected by the users. Therefore, the need for a completely automated process excludes the use of the graph observed in relevant research showing the relationship between vegetation and average precipitation and temperature, as there is no simple method for automatically calculating the two parameters depending on the geographic coordinates. Instead, a new method should be designed that ensures a small amount of user input variables.

II. Procedural World Generation for Mixed Reality Design Experiences



3 Contextual Design of a Procedural Environment Generation Framework for the Aesthetic Automotive Design

To be able to fulfil the designers' need to create virtual environments that are suitable for design experiences and require minimal resources, we decided to apply Contextual Design [6]. This design method is user-centered and offered us the flexibility to share all design and development steps with the automotive designers and developers. Since our goal is to develop a framework that will support both disciplines and especially the designers, this method was suitable to keep the end user of the framework and their needs in focus. Throughout the complete thesis, the designers were considered experts and we welcomed their expertise to guide us step by step to the final output.

As expected from Contextual Design, we adapted its steps to the automotive designers and developers. Consequently, the Contextual Inquiry was divided into three studies which combine phenomenological and ethnographic research. The aim of these studies was to get to know both groups and collect the parameters and use cases of the virtual environments. This information was interpreted with the help of the designers and developers. The results of the interpretation were analyzed using Affinity Diagramming and visualized in Work Models showing the current process of defining and generating the environment. From the results, specific design goals were derived that showed the development direction for the suggested framework. An additional Work Model for a proposed process was created, combined with a Storyboard that shows the steps of a designer applying our framework for the definition and generation of new environments. Based on the acquired information, the framework was designed and developed, with feedback sought from designers and developers in small informal meetings and in a final formal series of evaluation studies.

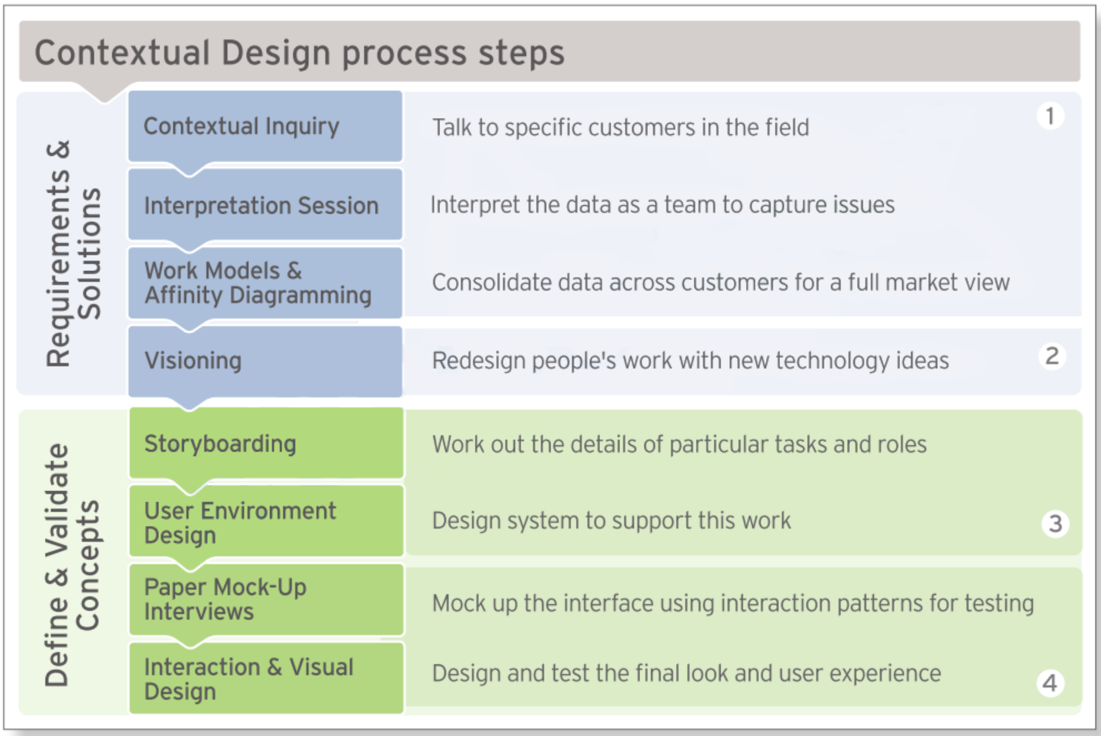


Figure 19: Contextual Design Process [6].

3.1 Contextual Inquiry, Interpretation and Affinity Diagramming

In the first step of our contextual design, we combined ethnographic and phenomenological research as a method to collect data from and about the two focus groups, designers and developers. This data includes the current process of context generation and the requirements for the ideal experience context. So far, the context has been generated by a team of developers according to the designers' descriptions. Therefore, we decided to also include the developers in our search for information about the communication flow between the two groups. We conducted two interview studies, one with automotive designers and one with developers, as well as a series of fly-on-the-wall observations to get first-hand experience of the context creation process. The fact that we worked with the development team for three years was a great advantage for the research. This gave us the opportunity to observe the efforts and outcomes of the developers who had to meet the designers' requirements within the limits of the available resources. Combined, these studies explore the design goals that need to be considered in order to develop a generation process for the experience context, that leads to the desired outcome without the need for modelling skills and that provides a positive experience for automotive designers. Therefore, we seek to answer the following research questions:

RQ1: What design goals should be pursued to transform the current context generation process into a new more efficient way of defining, generating and evaluating the design experience context that is suitable for aesthetic automotive design experiences?

RQ2: What requirements must be fulfilled in order to model the virtual environment as a context of design experience and its conditions?

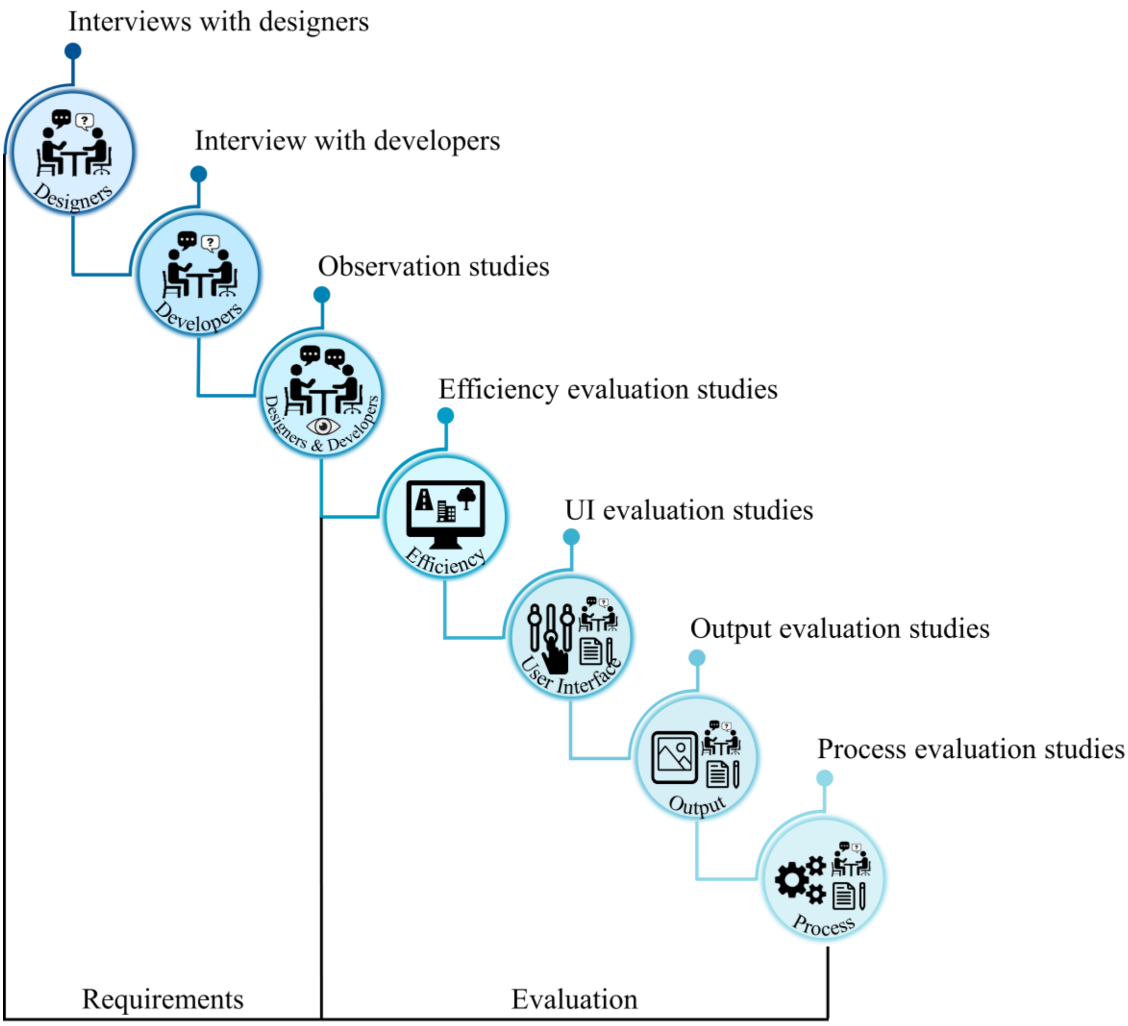


Figure 20: Overview of all conducted studies.

3.1.1 Automotive Designers

In this section, we present results of the first interviews with designers about the application of design experiences in different departments of the automotive design.

Participants and Procedure: Five automotive designers (2 females and 3 males; age ranging from 26 to 35, mean 30.5) were recruited. This number of participants was considered sufficient due to the narrow scope of the research and the small number of participants who had already used such prototyping experiences [268]. They were suggested by the development team which collaborates with the designers to generate the experience context for them. Each designer works in a different design department and has 5 to 10 years of design experience with one to two automotive companies.

Two of the semi-structured interviews were conducted in person at the designers' workplace and the other three in an online session due to corona-induced limitations. Each interview lasted about 60 minutes and started with a demographic survey on age, gender and design experience. After that, we explained the aim of the study to the designers and asked them the following questions without making any suggestions about the answers:

1) What is the current design presentation process, used in your department?
2) What software do you use to prepare the design presentation?
3) Have you prepared a Mixed Reality design experience with the help of the development team to present the design?
4) If yes: Can you describe such an experience and are there any challenges that arose during the preparation process and/or the presentation?
5) If no: If you had the opportunity to present and evaluate the design within a Mixed Reality experience, what would that experience be like?

Data Analysis: The data was analyzed using Affinity Diagramming to identify the parameters of the design experience by clustering down to the smallest details offered by the designers. We gathered the designers' statements on the interview questions, as well as other topics raised by the designers themselves during the interview. We categorized 53 statements with the help of the development team. After two rounds of applying Affinity Diagramming, several subthemes emerged. In the second round, only minimal changes were made to the sub-themes. We interpret this fact as an indicator that all team members, who carried out the interpretation and analysis had the same understanding of the statements and their relationship to the focus subject.

The final categorization includes 1) the *Environment Visualization*, 2) the *Vehicle Visualization* and 3) the *Experience Participants*.

1. The *Environment Visualization*:

Terrain Geometry: Six participants made statements about the terrain geometry of the virtual environment. Two of them (UI designers, PUI1 and PUI2) expressed the need to have a choice between an urban and a suburban environment. According to the same participants, the terrain should correspond to the appropriate road type and the design story taking place at this specific area. In addition, they emphasized the importance that the environment should elicit in the driver the feeling of being in a real location of the target market. According to the same participants, the virtual environment should correspond to the functional and aesthetic context of the design stories.

Two different participants (2 Exterior Designers, PE1 and PE2) stated that they need the possibility to switch between different environments during a design presentation. They also mentioned that exterior designers watch films produced in a real location of the target market and edited to show the new design over the actual vehicle used in the scene.

Another designer (Scent Designer, PS) stated that the virtual scene they use for their design presentations is the same as the one used for the UI-design presentations.

On the one hand, most designers use XR design experiences and interact with the interior of the vehicle. On the other hand, the exterior designers use films and focus on the exterior with corresponding interactions. Despite these differences in the medium of the experience and the type of interaction, the requirements for the virtual environment point in the same direction. Both groups of designers need environments that reflect the vehicle's character and represent real locations of the target market.

Finally, another participant (Color & Material Designer, PC&M) stressed the importance of a closed, color-neutral environment for their presentations, as it does not allow distractions from the vehicle's colors.

Road Geometry: Two participants (UI designers, PUI1 and PUI2) mentioned parameters they use to describe the road geometry to the developers. These include the road type (e.g., primary/country road), the road form (e.g., curvy) and the driving maneuver (e.g., turns which influence the form of the required intersections). Since the terrain geometry is an indicator of the roads it contains, the requirements for the terrain influence the shape of the road geometry. For example, an urban area tends to have different road types than a suburban area.

Building Geometry: Two participants (PUI1 and PUI2) emphasized the importance of buildings as indicators of the area (e.g., urban) and the road type (e.g., country road). It is equally important that the buildings present a certain degree of diversity, not only between buildings with different uses (e.g., a residential building should look different from a commercial building), but also between buildings with the same use (e.g., two residential buildings should look different). One of them (PUI1) also mentioned the case of "style guide-buildings". In such cases, they would like to have the freedom to change the textures of the buildings to fit a certain style. The other (PUI2) did not mention such a case, but still expressed the need to change the building textures as a requirement. The statements on this sub-theme came from only two participants (PUI1 and PUI2). However, since the building geometry influences the appearance of a given location and consequently the sense of presence, most of the requirements for the building geometry should be relevant for all types of design presentations, except for Color & Material.

Vegetation: The only statement on this sub-theme was made by two participants (PUI1 and PUI2). According to them, the vegetation needs to match the real location reflected in the virtual environment.

Lighting Conditions: The lighting conditions are relevant for all participants. One participant (PC&M) expressed that lighting is crucial, as they explore the experience of color and material during their presentations. This participant needs the same lighting conditions as in the corresponding aesthetic presentations and evaluations with physical prototypes. Another participant (PE1) mentioned the need to study the form of the vehicle under the lighting conditions at the target market location. They characteristically added that a car in Los Angeles looks different than the same car in Munich. According to a third participant (PE2), realistic daylight is necessary for their design presentations.

Time of Day: Three participants (PE1, PUI1 and PUI2) specified the time of day as a necessary parameter of the environment for both the functional (e.g., design stories occurring at night) and the aesthetic context (e.g., the time of day influences the atmosphere of the environment through the lighting). Another participant (PE2) was satisfied with the virtual scene being lit by daylight as a default and only option. This is an example of how differently designers even from the same design department prepare their presentations. This is the result of the different types of stories that different designers within a department are responsible for.

Weather: Two participants (PUI1 and PUI2) agreed that the weather conditions and the type of visibility (i.e., good, or poor) must be able to be selected. Both parameters should be defined according to the requirements of the design story.

Environmental Simulation: The same two participants (PUI1 and PUI2) also had consistent statements about the need to choose the type of traffic situation (i.e., no traffic, flowing traffic or traffic jam). This parameter serves the functional context of the design story.

2. The *Vehicle Visualization*:

Visual Representation of the Vehicle: All participants (PC&M, PE1, PE2, PUI1 and PUI2) except one (PS) expressed the desire to switch between different design focus parts of the vehicle. Those focusing on exterior design (PE1 and PE2), need to evaluate more than one exterior part during the presentation and correspondingly experience the same design stories, but each time with a different exterior design element. They must also be able to move the exterior form lines of the vehicle, as well as save their new positions. The one focusing on the Color & Material design (PC&M) must be able to switch between colors and material samples, as well as have a library of scanned materials available. Moreover, a color change of the exterior should also be offered as an option.

Functional Representation of the Vehicle: Two participants (PUI1 and PUI2) requested the possibility to choose between autonomous and manual driving of the virtual vehicle to cover the functional part of the design stories.

Reference Vehicle Models: This sub-theme includes statements from two participants (PE1 and PE2) whose design work focuses on the exterior. They described the challenge of fitting many CAD-sketches, photos or pictures into one slide of a PowerPoint-presentation in order to see all the reference vehicle models simultaneously and compare them. They also found that it was impractical and tedious to switch between several software tools during the presentation to show different types of 2D and 3D media (e.g., CAD-sketches, photos, films). These two challenges were their motivation to propose large posters in a virtual space as part of their Mixed Reality design presentation and evaluation.

Driving Simulation: Two participants (PUI1 and PUI2) defined the parameter of the vehicle's state of motion (i.e., parked or in movement) as an important available option, depending on the design story and type of evaluation.

3. The *Experience Participants*:

Avatar: Two participants (PUI1 and PE2) stated that it was important to have a virtual representation of themselves and the other participants. One of them (PUI1) suggested the selection of an avatar and its virtual clothes as an acceptable solution if the ideal solution of scanning the experience participants is not possible. The same participant stressed the importance of highly accurate tracking of the avatar during the XR experience, as finger interaction with the user interface (e.g., swiping across a display, pressing buttons) needs to be shown.

Camera: Two participants (PUI1 and PUI2) stated that they need the possibility to change the camera perspective within the virtual scene. One of them (PUI1) indicated more detailed camera positions: "behind the driver's shoulder, in the front of the driver's face and a camera perspective to capture the complete driving situation". Furthermore, they added the possibility to move freely around the virtual scene in bird-view ("God Mode") to see the action from different perspectives.

Approach to VR/AR: One participant (PE1) showed a negative reaction to VR and AR experiences due to the motion sickness they experienced in the past. In addition, they mentioned the corona-related limitations as another factor that complicates the use of VR and AR glasses. According to this participant, the result of such limitations is that designers are not allowed to go to the office and consequently cannot use the available glasses, or if they are allowed to work at the office, it is difficult to disinfect the glasses after use. However, other participants were open to the idea and welcomed the use of XR experiences for design presentations and evaluations. Two participants (PUI1 and PUI2) already prepare these XR-experiences systematically with the help of developers.

Another participant (PC&M) expressed curiosity and enthusiasm for such a design experience. Another participant (PE2) immediately recognized the potential of switching from 2D films to interactive 3D experiences with a realistic environment, also as a method to reduce the cost of filming in a particular location.

Hardware Options: Three participants (PUI1, PUI2 and PC&M) already use a driving simulator for their design presentation, both in a static (i.e., vehicle is parked) and in a dynamic state (i.e., driving vehicle). In addition, one participant (PUI1) expressed the need to be able to switch between different hardware parts of the simulator in preparation for the presentation. Another participant (PS) uses perfume dispensers to release and experience the scent.

Collaboration: This sub-theme includes a statement shared by three participants (PUI1, PUI2 and PE2). After observing the consequences of corona-induced limitations, they suggested the concept of multiplayer participation in an XR experience in the form of a web application. This method would allow all designers to participate in the design experience regardless of their location. Another participant (PUI1) expressed discomfort about having to contact too many people to generate a design experience in order to use it in their presentation.

3.1.2 Automotive Developers

In this section, we present the results of the interviews with the developers of the virtual environments used in design experiences. These interviews explore the current context development process and its main challenges from their perspective. We aim to uncover information about the communication flow between designers and developers, as well as to detect accompanying challenges.

Participants and Procedure: Five automotive developers (all males, age between 25 and 50, mean 35.4) were recruited. They are part of the development team, that has been working with the automotive designers for three years to deliver the full virtual design experience and consequently also its context. We contacted them by email or phone and invited them to an online session, due to corona-induced limitations. Each interview lasted about 30 minutes and started with a demographic survey on age, gender, developer experience and experience of working with automotive designers. We then explained the aim of the study and asked the following questions:

1) What is your role in the development team?
2) Can you please describe the development process of a design experience and its context from the beginning to the end of a project?
3) Are there any challenges that arise during the development process?
4) How do designers describe the virtual environment they want you to model?

Data Analysis: The interview results were also analyzed using Affinity Diagramming. We extracted 30 statements from the developers' answers, which we bundled into 4 sub-themes:

Process of Environment Development: Three participants (P2, P3, P5) highlighted the lack of a standard process in the development of virtual environments for the design experiences (P2: "It is difficult to describe a process because it is a very vague concept", P3: "There still is no official blueprint for this process", P5: "We are currently at a discovery phase"). According to one participant (P4), the designers use the environment as a context to visualize vehicle functions. Another participant (P1) emphasized the importance of the environment as a factor in defining the context and message of the product. About the beginning of the process, one participant (P2) stated that the designers communicate the new concept through a series of online meetings. Another participant (P4) added that the designers start from an abstract vision that they present on paper or other media (e.g., PowerPoint presentations) and try to communicate during these early meetings.

According to two participants (P1 and P2), important questions are answered, such as "How can we stage the context and the message of the product?", "What impressions and experiences should the user receive from the product?", "How long should each use case last?" and "How can we support these impressions and experiences with the help of the environment?". These meetings cover about 70% of the environment development (P2).

One participant (P4) summarized the nature of communication with the designers as formal, through weekly meetings, and informal, through phone calls and meetups.

The second phase of the development process is characterized by the search for an environment in the developers' portfolio that contains the best way to visualize the scenarios (P3). According to one participant (P1), the environment should fulfill two roles. The first one is to provide the designers with the right conditions to test the experience facilitated by the functions of the vehicle, as well as to capture problems by experiencing the vehicle under realistic conditions. E.g., to evaluate the experience of driving with a head-up display in low sun, a suitable environment would offer a high proportion of shade, such as a forest. The second is to offer the right level of distraction so as not to influence the outcome of the assessment. E.g., to evaluate the color scheme of the head-up display and its ability to guide on a dynamic road, a suitable environment would include winding streets, as well as flora and fauna that are not so distracting.

Challenges in the Communication: One participant (P4) noted that designers have no standard way of describing to the developers the environment they wish for ("It is very individual, depending on the designer"). Another participant (P5) commented upon the designers' description of the environment by highlighting the difficulty it poses in capturing their vision. A third participant (P2) expressed the impression that designers do not have the most efficient tools or terms available in order to provide developers with an accurate description of the environment they are designing.

This leads to an iterative process, where in each loop small pieces of the environment's progress are shown to the designers and based on their evaluation, the next development step is decided. The same iterative process was also mentioned by other participants (P4: "a ping pong between designers and developers"). Two other participants (P1 and P3) highlighted the difficulties resulting from the increasing number of requirements requested by the designers until the last minute before the presentation.

Three participants (P1, P2 and P4) stated that the development team "lacks the ability to generate an environment in an automated and fast way, the ability to do it without having to spend a big budget and the ability to do it based on standard designer attributes and templates" (P4). This leads to time pressure (P2 and P3) and sometimes, at the end of the preparation phase, to the environment being "counterproductive to the ever-changing new requirements".

Challenges in the Resources: This sub-theme includes statements from all participants. Two participants (P3 and P5) mentioned the lack of environments which leads to a large number of designers' wishes remaining unfulfilled. The fact that the environment has to be built from scratch for each vehicle model (P5: "from the ground up" and P4: "from zero") makes the development a very arduous job that requires a lot of human resources and budget. Two participants (P4 and P5) described the biggest challenge as trying to get the right result without using too many resources (P4: "The greatest challenge is staying in the Time-Costs-Content-Triangle").

Used terms for the Definition of the Environment: This sub-theme contains almost identical statements from all five participants. The designers used the following terms or parameters to describe a virtual environment:

- urban/suburban,
- modern,
- futuristic,
- more/less hilly,
- time of day,
- real locations as a reference (e.g., a coastal route like in California),
- vehicle design characteristics, which should be mirrored by the environment (e.g., a vehicle with a country character should be showcased on a country road),
- the driving maneuvers (intersection type, crosswalk, traffic lights, single- or multi-lane) and
- city names (e.g., New York).

3.1.3 Collaboration between Designers and Developers

In this section, we present the results of the observation we conducted during a series of meetings between the designers and the developers in the context of a specific project.

Participants and Procedure: In the first two meetings, four automotive UI designers (1 male and 3 females; ages ranging from 28 to 47, mean 37) and two developers (2 males; ages ranging from 25 to 36, mean 30.5) met to discuss the requirements for the virtual environment for a particular UI design presentation and to evaluate the overall generated experience. From the third meeting onwards, the number of designers was reduced to one. We only observed the meetings, by sitting next to the developers and taking notes. In case an unknown term or unclear expression was used by the designers and/or the developers, we noted it down and asked about it after the meeting in order not to disturb the communication flow between the participants.

Results: Before the sessions officially began, the designers contacted the developers by email and described the design stories to be presented. Based on the description of the design stories, the developers selected one environment from a list of three that had already been created manually. It was the environment that could potentially meet most of the designers' requirements. These existing environments are the result of both external companies and internal work. Two of them are a realistic representation of two different small municipalities in Bavaria, Germany, Haimhausen and Aschheim, while the third is a fantastic mega-city provided for free by Microsoft. Once the basic environment had been chosen, the first meeting was planned. The four automotive UI designers and the two developers met to discuss the detailed requirements for the virtual environment in a series of sessions. The first meeting took place at the workplace of the developers who sat at a large table with a big screen or moved to a simulator in the same room. At the end of the last meeting, we asked the designers to choose a real location in the world that would be suitable for their design stories. One designer replied that Hollywood would be such a location and the others agreed with the same enthusiasm. Then, they all started discussing the streets at the chosen location, that would be suitable for the different design stories. It was obvious that they used their memories of driving these roads to evaluate their suitability.

The first meeting lasted about one hour. At the beginning, we introduced ourselves and our goal to explain our presence at the meeting. After that, the developers showed the designers the virtual environment in bird view, with one developer guiding them through the different areas with the help of a mouse. At some points during this tour, the designers asked to see areas different than those visible on the screen, and the developer responded by moving to the desired location in the environment. These areas included a winding mountain road, followed by a mountain tunnel and finally ending in a large city. The design stories that had to be replicated in each of these different areas included making and receiving phone calls on the vehicle's middle console and performing different driving maneuvers, including parking maneuvers in different types of parking spaces in the city. The areas were shown by one of the developers in the order of the reproduction of the design stories, which the designers had defined via email. The designers requested changes to the originally planned order of the design stories and asked the designers to create “markers” for each story. These markers are spawning points to allow the user to transfer from the end of one design story to the starting point of the next without losing time driving the distance between them. In addition to the spawning points, the designers wished for a specific 3D text to be added to the markers. Designers also expressed a desire to place certain objects in certain places in the environment, such as parked scooters. These were used in design stories, e.g., so that the driver can use them after parking the vehicle.

For the second session, which also took about an hour, the developer had implemented the markers in the changed order and a simple car driving simulation. The designers were able to drive through the different design stories on the planned route. Each of these stories was scheduled by the designers for a specific amount of time. Therefore, if the designers noticed a difference between the planned and actual duration of the test drive, they requested to change the location of the markers or the driving speed to achieve the correct duration for the design story.

In the third meeting, after the developers had implemented the requested changes, the designers drove through the markers of the design stories again. The meeting lasted about half an hour. A problem arose when the designers drove the car on a cliff road. The slope of the road was so steep that the designer could not see how many degrees the wheel had to be turned to make the steep turn. The result was that the vehicle fell off the cliff. The designer made it clear that this was a critical incident that had to be avoided during the presentation. The designer responded by suggesting a solution via a waypoint system that followed the planned route. Using this system, the vehicle could automatically drive through the design stories and the designers could present without worrying about driving errors due to the unrealistic environmental geometry.

In the fourth session, which also lasted about half an hour, the designers tested the driving simulation one last time after the last big changes had been made. Moreover, they asked for the placement of road signs and other assets in certain locations in the environment, as well as changes to the location of objects, that were already present in the scene. Finally, they requested changes to the 3D texts of the markers.

After these four initial meetings, several more meetings took place in the crucial time immediately before the presentation, where more requirements were requested. This is a common phenomenon in the everyday life of automotive designers and leaves little time for the developers to react.

Data Analysis: The results of the observation were analyzed using Affinity Diagramming. We collected 12 facts about the communication between the designers and the developers, the implementation methods of the environment and the challenges in these two areas. Four sub-themes emerged from the categorization of the facts:

Challenges: The designers had to adapt their original requirements to the three existing environments offered by the developers. According to the developers, this is a common situation as building a new virtual environment requires many resources that are not always available, and the demand for environments is high. Another challenge arose when the designers drastically changed their requirements (i.e., request for another type of area) in the last weeks before the presentation. The reaction of the developers was to refuse any further changes to the environment, as time was too short.

Requirements for Modelling the Environment: The incident in which the vehicle fell off the mountain road was caused by an abrupt and high slope that is part of a fantastic environment. From this, we drew a conclusion about the requirements for the environment, namely ensuring realistic driving conditions and the corresponding terrain geometry. Finally, the designers must be able to place various objects at specific positions in the virtual environment.

Requirements for the Definition of the Environment: The influence of the duration of the design stories on the length of the route, the driving speed and the marker placement, shows its great importance for the design presentation. Therefore, we concluded that the duration of the design stories should be considered as a requirement for the definition of design stories and consequently of their environments. In addition, the frequency of the changes requested in the sequence of the design stories shows the importance of the designer's ability to change it even after the creation of the environment. Furthermore, we observed that moving in bird view helps the designers keep track of the environment and its different areas and offers them the ability to plan further details about the currently displayed area. Moreover, a subtle observation was made and confirmed by the developers that the designers needed the visual cue of the car driving past the buildings on the terrain to get a sense of the driving speed, additionally to the display of the speed value on the car's middle console. Another useful piece of information on this sub-theme is that designers immediately chose a real location that they had experienced themselves in the past. This is an anticipated response, but can be seen as helpful information, when designing an interface for automotive designers, as it enables them to choose a real location to generate a corresponding virtual environment. This information suggests the development of a feature that supports designers in finding the appropriate location in the world that meets their current needs, so that they feel comfortable selecting and testing unknown places.

Collaboration: The remaining observations were grouped in this sub-theme because they show parts of the process of preparing the virtual environment. A large part of this process is the collaboration between the designers and the developers, with the first group describing the environment and the second group understanding the requirements and implementing them. The communication between the two groups includes planning meetings, online meetings to describe the initial concepts and face-to-face meetings to test the implemented design experience. In addition, the designers send certain data (e.g., the markers' text) to the developers in digital form. Based on the developers' statement, some designers also send PowerPoint presentations to show the requirements for the virtual environment and the interaction sequence of the design stories. During the meetings with the developers, the designers still discussed among themselves the details for the design stories and their adaptation to the environment. This costs time but is necessary since the designers did not have the opportunity to get an overview of the environment beforehand.

3.2 Work Models

Based on the results of the studies, the current process of defining and generating the virtual environment can be depicted as follows:

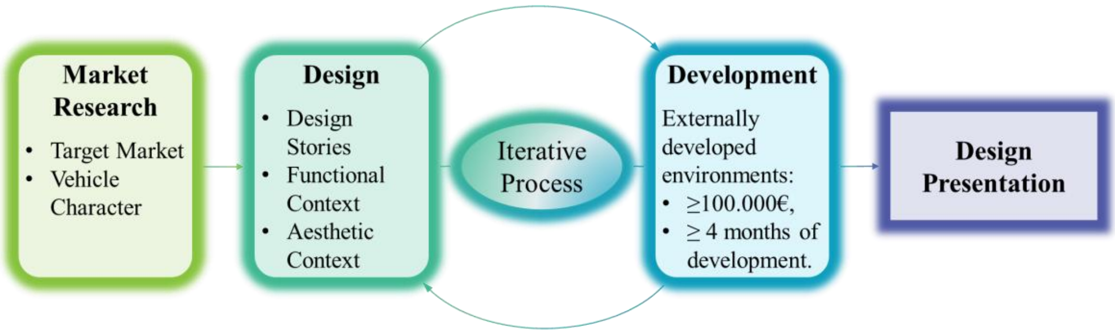


Figure 21: The current process of defining and generating the virtual environment for design experiences based on the results of the interview studies.

The information about the target market and the character of the next design is passed on to the designers by the market research department. They first conceptualize the new experiences that fit the newly acquired information and design accordingly. To evaluate and present the new design, they start a trial-and-error process to find the scenarios to be experienced with the new design and the appropriate context. During this process, they contact the developers and gradually share their requirements with them as they update them. The result is a series of feedback loops in which the design stories are tested in one of three pre-existing environments or, if they cannot be transformed into a more suitable version, an external company is hired to model a virtual environment. The resources required for the last option are at least 100.000€ and four to six months of development time. These feedback loops continue until the date of the design presentation.

3.3 Design Goals

Based on our findings from the interview and observation studies, we formulated 5 design goals. These were taken into account during the entire implementation of this work to support automotive designers in creating their own design experience context.

Design Goal 1: The context of a design experience plays a crucial role in its success as a prototype. Therefore, the first and most important goal is to create ***the right design experience context***. On the one hand, the environment as a modelled result and its conditions as a simulation should provide the right context for designers to visualize their design stories from a functional point of view (e.g., suitable streets and intersections). On the other hand, it should provide the right amount of distraction from the aesthetic design elements of the vehicle (e.g., colors, variety of building façades) by allowing the right level of variety and achieve the right atmosphere. In addition, realistic driving conditions are essential for designers, as the prototyping experience should allow them to test their product and discover complications they could not otherwise foresee. These are the necessary conditions to depict the design stories and to obtain valid results from the evaluation of these Mixed Reality design experiences.

Design Goal 2: The second design goal is to spend ***a minimum amount of resources*** while creating the right design experience context. Given the many challenges, associated with the development time, the budget and the corresponding results, an automated system that provides the ability to generate multiple environments is an ideal solution for many reasons. First, since the automated system can generate a new environment in a short period of time which cannot be compared to manual creation from scratch, last-minute changes are still possible to implement. Secondly, there would be no external costs, except in the case that scenario-specific smaller assets would have to be developed. To meet the need for realistic terrain and street conditions, this automated system should consist of a series of procedural generators for the terrain, the streets, the buildings and the vegetation that are based on real-world data. Finally, to create the feeling of presence in the real location, as well as the right atmosphere, the generation of buildings (e.g., architectural styles), street elements (e.g., type of lanterns) and natural elements (e.g., type of flora) should focus on the chosen world area.

Design Goal 3: Of course, an automated system saves the day when new requirements come up in the last days before the presentation, but it also needs to be accompanied by ***a suitable User Interface***. This should provide both the right parameters and the right level of creativity. The available options should optimize the capture of all the requirements for creating the basic environment, including terrain geometry, streets, buildings and objects that give the scene a sense of realism and presence (e.g., street lanterns, banks, bins, bus stations). They should also correspond to the most common parameters mentioned by the designers and the developers (e.g., urban/suburban). After the basic options have been selected, the more detailed requirements can be communicated, that depend on the design story and the characteristics of the vehicle.

Design Goal 4: This is a prerequisite for the success of the interface and the automated system mentioned in Design Goals 1 and 3. The ultimate purpose is to reach ***a standard process, that is suitable for designers and developers***. Gathering all this information about the communication between the two groups, as well as their perspective on the process they share to form the design experience, sheds light on various details that we can use as valuable information to develop a process, that is adapted to both groups of participants.

Design Goal 5: Two important factors to consider when developing technologies for automotive designers are ***individuality and scalability***. When defining and generating the environments, designers showed varying degrees of desired independence. On the one hand, some designers were content to let the developers generate the environment after collecting the requirements. On the other hand, other designers wanted to have the power to generate the experience context themselves. Therefore, the new process of experience context development should provide ***an adjustable degree of independence***, that depends on each individual designer. Moreover, there is always a need to create new design stories due to the frequent change of trends in the automotive industry. That leads to the need for new parameters to define and generate suitable environments as contexts for the new design experiences.

Consequently, a scalable list of environment parameters should be considered as a way to create a standard process that still leaves room for creative freedom to shape the design experience context.

3.4 Visioning & Storyboarding

Based on the gathered information, we created the blueprint of a process adapted to designers and developers, which is shown in Figure 23. The process consists of four phases: 1.) the User Input, 2.) the Preparation Phase I, 3.) the Procedural Generation and 4.) the Preparation Phase II. The User Input phase starts with the following methods to choose an environment: a) as an area from the world map, b) from a list of previously saved environments or c) search for selected parameters (e.g., street type, terrain elements) in an initial large area in the world map (probably defined by the target market). In the second phase, the details for each design story are defined, namely the duration, the route, the marker text and the speed. In the third phase, a series of procedural generators generate the environment by creating the terrain and adding the streets, the buildings and the natural elements. In the final phase, the user has the possibility to place objects and change textures, as well as environment conditions (e.g., time of day, traffic amount).

Consequently, the current process in Figure 21 becomes the process shown in Figure 22. In the suggested process, the series of loops is shifted towards the designers. In this case, these loops contain the search for the suitable area in the world. The designers can generate more than one virtual environment and test the route. The rest of the environmental conditions are still communicated to the developers, so that they can gradually prepare and implement them. When they are ready, the designers can inform the developers about the most suitable world area. Once the environment has been combined with the rest of the context, the developers can fine-tune the details in collaboration with the designers in the final feedback loops.

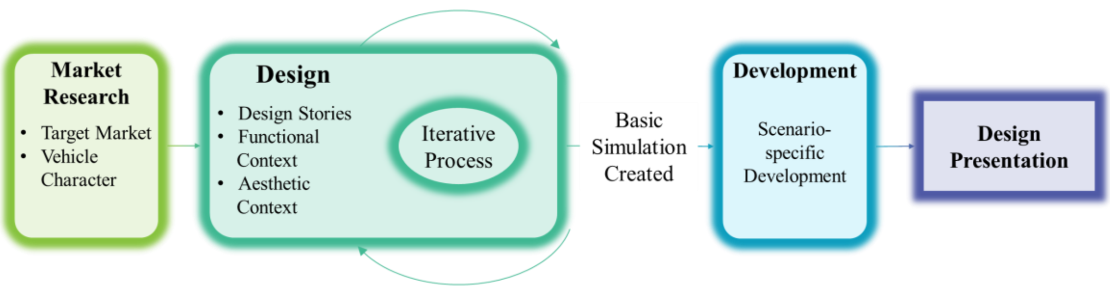


Figure 22: The suggested process of defining and generating the virtual environment for design experiences that contains an automated framework for the procedural environment generation based on map data.

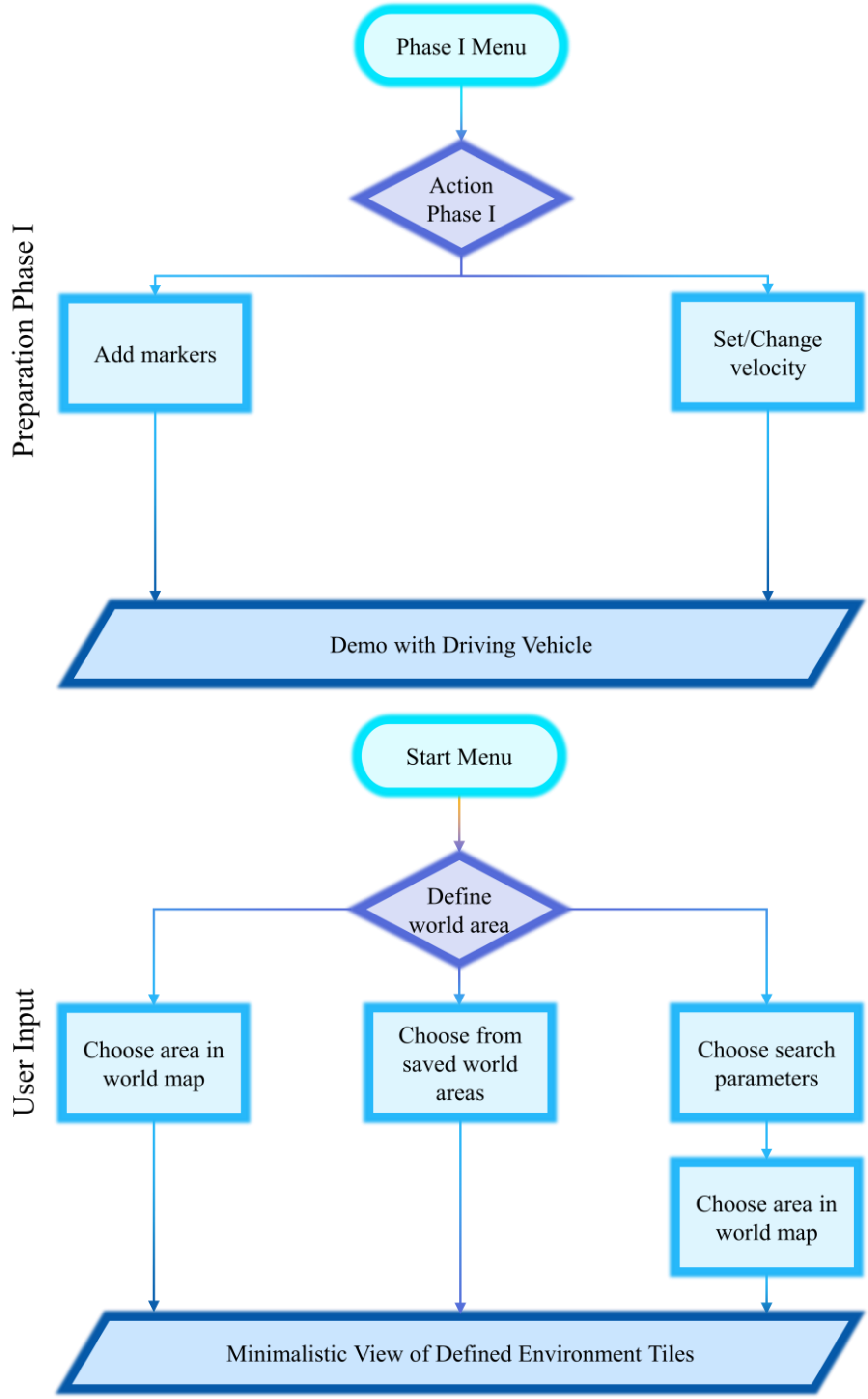


Figure 23: The two first phases of the suggested process of the User Interface to offer designers and developers all essential options to define the environment for the suggested framework (For a legend, see Appendix).

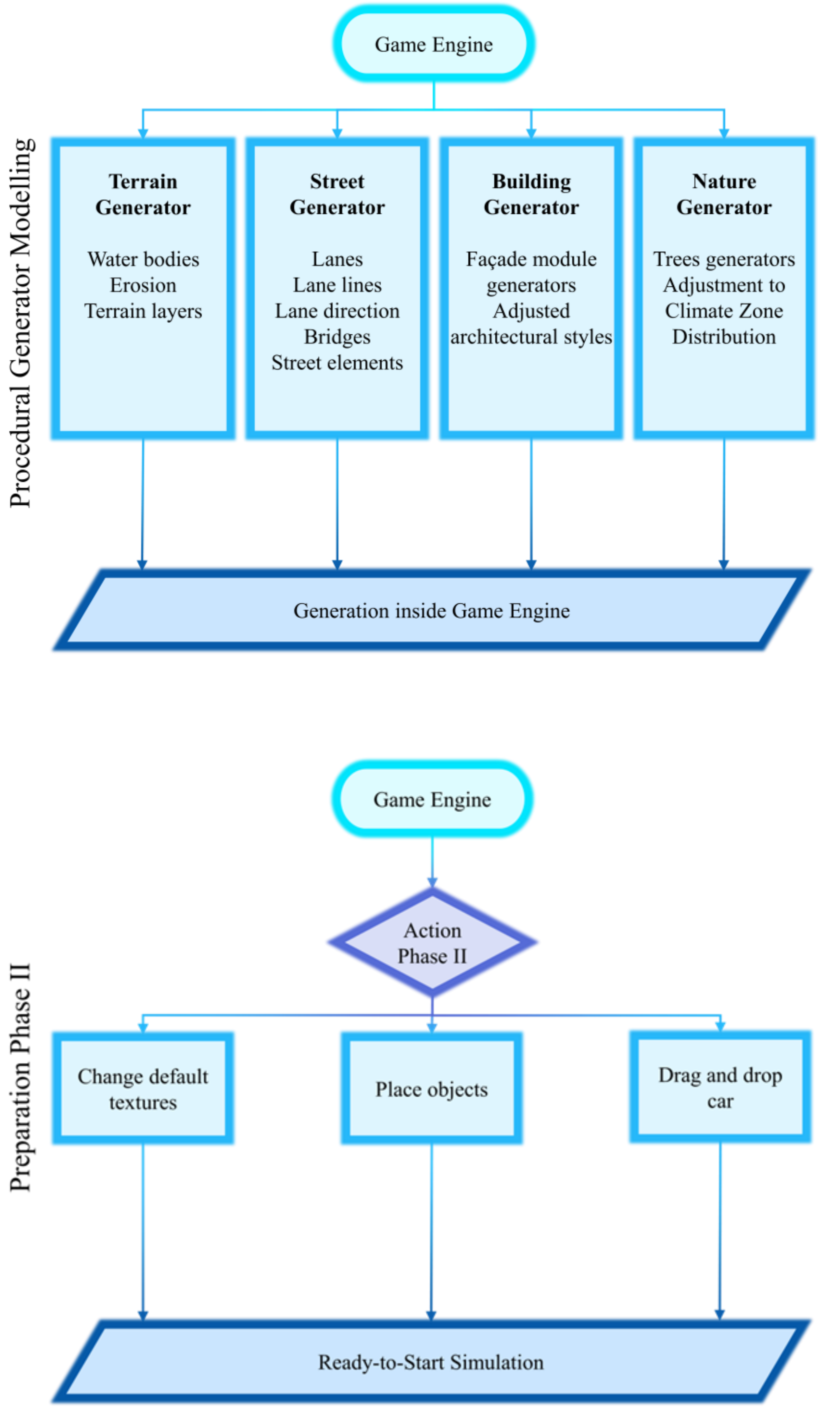


Figure 24: The two last phases of the suggested process of the User Interface to offer designers and developers all essential options to define the environment for the suggested framework (For a legend, see Appendix).

By creating a storyboard, we wanted to consolidate the suggested concept for a new process of defining and generating the virtual environment. After finishing the new design, the designers participate in a series of meetings with other designers and developers. During this time, the design stories start taking shape, as does their context and thus the concept of the suitable environment. The designers select an area in the world, as well as finer options that include the route, duration and vehicle speed. After adjusting the options to other environmental conditions (e.g., weather, time of day), they can activate the generation process of the entire simulation. They are informed with an email that their simulation is waiting for them on the server. When they open it, they can start the automated driving of the car on the selected route.

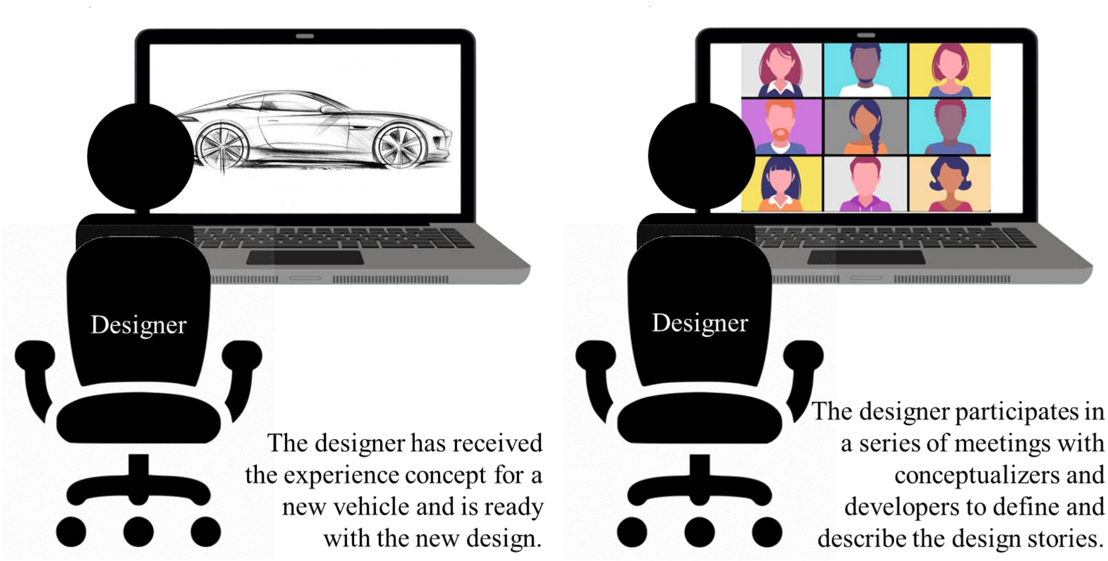


Figure 25: A storyboard depicting a designer going through the phases of the suggested process.

Figure 26: A storyboard depicting a designer going through the phases of the suggested process.

4 Procedural Generation Framework

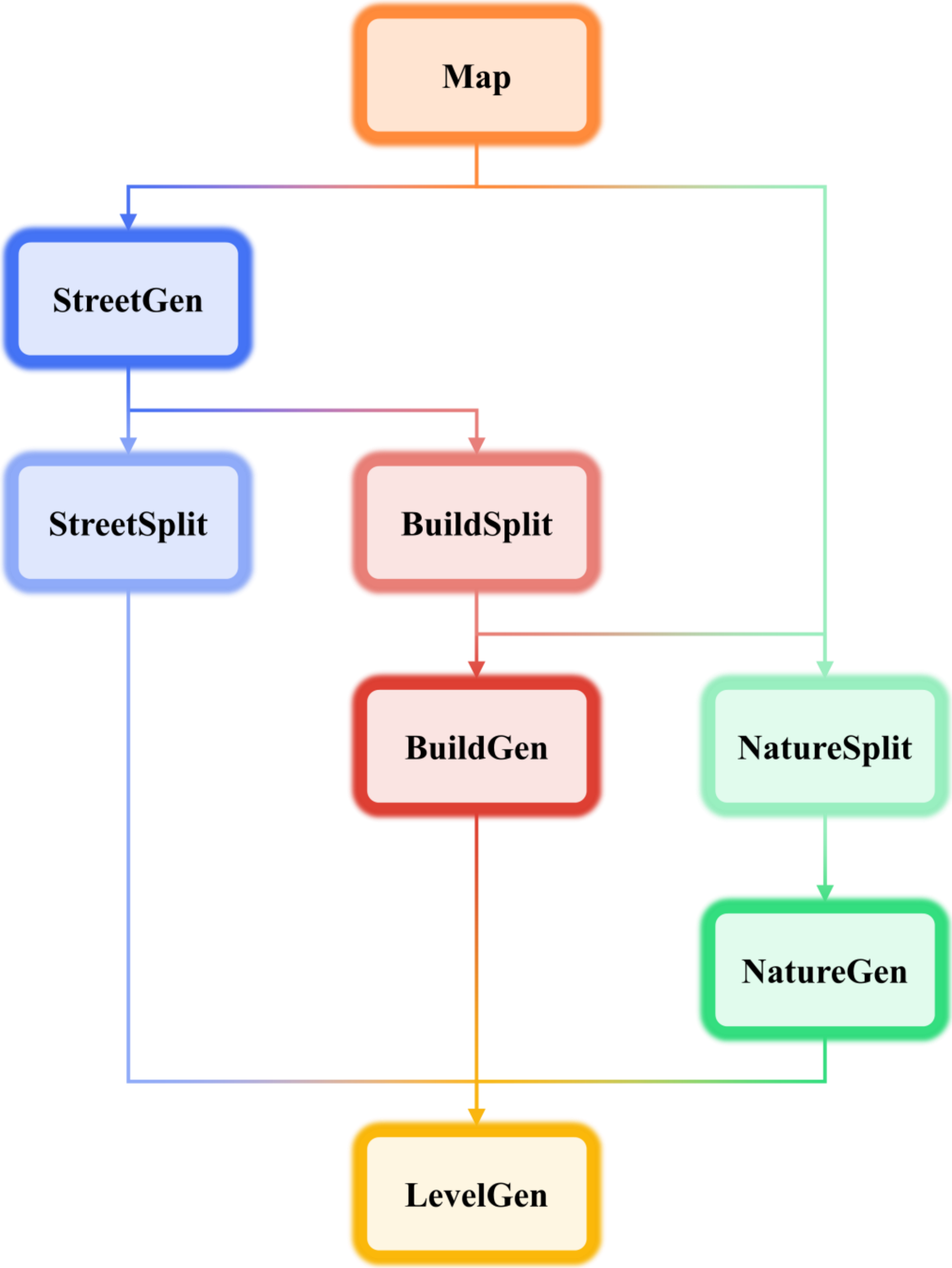


Figure 27: The input and output flow between the procedural generators created for the suggested framework.

This Chapter partly contains the topics covered in previous publications [269-271].

The proposed framework consists of a series of procedural generators. First, the network within the *Map* generator collects the necessary data for modelling the important geometries and distributes it to the generators, *StreetGen* (Chapter 4.3), *BuildSplit* (Chapter 4.6) and *NatureSplit* (Chapter 4.6). Inside *StreetGen*, the terrain is edited, and the street geometry is modelled. The first output is then processed further inside *BuildSplit*, while the second output is divided into levels inside *StreetSplit* (chapter 4.6) to leverage the potential of level streaming. Inside *BuildSplit*, the OSM information about the buildings is also divided into levels, before being transferred into *BuildGen* (Chapter 4.4) for modelling the buildings. The OSM data about the vegetation and other natural elements, as well as the three times edited terrain are the inputs for the *NatureSplit* generator. There, the OSM data is split into levels, before it ends up inside the *NatureGen* (chapter 4.5) to serve as the basis for the generation and distribution of the vegetation. Finally, all generated geometries are collected inside *LevelGen* (Chapter 4.7). There, they are combined with the previously defined levels and information is assigned to them which is used for level streaming within the Game Engine.

Houdini by SideFX [272] was the software used to generate the virtual environment as a game engine-neutral solution. Houdini’s tool, *Mapbox*, provides the ability to download the heightfield of any location in the world, combined with the street, building and vegetation data from Open Street Map. Another advantage and reason for choosing this software is the bidirectional connection between Houdini and the two Game Engines, Unity3D and Unreal. With the help of this connection, changes in the Game Engine lead to a re-processing of the Houdini asset and subsequent re-rendering in the Game Engine.

The following procedural framework consists of 50.000 code lines that were written in VEX, a programming language of the Houdini software based on C and C++.

4.1 Map Data

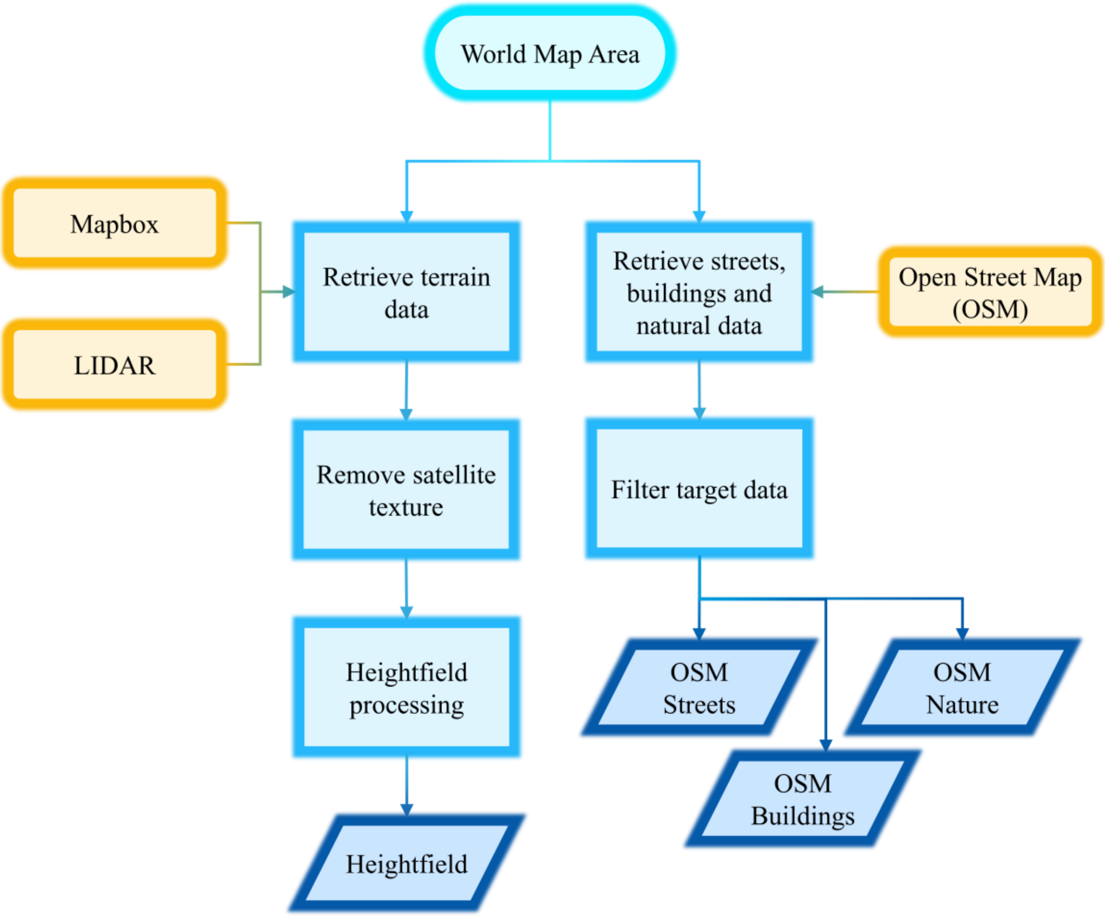


Figure 28: The process used inside the suggested framework for the collection of the map data and their distribution to other generators (For a legend, see Appendix).

The pipeline starts with the first procedural generator, *Map*, where the source data required to model the basic environmental elements (i.e., terrain, streets, buildings, and nature) are retrieved. Mapbox and OpenStreetMap (OSM) were selected as the main sources for the heightfield and the basic environmental elements, respectively. The decision for these two sources was based on the accessibility of their heightfield data and the variety, form, and level of detail of their information about the environmental elements.

Although Mapbox does not provide a high accuracy heightmap, heightmaps can be downloaded for any area of the world. In case a higher accuracy is required, a LIDAR heightfield can be manually searched and found for specific areas of the world on the relevant official websites (e.g., United States Geological Survey (USGS)). However, LIDAR data is not free for every area in the world and the efficiency of finding the correct geographic coordinates depends on the website's system.

OSM is a project that involves users in the creation of a world map. They can edit the world map by adding information based on certain rules. Consequently, the OSM data varies in the level of detail of the information it offers, depending on the area of the map, as well as the category of environmental elements. However, it is more than sufficient for the task at hand, as the main streets are always provided, and the buildings of some residential areas can be recreated with randomly generated footprints along the streets or patterns. In addition, the variety of elements (e.g., row of trees, swimming pools), the type of the provided data (i.e., lines for the streets and footprints for buildings and other elements), as well as the level of detail of information about the provided elements (e.g., lane number, lane direction and turn lanes for the streets) renders the OSM lines an efficient input for creating a design experience context.

After retrieving the data, the pipeline continues with some light processing steps to give the heightfield a first adjustment for the streets and buildings. On the one hand, this step is necessary to prepare the terrain for the adjustment after the respective geometries have been generated. On the other hand, it is important to straighten the topology of the Mapbox heightfield where erroneously heightened areas exist, e.g., due to the influence of tall buildings that are located at these positions in real life. In this first processing step, the primitive lines provided by OSM and representing the streets are extruded in their width by a fixed value and a heightfield mask is defined based on the created street primitives. Additionally, the footprint primitives provided by OSM and representing the buildings are also used to define a mask on the heightfield, which is then expanded by a minimum degree. The masked heightfield areas are then blurred by a representative small value to achieve the straightening effect, but still not change the general shape of the terrain topology. This first processing step marks the end of the *Map* network, which prepares the input data for future use inside the next procedural generators that create the streets, the buildings and the natural elements.

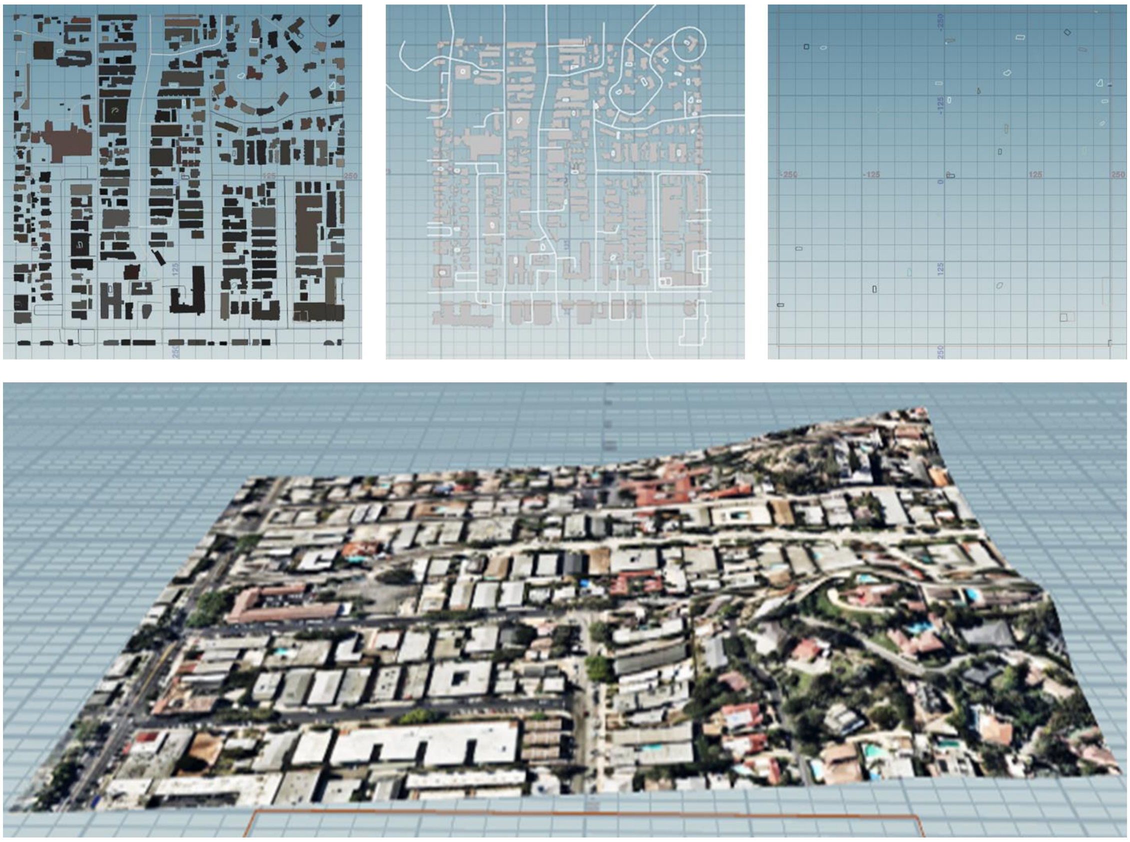


Figure 29: The output geometries of the *Map* generator. Up, left: OSM data used for the street generation. Up, middle: OSM data used for the building generation. Up, right: OSM data used for the vegetation distribution and generation. Down: Terrain edited inside the *Map* network.

4.2 Terrain

The part of the pipeline that contributes to the creation of the terrain is strongly intertwined with the network of the streets and buildings, as the heightfield needs to be adjusted for the placement of both environmental elements. Consequently, the original heightfield, as it comes from the *Map* network, is distributed as input to the *StreetGen* network. There, the heightfield goes through the necessary processing to achieve a good adjustment to the streets and definitions of heightfield masks for the assignment of various materials relevant to the Game Engine we ended up using. The resulting heightfield continuous its editing journey inside the *BuildSplit* network, where the heightfield is prepared for the adjustment to the buildings and masked areas are added to the existing material layers defined inside *StreetGen*.

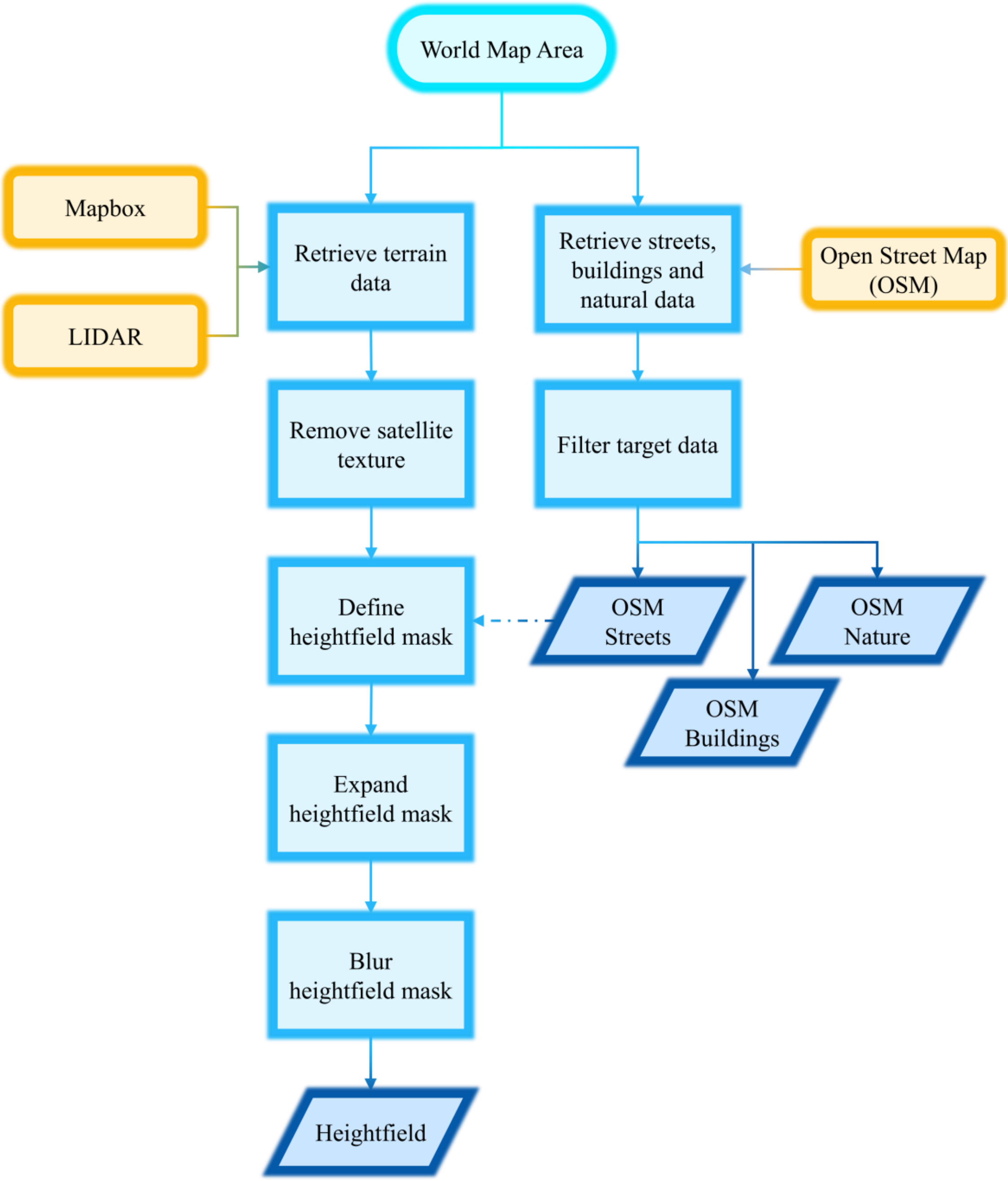


Figure 30: The process used inside *Map* for the collection of the Mapbox or LIDAR terrain and its first editing (For a legend, see Appendix).

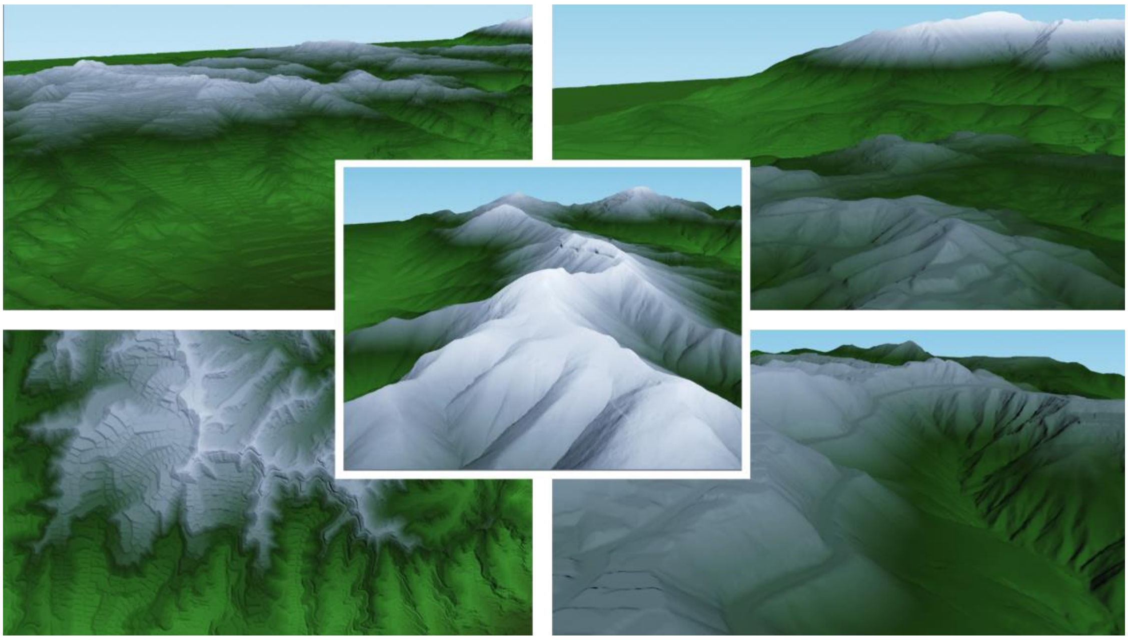


Figure 31: The terrain output of the *Map* generator (the colors were added only for visualization).

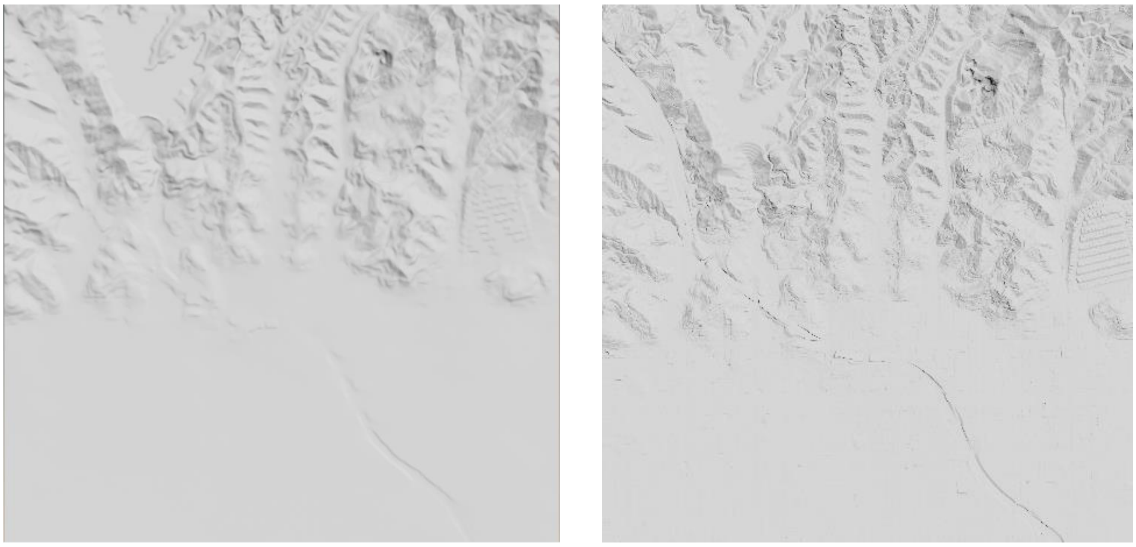


Figure 32: Left: Heightfield of the Hollywood area in Los Angeles, provided by Mapbox. Right: LIDAR heightfield of the same area downloaded from USGS.

Inside *StreetGen*, the heightfield is first edited to form concave areas that can accommodate bodies of water. The information about these areas is contained in the OSM data in the form of primitive lines that represent the outline of the corresponding body of water (e.g., river, lake, reservoir). These outlines are then filled with a single primitive and transferred slightly lower than their original position, which matches the level of the water surface. These areas, now covered by primitives, obtain a higher concavity by projecting them onto the heightfield. This step provides the opportunity to place a particle simulation within the formed areas representing the body of water. In case a high-quality water simulation is not required, the primitives representing the water surface can be used to define a heightfield mask and assign a suitable material.

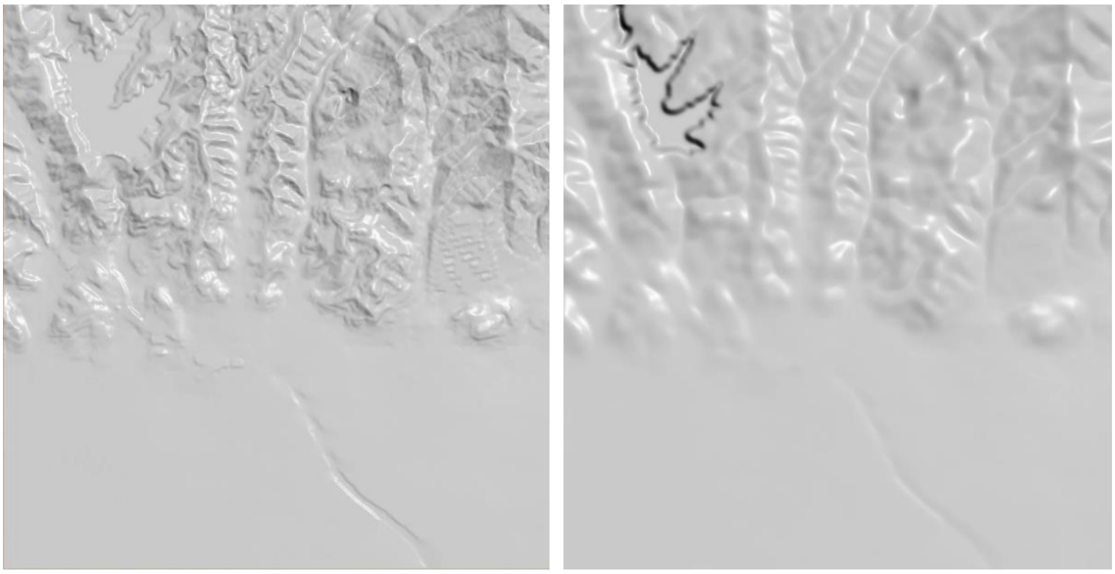


Figure 33: Left: Terrain without areas to accommodate bodies of water. Right: The same terrain with formed areas to accommodate bodies of water.

The next processing step aims to perfectly adjust the terrain to the streets. First, each street line is expanded in width by the average value of 3 (since the width of the street varies between 2.75 and 3.75 meters) multiplied by the number of lanes provided by OSM and increased by a small, fixed value so that the adjusted area extends a little beyond the edges of the street. These new primitives are projected onto the terrain to straighten the areas where the streets are to be placed.

After the projection of this first rough street geometry onto the heightfield, the terrain is not yet perfectly adjusted to the streets. The two types of terrain, Mapbox and LIDAR, require a different degree of adjustment. In the case of the Mapbox terrain, a more drastic method is required that applies Ray Casting of the terrain to the generated streets and transforms the corresponding terrain surfaces. To this end, the heightfield is first transformed into polygons and the polygons corresponding to the streets are removed. Only the terrain polygons remain, which model the features of the topology, that are not to be influenced by the Raycast. Finally, rays are cast from the heightfield polygons onto the street geometry and the terrain polygons that correspond to the features of the topology. The result is that the polygonal terrain takes the shape of the streets, while the shape of the natural elements remains intact. This newly edited terrain consisting of polygons is transformed back into a heightfield by projecting it onto a flat heightfield. In the case of the LIDAR terrain, the heightfield is simply projected onto the streets, with special care taken to achieve the minimum and maximum heights.

Before proceeding with the next steps, the size of the planar heightfield is calculated so that it corresponds to one of the permitted heightfield sizes of the Unreal Engine, which can also accommodate the entire area of the originally selected terrain. To achieve this, the OSM information stored as detail attributes, *map_height* and *map_length*, is used for the calculation.

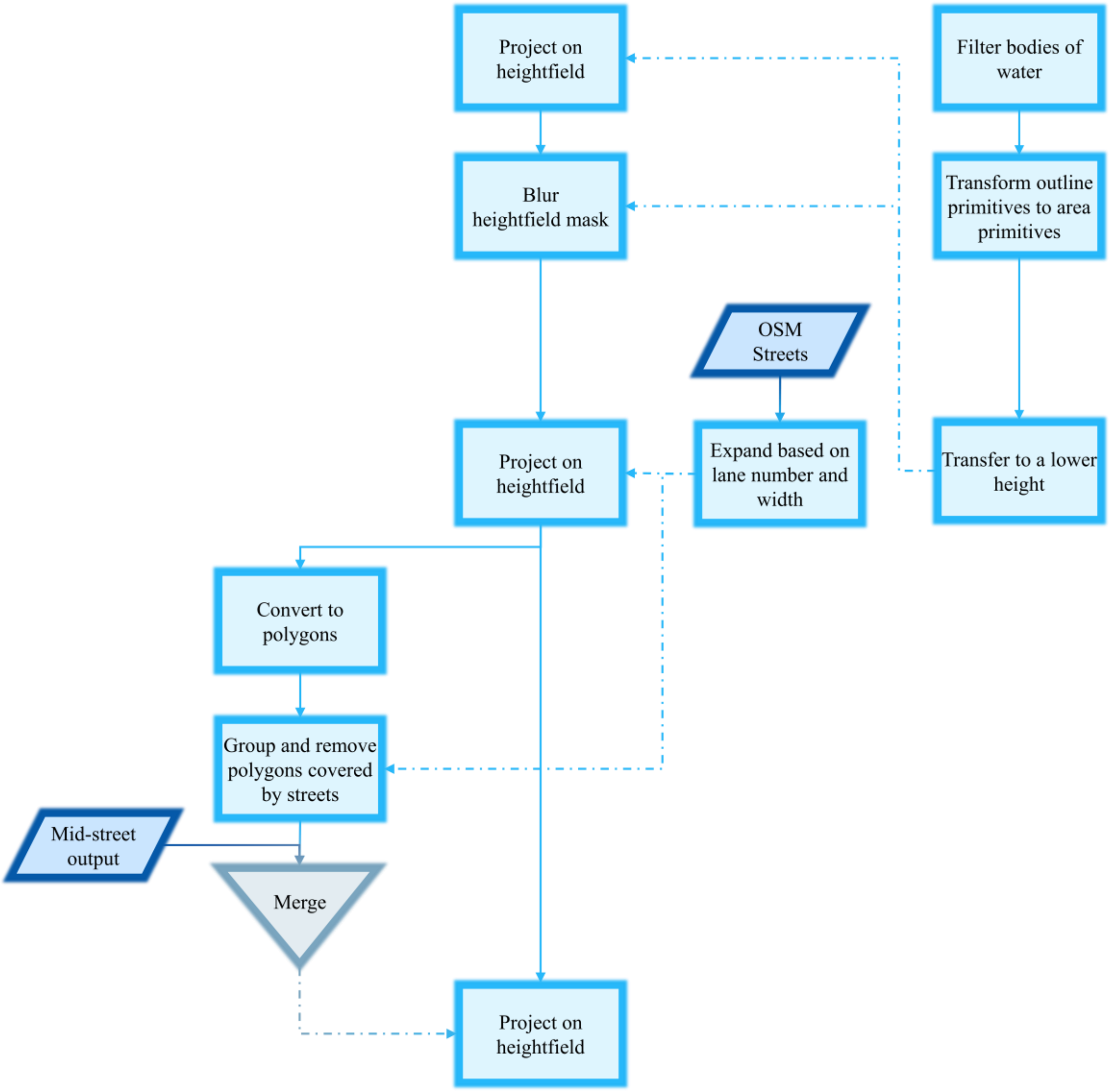


Figure 34: The process used inside *StreetGen* for the adjustment of the terrain on the streets (For a legend, see Appendix).

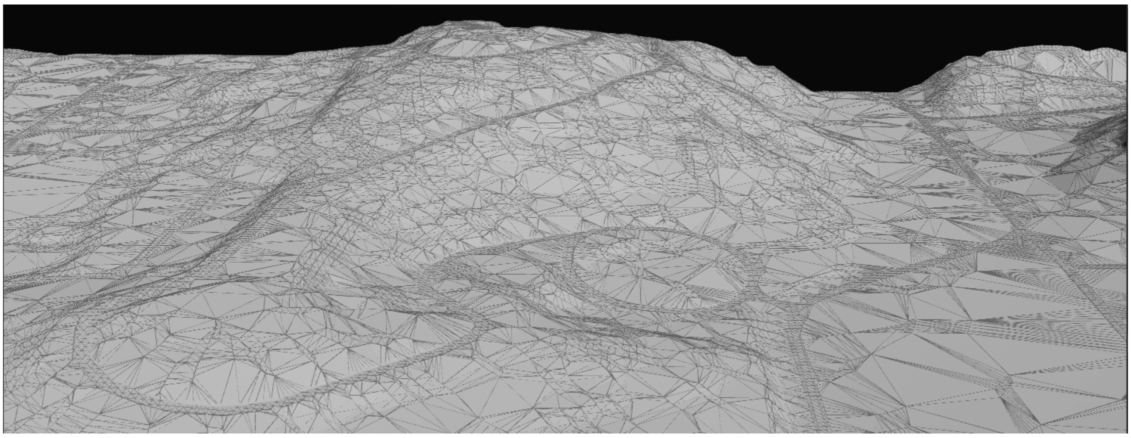


Figure 35: The adjusted terrain to the streets.

The next steps conducted for the terrain inside *StreetGen* aim to define layers and corresponding materials for the Game Engine. The first layer is defined by masking the areas based on slope where streams are usually formed. Using curvature to mask the heightfield results in including only those streamlines, that can be excluded by subtraction from the previous mask. The resulting mask is used to define the hill layer, which is assigned a material for soil. The second layer is defined using the same method as the first, with the only difference being that the corresponding mask has a higher slope range. This results in a terrain where hills are covered with a basic hill layer, while a rock layer is only visible in areas with a particularly steep slope. The third layer represents grass and is masked by inverting the rock layer, subtracting the hill layer after also inverting it and subtracting a mask based on height, after multiplying it by the values provided by a noise mask. The final natural layer represents light grass and is defined by masking the terrain based on occlusion and then multiplying this mask by the values provided by the grass layer. The pavement is also visualized by a suitable material on the terrain and therefore, the layer for the pavement and the streets must also be defined. For the pavement layer, the terrain is masked based on the primary, secondary, tertiary and residential streets. The mask is then expanded and the previous street mask is subtracted. For the street layer, the terrain is masked based on the streets and is named *visibility* layer, so that it functions as a heightfield hole with an invisible material within the Game Engine.

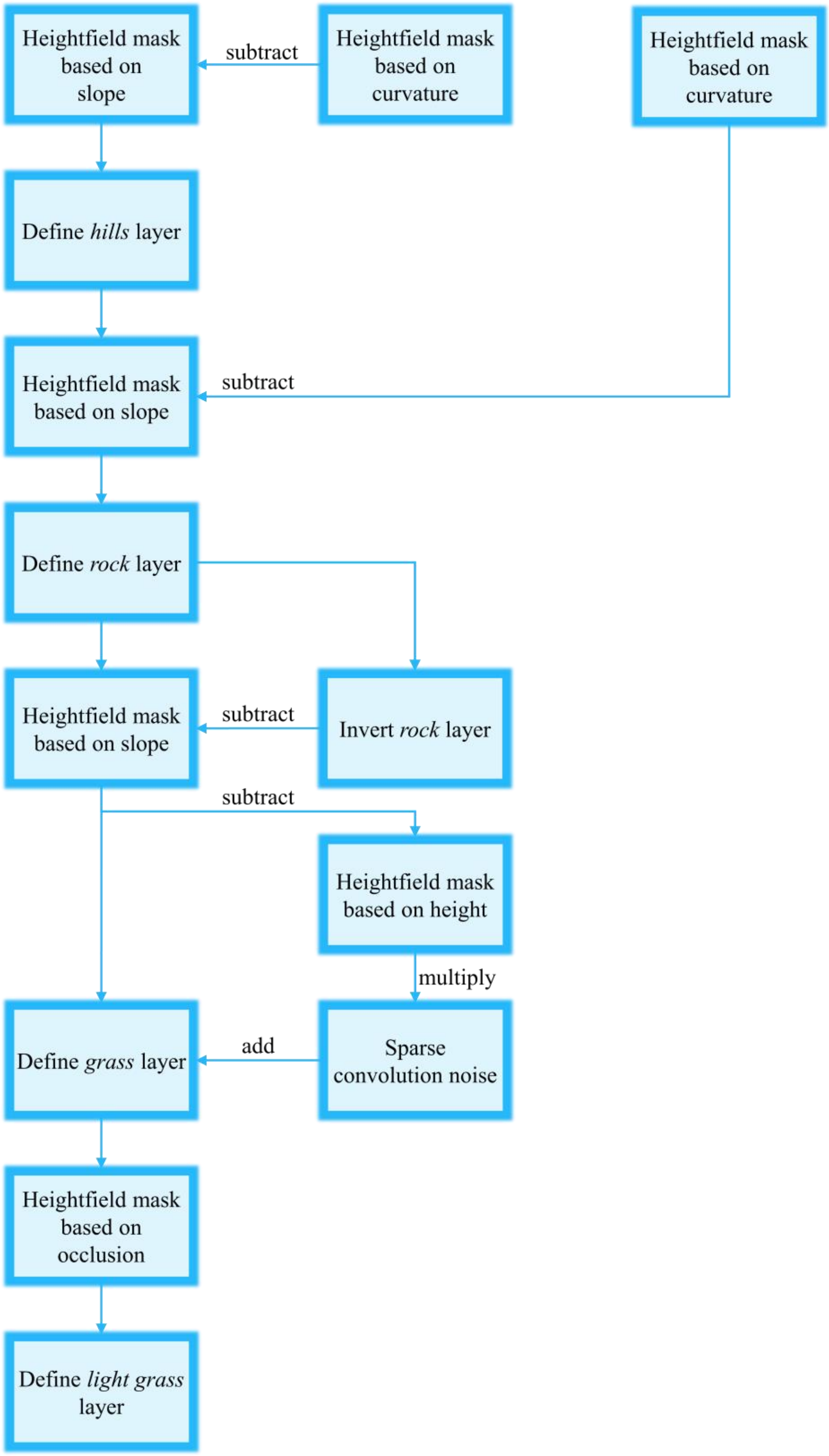


Figure 36: The process used inside *StreetGen* for the definition of terrain layers (For a legend, see Appendix).

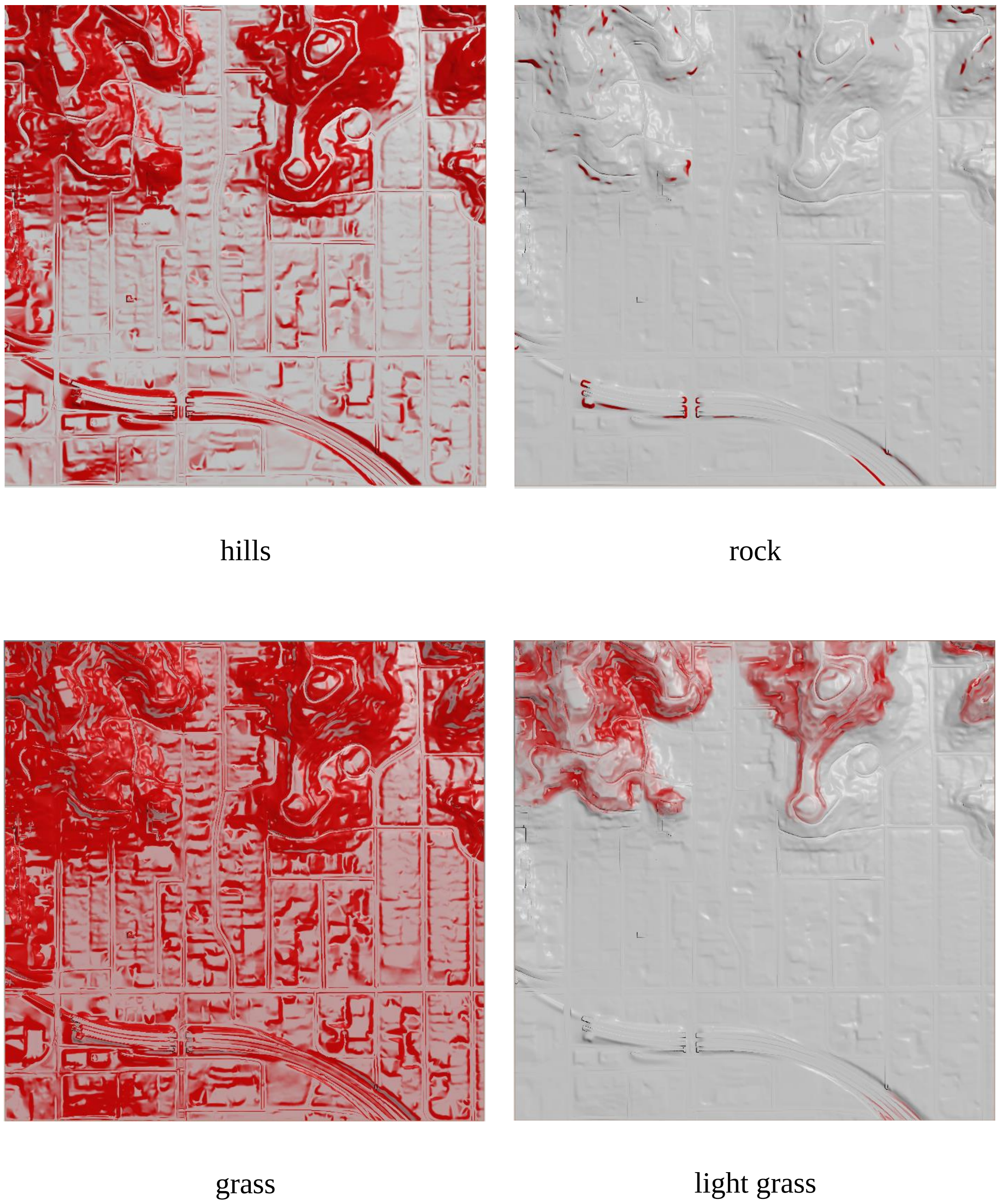


Figure 37: Different heightfield masks applied on the terrain inside *StreetGen*.

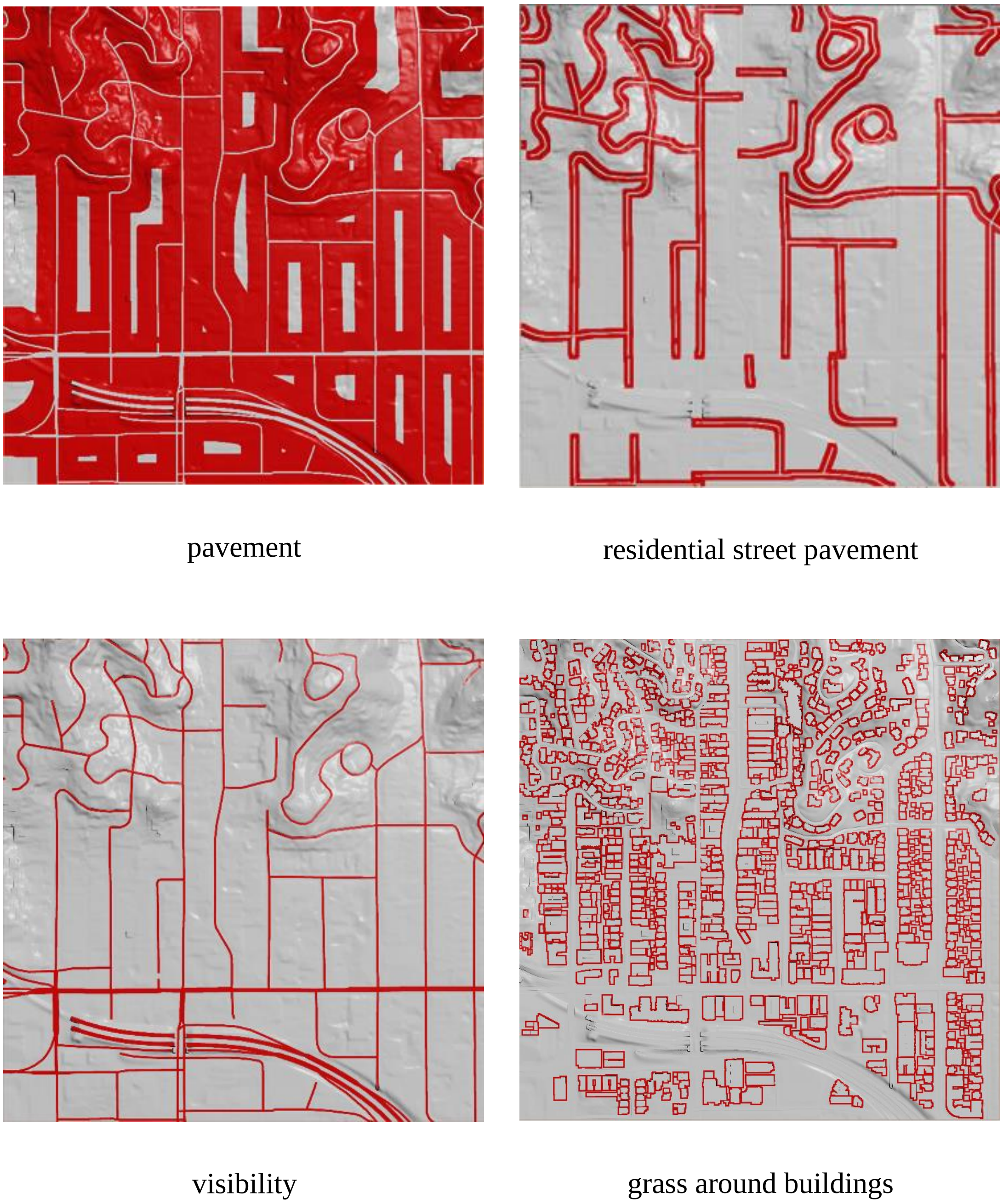


Figure 38: Different heightfield masks applied on the terrain inside *StreetGen*.

Inside *BuildSplit*, the building footprints are projected onto the terrain, thus editing it to adjust to the buildings seamlessly. The footprint primitives are also used to mask the terrain, expand this mask and add it to the grass layer. This step aims to provide the visual impression of grass around a building. The same mask is also used after subtracting the original, non-expanded building footprint mask to blur the terrain around the buildings in case the last editing step caused abrupt topography changes.

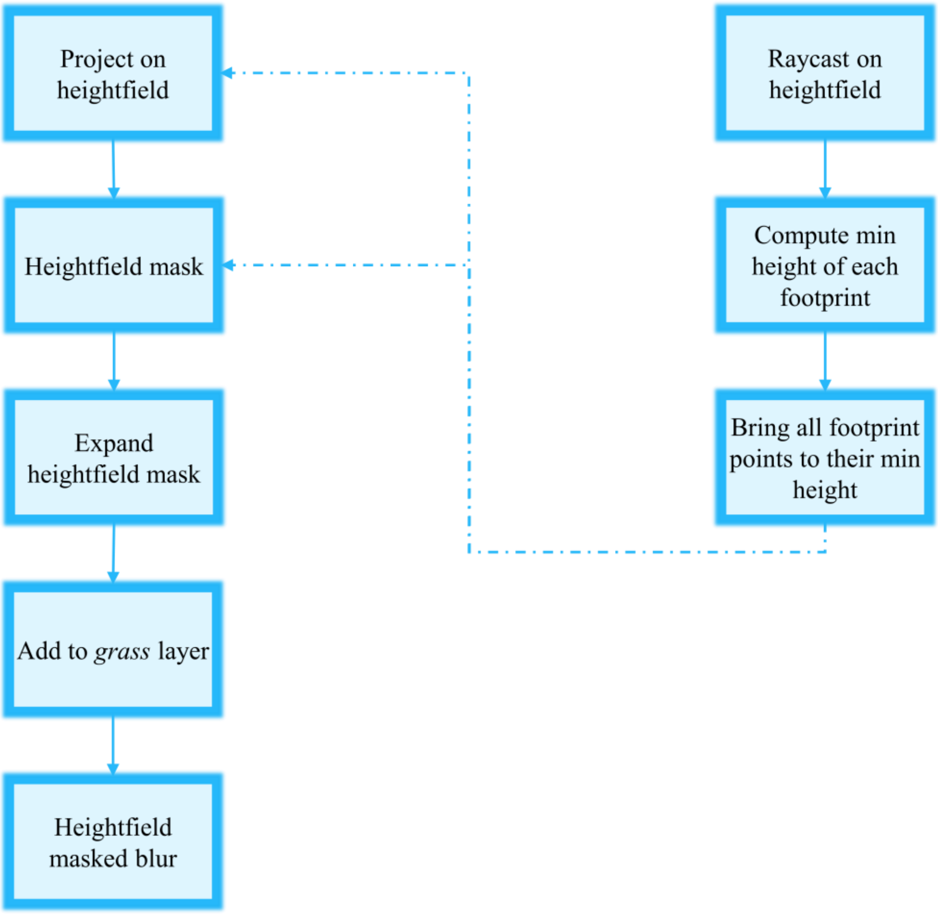


Figure 39: The process used inside *BuildSplit* for the adjustment of the terrain to the building footprints (For a legend, see Appendix).

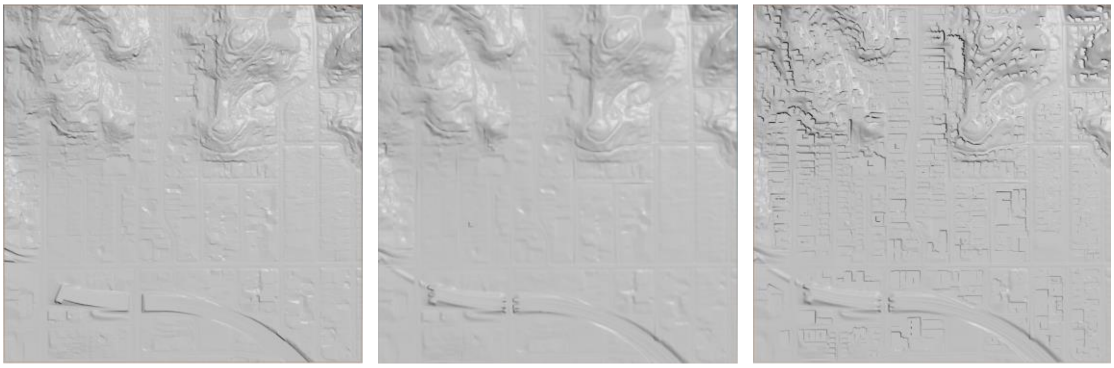


Figure 40: The terrain as the output of the *Map* (left), *StreetGen* (middle) and *BuildSplit* (right) generator.

The final processing step for the terrain occurs inside *BuildGen* and aims to adjust the terrain for the placement of the modelled porches and fence geometries. Applying the same method once more, these geometries are projected onto the terrain.

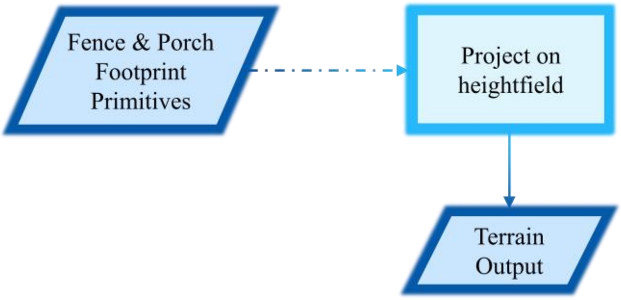


Figure 41: The process used inside *BuildGen* for the adjustment of the terrain on the rest of the generated building parts (For a legend, see Appendix).

Used OSM information		New defined variables	
Attribute name	**Type**	**Attribute name**	**Type**
highway	primitive attribute	name (for the terrain material layer)	primitive attribute
map_height	detail attribute	unreal_material	primitive attribute
map_length	detail attribute		
geometry for streets	street line primitives		
geometry for building footprints	footprint primitives		
geometry for nature	line primitives or points		

Table 1: OSM provided and new defined variables that were used to generate and edit the terrain.

4.3 Streets

In order to generate roads with drivable geometry, as well as to offer environments that reflect the character of real locations, OSM data was chosen as the input data for the generation. The data provided by OSM contains information about the streets, buildings, and nature elements, and thus covers all categories of elements mentioned in the designers' requirements for the environment.

The following steps describe the implementation inside *StreetGen*:

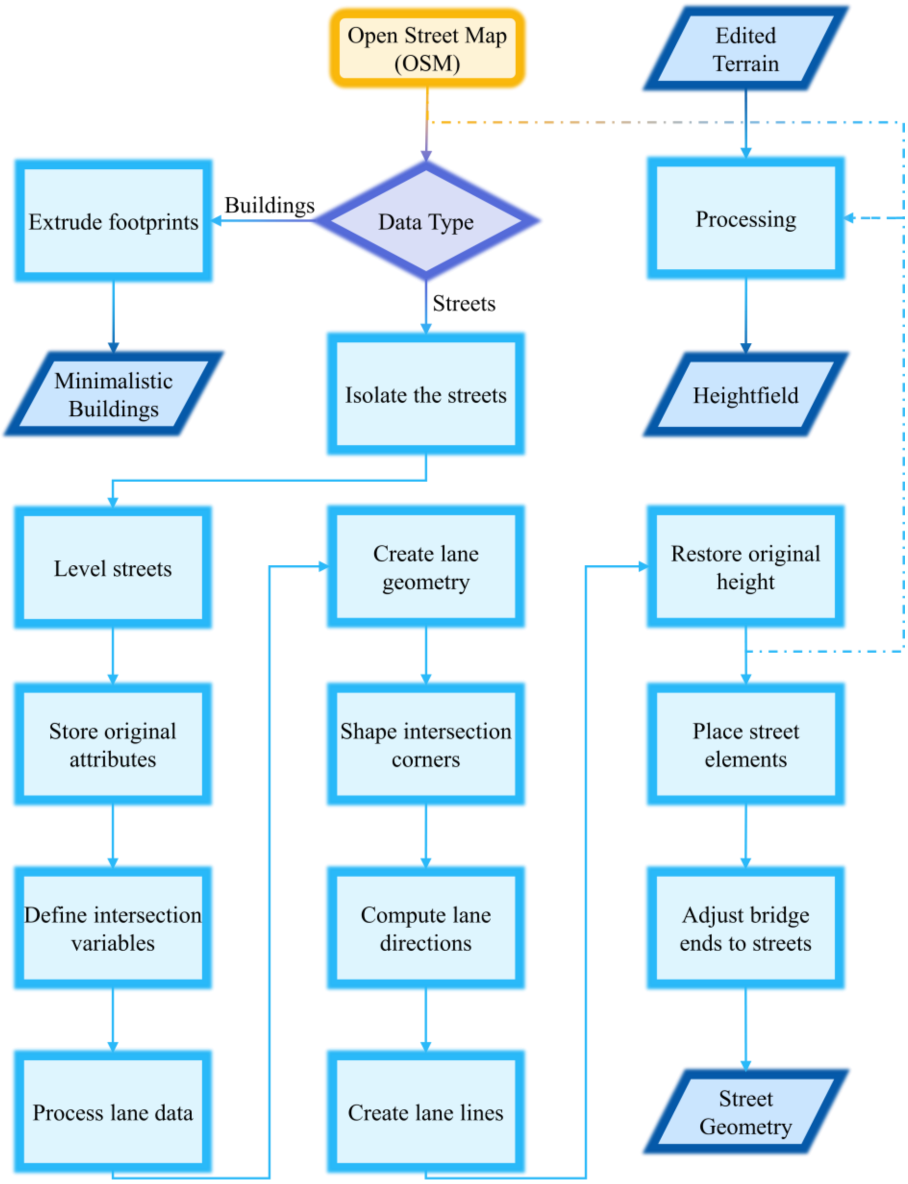


Figure 42: The process used inside the suggested framework to generate the street geometry (For a legend, see Appendix).

4.3.1 Processing the Input

The first step in the designed procedural generation process includes downloading the *Mapbox* heightfield and the corresponding OSM data using the *Mapbox* tool. The downloaded OSM data contains a middle line for each street, which is divided by points into segments with the same primitive attributes (e.g., street ID, street name, number of lanes and speed limit).

Inside Houdini, the imported OSM data receives a unique number for each point and primitive it contains. A recurring action that is performed several times during all further processing steps is the definition of attributes and groups. These are helpful pieces of information that enable the location of points and primitives with certain properties in a large geometry. The groups are actual attributes, that indicate whether a point or primitive belongs to the group (value: 1) or not (value: 0).

The input data already contains some repeating patterns, (e.g., intersections are formed when two or more lines cross each other). The repeating patterns need to be maintained throughout the processing of the geometry for the procedural algorithms to be successful. Therefore, the next major step aims to create an initial basic geometry, which allows the definition of variables that help with the location of specific points, primitives, edges, or vertices among the large set of objects present in the geometry. Since the target geometry consists of streets, these repeating patterns should ideally exist for each intersection and for each lane in relation to other lanes of the same or different segments with corresponding attributes.

Isolating the streets: To prepare the OSM curves for the creation of such a geometry, the first step is to filter out all scene elements except the streets. Depending on the selected area of the world, the current geometry may contain streets and/or bridges. The attribute offered by OSM that defines at which level above the ground an element exists is called *layer*. The layer has a value of zero for streets that are on the ground and a value higher than zero for streets that are above the ground.

Figure 43:Left: Original OSM lines representing the streets. Middle: The street lines after the first filter. Right: The street lines after the second filter.

Finding the bridge primitives: The next step requires branching from the main architecture. Before isolating the streets with a *layer* value of zero (streets on the ground) for further processing, important attributes are defined by fusing the lines representing the streets on the ground with the lines belonging to the bridges. The common points created by this fusion, connect the streets of the two different *layers* (*layer* zero and one). These points are grouped together to form the corresponding attribute. An additional attribute is defined by assigning the number of the above-ground primitives (bridges) to the corresponding ground primitives (the streets connected to them). The information contained in these two attributes provides support for the next processing steps e.g., by connecting the ends of the ground streets with the corresponding above-ground streets. As this is only a prototype, all layers above ground and higher than one are not considered or implemented.

Leveling of streets on the ground: The next step requires a return to the main branch of the architecture, where streets of all layers are present, and requires us to set aside every element that exists above the ground. After isolating the streets on the ground, the same y-component (height axis) is assigned to the remaining points, thus aligning them at the height of the lowest point in the geometry. From this point on, the third axis no longer plays a role, except at the end when the streets are raised to their original height.

Storing the original primitive numbers and their sequence of points (for later use in the calculation of lane directions): Next, it is necessary to fuse all existing points and connect all lines, as each original line has its own end points, which complicates further calculations (e.g., finding neighbor points to identify the intersection center points). The end points of each OSM line are assigned a number that indicates the direction of the corresponding street segment in ascending order. This direction corresponds to the *forward* direction, which is simultaneously a variable offered by OSM. It indicates the number of lanes that have the same direction as indicated by the ascending order of the end points, while the remaining lanes have the opposite or *backward* direction. If the points were fused, the number assigned to the end points of these lines would change. In order to keep the information and use it in the processed geometry for the later calculation of the direction of each street segment, the appropriate attributes should be defined at this exact point of the architecture. The first attribute (*originalPrimNum*) contains the number of each line primitive, and it is created to preserve information about the original constellation of the geometry. The second attribute (*originalPrimpointSeq*) defines the sequence of all points belonging to each line primitive. The start and end points of this sequence can be arbitrary.

Defining the intersection points: Two of the most important attributes that have to be defined are the group of intersection points (*interPoint*s) and the group of associated primitives (*interPrims*). After fusing all points and connecting all primitives, the number of neighboring points of each point is examined. In the case that a point possesses more than two neighboring points, the point belongs to the group of intersection points. The group of the primitives belonging to an intersection is determined by examining their points. If at least one of them belongs to the newly defined group of intersection points, the corresponding primitive is assigned to the group of intersection primitives. Furthermore, additional points are added to each primitive to maximize the number of sample points for the later adjustment of the streets to the terrain.

Correcting the lane numbers: The next steps are necessary to correct some of the attributes originating from OSM. Such attributes, which can lead to the generation of incorrect lane geometry, are those that specify the number of lanes (*lanes*), as well as their number in the forward (*lanes_forward*) and the backward direction (*lanes_backward*). Depending on the map area and the quality of the corresponding OSM attribute for the number of lanes, the following cases may arise:

(1) *Streets with zero lanes*: If not for all, then for most of the streets with one or even two lanes that do not have line markings, the value of the OSM attribute for the number of lanes is zero. An efficient way to assign a realistic value to this attribute, is to examine the type of street and the number of lanes of the neighboring segments and set an appropriate attribute value. Since the exact number of lanes is not important for the current purpose, but the driver's perception of the type of street they are driving on, the number of lanes is set to two for all streets with zero lanes and of type "residential" or "motorway_link" (*highway*).

(2) *Streets without parking lanes:* According to OSM, many streets have a smaller number of lanes than their actual lane number, because the corresponding attribute does not always contain the number of the parking lanes or because the information is inaccurate. However, parking lanes are of great importance for the task at hand because they contain parking spaces which are featured in many design stories. Furthermore, this inaccurate information leads to an unnatural overall appearance of the road network in certain street sections. A simple and efficient way to mitigate this effect is to collect the number of lanes of all street segments with the same street name and type, calculate their maximum value and find the difference between the number of lanes of each street segment and the maximum value. If this difference is greater than or equal to two, two more lanes are added to the corresponding number of lanes. If the difference is less than two, the corresponding number of lanes is increased by a number equal to the difference. Two lanes were chosen as suitable because a road can have a maximum of one parking lane on each side.

(3) *Narrow street segment between two wider ones:* According to OSM, road segments that contain lanes with both directions and where one lane switches from one direction to the other, contain one lane less than the actual lane number. This leads to the number

of lanes offered by OSM data being smaller than the actual width of the street. The same effect can also occur due to the mentioned changes in the number of lanes. Therefore, all street segments with fewer lanes than those of both their neighboring segments receive the same number of lanes, provided that the number of both neighboring lanes is the same.

In addition to these changes, some measures are taken to improve the overall appearance of the road network, as well as to simplify the later steps. The primitives belonging to the streets on ground which must be connected to the corresponding primitives belonging to the streets above ground are given the same number of lanes if that is not already the case. Furthermore, if a street segment has a neighbor with a right turn lane, it will receive one lane less to give the impression that a lane has been added on the right side of the last street segment before the intersection. Finally, as a safety measure, the difference between the number of lanes of each primitive and that of their neighboring primitives is calculated. Depending on the result, the number of lanes is adjusted to achieve a difference of two lanes. Consequently, only two lanes can be added or removed from one street segment to the next, ensuring a smooth change in street width.

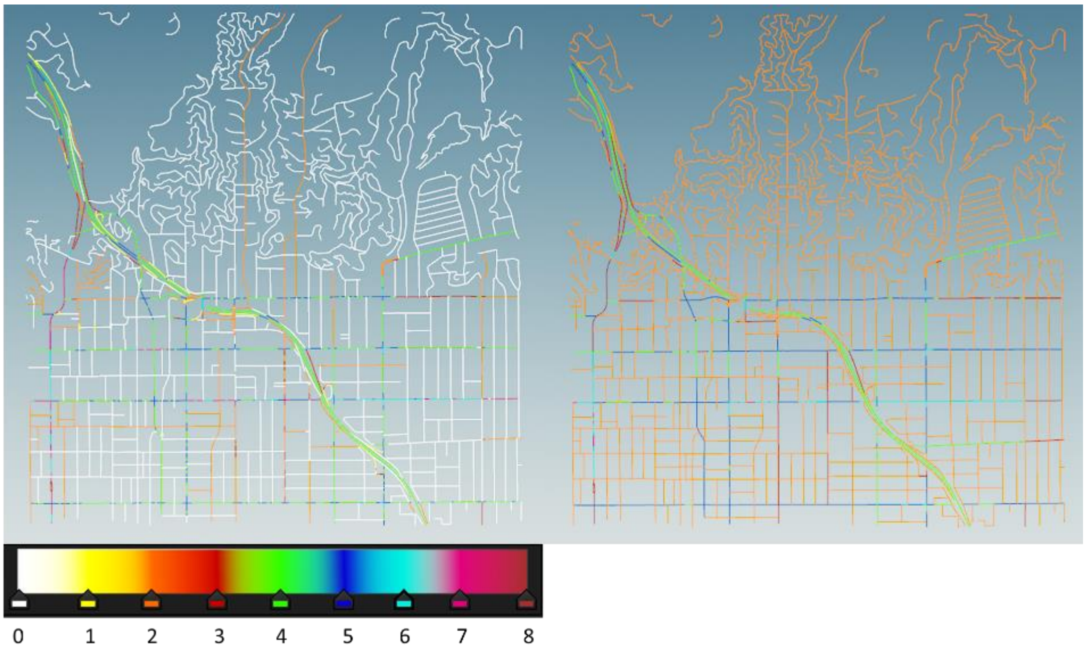


Figure 44: The street lines with colors which represent the number of lanes. Left: The colors represent the original number of lanes. Right: The colors represent the adjusted number of lanes.

Used OSM information		New defined variables	
Attribute name	**Type**	**Attribute name**	**Type**
highway	primitive attribute	originalPrimNum	primitive attribute
layer	primitive attribute	originalPrimpointSeq	primitive attribute
lanes	primitive attribute	interPoints	point group
lanes_forward	primitive attribute	interPrims	primitive group
lanes_backward	primitive attribute		
highway	primitive attribute		

Table 2: OSM provided and new defined variables that were used to process the input for the street geometry.

4.3.2 Creating the Lane Geometry

Using Houdini's *Polyexpand2D* tool, parallel lines are drawn from each given middle line. The parameters *offset* and *division* have to be specified to determine the shape of these parallel lines. The distance between the parallel lines depends on the *offset* parameter, while the *division* parameter determines the number of parallel lines to be created on each side of the middle line. The great advantage of this tool is not only the creation of parallel lines for even the most complicated intersections and the corresponding angles between the streets, but also the anatomy of the generated geometry, which contains repeating patterns for elements of the same type (e.g., intersection). In addition, every vertex of the generated geometry is assigned a new attribute (*vdist*) that plays a crucial role in the next processing steps. This attribute describes the distance of each vertex from the corresponding middle line and is converted into a point attribute (*pdist*) in the next step for easier use.

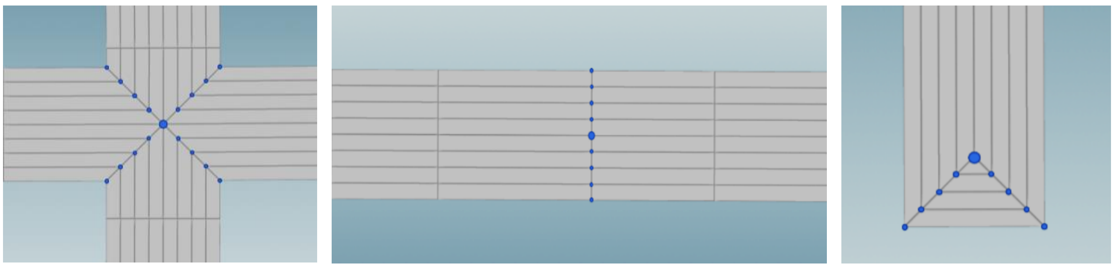


Figure 45: The geometry patterns for the intersection, the division of lanes with different attributes and the street ends.

The number of parallel lines that need to be created for each middle line depends on the number of lanes of each street segment. Unfortunately, the *PolyExpand2D* tool of Houdini accepts only one value for the "divisions" and adds it on each side of the middle line. Of the options tested, the most efficient is to set the "divisions" parameter to half of the maximum number of lanes that exist in the selected world area. With this solution, only one geometry element needs to be corrected, namely the number of lanes for all street segments that have fewer lanes than the maximum number of lanes in the geometry.

The excessive lane primitives are found in the geometry, then grouped and removed with the *Blast* node. Consequently, the point and primitive numbers change, rendering some of the important attributes invalid. After recalculating them, they are used to calculate the angle between each pair of streets at each intersection (*interAngles*). This enables the next processing step.

Important attributes that are defined in this step include the group of intersection points (*interPointsX*) that create the X-formation, shown in the left image of Figure 45, their corresponding intersection center point (*interPointZero*) and the middle point of each vertical line stored in its points (*middlePoint*).

New defined variables	
Attribute name	**Type**
pdist	point attribute
interAngles	detail attribute
interPointsX	point group
interPointZ	point attribute
middlePoint	point attribute

Table 3: New defined variables that were used to create the main lane geometry.

4.3.3 Correcting the Intersections

After removing the excessive lane primitives to obtain the right number of lanes and due to the original geometry, some of the intersection primitives are no longer connected to their intersection. Consequently, the lanes close to an intersection are interrupted by holes. To cover these holes and obtain a continuous geometry, a ray is cast from the outer point of the disconnected lane towards the intersection (Figure 46).

Creating the primitives that cover the holes is accompanied by its own difficulties, as each hole geometry has a different number of points and a different shape. After finding the points that belong to the new primitive (*newInterCorners*), line primitives are created to connect them and form the outline of the covering primitives. The line primitives are grouped and distributed as input to the *Polyfill* node. With its help, new primitives are created to cover the holes at the intersections.

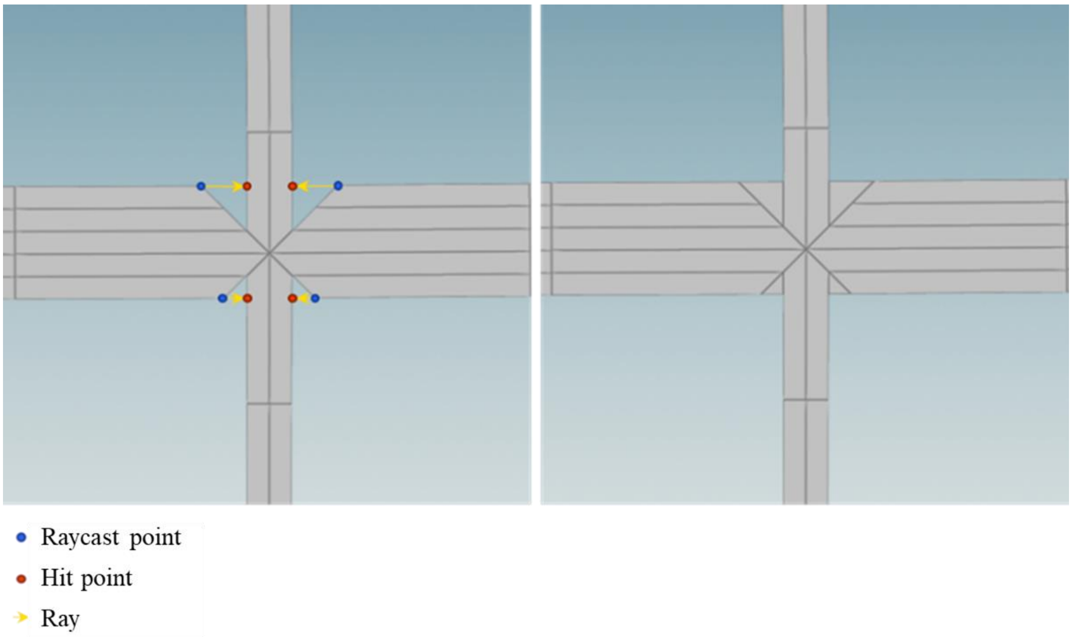


Figure 46: Left: An intersection with uncorrected geometry. Right: The same intersection with corrected geometry.

In further processing steps, the newly created primitives need to be incorporated into the rest of the geometry. This is particularly important for relocating the streets to their original position. However, the points created by the Raycast are not connected to all of the surrounding primitives. After calculating the correct order of the points in which they appear on the primitive created after the Raycast, a new primitive is created that now contains the old and the new points. Only this method can guarantee the creation of a manifold primitive. The new primitives are generated by collecting the old and new points that should belong to the new primitive and finding their sequence on the new primitive with the help of their UV values. As a result, a continuous geometry that contains all streets with the correct number of lanes for each street segment is achieved.

Further processing is required in order to round the intersection corners. The four corner points are detected for each intersection. Additional points are found that include their neighboring points on the outer side of the streets by leveraging variables that have been already found (e.g., *pdist*). A series of points is then created that forms an arc between the two neighboring points of each intersection corner. The radius of the arc depends on the angle between the two neighboring points, which was stored in a previously defined attribute (i.e., *interAngles*) or it can be calculated directly. Using the points of the arc, the two neighboring points and the intersection corner, a single primitive is created that is directly connected to the main geometry.

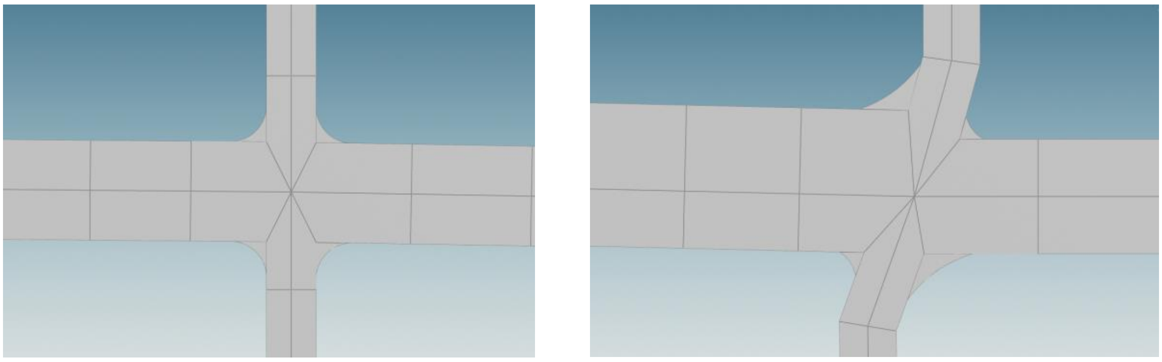


Figure 47: Rounded intersection corners.

New defined variables	
Attribute name	**Type**
newInterCorners	point group
interCornerRC_hitprim	point attribute

Table 4: New defined variables that were used to correct the intersection geometry.

4.3.4 Calculating the Direction of the Lanes

This step does not change the appearance of the geometry but stores important information in the form of the *direction* attribute for each lane. This attribute helps with the correct placement of roadside objects that are visible to the driver (e.g., road signs, traffic lights). Before the final direction of each lane can be calculated, other important attributes must first be defined. The first of these is the group of parking lanes that were added at the beginning of the street generation. These lanes are grouped as parking lanes (*parkLanes*) and the rest are considered as driving lanes. In addition, the difference between the original and the new number of lanes is calculated to determine the number of parking lanes in each street segment. Another important attribute is calculated that indicates the order of each lane from its corresponding middle line. This helps to identify the outer primitives as the parking lanes. By combining these and some of the already stored attributes, as well as several attributes provided by OSM, such as *oneway*, the final direction is calculated by mirroring the original points to the new points. This results in the definition of the incremental order of the point numbers. The direction attribute is stored in the vertices of each point of the lane primitives (*vdir*). The result can be seen in Figure 48.

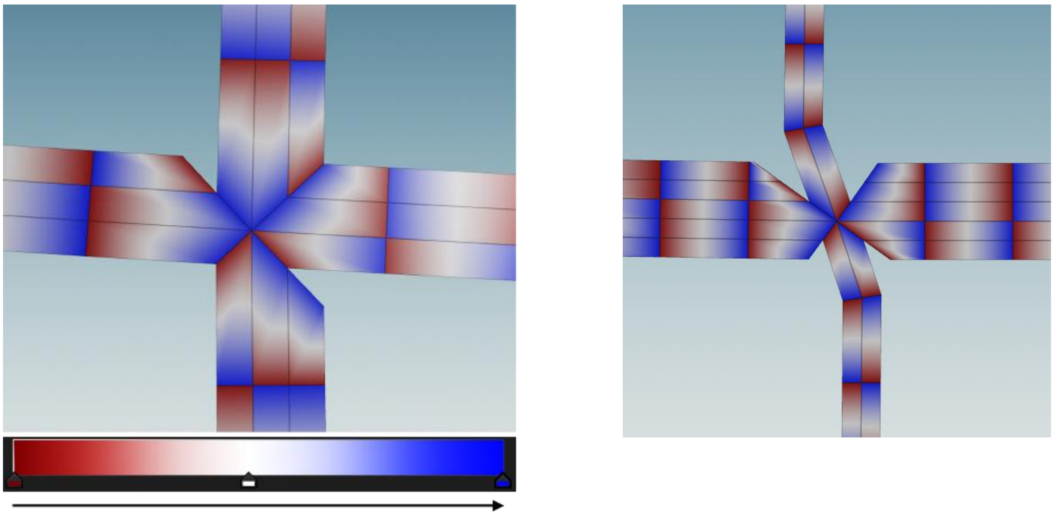


Figure 48: Streets with colors that represent the value of the direction attribute. The lane direction is from the red to the blue color.

Used OSM information		New defined variables	
Attribute name	**Type**	**Attribute name**	**Type**
oneway	primitive attribute	vdir	vertex attribute
lanes	primitive attribute		
lanes_forward	primitive attribute		
lanes_backward	primitive attribute		

Table 5: OSM provided and new defined variables that were used to calculate the lane directions.

4.3.5 Creating the Center, Lane, Stop, Turn Lane and Crossing Lines

The lane and the center lines are created by leveraging the existing parallel lines. Each point on these parallel lines has a vertex for each primitive that shares this point. The direction attribute is examined for all vertices belonging to each point. If all values are equal, the corresponding point lies between two primitives that have the same direction and thus belongs to a lane line. If the values are different, the point belongs to a center line. A group is defined for each type of line, and the points are grouped accordingly. The actual line primitives of these two groups are then formed by connecting the points of the same line type.

To create the curved shape of such a line primitive during a lane change, each point of these line primitives is shifted a certain distance from the others, while remaining on the parallel line. The result is a diagonal line that can be curved with the help of the *Polybevel* tool. As a final step of forming the lane and center lines, they need to be cut exactly at the stop lines. Therefore, this step is performed after the creation of the stop lines.

The challenge in creating the stop lines is to find the exact position of their two points. Real stop lines are not perpendicular to the direction of the street. Instead, they have the angle of the line connecting the two corner points of the intersection that belong to the same street. Therefore, for each intersection, all the corner points are found and stored as a group. Finally, we create the line primitives between the points of the defined group that belong to the same two streets.

The previous step results in lines that have the correct angle but not the correct position, because the stop lines are not directly at the end of the street before it connects with others at the intersection. Therefore, the generated lines must be transferred a certain distance in the direction of each street. The new position of each point belonging to these lines must be carefully calculated, because it must remain on the outer parallel lines, while the direction of the street may change too close to the intersection, e.g., in case of a curved road. For this purpose, each point must be moved in several steps from one point to the next on the same parallel line until the total distance is reached. Now the line primitives have both the correct angle and position but extend from one side of the street to the other. To cut them exactly at the center line, the center lines are isolated in one geometry along with the stop lines. With the help of the *Intersection Stitch* tool, the crossing lines are forced to intersect each other, which leads to the addition of new points. These points are used to find and remove the left side of each stop line primitive.

This process is used for all types of intersection except for T-intersections. In this case, the stop lines of the two streets that are at an angle of 180° to each other must be perpendicular to the direction of each street. This is achieved by calculating the distance required to allow an angle of 90° between the stop line and the direction of the street and using it to move one point of the stop line.

The created stop lines can then be used to edit the other street lines. The lanes and center lines cross the stop line and thus have some unnecessary parts near the intersection. These can be removed by intersecting them with the stop lines and defining the unnecessary parts. Additionally, the stop lines function as an indicator for the position of the turn arrows. These can be placed correctly by adding a middle point on each turn lane behind the stop line and copying the hand-modelled turn arrows to them.

The *highway* variable provided by OSM is leveraged for the creation of the crossing lines. In particular, primitives with the value *crossing* are of interest. These are points that are located on the middle line of the street where the crossing should be. These points are isolated before finding their correct position with respect to the generated street geometry. For each of these points, the nearest intersection corner point is determined, which is always on the right side looking in the direction of the intersection. This is where a corresponding geometry of crossing lines is copied previously customized to have the length of the street's width with the resulting number of lines.

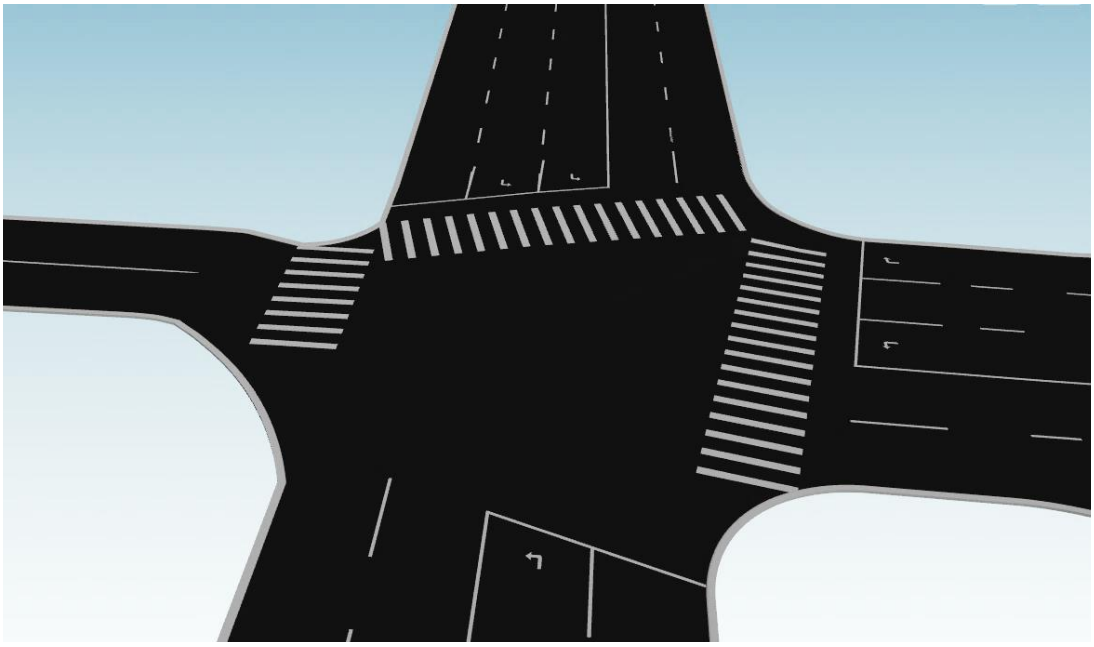


Figure 49: An intersection with stop, center, lane, turn and crossing lines.

Used OSM information	
Attribute name	**Type**
highway	primitive attribute
lanes	primitive attribute
turn_lanes_forward	primitive attribute
turn_lanes_backward	primitive attribute

Table 6: OSM provided information that was used to create street lines.

4.3.6 Bringing Streets to their Original Height

At the end of the processing of the street geometry, the streets are raised back to their original height. In the original geometry, the middle lines contain the height data, which must now be correctly distributed to the other points of the street geometry. Using Ray Casting, these points are transferred to their correct height. In order to obtain streets that are planar in width and thus drivable, all points on each line perpendicular to the street segment are assigned the same height as the line's middle point.

4.3.7 Bridges

A similar procedural generation process is followed for the creation of bridges with a *layer* value of one. The attributes defined at the beginning of the editing process that contain information about the connection points between streets and bridges is combined with a Raycast to bring the bridges to the height of the corresponding streets. A simple railing is modelled on both sides of each bridge.

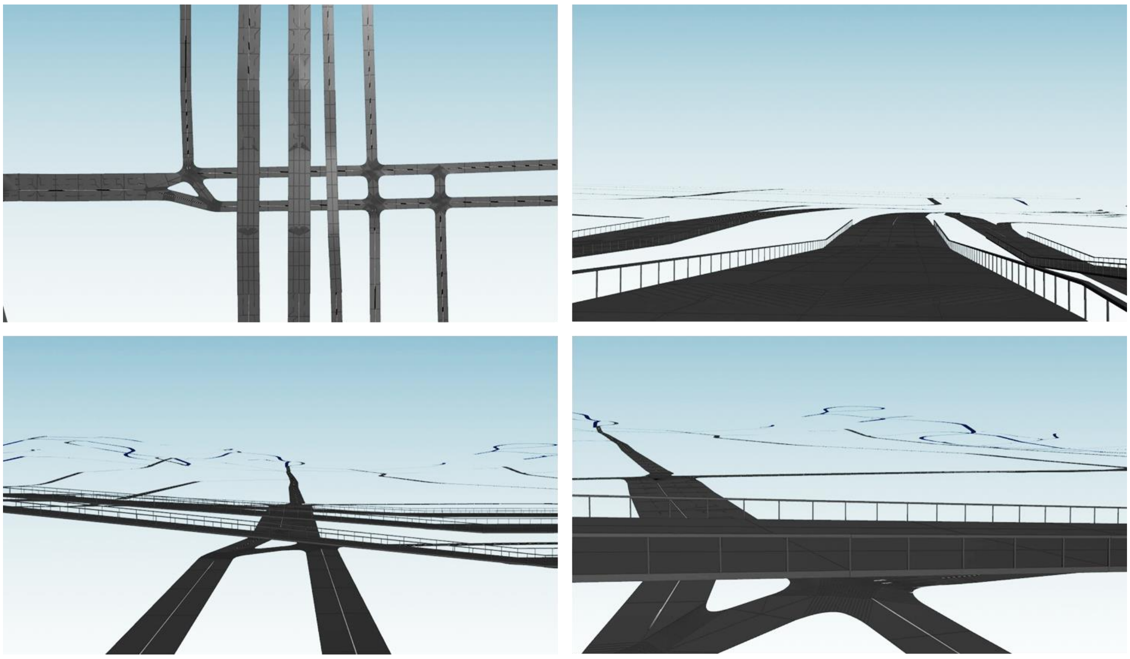


Figure 50: The output geometry for bridges.

4.3.8 Street Furniture

The street furniture, such as streetlamps, hydrants and bins, were placed along the streets with the help of copy points. To obtain the sides of the streets as line primitives, the generated street geometry is extruded and the primitives with a normal vector pointing up or down are removed. The remaining side primitives are converted to line primitives. The result is two sets of line primitives on two different height levels, one of which is removed. The remaining lines are expanded on the same plane to achieve the correct distance to the street side. These line primitives must satisfy two conditions for the next steps: Line primitives that are connected to each other should form one line and each primitive should contain the *highway* variable. After removing the points at the intersection corners, the remaining lines are processed in a loop. The sequence of the points of each connected part of lines is calculated and new primitives are created instead, which contain all the previous points that belonged to the connected part. Based on the *highway* variable and thus the road type, the points of each line are resampled to achieve a certain distance between them, which depends on the corresponding streetlamp. Primary, secondary, tertiary and highway roads are equipped with a different type of streetlamp than the residential roads. In addition to creating the correct copy points for the position of the street elements, their automated placement also requires determining their orientation. To achieve this, the original line that was shifted to create the correct distance to the street side was used by calculating the vector from the end copy point to the original copy point. This vector is directed towards the middle of the street segment. After the orientation is stored as a vector for each copy point, the corresponding assets are copied as instances, placed and oriented according to the calculated vectors.

A similar method is used for the automated placement of the traffic lights. First, the correct position at each intersection corner is determined, and then the correct orientation is calculated. For each traffic light, the correct position is obtained by creating two new points for each intersection corner, one to copy the instance of the right and one for the left traffic light. To obtain the position of these new points, the external neighboring points of each intersection corner are determined. With the help of these two points, the new positions are calculated by moving a certain distance in the direction of each neighboring point from the intersection corner. The orientation of each new point depends on the geographic coordinates. In Los Angeles and Shanghai, the traffic lights are aligned with the opposite street, whereas in Munich, they are aligned with the street they are next to. For the opposite street direction, the previously created lines connecting each pair of the intersection corner points play an important role. A vector is created that connects the intersection point with the opposite direction corner and serves as the orientation vector. For the other direction, the orientation vectors are defined using the neighboring points. An important step during the complete calculation of orientation vectors (*N*) is to assign the right type of traffic light (*streetEl*) to the corresponding new point. This is achieved by calculating the side of each new point and the corresponding neighboring points with respect to the center intersection point. Finally, the traffic lights can be copied as instances to the new points.

Other assets under this category that were automatically placed, include stop signs, bus and subway stops. For the stop signs, we leveraged the OSM information in the form of the *highway* variable for the value of *stop*. The points that contain this information are isolated and the middle point of the nearest street segment is detected. This helps in finding the nearest intersection corner, which is also located on the right side looking to the center of the intersection. A similar shift of the new copy point using the corner's neighboring points results in the correct position. The orientation is similarly calculated using the position vector of the neighboring point. OSM also provides information about the speed limit. This information can be used to position the appropriate sign next to the corresponding street segment. Bus and subway stops are also provided by OSM with the variables, *highway=bus_stop* and *subway=yes*. For each point containing these attributes, copy points are created by finding the nearest external primitive and storing an orientation that is directed towards the middle of the street.

Used OSM information		New defined variables	
Attribute name	**Type**	**Attribute name**	**Type**
highway	primitive attribute	streetEl	point attribute
subway	primitive attribute	N	point attribute

Table 7: OSM information and new defined variables that were used to automatically place the street elements.

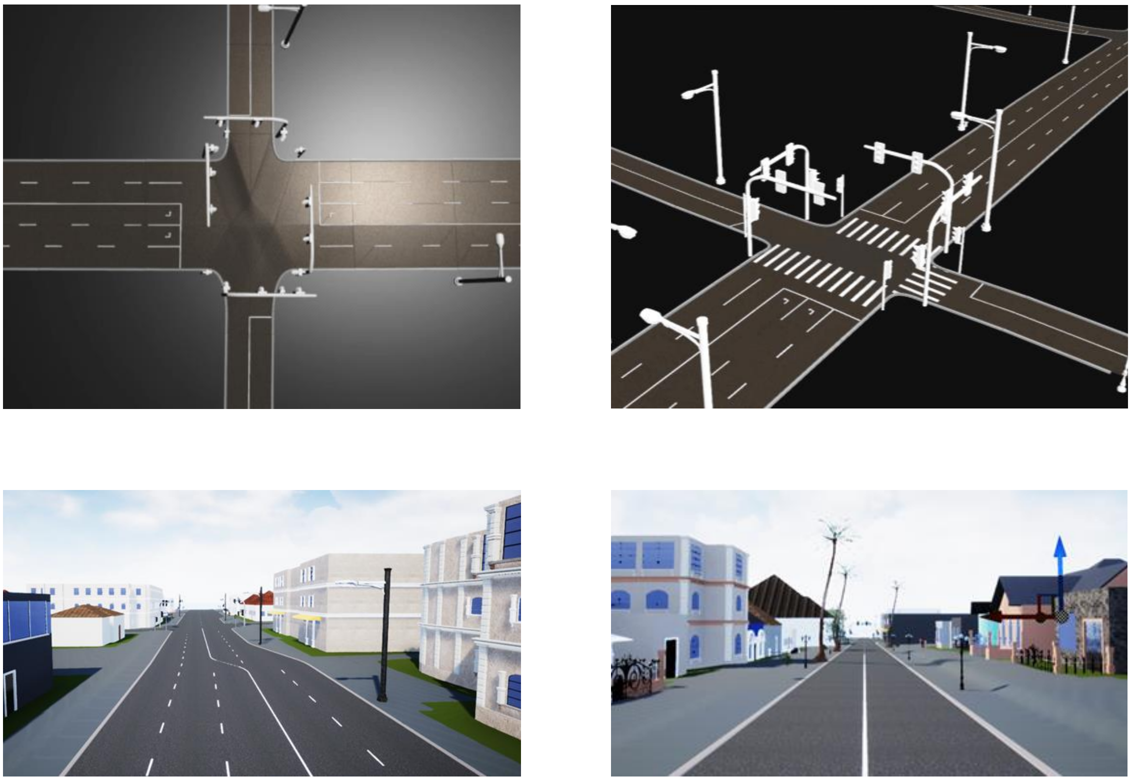


Figure 51: Up: A sample of the output geometry of *StreetGen* inside Houdini. Middle: A sample of the output geometry of *StreetGen* inside the Unreal Engine.

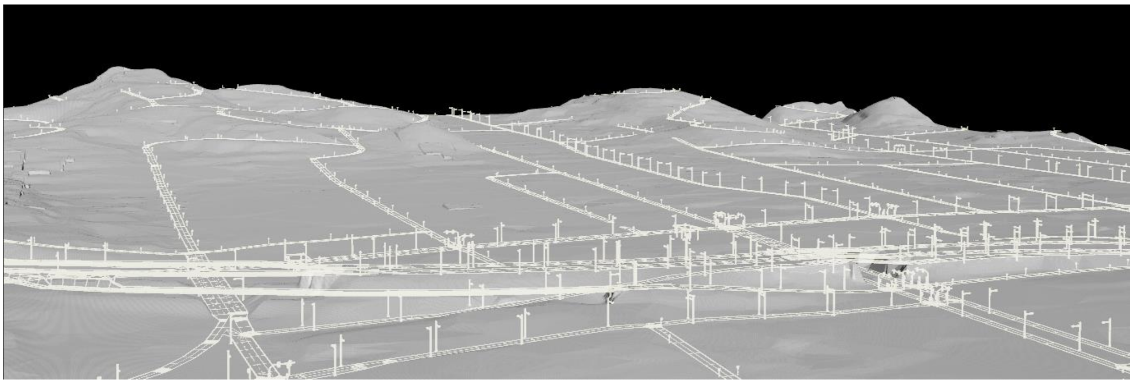


Figure 52: The street geometry in wireframe on the terrain.

4.3.9 Adding the Buildings

The footprints of the buildings contained in the selected area are offered by OSM as single primitives. To adjust them to the slightly edited terrain, rays are cast from the footprint primitives onto the terrain. Furthermore, they are extruded to obtain the appearance of 3D boxes using the *height* attribute offered by OSM. In the case that the height information is not available for all buildings, height values are calculated and assigned instead of the missing information. These height values are between the minimum and maximum heights representing the selected area, with the area of the footprint taken into account in the calculation.

This step aims at a minimalistic view of the buildings used in combination with the terrain for the User Interface, which is described in Chapter 4.8 after the procedural framework.

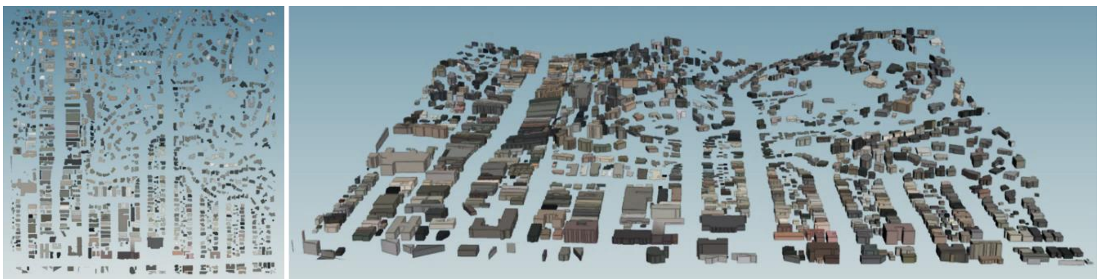


Figure 53: The simplistic version of the generated buildings as 3D-boxes based on data collected from Los Angeles.

Used OSM information	
Attribute name	**Type**
height	primitive attribute

Table 8: OSM provided information that was used to extrude the buildings.

4.4 Buildings

By using OSM for the generation of buildings, our output geometry reflects reality in terms of the buildings' basic shape and category, if the corresponding variable *building* is provided by OSM, which is required to achieve the latter. Additionally, the generation of the buildings focused on creating architectural styles characteristic of the selected map area. The implemented architectural styles were selected according to three important automotive markets: Los Angeles, Munich, and Shanghai.

The generation within *BuildGen* starts with the processing of the imported OSM data, which is the same as at the beginning of the framework. This processing prepares the original data by removing part of the input that should not be generated and defining variables for later use (*Processing the OSM Data*). A large loop then begins to isolate each individual building, with the first step involving the selection of the architectural style based on the geographic coordinates and the provided variables (*Choosing an Architectural Style*). The selection of the architectural style determines the available façade parameters that are imported for the generation of the building's façade and are randomly chosen within the available range to control the procedural generation of the façade modules (*The Façade Module Generators*). After the façade modules are modelled, they are automatically placed on the building walls. This step indicates the way the building walls should be modelled to accommodate the façade modules while forming a continuous surface. This step ends with the placement of the façade modules as instances (*Placing the Façade Modules*).

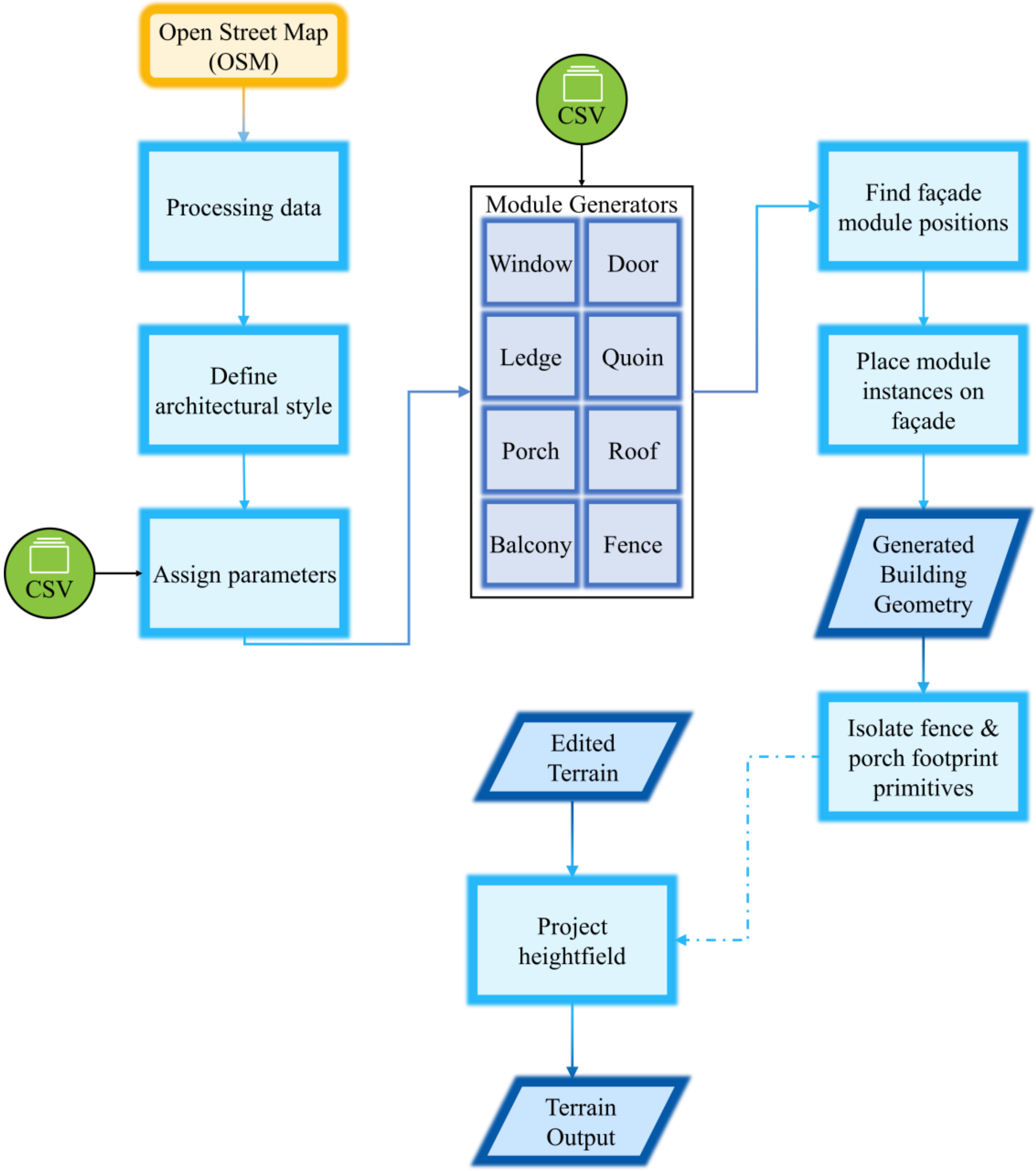


Figure 54: The process used inside the suggested framework for the generation of the building geometry (For a legend, see Appendix).

4.4.1 Processing the OSM Data

The initially downloaded and imported data provided by OSM is used for the generation of buildings after the corresponding primitives have been isolated. These can be automatically distinguished from the primitives representing the streets and nature with the help of the OSM variables *building* and *highway*. The primitives that do not have a *highway* value and whose *building* value is either “yes” or occupied by a building type (e.g., "apartments") represent the footprints of buildings and contain information that is used within *BuildGen* to generate them (e.g., *height*, *levels*). Not all of these primitives are used in the further generation steps. A variable that helps to remove the unnecessary primitives is *multipolygon*. The *multipolygon* attribute indicates that the footprint represents a hollow place (e.g., indoor garden) that is not visible from the street because it lies within the area of a larger building. If one of these primitives contains the *multipolygon* attribute or if it is not really a building but only a man-made structure (e.g., swimming pools) that is labelled as a building, it is removed from the geometry. Another useful attribute, *building_parts*, indicates that the corresponding footprint is part of the building and probably has a different height than the rest. All the footprints are extruded, each according to its individual height, giving the building its final basic shape.

During the building generation, the information about which building walls face a street is used to make more than one decision. Therefore, the corresponding variable is defined at this early phase in the generation process. To determine these specific building walls, a Raycast is performed after extruding the footprint and the street primitives from the initial OSM data along the height axis. The ray is cast from the top two points of each wall primitive towards the outside of the building. If the ray hits a street primitive, the corresponding wall primitive is grouped under *streetSide*. The first decision that is made based on the defined variable is which walls to cover with façade modules and which not (i.e., walls that face other building walls or are not visible from the street). The second decision is about the wall that receives a door and is made in a later generation step. In the case that more than one building side faces a street, the decision can be made by selecting the wall that faces a street with higher priority (primary > secondary > …). This is achieved by transferring the *highway* attribute, provided by OSM, from the street to the wall primitive during the Raycast.

Used OSM information		New defined variables	
Attribute name	**Type**	**Attribute name**	**Type**
highway	primitive attribute	streetSide	primitive attribute
multipolygon	primitive attribute		
building_parts	primitive attribute		

Table 9: OSM information and new defined variables that were used to process the OSM building footprints.

4.4.2 Choosing an Architectural Style

The decision for the architectural style is based on information received from the OSM variables and the newly defined attributes. The most important factor for the selection of the architectural style is the geographic coordinates. This already limits the selection of available styles leaving only those that are characteristic for the selected map area. The architectural styles that can be generated within *BuildGen* are characteristic of three specific cities that were chosen based on their automotive market and the designers' requirements: Los Angeles, Munich and Shanghai. Accordingly, the available architectural styles include:

- Skyscraper,
- Modern,
- Victorian,
- Craftsman,
- Spanish Colonial Revival,
- Beaux-Arts,
- Renaissance,
- Neoclassical,
- Tong Lau and
- Tang

In addition to the geographic coordinates, the height and area of the building, as well as the building type (e.g., office, apartments) are used to select the architectural style.

Characteristic of the city of Los Angeles, the following architectural styles were selected for the generation of the buildings based on the corresponding geographic coordinates:

- Skyscrapers: This architectural style, known for its great height, has the characteristic element of a steel framework that serves as support for the so-called curtain walls [273].
- Beaux-Arts: These buildings are usually richly decorated on the façade and roof with a central focal point. Common decorative elements include sculptures and intricate bas-reliefs [274]. Characteristic architectural elements include coupled columns, corner elements and entablatures [274].
- Victorian: This style has been built in many variations. The most recognizable of these is the Queen Anne style, which is known for its numerous steep roofs and porches with ornamental gables. Other characteristic architectural elements for this style are octagonal or round towers, highly decorated windows, corbelled chimneys, as well as doors with glass panels [275].
- Craftsman: Buildings in this architectural style include bungalows that have low-pitched, gabled roofs, grouped windows and oversized eaves with massive rafter tails [276].
- Spanish Colonial Revival: The façade of these buildings is characterized by deeply recessed windows, doors and balconies, while the roof can be gabled, hipped or flat with simply or non-ornamented cornices and eaves [277].

The following architectural styles have gradually shaped the appearance of the city of Munich throughout its history [278]:

- Neo-Renaissance: Characterized by great variety and symmetry, this architectural style assigns a great amount of decorative elements to its buildings: rusticated masonry, quoins, architraves framing the windows, as well as pediments and entablatures around the door. The top floor of these buildings is usually covered by small square windows.
- Neoclassicism: The façade of these buildings can be described by simple geometric shapes, columns and little to no decorative elements.
- The same architectural principles were applied to the skyscrapers, industrial and modern styles in Germany as to the corresponding Western styles.

From civil and industrial to religious and educational infrastructure, many buildings in Shanghai were built according to Western architectural styles [279]. There are some exceptions, especially in urban and rural areas with traditional styles, such as “Palace style” (gongdiau-shi), “Classical Chinese style” (Zhongguo guyou-shi) and “National style” (minzu-xingshi). A combination of European and American architecture can be seen in characteristic buildings, such as the Peking Union Hospital. This combination consists of elements of Neoclassicism and traditional Chinese roofs. Another characteristic style for modern Chinese architecture is evident in Shanghai with its industrial skyscrapers. An additional style that belongs to the traditional Chinese architectural styles is Tong Lau which is used for residential buildings in the working-class districts of Hong Kong. They are characterized by long and narrow poles that support the entire structure. The ground floor usually houses a shop, while the remaining floors are occupied by tenants. Finally, a common feature of both cities are the tall residential buildings that are divided into many small flats.

4.4.3 The Façade Module Generators

Once the architectural style has been selected, the generation process proceeds with the calculation of the correct parameters, which serve as input data and are sent to the Façade Module Generators. Each of these generators provides a degree of control that enables the generation of the characteristic elements for each architectural style we saw in the previous subchapter. This degree of control is achieved through the parameters that the generator can receive as input. To ensure sufficient variety for each façade module and consequently for each architectural style, each of these parameters is assigned a certain range depending on the architectural style. All these ranges for each parameter and each architectural style are written in a CSV table for each Façade Module Generator. For each parameter of each generator, a value is randomly selected following a uniform distribution in order to avoid too frequent selection of certain values, which results in too many modules of the same type to be generated.

The selected values for each input parameter of the generators are set in the corresponding Façade Module Generator, activating its workflow. Each of these generators was designed to enable further development and to simplify the addition of new assets to increase the variety in the appearance of each existing architectural style, and also to increase their number. The generation of each façade module includes the definition of the wall attribute that is stored in the corresponding primitives of the façade modules, which represent the exterior wall. Within *BuildGen*, there is a Façade Module Generator for each of the following modules:

4.4.3.1 The Window Generator

This generator creates windows based on the following input parameters:

- Form: This is an integer variable that specifies the shape of the window glass that should be selected. The frame is modelled with the same form. The available forms were converted from a corresponding black and white image into a single primitive.
- Grille: This is also an integer variable that controls the pattern of the panes, which separate the window glass [280]. The available patterns were partly pre-modelled manually and partly converted from a corresponding black and white image into a single primitive.
- Frame: This boolean variable determines whether the window frame should be modelled. This parameter helps to create the curtain walls of the skyscrapers, which look like windows without frames. The modelled frame automatically receives the form of the window glass and is thus influenced by the *form* variable.
- Number of additional windows: This is an integer variable that determines how many additional windows should be modelled, evenly distributed on both sides of the central window. If the central window is provided with decorative elements by the associated parameter, these elements are also assigned to the additional windows. This parameter helps to create grouped windows that are characteristic of, for example, the Craftsman style.
- Jamb: This is a float variable that defines the distance between the window glass and the building wall [280]. Increasing the value of this parameter results in a more recessed window, as is the case with windows of the Spanish Colonial Revival style, for example.

- Sill: This is an integer variable that determines the type of the modelled sill. The sill is the horizontal part in the lower section of the window frame [280]. The window can be modelled without a sill, with a simple sill or with a sill that carries decorative elements, if the *sill deco* parameter is set accordingly.
- Sill Deco: This integer variable controls the ornaments that decorate the sill. The available decorative elements consisted of Quixel Megascans [281] assets that were separated from the original asset of a complete window.
- Decorative Columns: This integer variable is used to select the type of columns that serve as decorative elements on both sides of the window. The available columns were also imported assets that were separated from the original window asset by Quixel Megascans.
- Pediment: This integer variable determines the type that should be used as decoration above the window [280]. The available pediment assets were also imported from Quixel Megascans after being detached from the same original assets of the complete windows.
- Glass Dimensions: This variable is a 2D float vector that defines the width and height of the window glass.
- Wall dimensions: This is also a 2D float vector that controls the width and height of the wall surrounding the modelled window. The wall dimensions include the area that extends between the decorative elements around the window and the end of the wall. The total dimensions of the window are calculated by adding the dimensions of all selected elements. By combining this parameter and the glass dimensions, the horizontal and vertical distance between two windows of the same or different floor can be controlled.
- Oriel Window: This is a boolean variable that can be used to transform the modelled window into an oriel window. An oriel window protrudes from the building wall in the form of a bay window [280].

For some architectural styles, a different window type in terms of form and/or size is selected for certain floors or a certain column of windows, which are generated using the same Window Generator but with different parameters.

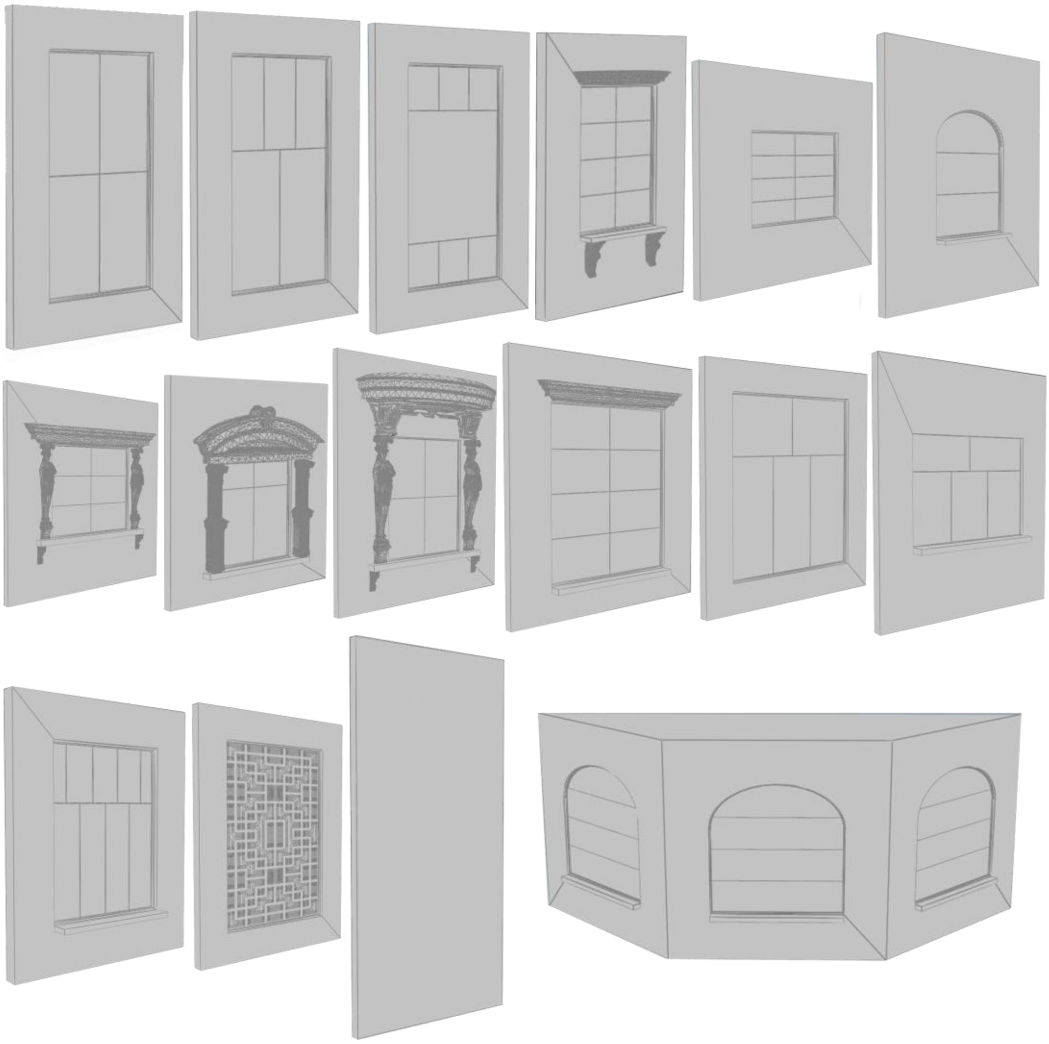


Figure 55: A sample of the windows created by the Window Generator.

4.4.3.2 The Door Generator

The following parameters can be used to control the generation of the doors.

- Form: This integer parameter determines the shape of the door. The available door shapes were converted from a black and white image into a single primitive.
- Entrance frame: This is an integer variable that can be used to select the type of entrance framing the door. The available entrance frames consist partly of manually pre-modelled geometries and partly of photogrammetric assets from Quixel Megascans that were separated from the complete model of a door.

- Single door: This boolean variable can be activated to select a single door. To model a double door, the single primitive selected by the *form* parameter is mirrored.
- Panel pattern: This integer variable defines the type of pattern that decorates the door. To achieve the impression of a door window, the material of the panel pattern can be changed accordingly. The available panel patterns are imported black and white images that were converted into a single primitive.
- Jamb: This is a float variable that specifies the distance of the door from the building wall [280]. This parameter can be used to model particularly recessed door frames that are common for the Spanish Colonial Revival style.

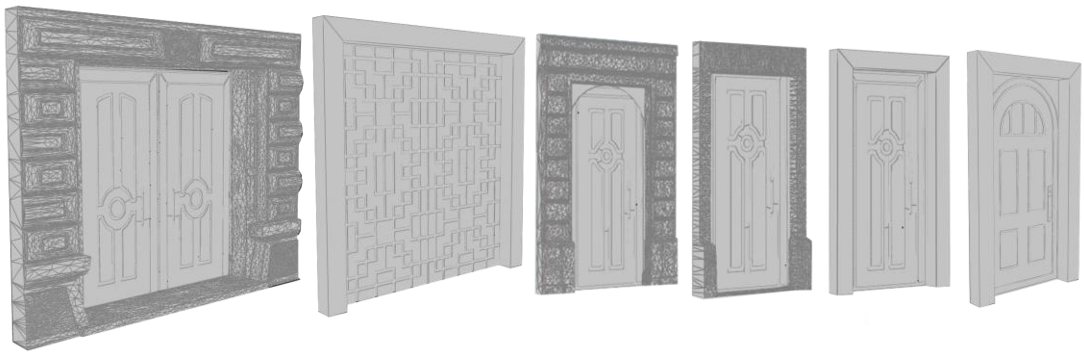


Figure 56: A sample of the doors created by the Door Generator.

4.4.3.3 The Balcony Generator

The input parameters of the Balcony Generator include:

- Form: This integer variable defines the shape of the balcony floor. The available forms are pre-modelled once manually.
- Handrail: This is an integer variable that determines the type of hand support that should be modelled on top of the balcony railing. The available handrail shapes range from none to cylindrical and follow the outline of the balcony.
- Railing: This is an integer variable that specifies the type of the balcony railing. The available railings are partly modelled with basic forms (e.g., cylinder) to more refined shapes (e.g., balustrade) which are imported from Quixel Megascans.
- Shed: This Boolean variable can be activated to place a shed over the balcony after taking into account the floor height and adopting the shape of the balcony's footprint.

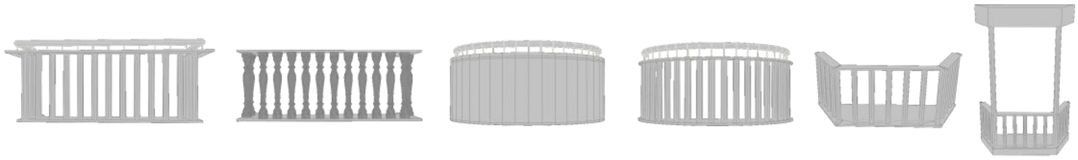


Figure 57: A sample of the balconies created by the Balcony Generator.

4.4.3.4 The Quoin Generator

Quoins are decorative elements on the corners of a building's façade [280]. The Quoin Generator adjusts the photogrammetric assets of quoins imported from Quixel Megascans on the corners of the building. This generator has only one input parameter:

- Type: This integer variable defines the type of quoin to place on the building's corners from no quoin to one of the available assets.

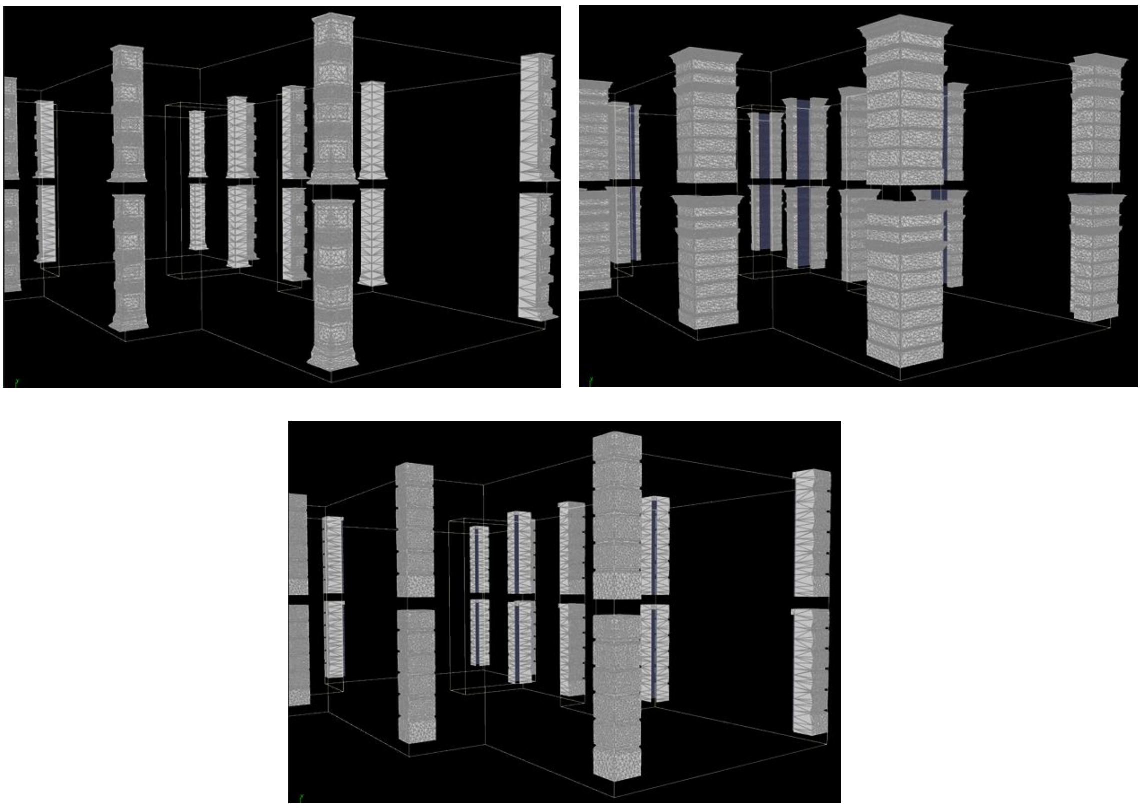


Figure 58: A sample of the quoins created by the Quoin Generator.

4.4.3.5 The Ledge Generator

Ledges are protrusions on the building wall that are both a decorative and functional feature of the façade, as ledges decorate the façade while leading the rainwater away from walls and windows. The following parameters control the generation of the ledges:

- Type: This integer variable defines the type of ledge to be adjusted on the building façade. The available models for the ledges that are created with the first method are imported from Quixel Megascans, while the rest consists of a single primitive that acts as a cross section and was converted from a black and white image.
- Method: The first method of creating the ledges is to import a modular piece and place it repeatedly as an instance along the outline of the building. This method is more efficient if a more refined asset of a ledge with decorative elements should be used. The second method was designed to simplify the process of adding new ledges. In this method, a single primitive of the cross section of the ledge shape is sufficient as it is expanded along the outline of the building. Decorative elements can be added with the next parameter, which places imported assets of ornaments at equal distances from each other on the ledge.
- Deco: This is an integer variable that specifies the type of decorative element to be added to a simple ledge. The available assets were imported from Quixel Megascans.

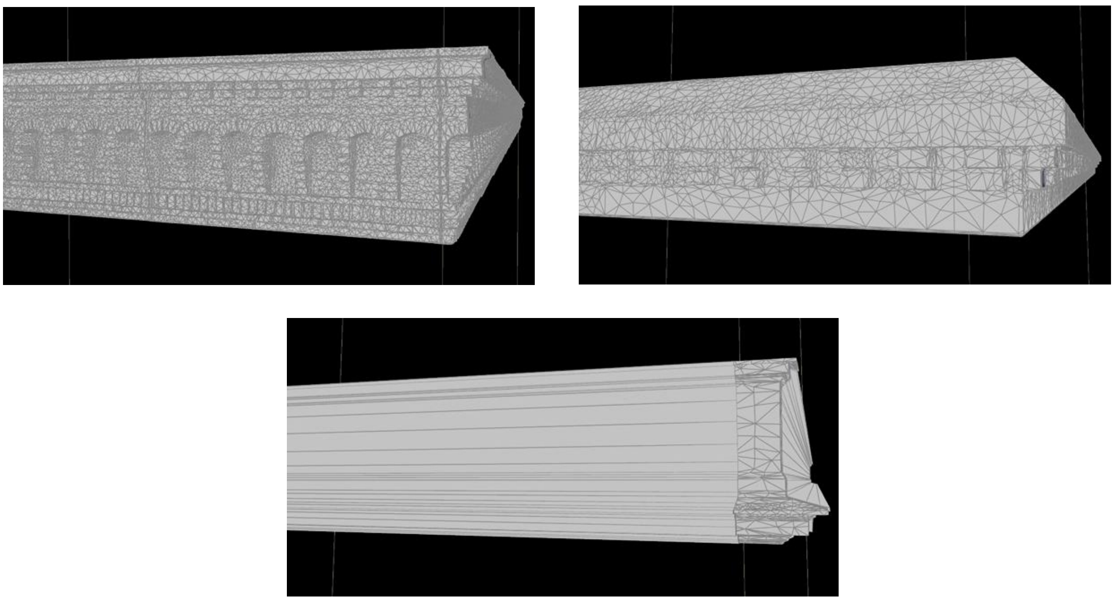


Figure 59: A sample of the ledges created by the Ledge Generator.

4.4.3.6 The Porch Generator

The Porch Generator creates the porch that surrounds the building's front door. To increase variety, it uses a random distribution for the placement of the porch entrance and is controlled by one parameter for changing the type of railing:

- Railing: This is an integer variable that specifies the type of railing and matching columns of the modelled porch. The available railing types consist of repeated instances of individual assets that were imported from Quixel Megascans.

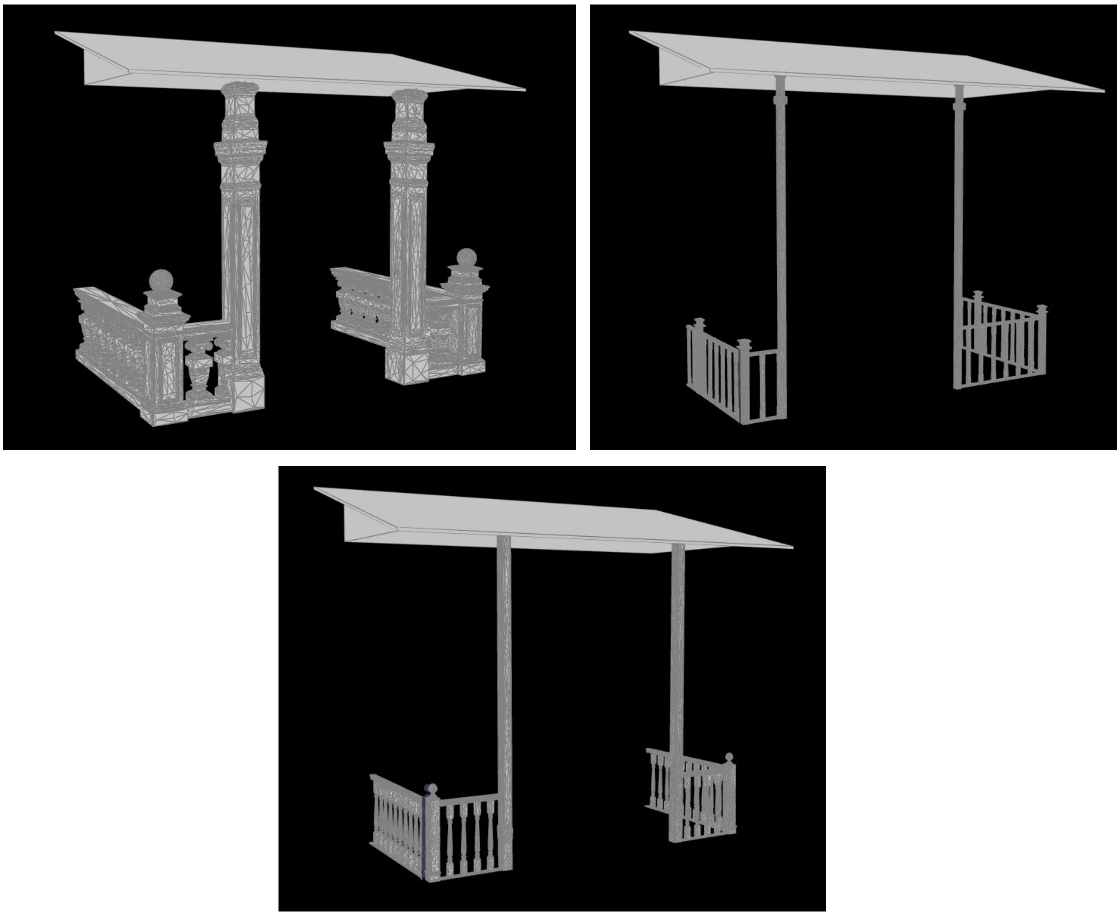


Figure 60: A sample of the porches created by the Porch Generator.

4.4.3.7 The Fence Generator

The Fence Generator creates the fence surrounding the wall that holds the front door. The following parameters can be used to control the generated output:

- Railing: This integer variable determines the type of railing to be placed along the fence. The railing consists of an imported asset that is copied as an instance at equal intervals along the fence.
- Door: This is an integer variable that can be used to select the type of entrance for the fence. The available entrances are imported assets that are adjusted to the height of the fence and the available width on the front side of the fence.
- Lower wall: This is a boolean variable that can be activated to convert the lower part of the fence into a wall.

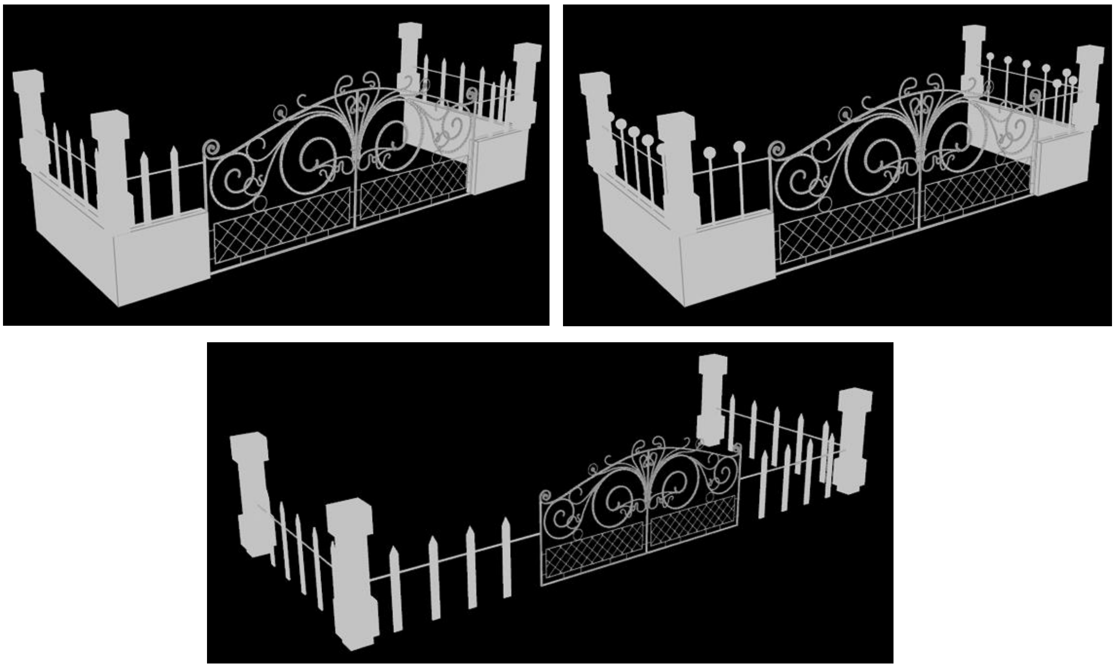


Figure 61: A sample of the fences created by the Fence Generator.

4.4.3.8 The Ground Façade Generator

The Ground Façade Generator creates the façade of the ground floor. The following parameters control the generation of the ground floor façade:

- Type: This is an integer variable that defines the type of façade to model on the ground floor. The ground floor façade can be a simple wall, a shop or a wall with windows as in the other floors. The shop fronts were converted from a black and white image into a geometry with surfaces that are extruded in different degrees to achieve the impression of a 3D structure. This method of generating the shop front was used to simplify the addition of new shop fronts. These geometries can be switched for photogrammetric shop façades to increase the graphics quality.

- Awning type: This is an integer variable that can be used to change the type of awning modelled above the shop. The available awnings are imported assets.
- Sign type: This is an integer variable that controls the type of sign that hangs on the side of the shop. It consists of a simple geometry that represents the basic shape of a sign with different options for shop names available.

Figure 62: A sample of the façades at the ground floor created by the Ground Façade Generator.

4.4.3.9 The Roof Generator

The Roof Generator uses two input parameters that influence the output geometry:

- Type: This integer variable defines the type of roof to be generated based on the building's footprint and placed on top of the building. The available roof types are hip, gable, mansard, gambrel, flat and xieshan type.
- With windows: This boolean variable can be activated to add windows to the appropriate roof types.

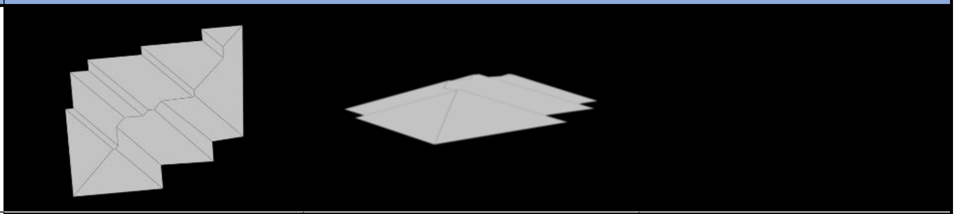
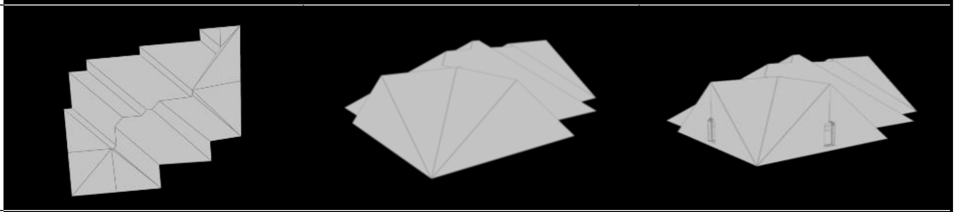
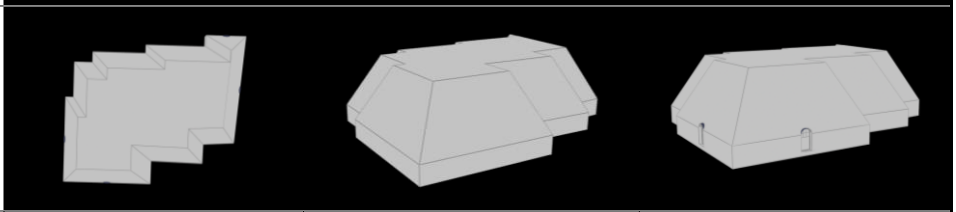
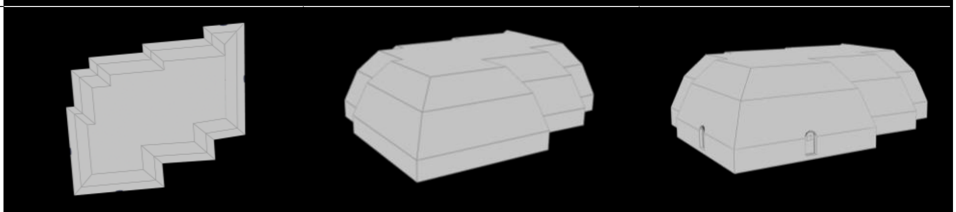
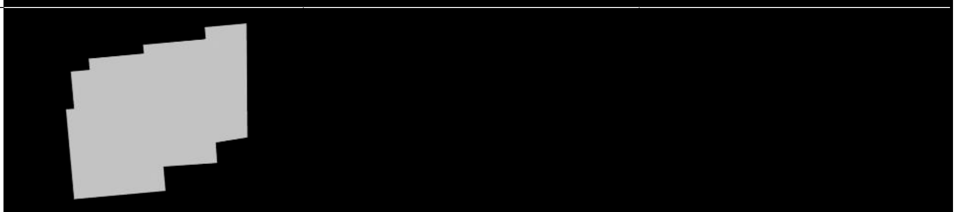
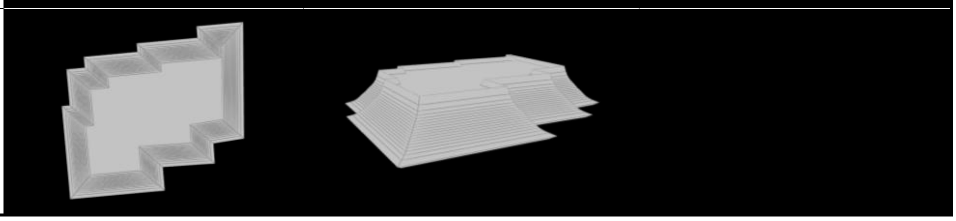
Roof Type	Various Perspectives
hip	
gable	
mansard	
gambrel	
flat	
xieshan	

Table 10: The possible roof styles created by the Roof Generator.

4.4.4 Placing the Façade Modules

After the generation of the façade modules is completed, the generation process continues with their automatic placement. For this purpose, Houdini's *building generator* was used after its workflow was modified to enable the placement of more than one type of window, such as different windows for different floors or columns. In the end of this step, the output copy points of the window instances from Houdini's *building generator* are stored for later use, as are the output geometries after they have been reduced to the primitives that are defined by the *wall* attribute to save performance. The method used in this tool for covering the building wall with modular primitives is used to identify the wall surfaces that do not need to be generated with façade modules. These wall primitives either do not have the *streetSide* attribute and are therefore not visible from the street or they are too close to another wall.

To place the building's front door, the wall modules belonging to the side of the building that is characterized by the *streetSide* attribute are isolated. If more than one street priority is involved (primary > secondary > …), only the modules with the highest priority are kept. To place the main entrance in the middle of the remaining wall modules, the central modules are selected, the number of which depends on the width of the door. Then, the door must be placed in the right location and in the right direction on the wall modules. To this end, the wall module that is to contain the lower right point of the door is found. To determine which of the two lower points of the primitive is the correct one and should be chosen as the copy point, a vector is calculated that starts at each of the lower points of the wall module and ends at the other. The cross product between this and the upward vector is then calculated and compared with the normal vector of the wall module. If the direction of the two vectors is the same, the corresponding lower point is stored as the copy point for the generated door. The copy direction is equal to the direction of the normal vector of the wall module.

The placement of the balconies depends on the position of the windows. The generated balconies are first adjusted in width to accommodate the window, under which they are placed. In this case, the copy point and copy direction are calculated based on the wall of the corresponding window and the same method as for the placement of the door is used to find the lower right corner. To obtain a more realistic façade, not all windows are assigned a balcony.

In order to place the quoins, the type of each building corner must be specified. To this end, the point sequence of the footprint primitive is determined. This is then used to create the outline of the footprint primitive by creating line primitives between its points. For each point in the sequence, the side on which it is located relative to the line formed by the previous two points in the sequence is determined. The normal vector of the wall primitive to which the two previous points belong is obtained and used to determine the position of a theoretical point on this vector. The side of this new point relative to the same line formed by the two previous points in the sequence is determined and compared with the side of the next point. If the two points lie on the same side, the corner is concave, otherwise it is convex. If the corner is convex, a point is created, and a normal vector is calculated, which serve as the copy point and copy direction of the quoin instance.

The ledges are placed using the first method of adjustment by measuring the length of the modular part, calculating the positions of copy points along the building's outline and creating the copy points. Instances of the modular ledge asset are then copied to these points, covering the entire outline of the building. In the second method, the cross section of the ledge is expanded along the building's outline. In both methods, the corners of the ledge are cut at the angle of the corners of the building after they have been measured. The ledges are automatically placed at the top of each floor based on the corresponding floor footprint, which is also an output of Houdini's *building generator* that has no other use in our case.

For the placement of the porch, the same wall modules are used that were selected for the placement of the building's front door. The wall modules of the entire side of the building that corresponds to the side of the entrance door are used to place the fence. The two generated modules do not need to be copied because they are generated directly on the wall modules selected for the door. That means that their generation takes place after the generation and placement of the windows and door.

To achieve a continuous surface of the building wall and facilitate the assignment of materials on the wall, the wall modules are used to determine the outlines of the façade modules before they are removed. For each side of the building, a new primitive is created whose surface covers the entire side and a simple geometry with the same outline is created for each of the outlines. These geometries are used to remove the part of the new wall surface, which is covered by them, in a Boolean operation. The result is a continuous wall for each side of the building that is interrupted by holes for the façade modules. The generated façade modules can then be copied to the holes using the copy points that were stored during their corresponding placement after the primitives with the *wall* attribute have been removed.

The UVs of the primitives are orthographically projected for each category of façade modules and the wall. The area of each primitive group is taken into account to automatically resize them. The assignment of materials is achieved based on the same concept of randomly selecting and assigning the parameters of the façade module. A material library is written in a CSV table. For each architectural style, an additional CSV table defines the range of materials, which are described using the order they have inside the material library, for each category of façade elements (e.g., glass, window frame, window grill). One material is randomly selected for each façade element category and assigned accordingly.

The following table shows the generated output of a complete building for each architectural style and corresponding city, together with some variations for the same building footprint.

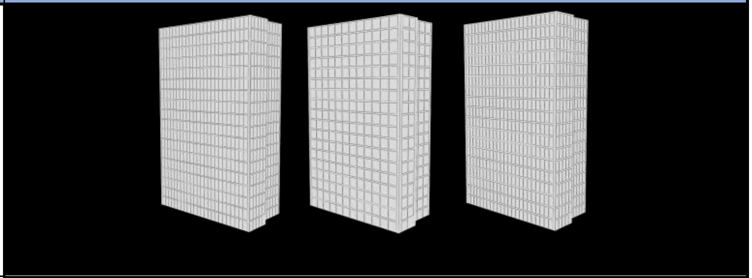
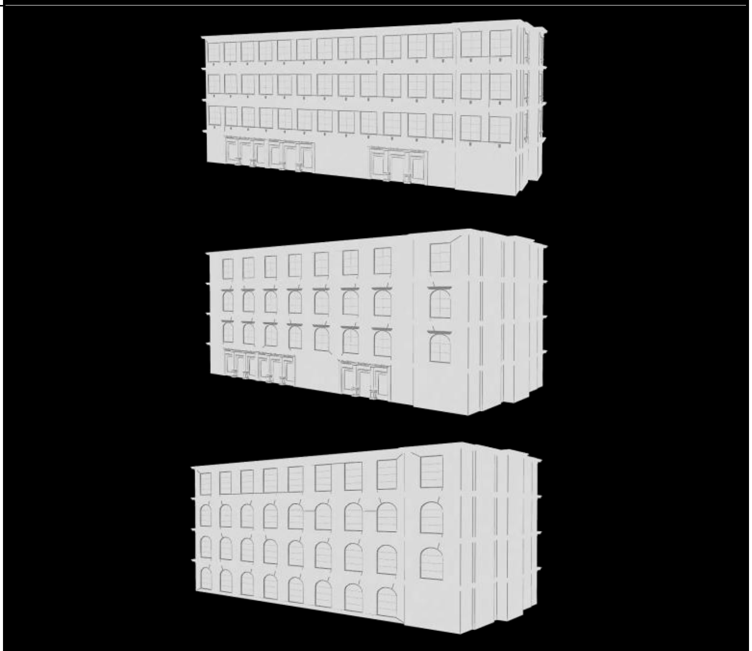
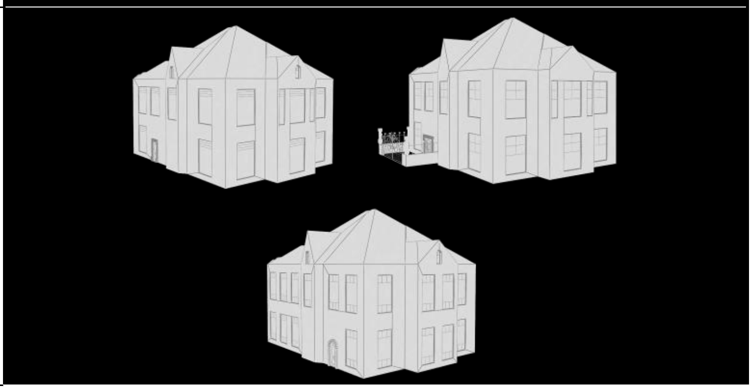
City	Architectural Style	Generated Building & Variations
Los Angeles /Munich /Shanghai	Skyscraper	
Los Angeles	Beaux-Arts	
Los Angeles	Victorian	

Table 11: A sample of the generated buildings in the different architectural styles.

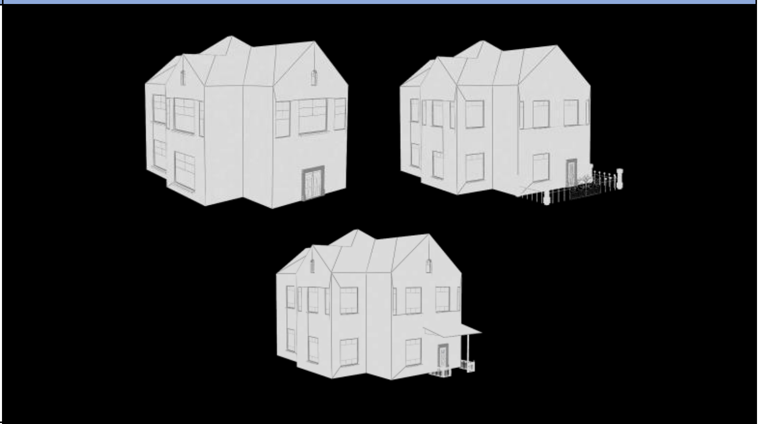
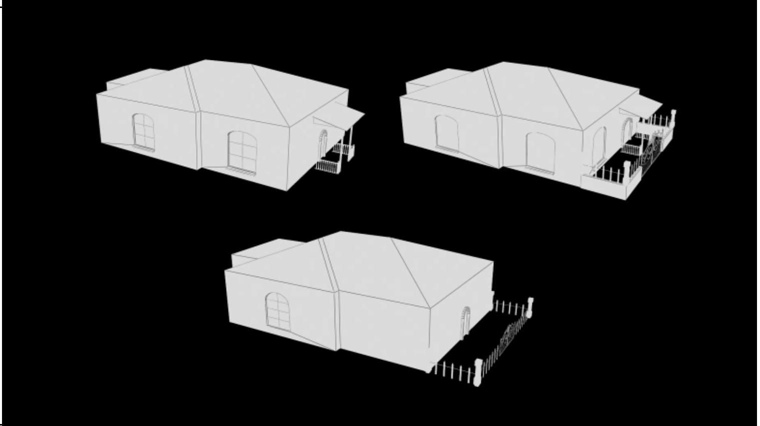
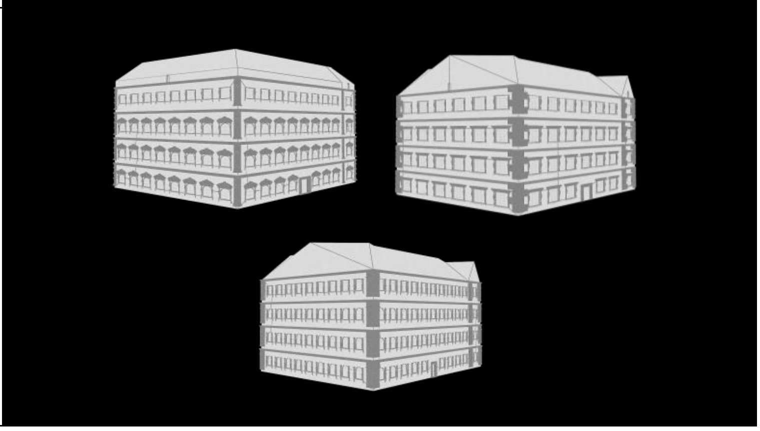
City	Architectural Style	Generated Building & Variations
Los Angeles	Craftsman	
Los Angeles	Spanish Colonial Revival	
Munich	Neo-Renaissance	

Table 12: A sample of the generated buildings in the different architectural styles.

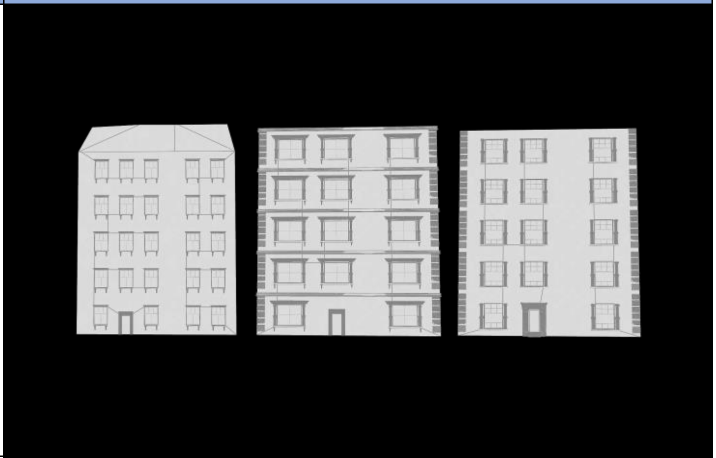
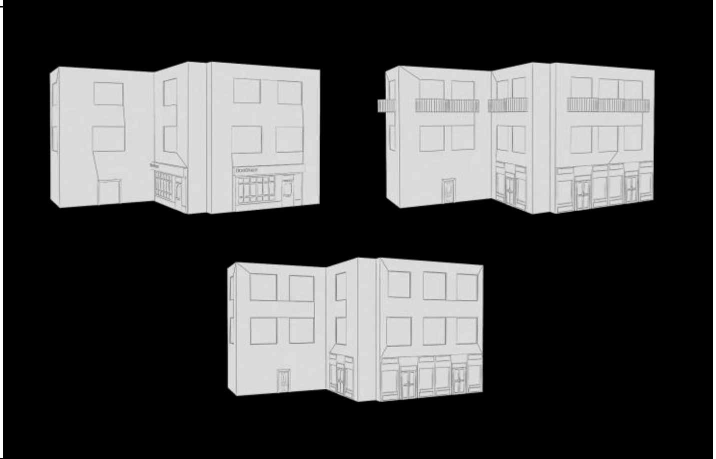
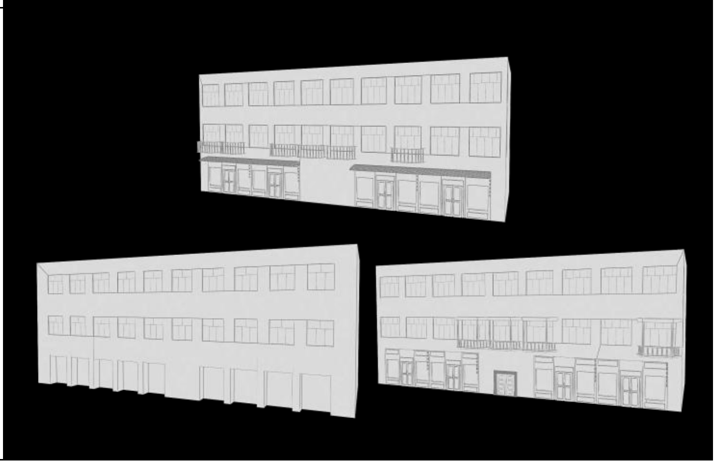
City	Architectural Style	Generated Building & Variations
Munich	Neo-Classicism	
Los Angeles/ Munich	Modern	
Shanghai	Tong Lau	

Table 13: A sample of the generated buildings in the different architectural styles.

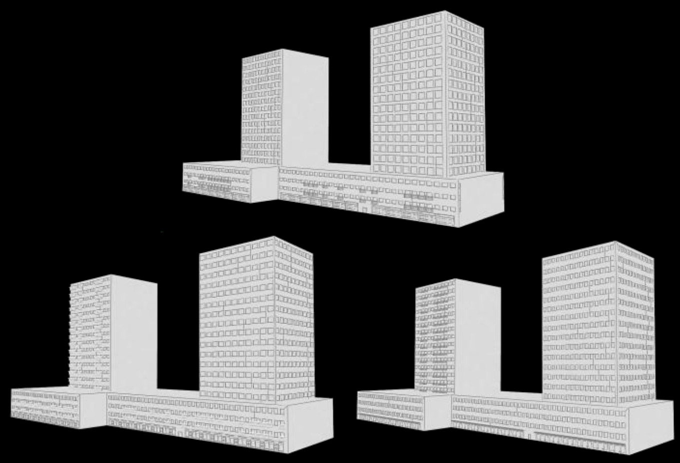
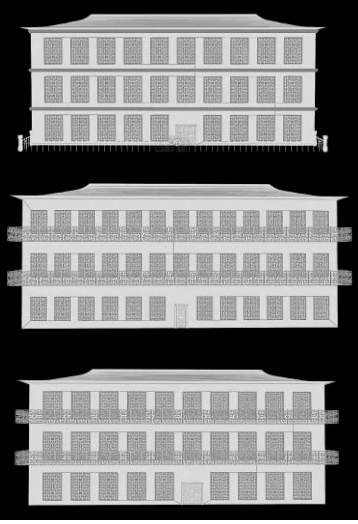
City	Architectural Style	Generated Building & Variations
Shanghai	Modern	
Shanghai	Tang	

Table 14: A sample of the generated buildings in the different architectural styles.

Figure 63: A sample of the output geometry of *BuildGen* inside the Unreal Engine.

4.5 Vegetation

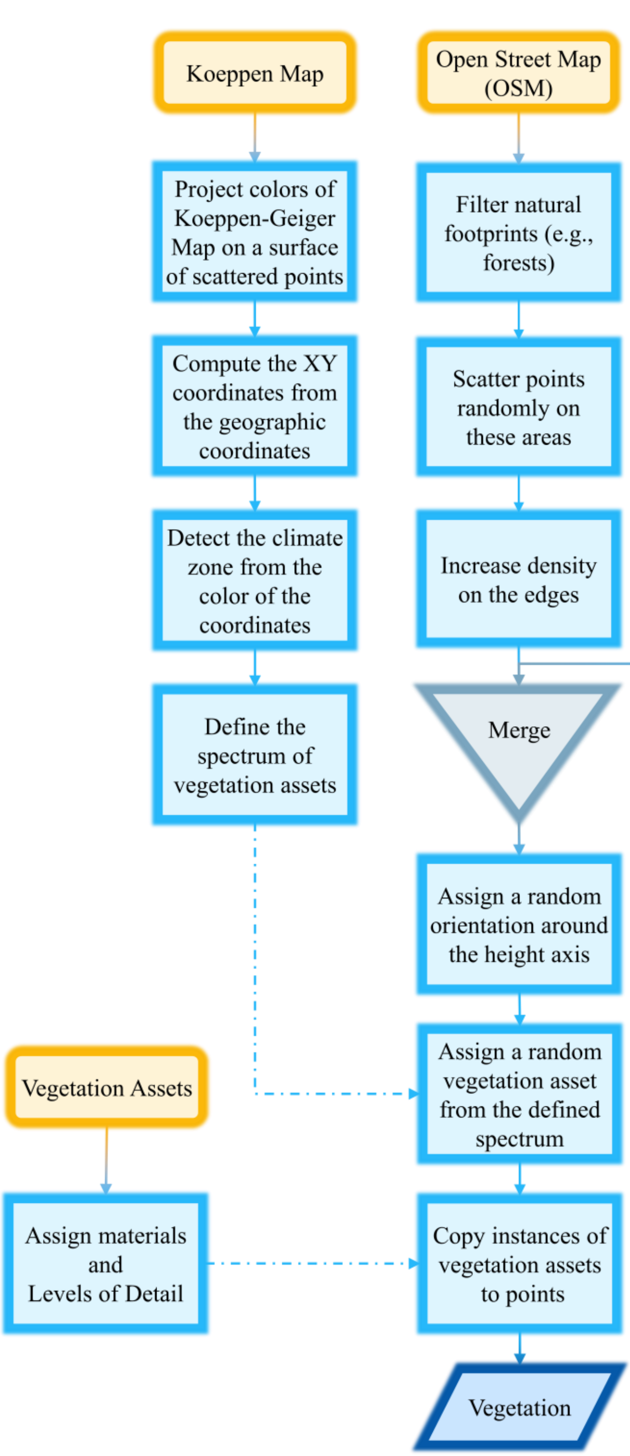


Figure 64: The process used inside the suggested framework for the generation of the vegetation (For a legend, see Appendix).

Inside *NatureGen*, there are two main steps for adding vegetation to the scene. The first step leads to the recognition of the biome where the geographic coordinates are located. The second step places the points that serve as the positions for copying the instances of the vegetation elements. To find the biome, other researchers have used the annual precipitation and annual average temperature to control the modelling of vegetation [246]. Due to the lack of a practical formula that provides this information as a function of the geographic coordinates and since there is a map, which is vegetation-oriented that also uses precipitation and temperature as criteria for dividing the world into different climate zones, the vegetation elements are selected according to the climate zone of the Koeppen-Geiger Map [282] and the geographic coordinates. To this end, the image of the Koeppen Map is imported and transformed into colored points. This is achieved by scattering the points across a grid of the same size as the image and by assigning a color to the points relative to their position within the image. After that, the geographic coordinates of the center point of the selected map area are transformed to XY-coordinates on the area of the colored points. The function used for this purpose, is suitable for maps with equirectangular projection or la carte parallelogrammatique projection that is the projection type with which the used Koeppen Map was created. The following function calculates the x- and y-coordinate from a longitude λ and a latitude φ (in rad) [283]:

$$x = R(\lambda - \lambda_0)cos\varphi_1$$

$$y = R(\varphi - \varphi_0)$$

where R is the radius of the globe (6.371km), $\lambda_0$is the central meridian of the map (0 rad), φ_0 is the central parallel of the map (0 rad) and φ_1 are the standard parallels (north and south of the equator) where the scale of the projection is applied (0.524 rad). After calculating the position of the selected area within the Koeppen Map, the nearest colored point is found. Based on a predefined table, the correct climate zone is assigned to the color, according to the colors of the Koeppen-Geiger map for each climate [282]. Each of these climates can facilitate the right conditions to accommodate specific vegetation.

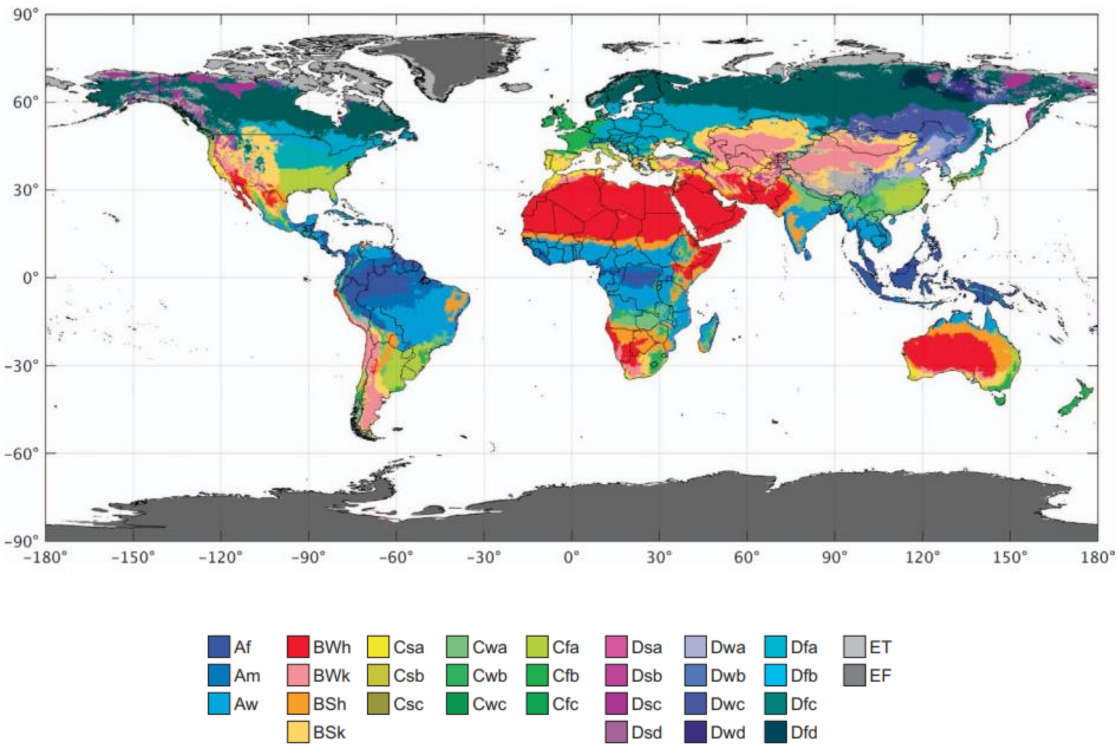


Figure 65: World map of Koeppen-Geiger climate classification [282].

The Koeppen-Geiger classification system divides the world into five different climate zones, using colors and shades [282]. This system uses criteria related to temperature and aridity (relationship between the precipitation input to the soil and the evaporative losses). The climate types are named with a letter: A, B, C, D and E and each of them is further divided into subtypes:

I. A: tropical climates characterized by constant high temperatures (with an average of 18°C) and high precipitation.
   - Af: tropical rainforest climate: This is a hot, humid and wet climate with high temperatures, that hardly change, and constant rainfall [284]. In terms of vegetation [285], the angiosperms have a particular biodiversity. These flowering plants reach in their majority great heights and have trunks that have an accordingly large diameter. In tropical rainforests, only a few gymnosperms (conifers and their relatives) exist, while the Arecaceae plant families (palms) are abundantly represented with various species characteristic of each region. Other plant families are restricted to certain areas, such as the Dipterocarpaceae (dipterocarps), which grow

mainly in western Malesia and the Bromeliaceae (bromeliads) that are found only in the New World. Smaller plants, such as ferns, mosses, liverworts, lichens, algae and fungi, are also abundant and show great diversity.

- Am: tropical monsoon climate: It is influenced by the monsoon winds, which change direction in every season. It is drier than the Af climate with the driest month having less rainfall than the average rainfall of the Af climate and at least 75% of the total annual precipitation [284]. These forests have evergreen species, many lianas (woody vines) and herbaceous epiphytes (air plants, e.g., orchids) [286].
- Aw: tropical savanna climate with dry-winter characteristics: It is the driest of the three subtypes of tropical climate. The driest month has less rainfall than the average amount of the Af climate and less than 75% of its total annual precipitation [284]. American savannas are characterized by broad-leaved trees, such as Curatella, locustberries, maricao cimarrons and Bowdichia, which are seasonally replaced by the palms Copernica and Mauritia. Smaller plants include cutgrass (Leersia), bahia grass (Papsalum) and the bean relative, Prosopis. In Africa, savannas are mainly covered by acacias (Acacia) and bushwillows (Combretum), baobabs (Adansonia digitata), palms (Borassus) and spurge (Euphorbia). A wide variety of spiny shrubs, as well as grass species, such as bluestem (Andropogon), thatching grass (Hyperrhenia) and kangaroo grass (Themeda). Indian savannas are known for thorny trees, such as Acacia, Mimosa and Ziziphus, and grass species, such as Sehima and Dichanthium. Finally, in Australia, savanna flora consists of sclerophyllous plants, evergreen trees, such as Eucalyptus and tall shrubs, such as Acacia, Bauhinia and screwpine (Pandanus). Other plants include baobabs, spear grass (Heteropogon), kangaroo grass (Themeda) and spinifex grass (Plectrachne Triodia) [287].

II. B: dry climates that are divided into desert and semi-arid. They are characterized by very low annual precipitation below a certain threshold which is close to the potential evapotranspiration. This threshold value is used to distinguish between arid or desert climates and semi-arid or steppe climates [284].
   - BW: Arid climate: It is divided into hot (BWh) and cold desert climate [284]. Vegetation common in desert climate areas includes the tumboa or welwitschia, the creosote bush (Larrea tridentata), the saltbush and prickly saltwort [288]. The daisy family shows great diversity, while Artemisia (wormwood), Senecio and the bean family are widespread. Only in desert areas are the chenopod or saltbush family, the cactus and the Frankeniaceae family common. Other trees and shrubs that are also

found in desert environments but less abundant, are conifers (Pinus, Callitris and Cypressus) and tamarisks. Smaller shrubs are the Artemisia, Ephedra, Atriplex and Larrea. Perennial grasses found in most deserts include Aristida, Panicum and Stipa.

- BS: Semi-arid or steppe climate: It is also divided into hot (BSh) semi-arid and cold (BSk) semi-arid climate [284] according to the average temperature of the coldest month. The steppes are predominately covered with grass and shrubs [289]. The predominant grass species include the Festuca pallescens, Rytidosperma and Lathyrus magellanicus, while Agrostis pyrogea, Senecio sericeonitens and Nassauvia aculeata are also very common in steppes [290]. Among shrubs, the Mulinum spinosum, Acaena splendens, Senecio and Rumex acetosella are the most common. Further species of plants, such as graminoids also grow in great diversity in these areas (e.g., the Bromus, Elymus, Agropyron, Festuca, and Stipa species). Additionally, herbaceous flowering plants, such as Achillea, Artemisia, Aster, Astragalus, Centaurea, Inula, Linum, and Salvia are also found in areas with semi-arid climate.

III. C: temperate/mesothermal climates: They are characterized by an average temperature between 0°C and 18°C during its coldest month. They are divided into the following subcategories [284]:

- Csa: Mediterranean hot summer climates,
- Csb: Mediterranean warm/cool summer climates,
- Csc: Mediterranean cold summer climates,
- Cfa: Humid subtropical climates,
- Cfb: Oceanic climate,
- Cfc: Subpolar oceanic climate,
- Cwa: Dry-winter humid subtropical climate,
- Cwb: Dry-winter subtropical highland climate,
- Cwc: Dry-winter cold subtropical highland climate.

The Mediterranean vegetation of the areas bordering the Mediterranean Sea consists of shrubs and small trees. Characteristic shrubs include the Lamiaceae, Laurus and Myrtus species, while olive, carob and fig trees are often seen in these areas [291]. The Mediterranean vegetation in North America presents a similar picture with various shrubs and small trees. These include evergreen oaks, sagebrush, chamise, mountain mahogany, red shanks, and manzanita [292].

Humid subtropical climates are suitable for the establishment of evergreen trees, bushes, and shrubs. The most common trees in these areas are palms and ferns.

IV. D: Continental/microthermal climates: They are characterized by an average temperature of below 0℃ during its coldest month and above 10℃ during its warmest months. Their subcategories include [284]:
   - Dfa/Dwa/Dsa: Hot summer continental climates,
   - Dfb/Dwb/Dsb: Warm summer continental or hemiboreal climates,
   - Dfc/Dwc/Dsc: Subarctic or boreal climates,
   - Dfd/Dwd/Dsd: Subarctic or boreal climates with severe winters.

   Areas with a continental climate are covered by coniferous or deciduous forests. Deciduous forests are home to trees, shrubs and smaller herbaceous and flowering plants. Among best-known trees are oaks, beeches, birches, chestnuts, aspens, elms, maples, and basswoods [293]. In coniferous forests, cone-bearing and needle-leaved or scale-leaved trees (e.g., pines, spruces, firs, and larches) grow, while the ground is covered by low shrubs and herbs, (e.g., mosses, liverworts, and lichens) [294].

   Boreal climates are suitable for taigas or boreal forests, where cone-bearing needle-leaved or scale-leaved evergreen trees are also predominant. These trees include pine, spruce, larch, fir, birch and poplar. The ground is covered by a high diversity of mosses and lichens. Other plants include twinflower, lingonberry, baneberry, Swedish and Canadian dwarf cornel, fireweed, Labrador tea, cloudberry, cotton grass, and crowberry [295].

V. E: polar climates: They are characterized by the warmest temperature in a month being below 10℃ [284].
   - ET: Tundra climate. In coastal areas with a tundra climate, sedges, cotton grass, and mosses including Sphagnum, are predominant. In higher areas, common plants include low willows, grasses, and rushes [296].
   - EF: Ice cap climate.

For the distribution of vegetation, the OSM information on this subject, such as forests and tree rows, is leveraged to define the areas where plants and trees need to be placed. These areas are provided as closed line primitives that precisely define their bounds. These primitives are used to scatter points within the corresponding areas where the instances of the plant assets are to be copied. These points are placed with particular attention to the boundary of the primitives, as they contain the first row of plants and trees visible to the driver. Accordingly, the boundary receives a higher density of vegetation points. The primitive is covered up to a certain percentage of its area, which is controlled by a number to quickly and easily increase the performance of the scene by reducing the number of vegetation assets.

Within a city and in cases where no information about the vegetation is available, the trees are placed randomly along the streets. To this end, the outlines of the street geometry calculated inside *StreetGen* are used as a guide for the placement of the vegetation along the streets. Naturally, these outlines need to have a certain distance from the actual street sides and, therefore, the process of expanding the lines that was used to place the street elements is followed again to shift the lines away from the streets. The points of these lines are then resampled to be placed at a specific distance from each other. This number is manipulated to control the number of plants placed along the streets to increase performance. The condition for resampling these lines is that the connected line primitives become one. To achieve this, the sequence of the points belonging to all connected line primitives is calculated and new line primitives are created based on the sequence of all connected points.

These points are given a certain random rotation around the height axis (*N*) to elicit the impression of a variety of the same plant asset. Additionally, a tree or a plant is randomly selected from a certain spectrum of trees and plants that are suitable for the previously detected climate (*tree*, *plant* and *vegAsset*). After assigning these parameters, each point has the information about the position, the orientation and the type of tree or smaller plant whose instance needs to be copied.

Before finally copying instances of the vegetation assets to the generated points, these points should be split into the levels that we want to stream within the Game Engine. After these operations have taken place inside *NatureSplit*, multiple levels of detail are automatically created for each vegetation asset and the parameter for screen percentage and corresponding material is assigned to each of them. Finally, the vegetation assets are copied as instances to the respective points.

Used OSM information		New defined variables	
Attribute name	**Type**	**Attribute name**	**Type**
natural	primitive attribute	biome	detail attribute
leisure	primitive attribute	tree	primitive group
landuse	primitive attribute	plant	primitive group
boundary	primitive attribute	vegAsset	point attribute
		N	point attribute

Table 15: OSM provided information and new defined variables that were used to automatically place vegetation assets.

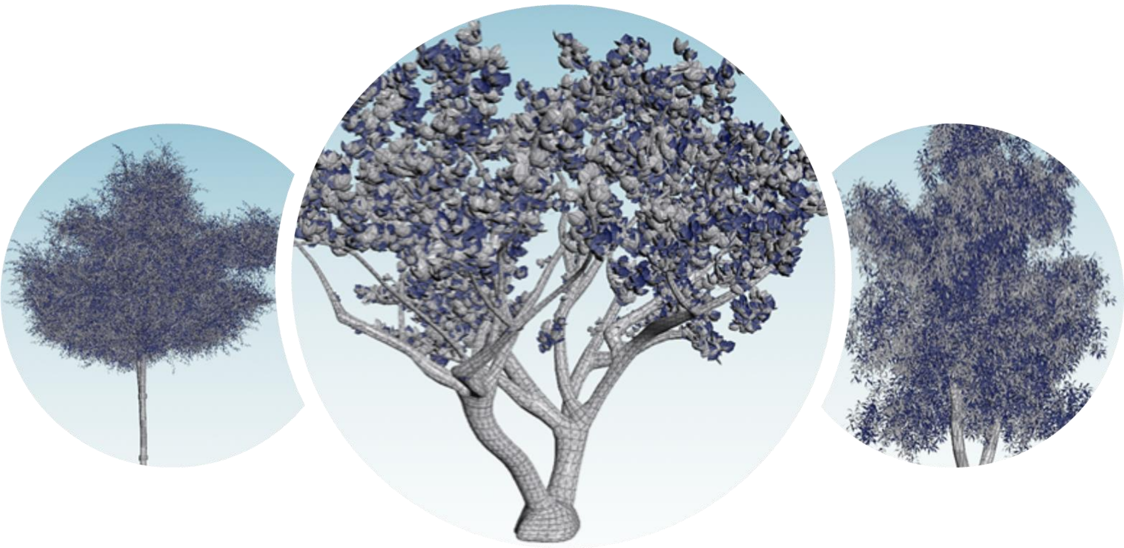


Figure 66: A sample of the used plants for the generation of the vegetation.

Figure 67: A sample of the used plants for the generation of the vegetation.

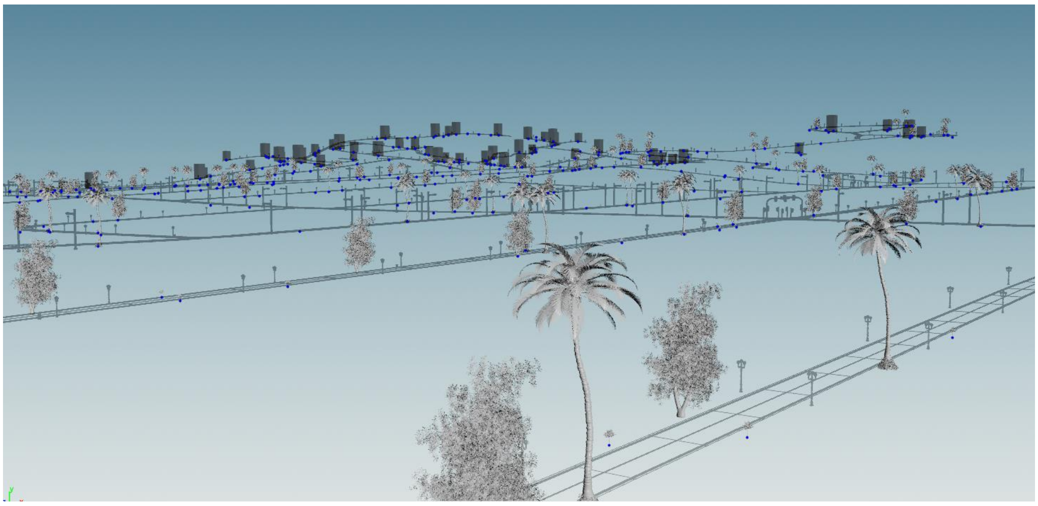


Figure 68: Up: A sample of the used plants for the generation of the vegetation. Down: The generated streets and vegetation in wireframe.

4.6 Split Generators

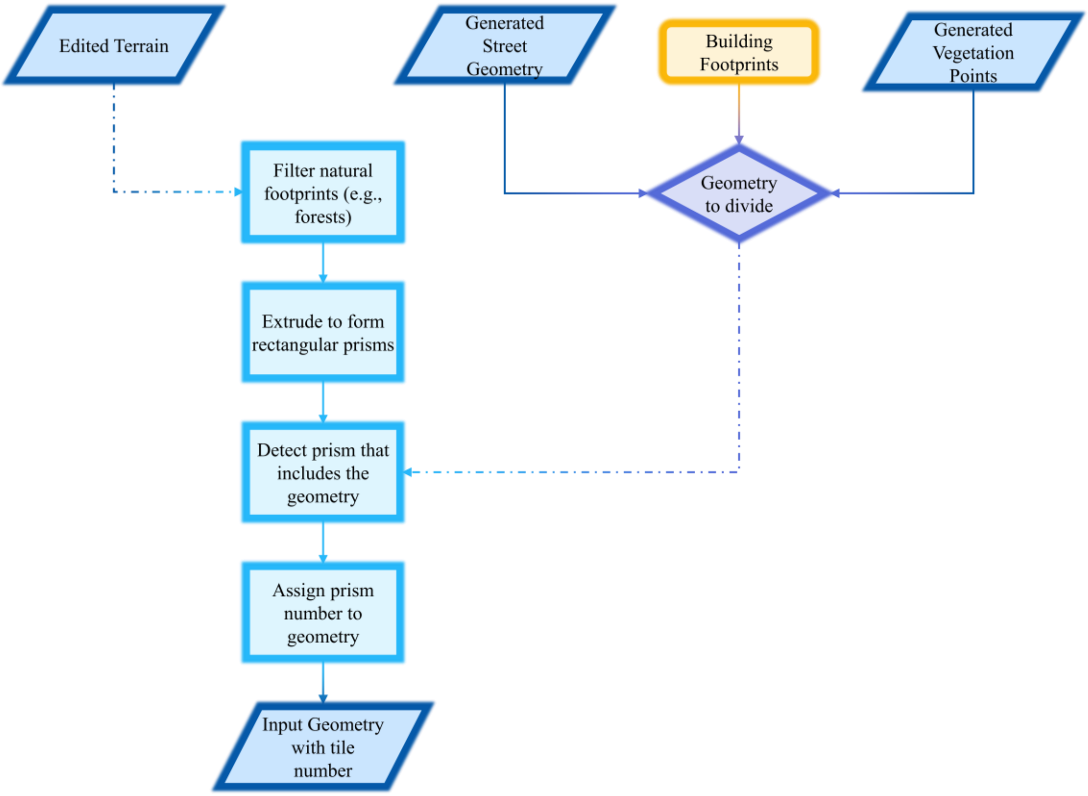


Figure 69: The process used inside the suggested framework for splitting the generated geometry in levels (For a legend, see Appendix).

The Split Generators were created to divide the generated geometries for the streets, buildings and vegetation into levels so that Level Streaming can be applied inside the Game Engine. The ability to apply Level Streaming is important because it enables the loading and unloading of map tiles into memory, so that only the map tiles that are visible to the user consume resources and are rendered during gameplay [297]. All three Split Generators (*StreetSplit*, *BuildSplit*, *NatureSplit*) work in the same way. They receive the terrain and the corresponding generated geometry as input, as well as a parameter, *tile amount*, which defines the number of tiles into which the output environment is to be divided. First, a horizontal grid is created with the same size and height as the terrain and with the same number of subdivisions as *tile amount*. Each subdivision of this grid is a primitive with a specific primitive number. This number is additionally assigned as a variable to each primitive. The grid is then transferred to the minimum height of the terrain and each primitive is extruded to form rectangular prisms with a very large height to ensure that the generated geometry lies within their bounds. After the extrusion, all primitives of each prism automatically have the primitive number as a defined variable, which makes it easier to identify the primitives belonging to each prism. Within a loop, each prism is isolated and the primitives of the generated geometry that lie within it are detected and given a variable, *tile number*, which corresponds to the number of the prism.

While *StreetGen* splits the fully generated street geometry, *BuildSplit* and *NatureSplit* assign a *tile number* to the primitives of the building footprints and the vegetation copy points, respectively.

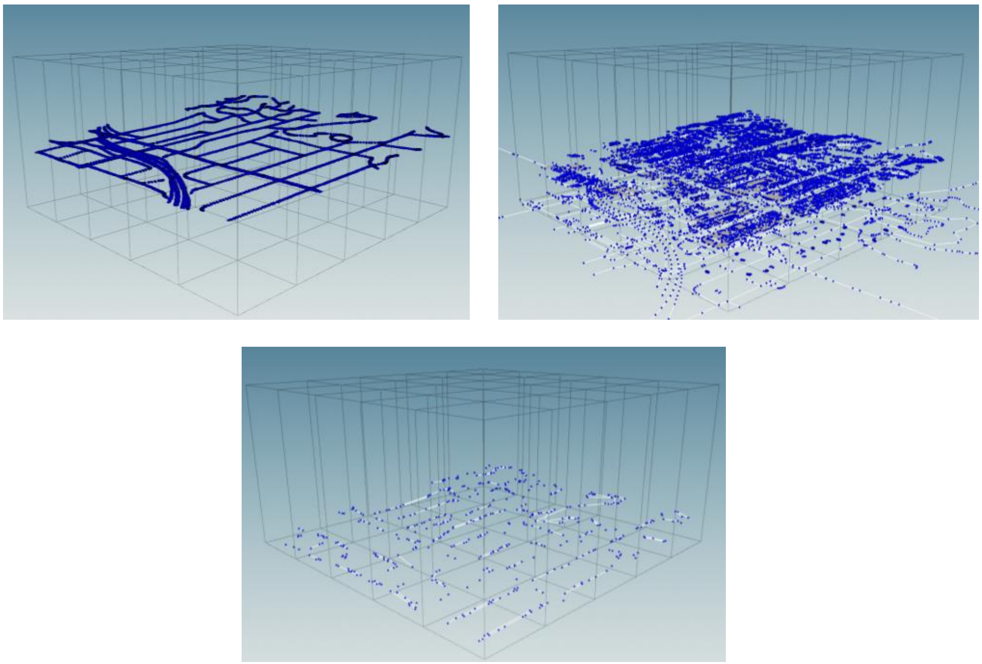


Figure 70: The generated streets (up, left), buildings (up, right) and vegetation copy points (down, middle) are split based on the terrain levels.

4.7 The Level Generator

After the generated geometry of the streets, buildings and vegetation has been divided into levels, they are provided to the final generator. Inside the Level Generator (*LevelGen*), geometries of the same level are merged for the application of the World Composition feature of the Unreal Engine. This implies the assignment of the parameters that the Unreal Engine can recognize and interpret. These include the parameter that defines the attribute containing the level number (*unreal_split_attr*), the parameter that defines the path within the Game Engine that is used to store the levels (*unreal_level_path*) and the parameter that defines the output of the baking process, which can be a landscape actor or a landscape streaming proxy (*unreal_landscape_tile_actor_type*).

New defined variables	
Attribute name	**Type**
unreal_split_attr	detail attribute
unreal_level_path	primitive attribute
unreal_landscape_tile _actor_type	primitive attribute

Table 16: New defined variables that were used to combine the generated geometries into levels for the application of World Composition inside the Unreal Engine.

Figure 71: A part of the Hollywood area in Los Angeles generated with the suggested procedural framework inside the Unreal Engine.

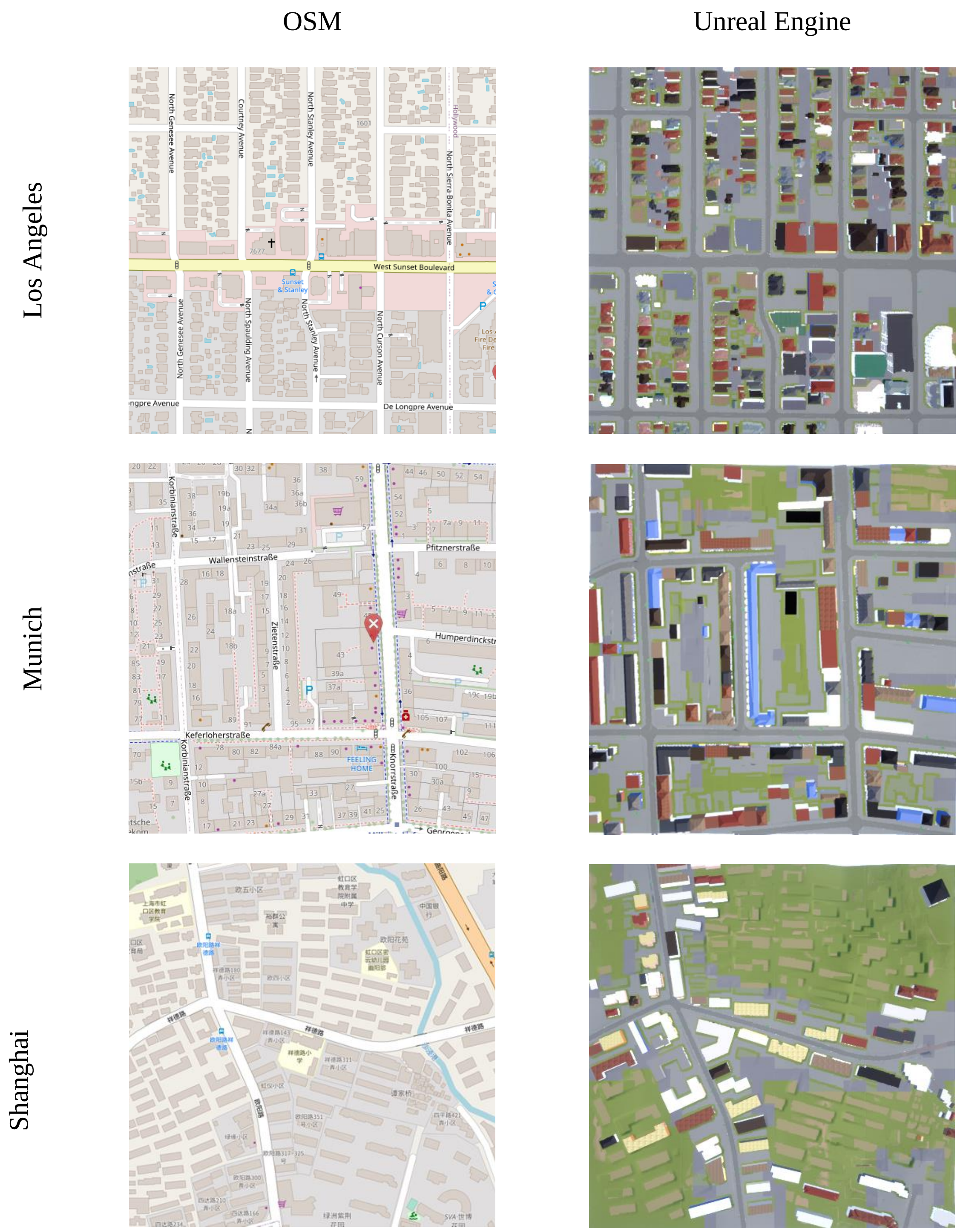


Figure 72: Screenshots of the OSM map of three different locations in Los Angeles, Munich and Shanghai and the corresponding generated environment in the Unreal Engine.

4.8 The User Interface

During the current process of preparing the prototyping experience, designers receive the ready-to-use simulation after just describing the design stories and the environment to the developers. Switching from this process to one that requires more independence from the developers should aim at minimizing the time invested in the designers' interaction with the developed framework to achieve their willingness to use the new process. Based on this design goal that was acquired after the initial studies, a User Interface (UI) was created using the Unreal Engine.

The UI was based on the flowcharts in Figures 23 and 24, but it was extended by adding some basic options to generate a complete driving simulation. As a first prototype, the UI follows a serial structure, where one menu follows the other, starting with the environment, its weather and time conditions and continuing with the characters, the vehicle and the camera position. In each of these menus, there is a *Back* button that allows the user to change their options selected in the previous menu.

To this end, design stories are defined in the first menu the user sees by clicking on the only existing button on the screen. Further buttons are then displayed, each representing a design story. To begin the description of each design story, the user must click on it. This leads to the next menu, where the user can choose between three options for selecting an environment.

The first option shows a map that was created using the API of Mapbox and loaded into the UI from a local HTML file. Radio buttons in the upper right corner of the map window can be used to change the style of the map, and a route navigation widget on the upper left corner can be used to move the map view directly to the desired location, as well as define a certain route they wish to experience.

After defining the route, a list of maneuvers possible on this route and their respective duration is displayed. As concluded from the studies, time is crucial during a presentation and information about the duration of each maneuver can be helpful for designers. The user can also manually move the map and adjust it so that that the part visible through the map window is the desired area. Then the user can click on the *Proceed* button to officially select the area and continue with the description of the design story.

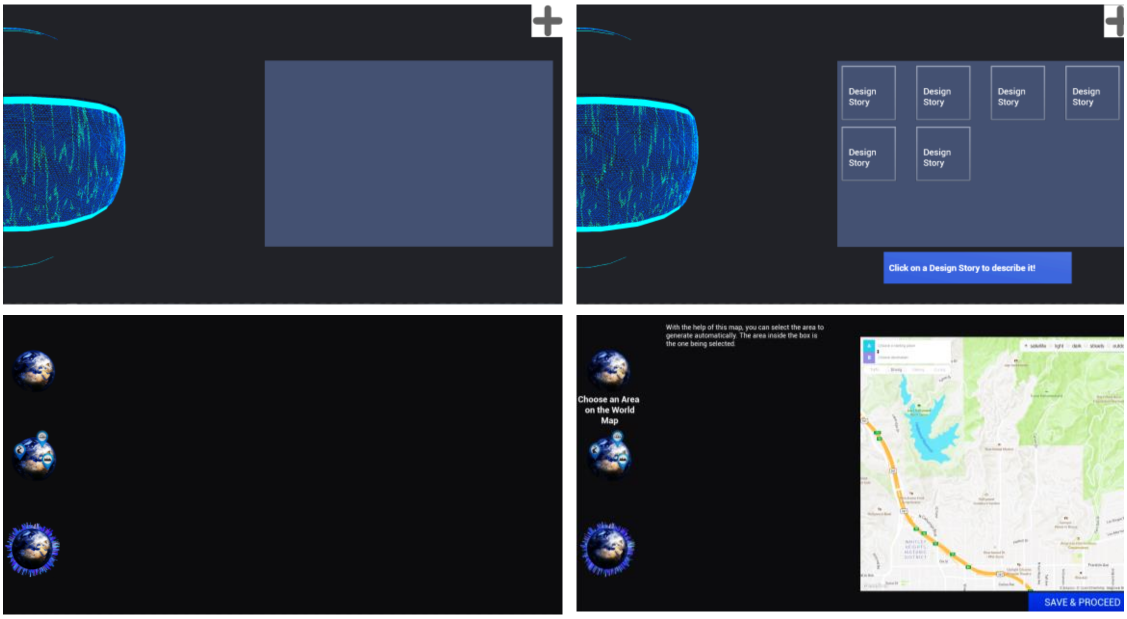


Figure 73: Up: Menu to define new design stories. Down: Menu with three options to select the world area.

The second option to select the world map area was created to support designers in finding the area in the world that best serves their design story if the designers are not sure which world area to choose or in exploring new potentially suitable areas. As the studies showed, such an environment should have the right aesthetic context by being a location within the target market and the right functional context by providing the right situation (e.g., facilitating a certain driving maneuver or providing a certain type of road). Based on the used terms inquired during the first studies, search options are offered within this second option for the environment selection that are divided into two categories. The first type of search parameters refers to natural elements, i.e., mountains, forests and bodies of water, with corresponding sliders for the size of the area. The second type of search parameters includes street-related options. The first helps the user to describe a desired intersection by selecting the number of intersecting streets and the types of the two streets (e.g., primary, residential, motorway) that are important for the maneuver (i.e., street from which the driver enters the intersection and the street on which the driver leaves it). The second option is a slider that allows the user to select how curvy the desired roads should be while the third option defines the height level of the bridge required for the design story. The same map window used also for the first option of environment selection is on the right side of the menu and shows the area of the world to be searched for the selected search elements. After the search, the set search parameters can be seen on the map with their respective symbol and with a legend that shows the symbol assignment. After finding the important elements at the desired location and optionally fine shifting the map to the exact center point of choice, the user can proceed with the same *Proceed* button.

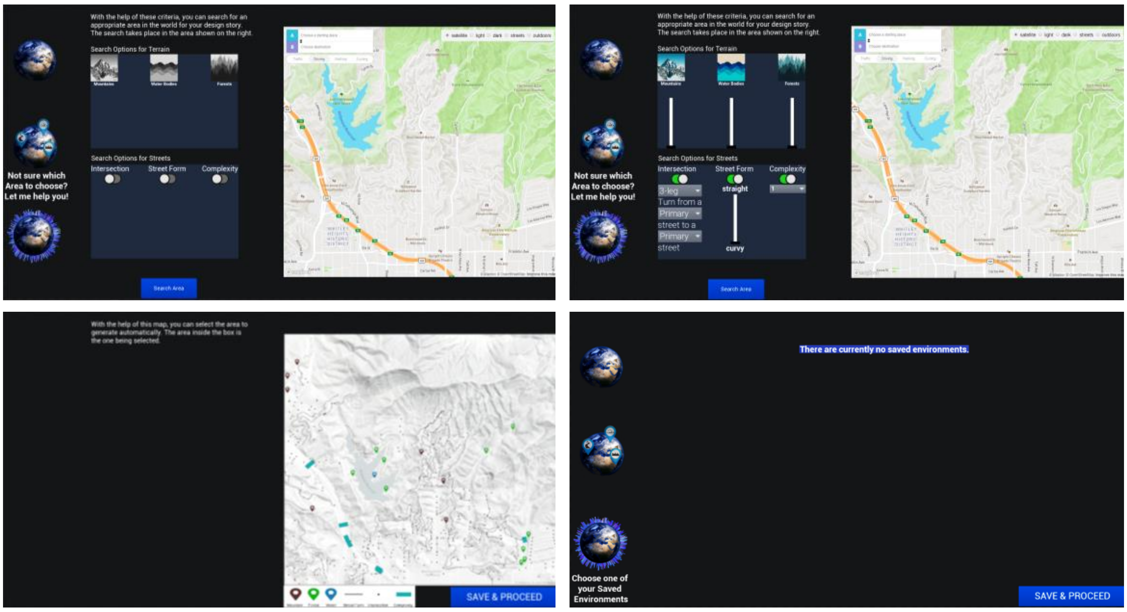


Figure 74: Up: The menu that enables the search for certain features in the selected area. Down, left: An example of the search result. Down, right: The third option of selecting a world area.

After clicking the *Proceed* button, the geographic coordinates which are programmed to be updated in the URL of the HTML file, are stored as a variable for future use.

The third and last option for the environment selection provides the user with the possibility to select one of previously selected environments.

The right color temperature, created by the right sun position and the way the light is scattered by the atmosphere and its particles, is important for the designers, as the exterior colors and lighting influence the evaluation of colors of the car's interior and exterior. Therefore, after moving on to the next menu, the user is confronted with three drop-down buttons. These can be customized to select a specific date and time, as well as specific weather conditions. The *Sun Position Calculator* plugin of the Unreal Engine [298] was used to reproduce the correct sun position with an accuracy that correlates with the geographic coordinates, date and time.

The next menu allows the user to select the characters that participate in the design story. The characters the user can choose from were created using the Unreal Metahuman Creator [299] to cover many ethnicities and age groups, as well as both genders. The purpose of this extensible library of metahuman avatars is to cover most, if not all use cases by adding new avatars as needed. As a prototype to provide the designers with a vision of the complete process, the selected avatars are spawned directly inside the vehicle after the description of the design stories is completed. Within the menu, a further option is offered to define the avatar to be used as the driver.

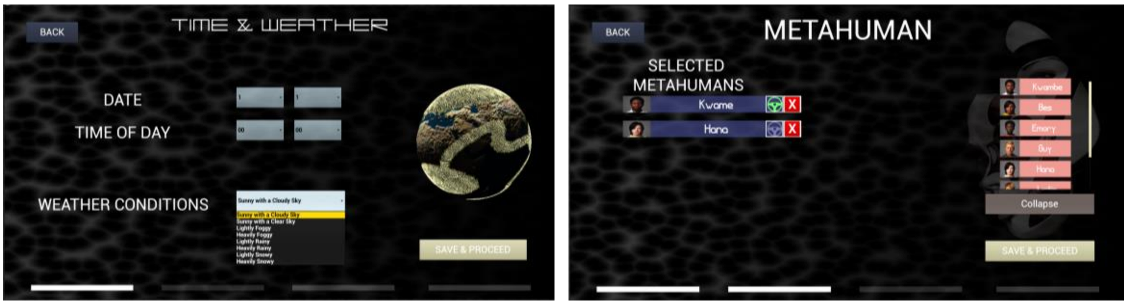


Figure 75: Left: The menu for the selection of the date, time of day and weather conditions. Right: The menu for the selection of the characters.

In the last menu, the user can select the desired vehicle from an extensible library using a drop-down menu. The selected vehicle is displayed in the 3D view next to the option. Below this, a second drop-down menu offers the user the selection of a camera perspective from three standard options: a) first person perspective, b) shoulder perspective and c) dashboard view from above. The three default options are based on the designers' responses in the interview studies. By selecting the camera perspective, the 3D view of the vehicle is updated accordingly. In the event that none of the default options fulfils the designers' requirements for the camera perspective, another menu can be activated that offers the user the possibility to move and rotate the camera relative to the car. The new camera perspective then becomes available in the drop-down menu.

Proceeding further leads the user to a demo that contains a terrain, simple streets and buildings in box shape. The simulated weather conditions and time of day correspond to the user's selected options. Finally, the selected avatars are placed in the corresponding car, which the user can drive with the up, down, right and left arrow keys.

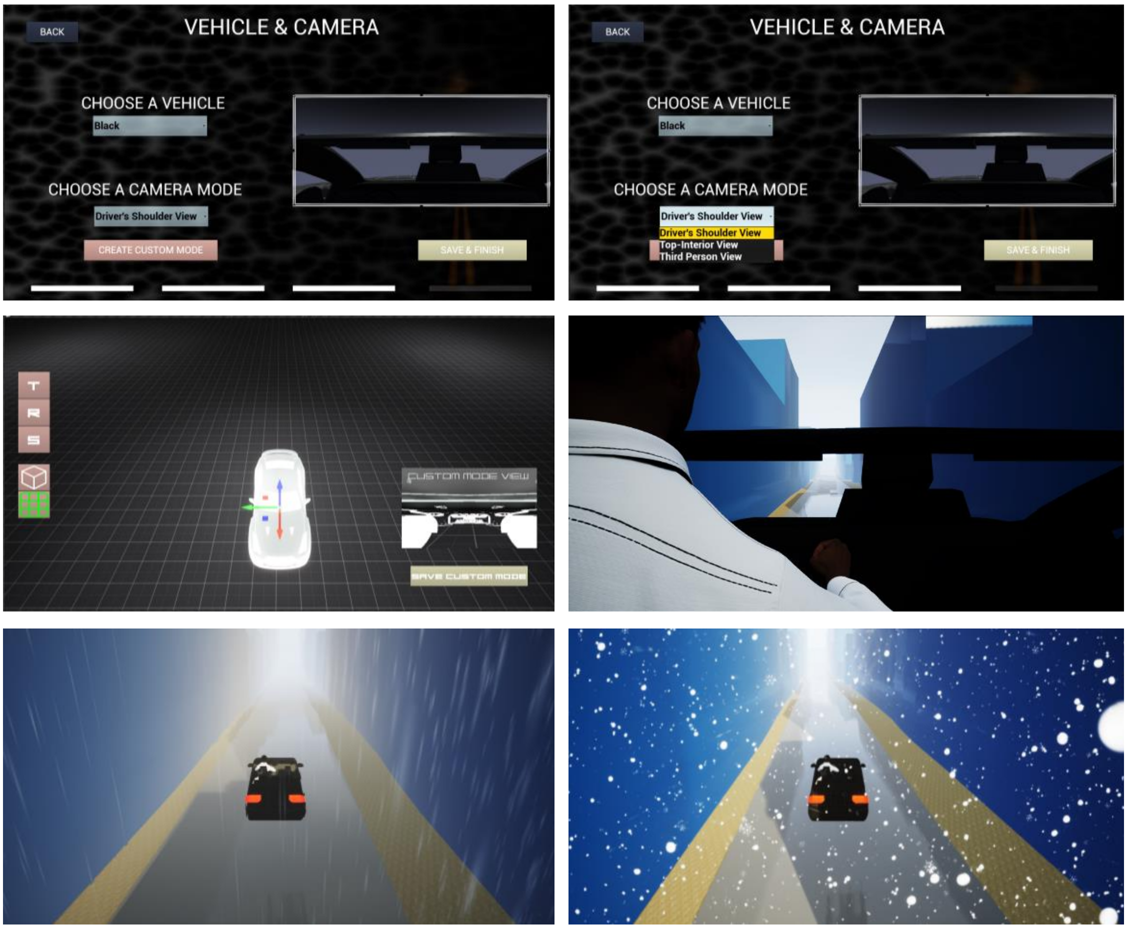


Figure 76: Up: The menu for the selection of the vehicle and the camera perspective. Middle, left: The menu for the customizable camera perspective. Middle, right and down: A sample of output simulations.

5 Experimentation & Results

After creating a framework that is capable of generating an environment with streets, buildings and vegetation and provides developers with the potential to quickly adjust the containing assets, certain important parameters need to be examined. The first is the robustness of the procedural framework that should allow for no or at least a remarkably low number of geometry errors that can be easily fixed after the generation. The second parameter defining the created framework should be the suitability of its results for use in design presentations and evaluations. Finally, one last parameter should not be missing, namely the ability of the framework to be used by automotive designers and developers. These parameters correspond to the following research questions:

- How robust are the procedural generators?
- Are the virtual environments generated with the created framework suitable for automotive design experiences?
- Does the interface allow designers to generate their own virtual environments, and does it provide all essential functionalities?
- Do designers want to use the proposed process?
- Are the resources required by the proposed process to generate virtual environments less than the resources spent during the current process?

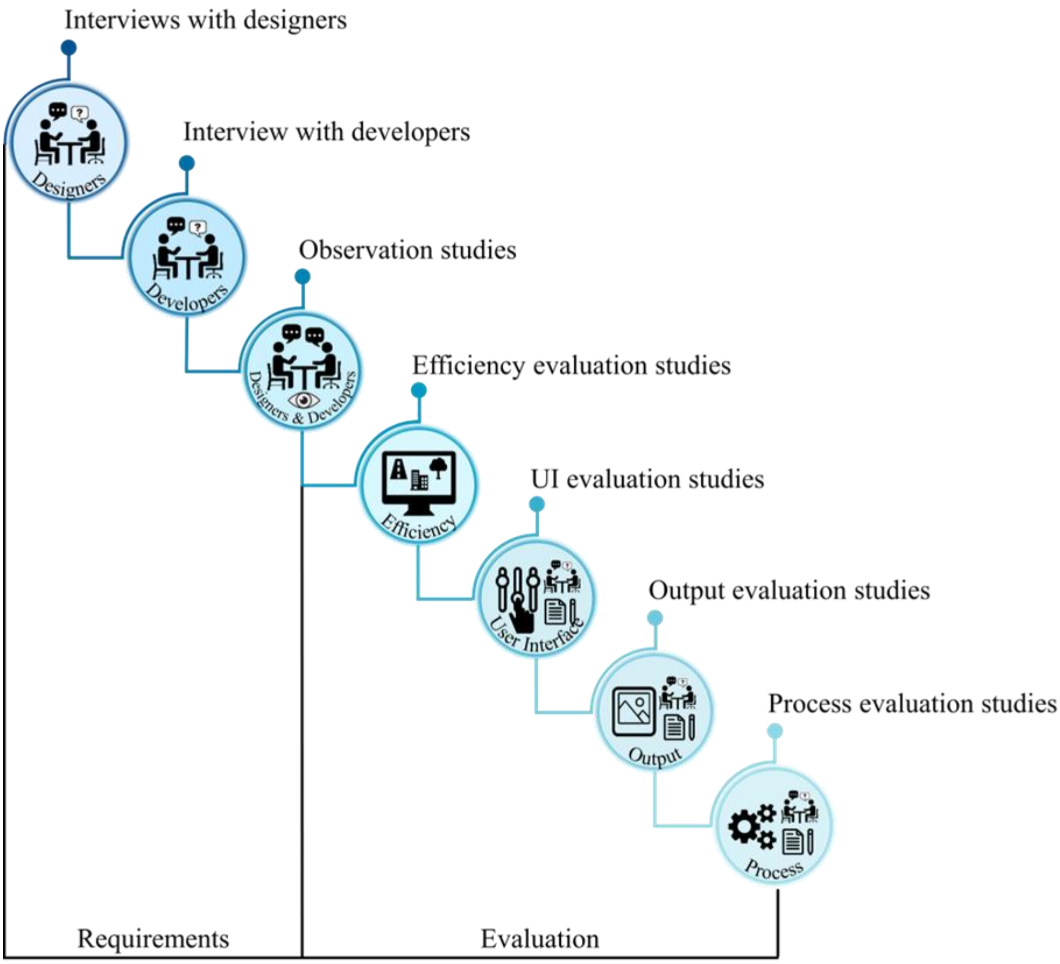


Figure 77: Overview of all conducted studies.

5.1 Procedural Generator Efficiency

To evaluate the efficiency and robustness of the generators as procedural tools, geometry samples were created using the street and building generator, *StreetGen* and *BuildGen*, and checked for modelling errors.

5.1.1 Efficiency of the Street Generator

Procedure and Measurement: Thirty randomly selected areas were generated using the described procedural generator framework. These areas belong to the three most requested countries, i.e., USA, Germany and China. The procedural generation for these 2.25 km² large areas lasted about 25 minutes, depending on the amount of contained elements, on a laptop with the following technical features: Intel(R) Core(TM) i7-8750H CPU (2.21 GHz), 64-bit operating system, x64-based processor, 32.0 GB RAM and NVIDIA GeForce RTX 2080 with Max-Q Design. The generated samples were evaluated according to the number of errors they were created with. The investigation focused on errors related to the consistency of the street geometry (e.g., holes, misplaced street elements), as well as to geometry errors (e.g., non-manifold geometry). To quantify the total sum of errors, the parts into which the development of the generator was naturally divided were used as error categories. These are: 1) Lane geometry, 2) intersection corners, 3) lane directions, 4) lane markings, 5) street elements, 6) terrain adjustment and 7) bridges.

Results: The results showed a congruent and clean street geometry. A possible reason for this is the fact that the generator can only provide one result, which minimizes the potential for error. Although this means that the generator has a narrow expressive range, this is still an important prerequisite for the street generator of our case. For each input, only one and always the same result needs to be generated which fulfills the requirement for realistic driving conditions, as this result is based on real map data.

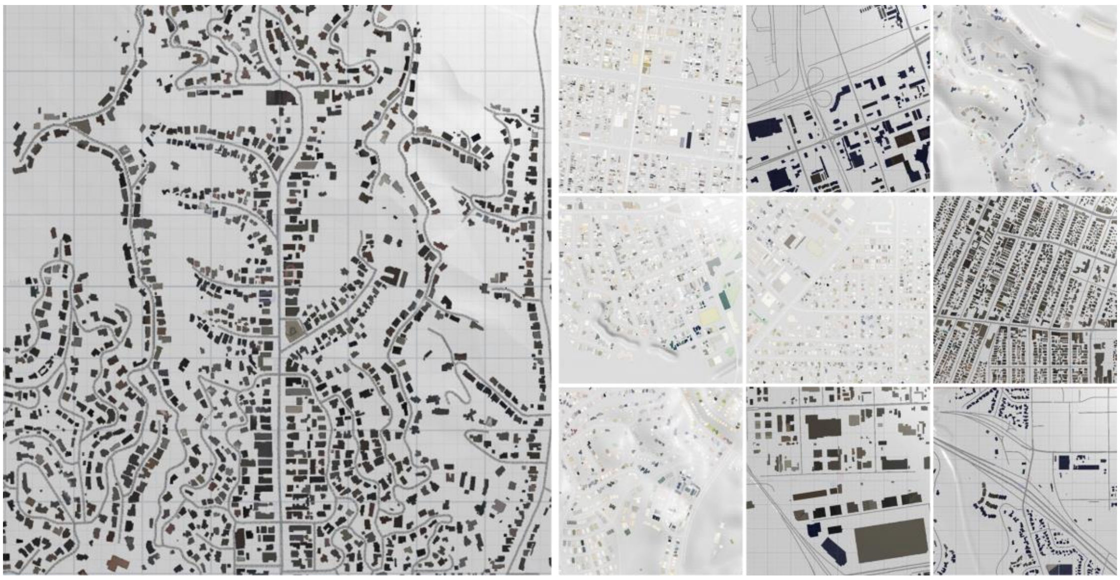


Figure 78: An overview of the test samples of generated environments located in Los Angeles.

Figure 79: An overview of the test samples of generated environments located in Munich (up) and Hongkong (down).

5.1.2 Efficiency of the Building Generator

Important goals for the efficiency of the Building Generator (*BuildGen*) included the ability to create congruent buildings and characteristic façades of representative architectural styles. The resources needed for generating and rendering the buildings, as well as the required input should be kept as low as possible. Finally, the Building Generator should facilitate the easy addition of new architectural styles and assets.

Procedure. To evaluate the extent to which these goals are achieved by the developed Building Generator and its workflow, a computation study was conducted. During this study, 1000 buildings were generated from a random area using *BuildGen* on a laptop with the following characteristics: Intel(R) Core(TM) i7-8750H CPU (2.21 GHz), x64-based processor, 64bit operating system, 32.0 GB RAM and NVIDIA GeForce RTX 2080 with Max-Q Design. The generated buildings were evaluated based on their congruence as defined by Tutenel et al. [226]. They performed an evaluation of complete buildings because they wanted their buildings to be both congruent and consistent. In this case, the generated buildings are not intended to be entered, which leaves us with the definition of congruent buildings. Therefore, after the buildings were generated, they were examined for inconsistencies on the façades. These inconsistencies occur if façade modules have not been placed correctly or overlap. Additionally, due to our focus on architectural styles, the buildings were also examined for the presence of the characteristic architectural elements for the corresponding style.

To be able to observe the different architectural styles and their variations, a second computation study was carried out on the same laptop. For each architectural style, a building was generated in three variations of the same style.

Another computation study was performed on the same laptop to count the computation time and number of primitives for each architectural style on the same generated building with a 991.146 m^2 footprint and three floors.

Congruent Buildings. The generated buildings showed no signs of incongruence. This is the result of three factors. The first is the exact calculation of all position vectors for each façade module. During the calculation, the surface of each wall is divided into the individual areas for each façade module. The next factor is that additional care is taken to write parameter values in the CSV tables in a way that ensures the variety of the façade modules, but also avoids incongruities (e.g., the overall dimensions of the exception window must not differ too much from those of the standard window). Finally, even in the case that something goes wrong (e.g., the geometry of a roof is not suitable for the correct placement of roof windows), rules were defined to identify the problem and find a solution that achieves a congruent output (e.g., the roof is generated without windows). These factors keep the potential for inconsistent, overlapping or incorrectly sized elements low.

Architectural Styles. *BuildGen* was designed to generate buildings of different architectural styles using different façade modules. There are, of course, many more variations of each architectural style that we selected, and the part of the characteristic appearance also depends on the shape of the building which is provided by the OSM footprint. The results of this study are shown on Tables 11-14 from Chapter 4.4.4. This is an evaluation that also needs to be conducted by interviewing the designers, as in Chapter 5.2.3. Nevertheless, there is a clear difference between generated buildings of different styles and the characteristic style elements that we explored in Chapter 4.4.2 are present.

Computation Time and Polygon Count. The results of this study can be seen in the next Table (Table 17). The 1000 generated buildings from the first computation study lasted 45 minutes and the generation time of each architectural style for the used building varies from 1.09 seconds for skyscrapers to 19.59 seconds for the modern style. We observed that the computation time is affected by the number of the same assets, as well as the number of different façade module categories that are placed on the façade. The polygon count is influenced by the assets used for the generation of the façade modules, most of which were photogrammetric assets that have a high number of polygons. The polygon count of each architectural style ranged from 166 for skyscrapers to 24629 for the Neo-Renaissance style, which contains many decorative elements.

City	Architectural Style	Computation Time (s)	Polygon Count
Los Angeles/ Munich/Shanghai	Skyscraper	1.09	166
Los Angeles	Beaux-Arts	10.976	6000
Los Angeles	Victorian	1.725	4942
Los Angeles	Craftsman	2.408	32493
Los Angeles	Spanish Colonial Revival	2.351	14846
Munich	Neo-Renaissance	1.985	24629
Munich	Neo-Classicism	16.466	17714
Los Angeles/Munich	Modern	19.593	15380
Shanghai	Modern	3.14	6617
Shanghai	Tong Lau	10.805	8898
Shanghai	Tang	8.068	5977

Table 17: Generation time and polygon count for each architectural style for a building with three floors and an area of 991.146 m^2.

Automotive Designer's Effort. This is also an evaluation that must be conducted with designers and developers, as we did, the results of which can be seen in the next subchapters. Nevertheless, we can already conclude that the effort of the automotive designer during the generation remains minimal, as not only *BuildGen* but the entire procedural generation framework only requires the geographic coordinates and the zoom factor to generate a default environment. In addition, there are all the methods we described in 4.4.3 that show the simple and quick ways to add new assets and define new architectural styles that also keep the developer's effort at a minimal level.

Time and Cost for the Addition of Styles. Comparing the resources required to manually generate a complete environment from scratch with those of an environment using this procedural framework, we can conclude that this procedural framework allows a great reduction of the required resources. Even if the graphics quality is not yet as good as in a manually generated environment and more detail has to be added manually after the generation, this difference remains great. Finally, the fast and easy methods to add new assets were designed to offer this exact advantage.

5.1.3 Efficiency of the Nature Generator

The two most important steps inside *NatureGen* are the detection of the climate zone and the automated placement of the vegetation assets. In the first step, the climate zone is determined by calculating the geographic position in the Koeppen-Geiger map. Consequently, it depends on an equation, that keeps the error rate minimal. The second step places the vegetation assets either based on the corresponding areas offered by OSM or next to the street based on the corresponding generated geometry. In the case that the OSM data is used for placing the assets, the risk of collision is automatically minimized, because the data is created in such a way that does not cause overlaps. Even in the case where we do not place assets using the OSM data, we optimized the process by checking for collisions with the generated geometries. We did not encounter any such errors during the debugging and the other efficiency evaluations.

5.2 Suitability of the Process, User Interface and Generated Output

To assess the suitability of the generated output, the User Interface and the whole new process of defining and generating the experience context, we conducted a study consisting of five parts:

1. The user experience evaluation of the current process
2. The pragmatic and aesthetic evaluation of the User Interface
3. The qualitative evaluation of the generated environments
4. The user experience evaluation of the suggested process
5. The technology acceptance evaluation of the suggested process

Participants: Nine designers and six developers (4 females and 11 males; age ranging from 23 to 40, mean 33.3) were recruited. They were also proposed by the development team who collaborates with the designers to generate the experience context. All designers worked in the UI department and had collected 1 to 15 years design experience.

The aim of involving both perspectives represented by the two groups of participants was to increase triangulation [300], as both groups play an important role in preparing the design experience.

Overall Procedure: All studies were conducted in an online session, due to corona-induced limitations and working conditions. Each interview lasted about 60 minutes and started with a demographic survey on age, gender and design experience. After that, we asked the participants to describe the current process of defining and generating the environment. With this in mind, we sent them a User Experience Questionnaire to complete and evaluate the current process (1). The next part of the study was to evaluate the User Interface with the AttrakDiff Questionnaire (2), that was followed by the evaluation of the generated environments with expert interviews (3). The proposed process of defining and generating the environment was evaluated with the User Experience Questionnaire (4), as well as the Technology Acceptance Questionnaire (5). The following subchapters describe the procedure and results of the individual parts of the study.

5.2.1 Suggested Process of Defining and Generating the Virtual Environment: Usability of the Current Process

To find out, whether the created procedural framework and the associated proposed process are a welcome method of creating the Mixed Reality experience by themselves, the participants were also asked to complete the User Experience Questionnaire [301] (see Appendix) and the Technology Acceptance Questionnaire [302] (see Appendix) for the current and the proposed process.

The development of the new process for generating the virtual environments for design presentations and evaluations was primarily aimed at transforming the current process into a method that proves to be more efficient and effective, while providing a positive experience for the participants. The User Experience Questionnaire (UEQ) [301] is used to evaluate the user experience of interactive products and consists of adjective pairs that belong to two scales, the Pragmatic and the Hedonic Quality. Since the proposed process is based on the use of our procedural pipeline and we want to evaluate both usability and user experience aspects of this process, we selected the UEQ. To be able to compare the proposed with the current process, the designers were first asked to evaluate the current process on the questionnaire and its criteria with the help of the Likert scale (values ranged from 1 to 7 and were converted to the range of -3 to 3 for the analysis).

Item	Mean	Variance	Std. Dev.	No.	Negative	Positive	Scale
1	-0.8	2.0	1.4	14	obstructive	supportive	Pragmatic Quality
2	-1.2	0.5	0.7	14	complicated	easy	Pragmatic Quality
3	-1.8	0.8	0.9	14	inefficient	efficient	Pragmatic Quality
4	-0.8	1.6	1.3	14	confusing	clear	Pragmatic Quality
5	-0.6	1.2	1.1	14	boring	exciting	Hedonic Quality
6	-0.6	0.9	0.9	14	not interesting	interesting	Hedonic Quality
7	-1.6	1.3	1.2	14	conventional	inventive	Hedonic Quality
8	-1.6	1.0	1.0	14	usual	leading edge	Hedonic Quality

Table 18: The mean, variance and standard deviation for each scale of the User Experience Questionnaire for the evaluation of the current process of defining and generating the virtual environment.

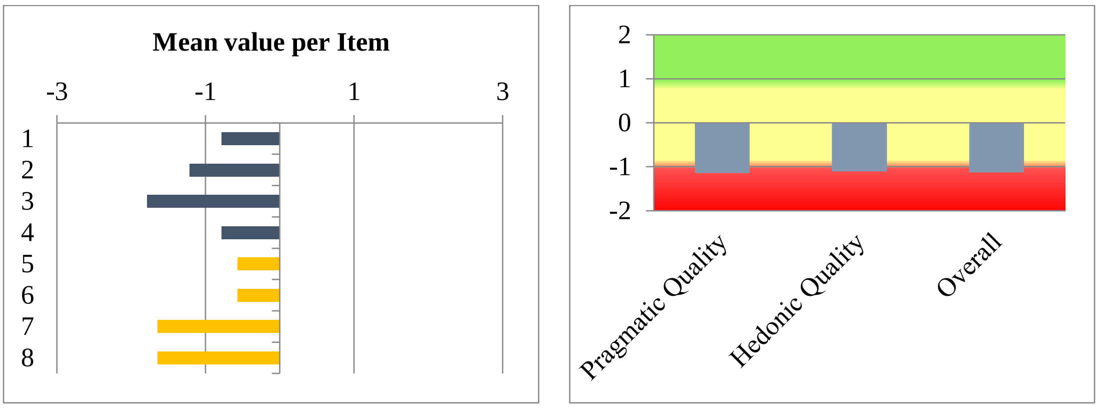


Figure 80: Left: The diagram of mean values of every scale for the current process. Right: The bar diagram of the mean value of the pragmatic and hedonic quality for the current process.

As shown in Table 18 and Figure 80, the UEQ consists of two scales, the pragmatic and hedonic quality, and the validity of the results was verified by calculating the consistencies of the participants' responses between the parameters of each scale to the following resulting reliability coefficients or Cronbach's alpha values:

Pragmatic Quality	
Items	**Correlation**
1.2	0.59
1.3	0.75
1.4	0.71
2.3	0.70
2.4	0.41
3.4	0.44
Average	0.60
Alpha	**0.86**

Hedonic Quality	
Items	**Correlation**
5.6	0.79
5.7	0.60
5.8	0.62
6.7	0.49
6.8	0.40
7.8	0.88
Average	0.63
Alpha	**0.87**

Table 19: The Cronbach's alpha values for the pragmatic and hedonic quality of the current process.

Cronbach's Alpha Score	Level of Reliability
0.0 – 0.20	Less Reliable
>0.20 – 0.40	Rather Reliable
>0.40 – 0.60	Quite Reliable
>0.60 – 0.80	Reliable
>0.80 – 1.00	Very Reliable

Figure 81: The level of reliability assigned to Cronbach's alpha values [303].

These coefficients must be above 0.6 (Figure 81, [303]) to consider the results of the questionnaire as reliable. In this case, the results with coefficients above 0.8 (Table 19) can be interpreted as very reliable.

The designers and developers who participated in this study evaluated the current process with rather negative scores. This was not surprising, as the majority of participants also commented on the current process with corresponding remarks and stories, during the initial interview studies, which were aimed at gathering requirements and analyzing the current situation, as well as during the evaluation studies. This is a natural consequence of the fact that this process has slowly crystallized into its current form based on individual needs and preferences rather than on an official blueprint, that has been processed and agreed upon by all participants.

5.2.2 Suitability of the User Interface

We collected qualitative feedback from the designers and developers that can help optimize the created prototype and quantitative feedback that can describe the pragmatic and hedonic qualities of the User Interface. In this study, the participants were asked to use the interface to create a driving simulation for a design experience in two different scenarios, while thinking aloud. In the first scenario, the participants were asked to use the normal method of selecting a world map area, while in the second scenario, they were asked to search for a suitable world map area with the help of the search feature.

5.2.2.1 Qualitative Feedback

After both objectives were achieved, the qualitative feedback from the designers and the developers was collected in a series of structured expert interviews. Due to the nature of the interviews and the limited number of experts in this new prototyping method, participants were selected from the original group for the collection of feedback based on the “n=12+3” sample size principle (nine designers and six developers: 4 females and 11 males; age ranging from 23 to 40, mean 33.3) [304]. We achieved saturation of the participants’ verbal codes by analyzing the responses of twelve participants and reviewing them based on the interview results of the remaining three participants [305]. The participants were asked the following questions:

1) What did you like or dislike about using the Interface?
2) Would you add more features or functions to the UI?
3) Would you remove features or functions that you consider redundant?

Content Analysis was used to analyze the collected data which was interpreted with the help of the developers’ experience they gained by working with the designers and developing design experiences. This method was considered suitable for the interpretation and presentation of interview data [306]. The statements were categorized as follows:

Additional Features: One developer (PDev1) had a suggestion for an additional option as a criterion for the search feature, namely a motorway with a certain number of lanes (e.g., a 5-lane motorway). They also suggested several options to be added to the UI and considered when creating the driving simulation. These included the option to select the level of traffic and autonomous or manual driving of the vehicle within the simulation. According to the same participant, the route that the user can create in the environment selection menu can be used to create a demo video that can be played after interacting with the UI in which the car can drive autonomously along the selected route. The option to control the level of traffic was also suggested by a designer (PD4), along with the option to select the level of pedestrian flow, as well as the ability to describe a specific narrative with secondary traffic participants.

Current Features: One designer (PD1) liked the sequential menus of the interface: "They are very clear and simple steps. This gradual process from defining the story to then describing it by starting with the environment and then adding the details is very good. It is a very good way to describe complicated things in a simple way". They also suggested that the camera selection mode should be provided within the already generated environment and that the user should be able to move the camera freely. Another designer (PD2) liked the variety of the weather options and referred to their potential of covering all the use cases. The same designer even found the options provided by the UI sufficient and would not suggest adding more. They explained that "the current amount of options is pleasant. It can get too confusing very quickly if the user has to click many buttons". Furthermore, they commented on the usability of the tool: "It is very clear and understandable what the tool is trying to achieve with a few clicks". Finally, another designer (PD4) commented on the current number of options: "I was impressed by how detailed the UI is in terms of options, especially the search function with the criteria to choose from".

Aesthetic Quality: A developer (Pdev1) expressed the wish for a more neutral design. Additionally, one designer (PD1) thought that the company's official font should be used for all menus. Another designer (PD2) suggested more bright colors for a friendlier design. Finally, one developer (Pdev2) suggested that the menus should be aesthetically optimized as they give the impression of a game menu.

5.2.2.2 Quantitative Feedback

To evaluate the pragmatic and hedonic qualities of the interface, the designers were asked to complete the AttrakDiff Questionnaire [307] (see Appendix).

The User Interface was also evaluated by the participants of this study based on its pragmatic and hedonic qualities. To this end, the participants were asked to complete the AttrakDiff Questionnaire [307]. It consists of four scales, Pragmatic Qualities (PQ), Hedonic Qualities I (HQ I), Hedonic Qualities II (HQ II) and Attractiveness, which were evaluated based on a series of adjective pairs (see Figure 84 and Appendix) and the Likert scale (values ranged from 1 to 7 and were converted to the range from -3 to 3 for the analysis). The results can be seen in the following graphs of the mean value for each scale, the mean value of the two primary scales and the mean value for each individual adjective pair.

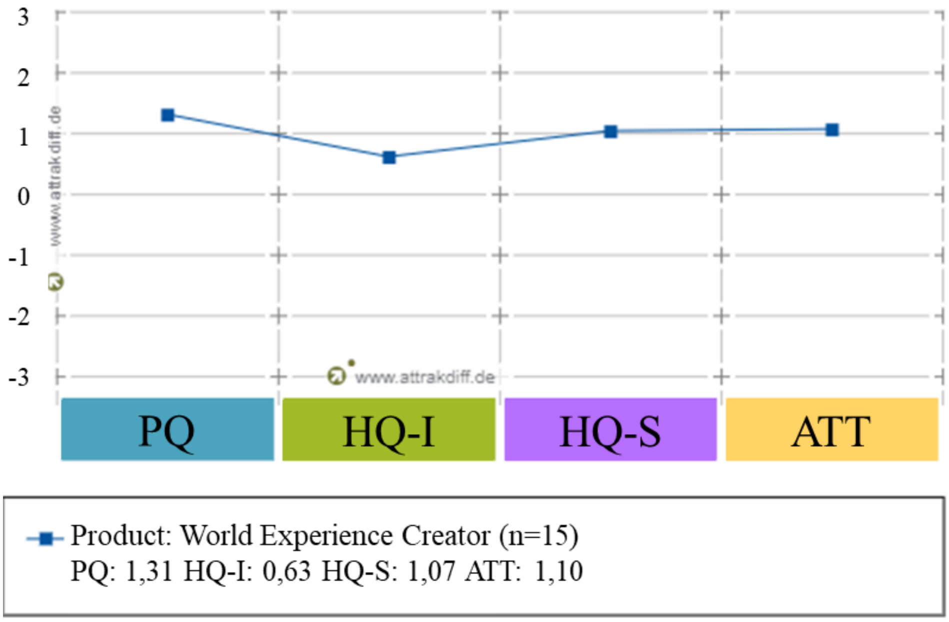


Figure 82: Diagram of mean values for every scale of the AttrakDiff Questionnaire (Pragmatic Qualities (PQ), Hedonic Qualities I (HQ I), Hedonic Qualities II (HQ II) and Attractiveness).

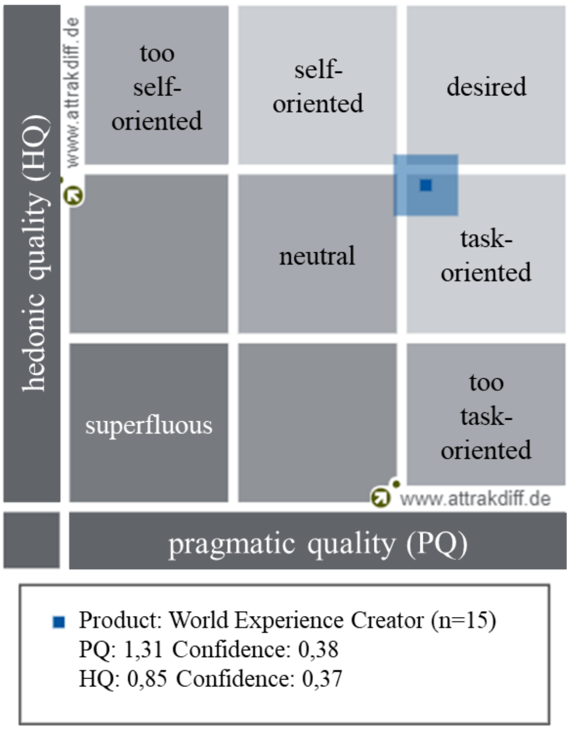


Figure 83: The average values of the dimensions PQ and HQ and confidence rectangle.

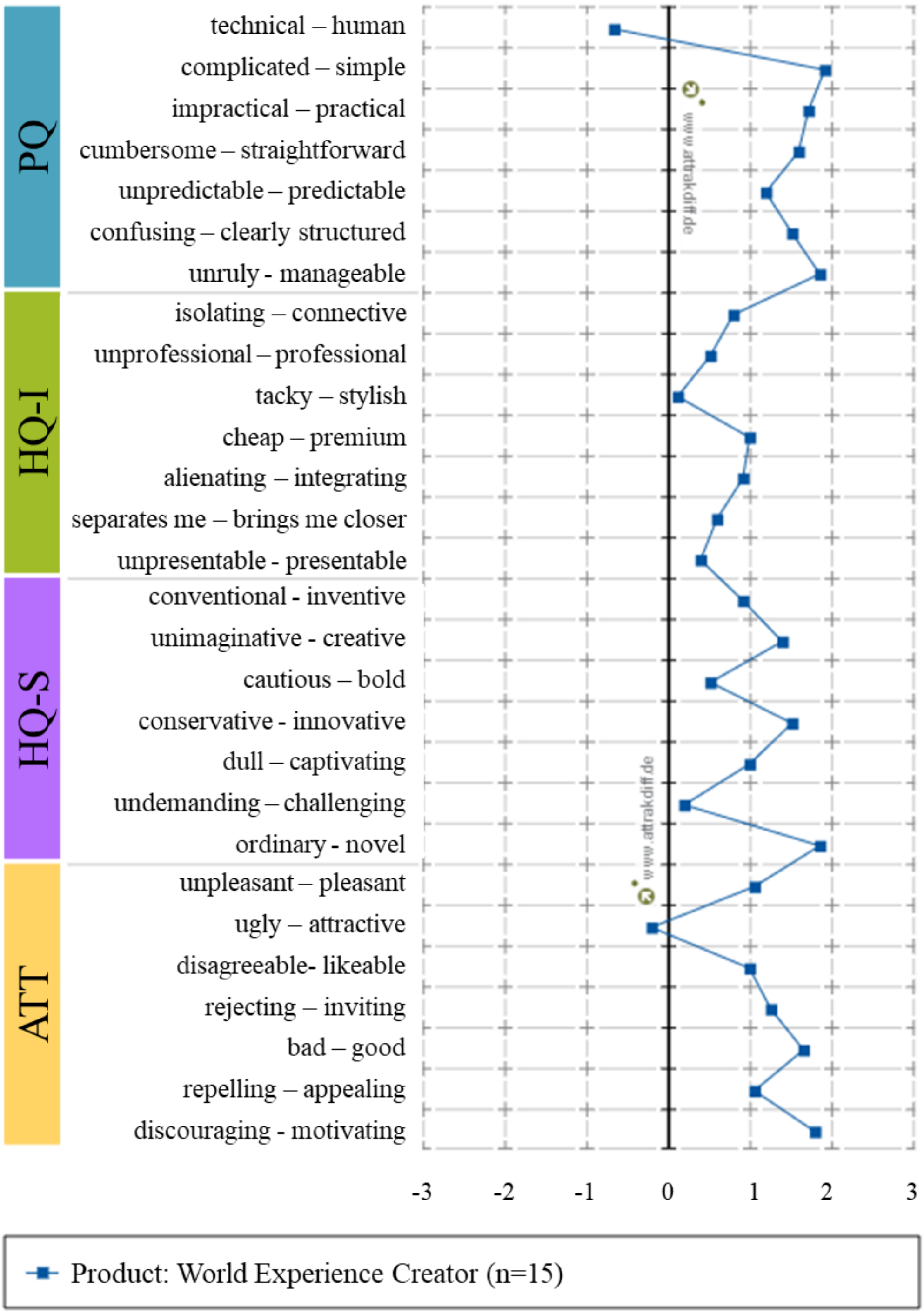


Figure 84: The average value for each adjective pair of the AttrakDiff Questionnaire calculated from a [1;7] - value range to a [-3;3] - value range.

The participants' responses indicate a visible difference between the pragmatic and hedonic qualities of the User Interface. Such a result is in line with our expectations, as the focus of the implementation was on the pragmatic qualities. The User Interface was intended to arouse the interest with decorative elements and icons. However, it was a first prototype that primarily aimed to provide all the functionalities, required for the selection and generation of the virtual environment. The results are also consistent with the qualitative feedback of the participants, which was positive towards the provided options and their number, as well as the functional concept and interaction design. However, as shown in the quantitative results, participants suggested a different aesthetic design that is more neutral and friendly.

5.2.3 Suitability of the Generated Output

After evaluating the UI, we showed the participants three generated environments from Munich, Los Angeles and Shanghai and asked them the following questions:

1) What did they like or dislike about the geometry of the terrain, street, building and vegetation and why?
2) What elements would they add, change or remove from these geometries and why?
3) What do they think of the proposed procedural framework as a tool to create their own environments based on the selection of a map area?

Again, the participants' responses were analyzed using Content Analysis. After discussing the participants' responses with the developers, the following categories of comments and suggestions emerged:

Street Geometry: In terms of structure and visualization, two developers (PDev1 and PDev3) praised the generated street geometry. In their opinion, there is a lot of potential to increase the variety of street materials and textures to improve the quality of visualization. They also proposed the use of materials with a normal map that provide a 3D asphalt effect. According to a developer (PDev1), if no scenario-specific changes are needed, e.g., parking spaces, the generated geometry is sufficient for any design scenario. An important feature is the ability to edit the environment after the generation (e.g., changing textures), as proposed by another developer (PDev2). This option is already available in the proposed environment generation process due to Houdini's pipeline that allows the generation within a Game Engine. However, these changes still need to be made by developers. Another proposal from the same developer (PDev2) was about refining the distance threshold for the Level of Detail of certain assets. No further comments were made other than being satisfied with the street geometry, although there is still a lot of potential to improve the graphics quality.

Building Geometry: The variety presented by the building façades was found to be sufficient by one developer (PDev1). The same opinion was expressed by a designer

(PD4), who was intrigued by the level of detail achieved through the façade modules and the different architectural styles. Suggestions for possible optimizations included the protrusion of roofs from the building walls (PDev3), which was not the case for a particular roof type.

Materials: A further potential for optimization was proposed by three participants (PD1, PD2 and PD3) and was also to be expected due to the short development time. This contained the use of a higher variety of materials, especially for the building walls.

Use Cases: A suggestion for a possible use case in terms of aesthetic context was made by a designer (PDS1). According to this use case, the street should have the appearance of a so-called “party mile” with many neon signs that light up in the night. This can be implemented with the appropriate UI option and the addition of more ground floor façades that are associated with commercial and entertainment uses.

The New Design Experience Development Process: The participants’ reactions to the proposed process were unanimously enthusiastic, which they expressed immediately after experiencing it (e.g., “This is a very powerful tool). An important statement from one designer (PD2) revealed information about the current process of environment generation. They emphasized the trial-and-error nature of the process that designers follow to define the virtual environment, leading to these frequent changes in requirements mentioned by developers in the initial interview studies. They emphasized the inevitability of this process and thus the great compatibility of the generated tool which has the potential to solve the resulting situation by providing the ability to generate an environment in a short amount of time, compared to waiting four to six months. They also appreciated the additional creativity the tool provides by offering a chance to test a design story with different environments. The advantage of reducing the time needed to create a new environment was also highlighted by another designer (PD1). Finally, two designers (PD2 and PD5) felt that the designers’ needs were heard and taken into account during the implementation of the framework.

In summary, all designers and developers were enthusiastic and satisfied with the generated output and expressed the impression that the generation is going exactly in the right development direction. The concept of selecting a map area and automatically obtaining the corresponding virtual environment in a short amount of time was fascinating for all participants, as the current process does not cover all use cases in an efficient amount of time.

5.2.4 Suggested Process of Defining and Generating the Virtual Environment: Usability of the Suggested Process

The same method used to evaluate the current process, was also applied to evaluate the proposed framework as a complete process of selecting, generating and testing the virtual environment. The validity of the results was confirmed with the same calculation of the

Cronbach's alpha values, which showed an acceptable reliability of over 0.7 (Table 21Table 21, [303]). The results show a positive attitude of the participants towards the proposed process, both in terms of its pragmatic and hedonic qualities. Again, this was to be expected based on their responses during the qualitative feedback studies. Some participants wanted such a framework even before they were shown its implementation and were particularly excited to see it realized, as well as its future development.

Item	Mean	Variance	Std. Dev.	No.	Negative	Positive	Scale	
1	2.1	0.4	0.6	15	obstructive	supportive	Pragmatic Quality	
2	2.1	0.1	0.4	15	complicated	easy	Pragmatic Quality	
3	2.1	0.3	0.5	15	inefficient	efficient	Pragmatic Quality	
4	2.3	0.2	0.5	15	confusing	clear	Pragmatic Quality	
5	2.3	0.4	0.6	15	boring	exciting	Hedonic Quality	
6	2.4	0.7	0.8	15	not interesting	interesting	Hedonic Quality	
7	2.5	0.4	0.6	15	conventional	inventive	Hedonic Quality	
8	2.4	0.3	0.5	15	usual	leading edge	Hedonic Quality	

Table 20: The mean, variance and standard deviation for each scale of the User Experience Questionnaire for the evaluation of the suggested process of defining and generating the virtual environment.

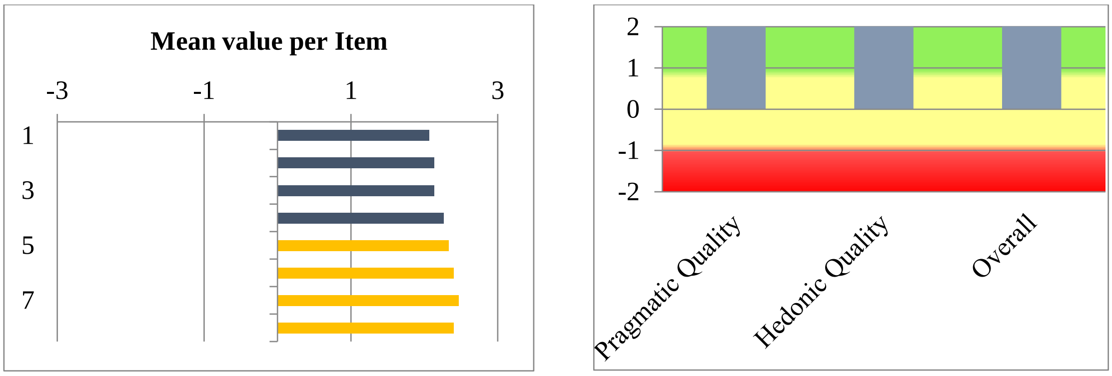


Figure 85: Left: The diagram of mean values of every scale for the suggested process. Right: The bar diagram of the mean value of the pragmatic and hedonic quality for the suggested process.

Pragmatic Quality	
Items	**Correlation**
1.2	0.52
1.3	0.52
1.4	0.46
2.3	0.29
2.4	0.21
3.4	0.44
Average	0.41
Alpha	**0.73**

Hedonic Quality	
Items	**Correlation**
5.6	0.56
5.7	0.30
5.8	0.68
6.7	0.30
6.8	0.27
7.8	0.26
Average	0.40
Alpha	**0.72**

Table 21: Left: The Cronbach's alpha values for each quality of the suggested process. Right: The level of reliability assigned to Cronbach's alpha values [303].

5.2.5 Suggested Process of Defining and Generating the Virtual Environment: Technology Acceptance

To obtain a quantitative assessment of the technical acceptance of the created framework, participants were asked to complete an additional questionnaire. This consisted of questions from the long version of the User Experience Questionnaire and the Technology Acceptance Questionnaires, TAM3 and UTAUT as in [302]. We used this particular method because it takes into consideration the user experience as a determinant of the technology acceptance. Since our goal is not only to increase the efficiency and effectiveness of the process, but also to make it a positive experience for the participants, the questions used in the selected method better fitted the purpose of our study. The Likert scale was also used for this questionnaire (scores ranged from 1 to 7 and were converted to a range of -3 to 3 for the analysis). The results were again valid with coefficients above 0.7 (Table 22Table 21, [303]).

Representative of the qualitative feedback, the results of all scales were similarly high with a lower result for the *Output Quality*. As the designers are used to high quality environments that are manually created in four to six months with the corresponding result, this was an anticipated reaction. Nevertheless, they understood that the value of the framework lies in the automated generation, the parametrizability of the generated environment and the minimal resources required for the generation. As can be seen by the graph (Figure 86), *Efficiency* and *Novelty* were rated highest, followed by perceived *Ease of Use*, *Perspicuity* and *Stimulation*. Based on the participants' excitement and qualitative feedback, they understood the aim and method of using this framework, observed it as a new and innovative method and found it easy to use. They are not all ready to lose their dependency on the developers and need some time of adjustment to start applying this method in projects, as can be concluded from the *Dependability* score.

Finally, most participants did not know when and how often they would need to use this framework in the future. This leads to the lower scores of *Behavioral Intention* and *Use*.

	M	SD	1	2	3	4	5	6	7	8	9
1. Efficiency	6.2	0.56	*0.74*								
2. Output quality	5	0.93	0.41	*0.71*							
3. Perspicuity	6.07	0.26	0.39	0	*0.76*						
4. Dependability	5.93	0.59	0.47	0.52	0.5	*0.7*					
5. Stimulation	6.27	0.46	0.33	0.34	0.44	0.33	*0.73*				
6. Novelty	6.4	0.51	0.45	0.46	0.33	0.33	0.74	*0.7*			
7. Perceived usefulness	6.13	0.74	0.27	0.62	0.32	0.18	0.31	0.42	*0.7*		
8. Perceived ease of use	6.13	0.52	0.39	0.15	0.46	0.5	0.44	0.33	0.14	*0.81*	
9. Behavioral intention	5.73	0.7	0.51	0.44	0.5	0.64	0.46	0.52	0.35	0.69	*0.81*
10. Use	5.71	0.73	0.53	0.53	0.51	0.44	0.07	0.19	0.62	0.32	0.51

Table 22: The mean, standard variation and correlation coefficients of the Technology Acceptance Questionnaire scales (The Cronbach's alpha values are the first number of each column in italic).

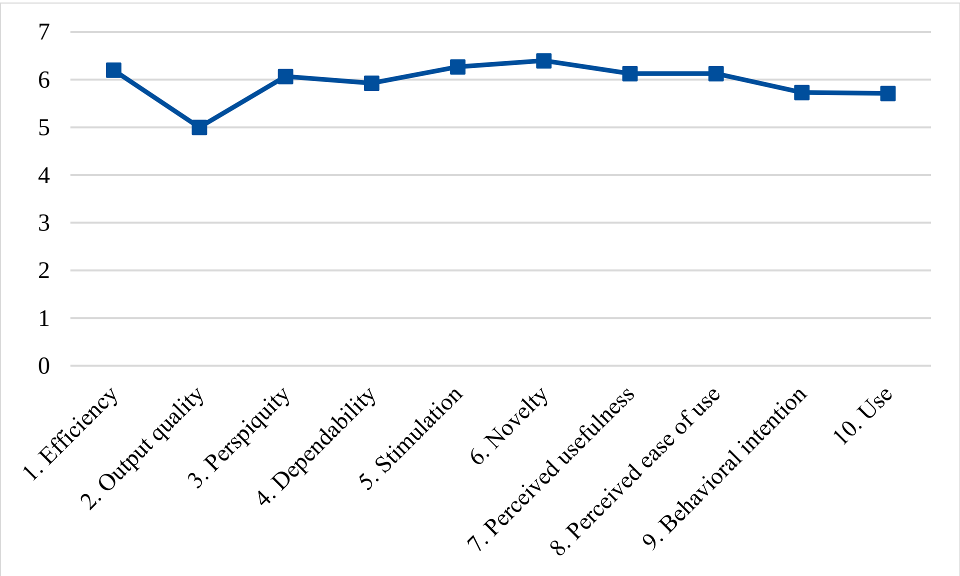
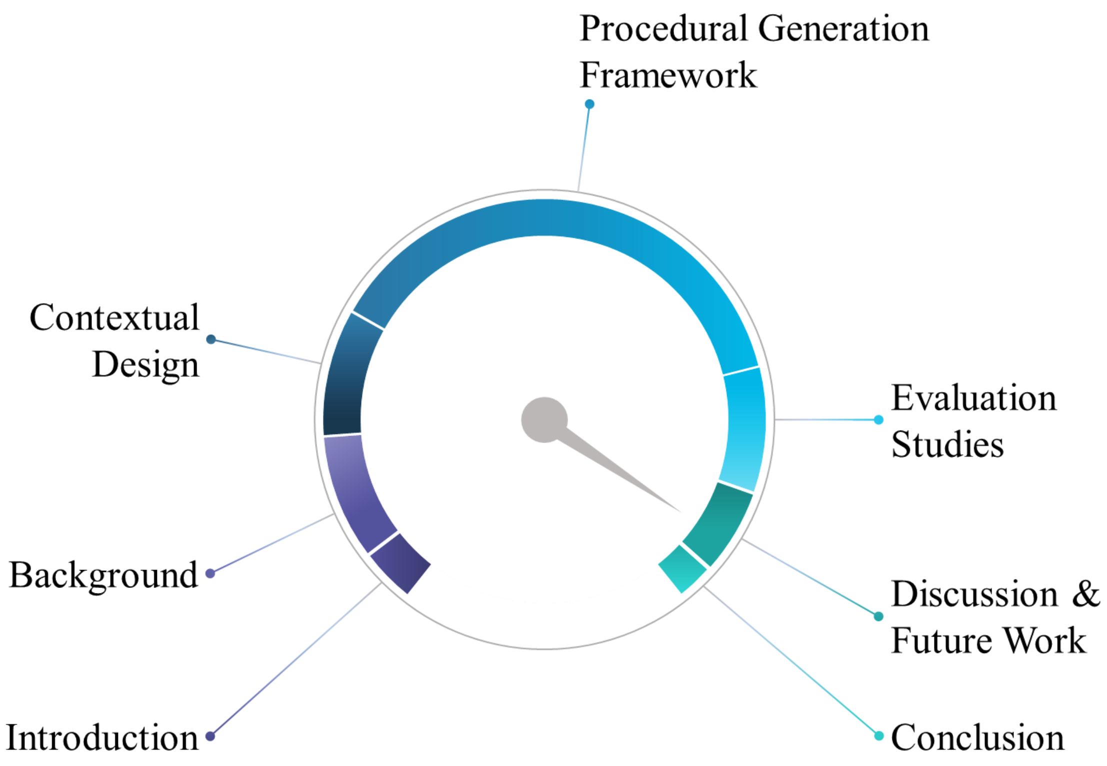


Figure 86: The mean diagram of each scale of the Technology Acceptance Questionnaire for the evaluation of the suggested process.

III. Discussion, Future Work and Conclusion



6 Discussion & Future Steps

With the help of procedural generation, the created framework executes a series of algorithms for modelling the terrain, the corresponding streets, the buildings and the vegetation using OSM data as input. In this chapter, the extent to which the created framework fulfils the collected requirements is discussed based on the results of the conducted studies and computation tests.

6.1 Procedural Generation Efficiency

6.1.1 Location of the Target Market and Realistic Conditions

We can confirm that the use of OSM and Mapbox data was the right decision for several reasons. On the one hand, it provides realistic driving conditions as it contains representative data of the real street network with sufficient information (e.g., lane number, direction, markings and traffic signs) to reproduce it. Even if the data is insufficient for some areas in the world, the procedural algorithms we created can still produce a result that has the representative look and fill of the real location.

On the other hand, the combination of Mapbox and the OSM data provides the designers with a complete world map, which contains all the possible use cases of virtual environments they require, as their use cases are always connected to the real world. In terms of the functional context of the experience, the input data is sufficient, as it can facilitate an experience related to any street. In terms of the aesthetic context, which depends on the atmosphere created by the virtual environment, the input data is sufficient as it contains a variety of information about other environment elements (e.g., buildings, nature, street furniture, culture). The chosen input data provides the options that are essential to facilitate design stories and were mentioned as requirements in the initial interview studies:

- type of area (e.g., urban/suburban),
- road form (e.g., curvy)
- road type (e.g., primary highway),
- terrain form (e.g., hilly),
- driving maneuvers and
- an area reflecting the vehicle's character (e.g., mega city for the character of a gentleman).

The realistic driving conditions are also enabled by our method of procedurally generating only one possible output from the same input data. This reduces the expressive range of the generator to a minimum, which makes it not an efficient procedural generator in general, but a very efficient generator for creating virtual environments for aesthetic automotive experiences.

Nevertheless, the building generator can produce multiple outputs from one input depending on the defined parameters for the façade module generators and on the height and area of the building.

In the event that the OSM information is not complete, we enriched the framework with instructions to automatically place street elements without necessarily expecting the relevant information. In addition, in case certain residential buildings are missing, our plan for the future is to leverage the building footprint pattern of the same area by transforming it into other similar building patterns with the help of the Wave Function Collapse algorithm.

6.1.2 Reduced Resources

One of our design goals was to reduce the resources required to generate the environment. In terms of development, 4 to 6 months cannot be compared with the development time of 25 minutes, even with the additional manual work of adding details. This is a great advantage, as more designers can use this prototyping method simultaneously without overloading the development team and, as the designers noted, the framework can be used to generate and test more than one environment for one design story. The latter is important as designers need a specific amount of time to decide which environment is best for experiencing their design. Consequently, the suggested process has the potential to support designers already in this trial-and-error phase of finding the most suitable environment, thus further speeding up the complete process. Moreover, the process allows designers and developers to have more time to make the final subtle changes and additions, which are also important, as they increase the quality of the experience. The required resources in terms of cost are also reduced, as the budget for a project is not consumed for the generation of a complete environment, but only for additional assets and their placement in the scene.

In addition, the generators were created and stored as assets, which are used by a Procedural Generation Graph (PDG) system, that was also built inside Houdini. This is a graph system that is also based on nodes. Given an asset, it generates actionable tasks that it can then distribute across multiple cores of a machine, a computer farm or the cloud, thus further reducing the development time.

6.1.3 Common Language

With the new method of the environment generation, the environment does not need to be described. Instead, designers can simply point to the area in the map that they require. As a result, the communication challenge identified in the initial interview and observation studies is now minimized due to the inherently low potential of confusing a highlighted map area. Until now, designers have tried different ways of describing an environment, such as abstract sketches and mentions of real areas in the world, but the history of this collaboration has shown that these methods still lead to numerous development loops. Having the real-world map as a common language does not leave much room for misinterpretation of a description, and designers do not have to do extra work to create a document or sketch. This process has automatically the potential to reduce the number of loops between designers and developers, leaving time for refining the final result or implementing scenario-specific or special requirements.

During the studies, the designers observed this reduction in resources and the resulting benefits and expressed their appreciation. They immediately understood the importance of the framework for the design process and listed the advantages that the proposed process could have for their own individual tasks. This fact contributes also to their acceptance to apply the framework in their daily lives.

6.1.4 Scalability

Many mechanisms were developed throughout the procedural generation framework, such as the methods for easily adding new architectural styles and façade modules. These mechanisms, combined with the high degree of control over the generated environment enabled by the parametrization of most generated geometries ensure a high degree of scalability. Moreover, materials for different architectural styles and different primitive categories can be easily inserted into the corresponding Excel tables. Of course, the further development of the framework is not meant for the designers but the developer team. However, following these simple steps to add more elements and architectural styles also reduces their effort to add more diversity to the environment.

Diversity is not always a prerequisite for these environments. On the contrary, diversity is sometimes undesirable as it diverts too much attention from the aspects of the design that need to be evaluated. For those cases, a UI option that regulates the level of abstraction of the environment is necessary and will be added in future work. The geometry to serve as the output of this option is already available as the output of the intermediate steps of the framework provides exactly these different levels of abstraction (e.g., environment with modelled streets and buildings in 3D-box shape). In cases where more variety is required, more elements can be added to the environment to increase its realism, such as tables and chairs outside restaurants, grass patches on the pavement and special famous buildings visualized by manually modelled assets that are placed by leveraging the corresponding OSM attributes. This was also included in future work, along with increasing the diversity of materials.

6.1.5 Individuality

In cases where the designers do not feel ready to switch to the proposed process because of their attitude towards a more independent process of generating the environment, the developers can be the ones that interact with the UI based on the designers' requirements and start the generation process. This adjustment of the proposed process still provides the same advantages while achieving the level of independence that each individual designer feels comfortable with. Regardless of this level of independence, developers will always be needed to do some of the development work, even if in some cases it only consists of scenario-specific changes and additional elements.

6.1.6 Reliability

An automated process needs to have high reliability, which was confirmed in the computation studies with the successful generation of 30 street network samples and 1000 buildings with mixed architectural styles. This reliability exceeds that of other existing procedural generators, such as the one offered by Houdini (see Figure 16 for common modelling errors). This reliability enables the designers to use the framework through programming-free interaction. Nevertheless, the generated environment within the Game Engine can be edited by developers to add scenario-specific elements or replace already placed assets, if necessary.

6.1.7 Suitability for Addition of Simulations

The scalability of the procedural generation framework facilitates the addition of simulations. The contributing factor is the information stored in the generated geometry, which enables the addition of a traffic and pedestrian simulation with the corresponding possibility of adjusting the degree of traffic and pedestrian flow respectively within the UI. The Houdini software has the potential to simulate traffic and crowds, which is an additional advantage of its use.

6.1.8 Graphics Quality

Finally, in this prototyping version of the procedural framework, we did not focus so much on the graphics quality, but rather on functionality. However, PBR textures can be added to the generated geometries and the assets of the façade modules are easily replaceable with those of higher quality and more detail. In the future, more algorithms can be written to allow the generation of tunnels and footways, as well as adding more details to the façade (e.g., AC units on the exterior of buildings in warm areas).

Of course, increasing the quality by adding assets can easily lead to a decrease in performance. However, since Houdini is a software-neutral solution, the Nanite feature of the Unreal Engine 5 can be leveraged allowing an extremely high number of polygons to be rendered simultaneously without causing performance peaks.

6.2 Suitability of the UI

Keeping the user’s effort as low as possible in a highly complex task, such as environment generation, is equally complicated. However, we have achieved it by adjusting the input of the procedural generation framework to be minimal. Accordingly, the designers’ effort, who only have to select a map area, is also low. The change from a process where the designers just describe the environment to the developers and wait for them to model it, to a process where designers have to interact with a UI and generate the desired environment is a sensitive change in the designers’ everyday-life. The proposed process already has a major advantage for the designers as it offers them the desired output in a greatly reduced amount of time and makes them independent of the developers for most of the simulation. However, the proposed process may be considered less beneficial if it requires more effort and learning of new software. Therefore, the requirement of minimum designer effort was fulfilled by requiring as input only the selection of the world map area to replace the designers’ environment description to the developers. In addition, to ensure that the switching from the current to the proposed process is as smooth as possible, the UI also provides support in finding a suitable environment in the world in case designers feel overwhelmed by too many options or not sure if they have found the best environment for their design story. However, this support can only offer them an indication of existing characteristic elements in a real environment. The designers still need to test the environment which leads to the same trial-and-error process to find the environment, only potentially more efficient, as they know where to look.

This balance between the minimum amount of effort, the right amount of options and the right options themselves led us to develop a linear UI. This is the first prototype of this UI and, as was evident from the interviews, the amount and the type of the options we offered to the designers proved to be right. However, further research could be conducted to test other forms of UI, which might be better suited to its main purpose.

Future work on the UI includes adding the options suggested by the designers and developers during the studies. These are the ability to control the level of traffic and pedestrian flow, as well as automated or manual driving. Furthermore, the proposed search parameters will also be added, as well as the ability to activate the demonstration of the vehicle following the selected route in the generated simulation. As shown by the results of the studies and expressed by the designers themselves, the aesthetic quality of the UI needs to be improved by a more neutral design and color scheme. An interesting task that will also be addressed in the future is the ability to describe a design scenario within the UI. While this is not directly related to the task of generating the environment, it is still part of the vision surrounding the use of these prototyping experiences.

6.3 User Experience and Technology Acceptance of the Suggested Process

From the large difference between the user experience survey results of the current and the proposed process, it can be concluded that the designers consider the proposed process to be more efficient and would welcome its application in preparing the environment for the prototyping design experiences. After using the UI to define and generate an environment, and looking at the generated output, the participants in this study found the concept promising and adjusted with their needs. Even though the generated output is not yet of the highest quality, the participants understand how valuable it is for their work and want to use this framework either with or without the developer support. Consequently, this is a great functional basis that we have achieved from a development perspective. It suggests that future work should include the addition of materials and high-quality assets to replace those already present and add more life to the scene with the ultimate goal of increasing the graphics quality of the environment.

6.4 Use of Procedural Generation

As we saw in Chapter 2, there are many methods available today to automatically generate an environment. Opting for procedural generation has allowed us to retain control over the input in case scenario-specific changes are needed (e.g., adding parking spaces along the road), during the generation (e.g., using certain assets and textures to achieve the comic-style buildings) and after generation (e.g., changing a texture, replacing a building).

The quality can be increased without changing the method by using assets and textures of higher quality, as well as by adding more details (e.g., on the building façades). However, we could also increase it by using machine learning methods to increase photorealism, for example [308]. Higher photorealism could be achieved by using scanned data. However, we rejected this method because the accessibility of the data depends on the budget and obtaining this data requires a lot of time. Additional proof that we are on the right direction with the method we have chosen, are the many great games which we also mentioned in Chapter 2 (e.g., Horizon Zero Dawn [198]), that have been developed using procedural generation tools and even the Houdini software.

7 Conclusion

In the first part of this thesis, a dive into the automotive design world showed the different types of challenges a designer has to face to adjust their design according to the needs of the end user. This indicates the importance of making design decisions based on experiencing the design already in the first phase of the design process. In addition, a look at the sensory modalities that a product can trigger and the resulting experience increases this importance by providing an overview of the way perception works and the way it influences the product experience. Further significance is given to the use of experience prototypes by designers after exploring the different aspects and multidimensionality of a multisensory Mixed Reality experience from different perspectives relevant to the method used to implement them.

In the second part of the thesis, interviews with automotive designers and developers shed light into their everyday life, collaboration, and challenges in their joined attempt to develop and apply these prototyping experiences in the design process. The collected information pointed to the need for a procedural generation framework that can automatically model a selected area in the world based on real map data. Such a framework was implemented with a series of procedural generators, one for each major element that exists in each environment (i.e., terrain, streets, buildings and vegetation). Additionally, a User Interface was implemented to match the needs of the designers for selecting an area in the world and finding one that can best visualize their design stories. It also allows designers to add a vehicle with the selection of the driver and passengers, adjust the date, time of day and weather conditions, as well as switch to another camera perspective or even create and apply a new perspective. To evaluate the suitability of the generated output, the usability of the UI and the user experience of the environment generation process for the prototyping experiences proposed by the use of the new framework and UI, relevant evaluation studies were conducted by collecting quantitative and qualitative feedback from designers and developers.

In the final part, the results of the evaluation studies are used to analyze the extent to which the collected requirements and observed challenges arising from the initial studies were fulfilled and resolved. The results of the questionnaires showed a particularly positive impression of the participants about the suggested process, the interface and the framework's output. This showed that the designed concept was the right decision, and that the development of the framework has reached a solid basis that can be expanded from this point on with more materials, assets and other environment details. Future work includes the use of high-quality materials and assets, the addition of a traffic and pedestrian simulation, as well as further options proposed for the UI.

Appendix

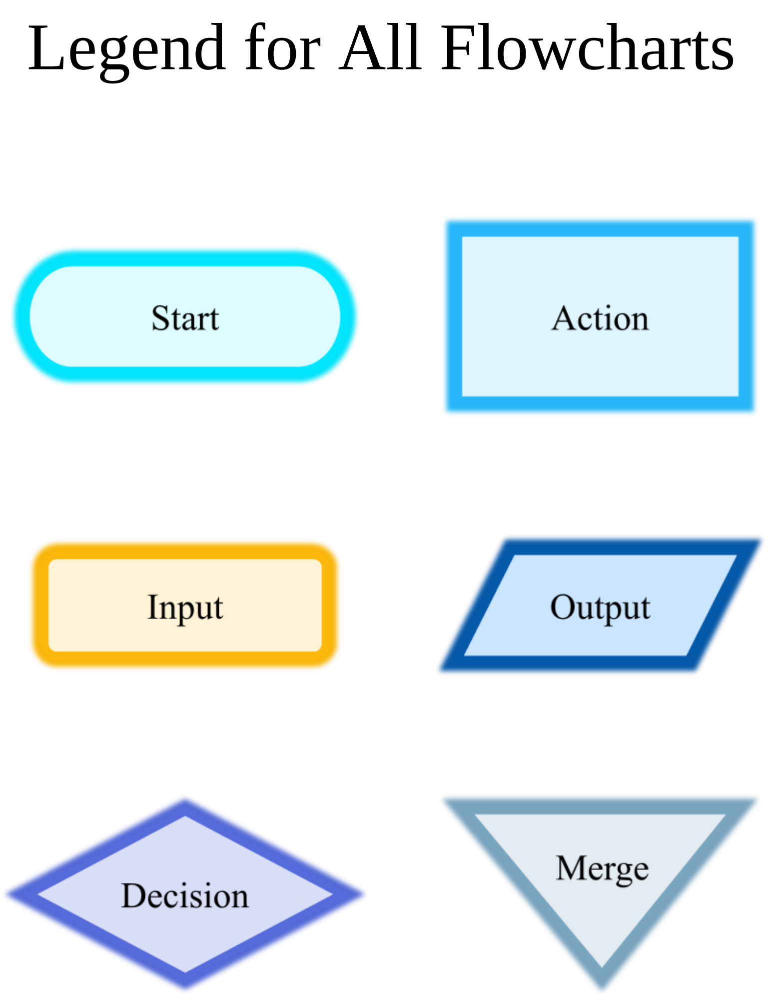


Figure 87: A legend for all flowcharts.

Process Evaluation Study: Current/Suggested Process of Preparing the Experience Context

I. Demographic Questions

1. Age

   ____

2. Gender

   ( ) female

   ( ) male

3. Design Experience (Time)

   ____

4. Job Description

   ____

II. User Experience Questionnaire

Please decide spontaneously which gradation between the following adjective pairs best reflects your opinion about the current/suggested process of preparing the virtual environment for design presentations and evaluations.

	1	2	3	4	5	6	7	
obstructive	◯	◯	◯	◯	◯	◯	◯	supportive

	1	2	3	4	5	6	7	
complicated	◯	◯	◯	◯	◯	◯	◯	easy

	1	2	3	4	5	6	7	
confusing	◯	◯	◯	◯	◯	◯	◯	clear

	1	2	3	4	5	6	7	
conventional	◯	◯	◯	◯	◯	◯	◯	inventive

	1	2	3	4	5	6	7	
inefficient	◯	◯	◯	◯	◯	◯	◯	efficient

	1	2	3	4	5	6	7	
not interesting	◯	◯	◯	◯	◯	◯	◯	interesting

	1	2	3	4	5	6	7	
boring	◯	◯	◯	◯	◯	◯	◯	exciting

	1	2	3	4	5	6	7	
usual	◯	◯	◯	◯	◯	◯	◯	leading edge

UI Evaluation Study

I. AttrakDiff Questionnaire

	1	2	3	4	5	6	7	
human	◯	◯	◯	◯	◯	◯	◯	technical
isolating	◯	◯	◯	◯	◯	◯	◯	connective
pleasant	◯	◯	◯	◯	◯	◯	◯	unpleasant
inventive	◯	◯	◯	◯	◯	◯	◯	conventional
professional	◯	◯	◯	◯	◯	◯	◯	unprofessional
simple	◯	◯	◯	◯	◯	◯	◯	complicated
ugly	◯	◯	◯	◯	◯	◯	◯	attractive
practical	◯	◯	◯	◯	◯	◯	◯	impractical

	1	2	3	4	5	6	7	
likeable	○	○	○	○	○	○	○	disagreeable
cumbersome	○	○	○	○	○	○	○	straightforward
stylish	○	○	○	○	○	○	○	tacky
predictable	○	○	○	○	○	○	○	unpredictable
cheap	○	○	○	○	○	○	○	premium
alienating	○	○	○	○	○	○	○	integrating
brings me closer to people	○	○	○	○	○	○	○	separates me from people
unpresentable	○	○	○	○	○	○	○	presentable
rejecting	○	○	○	○	○	○	○	inviting
unimaginative	○	○	○	○	○	○	○	creative

	1	2	3	4	5	6	7	
good	◯	◯	◯	◯	◯	◯	◯	bad
confusing	◯	◯	◯	◯	◯	◯	◯	clearly structured
repelling	◯	◯	◯	◯	◯	◯	◯	appealing
bold	◯	◯	◯	◯	◯	◯	◯	cautious
innovative	◯	◯	◯	◯	◯	◯	◯	conservative
dull	◯	◯	◯	◯	◯	◯	◯	captivating
undemanding	◯	◯	◯	◯	◯	◯	◯	challenging
motivating	◯	◯	◯	◯	◯	◯	◯	discouraging
novel	◯	◯	◯	◯	◯	◯	◯	ordinary
unruly	◯	◯	◯	◯	◯	◯	◯	manageable

Technology Acceptance Evaluation Study: Suggested Process of Preparing the Experience Context

I. User Experience

	1	2	3	4	5	6	7	
not understandable	◯	◯	◯	◯	◯	◯	◯	understandable
creative	◯	◯	◯	◯	◯	◯	◯	dull
easy to learn	◯	◯	◯	◯	◯	◯	◯	difficult to learn
valuable	◯	◯	◯	◯	◯	◯	◯	inferior
boring	◯	◯	◯	◯	◯	◯	◯	exciting
not interesting	◯	◯	◯	◯	◯	◯	◯	interesting
unpredictable	◯	◯	◯	◯	◯	◯	◯	predictable

	1	2	3	4	5	6	7	
fast	◯	◯	◯	◯	◯	◯	◯	slow
inventive	◯	◯	◯	◯	◯	◯	◯	conventional
obstructive	◯	◯	◯	◯	◯	◯	◯	supportive
complicated	◯	◯	◯	◯	◯	◯	◯	easy
usual	◯	◯	◯	◯	◯	◯	◯	leading edge
secure	◯	◯	◯	◯	◯	◯	◯	not secure
motivating	◯	◯	◯	◯	◯	◯	◯	demotivating
meets expectations	◯	◯	◯	◯	◯	◯	◯	does not meet expectations
inefficient	◯	◯	◯	◯	◯	◯	◯	efficient
clear	◯	◯	◯	◯	◯	◯	◯	confusing

	1	2	3	4	5	6	7	
impractical	◯	◯	◯	◯	◯	◯	◯	practical

	1	2	3	4	5	6	7	
organized	◯	◯	◯	◯	◯	◯	◯	cluttered

	1	2	3	4	5	6	7	
conservative	◯	◯	◯	◯	◯	◯	◯	innovative

II. Technology Acceptance

1. The quality of the output I get from the proposed process of preparing the virtual environment for design presentation and evaluation is high.

	1	2	3	4	5	6	7	
strongly disagree	◯	◯	◯	◯	◯	◯	◯	strongly agree

2. I have no problem with the quality of the proposed process's output.

	1	2	3	4	5	6	7	
strongly disagree	◯	◯	◯	◯	◯	◯	◯	strongly agree

3. I rate the results from the proposed process to be excellent.

	1	2	3	4	5	6	7	
strongly disagree	◯	◯	◯	◯	◯	◯	◯	strongly agree

4. Using the proposed process improves my performance in my job.

	1	2	3	4	5	6	7	
strongly disagree	◯	◯	◯	◯	◯	◯	◯	strongly agree

5. Using the proposed process in my job increases my productivity. i.e., I achieve more results, higher performance.

	1	2	3	4	5	6	7	
strongly disagree	◯	◯	◯	◯	◯	◯	◯	strongly agree

6. Using the proposed process enhances my effectiveness in my job, i.e., I achieve better results.

	1	2	3	4	5	6	7	
strongly disagree	◯	◯	◯	◯	◯	◯	◯	strongly agree

7. I find the proposed process to be useful in my job.

	1	2	3	4	5	6	7	
strongly disagree	◯	◯	◯	◯	◯	◯	◯	strongly agree

8. My interaction with the proposed process is clear and understandable.

	1	2	3	4	5	6	7	
strongly disagree	◯	◯	◯	◯	◯	◯	◯	strongly agree

9. Interacting with the proposed process does not require a lot of my mental effort.

	1	2	3	4	5	6	7	
strongly disagree	◯	◯	◯	◯	◯	◯	◯	strongly agree

10. I find the proposed process to be easy to use.

	1	2	3	4	5	6	7	
strongly disagree	◯	◯	◯	◯	◯	◯	◯	strongly agree

11. I find it easy to get the proposed process to do what I want it to do.

	1	2	3	4	5	6	7	
strongly disagree	◯	◯	◯	◯	◯	◯	◯	strongly agree

12. On average, how much time do you spend on the proposed process each day?

	1	2	3	4	5	6	7	
never	◯	◯	◯	◯	◯	◯	◯	all the time

13. I plan to use the proposed process in the next 3 months.

	1	2	3	4	5	6	7	
strongly disagree	◯	◯	◯	◯	◯	◯	◯	strongly agree

14. Given that I had access to the proposed process, I predict that I would use it.

	1	2	3	4	5	6	7	
strongly disagree	◯	◯	◯	◯	◯	◯	◯	strongly agree

15. Assuming I had access to the proposed process, I intend to use it.

	1	2	3	4	5	6	7	
strongly disagree	◯	◯	◯	◯	◯	◯	◯	strongly agree

Table of Figures

List of Tables

Bibliography

1. Aristoteles, “Μετα τα Φυσικα Α”, 981$^{\alpha}$, 3
2. Donald A. Norman and Andrew Ortony. 2006. Designers and Users: Two Perspectives on Emotion and Design. In Proceedings of the Symposium on Foundations of Interaction Design at the Interaction Design Institute, Ivrea, Italy.
3. Aristotle. 1935. Aristotle in 23 Volumes, 17, 18, translated by Hugh Tredennick. Cambridge MA, Harvard University Press, London, William Heinemann.
4. Sebastian Loehmann. 2015. Experience Prototyping for Automotive Applications. Dissertation, LMU München: Faculty of Mathematics, Computer Science and Statistics.
5. Izumi Satake, John H. Xin, Tu Tianming, Aran Hansuebsai, Kenji Ando, Tetsuya Sato, Kanji Kajiwara and Seiji Ohsawa. 2011. A comparative study of the emotional assessment of automotive exterior colors in Asia. Progress in Organic Coatings - PROG ORG COATING. 72, 528-540. DOI: 10.1016/j.porgcoat.2011.06.013
6. Bella Martin and Bruce M. Hanington. 2012. Universal Methods of Design: 100 Ways to Research Complex Problems, Develop Innovative Ideas, and Design Effective Solutions. Rockport Publishers.
7. Everett McKay. 2013. UI is Communication. Elsevier Inc. DOI: https://doi.org/10.1016/C2011-0-07788-8
8. Marlene Weber, Joseph Giacomin, Alessio Malizia, Lee Skrypchuk, Voula Gkatzidou and Alexandros Mouzakitis. 2019. Investigation of the dependency of the drivers’ emotional experience on different road types and driving conditions. Transportation Research Part F: Traffic Psychology and Behaviour 65, 3. DOI: 10.1016/j.trf.2019.06.001
9. Kim B. Clark and Takahiro Fujimoto. 1991. Product Development Performance: Strategy, Organization, and Management in the World Auto Industry. Harvard Business School Press.
10. Roland Mandel, Philipp Pomiersky and Thomas Maier. 2015. Der vollvariable Fahrzeug-Ergonomieprüfstand - Absicherung des digitalen Auslegungsprozesses. In Proceedings of the Binz, H.; Bertsche, B.; Bauer, W.; Roth, D. (Hrsg.): Stuttgarter Symposium für Produktentwicklung 2015, Stuttgart.
11. Roman Raabe. 2013. Ein rechnergestütztes Werkzeug zur Generierung konsistenter PKW-Maßkonzepte und parametrischer Designvorgaben. Stuttgart: Inst. für Konstruktionstechnik und Techn. Design.
12. Julian Weber. 2009. Automotive Development Processes: Processes for Successful Customer Oriented Vehicle Development. Springer.
13. Hans-Hermann Braess and Ulrich Seiffert. 2013. ATZ/MTZ-Fachbuch, Vieweg Handbuch Kraftfahrzeugtechnik. Springer Fachmedien Wiesbaden. DOI: https://doi.org/10.1007/978-3-658-01691-3
14. Eduardo S. Schwartz. 2004. Patents and R&D as Real Options. Economic Notes, 33(1), 23–54. DOI:10.1111/j.0391-5026.2004.00124.x
15. Ramchandran Jaikumar. 1986. Postindustrial Manufacturing, Harvard Business Review, 69–76.

16. Daniel Sörensen. 2006. The automotive development process: A real options analysis (1st edn, Gabler Edition Wissenschaft). Deutscher Universitätsverlag Wiesbaden. DOI: https://doi.org/10.1007/978-3-8350-9339-3
17. Ferdinando Milella. 2015. Problem-Solving by Immersive Virtual Reality: Towards a More Efficient Product Emergence Process in Automotive. Journal of Multidisciplinary Engineering Science and Technology (JMEST), 2 (4). 860 – 867.
18. Chris Argyris. 1984. Reasoning, Learning, and Action: Individual and Organizational San Francisco: Jossey-Bass, 1982, 499 pages. Organization Studies, 5(3):283-284. DOI:10.1177/017084068400500316
19. Henry Petroski. 1992. To engineer is human: The role of failure in successful design. New York: Vintage.
20. Richard Coyne, Michael Rosenman, Anthony Radford, M. Balachandran and John Gero. 1989. Knowledge-Based Design Systems. Addison Wesley.
21. Andre Liem, Shahriman Zainal Abidin and Anders Warell. 2011. Designers' Perceptions of Typical Characteristics of Form Treatment in Automobile Styling. Paper presented at 5th International Workshop on Design & Semantics of Form and Movement, DesForm 2009, 144-155, Taipei, Taiwan.
22. Michael Tovey, S. Porter and Robert Newman. 2003. Sketching, concept development and automotive design. Design Studies, 24(2), 135-153. DOI: https://doi.org/10.1016/S0142-694X(02)00035-2
23. Guangdong Tian, Honghao Zhang, Hongfei Jia, Yumei Liu, Guan Xu and Jing Wang. 2016. Automotive style design assessment and sensitivity analysis using integrated analytic hierarchy process and technique for order preference by similarity to ideal solution. Advances in Mechanical Engineering, 8(5). DOI: https://doi.org/10.1177%2F1687814016649885
24. Ignacio Alvarez and Laura Rumbel. 2017. Skyline: A Platform Towards Scalable UX-Centric In-Vehicle HMI Development. Int. J. Mob. Hum. Comput. Interact. 9, 3 (July 2017), 34–53. DOI: https://doi.org/10.4018/ijmhci.2017070103
25. Monica Bordegoni and Giandomenico Caruso. 2012. Mixed reality distributed platform for collaborative design review of automotive interiors. Virtual and Physical Prototyping, 7:4, 243-259, DOI: 10.1080/17452759.2012.721605
26. Anders Warell and Kenneth Young. 2011. Interior aesthetics: an experience-focused approach for the design of brand-specific automotive identity. International Journal of Vehicle Design, 55(2/3/4), 278. DOI:10.1504/IJVD.2011.040587
27. Marion Buchenau and Jane Fulton Suri. 2000. Experience prototyping. In Proceedings of the 3rd Conference on Designing interactive systems: processes, practices, methods, and techniques (DIS '00). Association for Computing Machinery, New York, NY, USA, 424–433. DOI: https://doi.org/10.1145/347642.347802
28. Anthony Dunne. 2006. Hertzian Tales: Electronic Products, Aesthetic Experience, and Critical Design. The MIT Press.
29. Pieter Desmet and Paul Hekkert. 2009. Special Issue Editorial: Design & Emotion. International Journal of Design, 3(2), 1–6.
30. Amic G. Ho. 2014. The New Relationship between Emotion and the Design Process for Designers. Archives of Design Research. DOI:10.15187/adr.2014.05.110.2.45

31. Jeffrey W. Herrmann and Linda C. Schmidt. 2002. Viewing Product Development as a Decision Production System. In Proceedings of the ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 4: 14th International Conference on Design Theory and Methodology, Integrated Systems Design, and Engineering Design and Culture. Montreal, Quebec, Canada. September 29–October 2, 2002, pp. 323-332. ASME. DOI: https://doi.org/10.1115/DETC2002/DTM-34030
32. Rita A. Almendra and Henri Christiaans. 2009. Improving Design Processes through better Decision-Making: an experiment with a decision making support tool. In K. Lee, J. Kim, & LL. Chen (Eds.), IASDR 2009 Proceedings - International association of societies of design research 2009 (pp. 1-11), Seoul, Korea.
33. Joan Ernst van Aken. 2005. Valid knowledge for the professional design of large and complex design processes. Design Studies, 26(4), 379-404. DOI: https://doi.org/10.1016/j.destud.2004.11.004
34. Jasmin Enayati. 2002. The Research: Effective Communication and Decision-making in Diverse Groups. In Minu Hemmati, Jasmin Enayati, Jan McHarry, Felix Dodds (Eds.), Multi-stakeholder processes for governance and sustainability: Beyond deadlock and conflict, 73-95. Earthscan by Routledge.
35. Jodi Forlizzi and Katja Battarbee. 2004. Understanding experience in interactive systems. In Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques (DIS '04). Association for Computing Machinery, New York, NY, USA, 261–268. DOI: https://doi.org/10.1145/1013115.1013152
36. Kees Overbeeke, Tom Djajadiningrat, Caroline Hummels, Stephan Wensveen and Joep Prens. 2003. Let’s Make Things Engaging. In Blythe, M.A., Overbeeke, K., Monk, A.F., Wright, P.C. (Eds.), Funology. Human-Computer Interaction Series, 3. Springer, Dordrecht. DOI: https://doi.org/10.1007/1-4020-2967-5_2
37. Barthélémy Longueville, Julie Stal Le Cardinal, Jean-Claude Bocquet and Pascal Daneau. 2003. TOWARD A PROJECT MEMORY FOR INNOVATIVE PRODUCT DESIGN, A DECISION-MAKING PROCESS MODEL. In Proceedings of the 14th International Conference on Engineering Design (ICED03), Stockholm.
38. Jodi Forlizzi, Carl Disalvo and Bruce Hanington. 2003. On the Relationship between Emotion, Experience and the Design of New Products. The Design Journal, 6(2): 29-38. DOI: 10.2752/146069203789355507
39. Ed S. Tan. 1999. Emotion and style as design. In P. Hekkert (Ed.), Proceedings of the First International Conference on Design and Emotion, 55-64. Delft University of Technology.
40. Uday Dandavate, Elizabeth B.-N. Sanders and Susan Stuart. 1996. Emotions Matter: User Empathy in the Product Development Process. In Proceedings of the Human Factors and Ergonomics Society 40th annual meeting, 40(7): 415-418.
41. Malka Ben-Peshat. 2004. Popular Design and Cultural Identities – Emotional Exchange: Study Cases in Israel. In Proceedings of the 4th International Conference on Design and Emotion. DOI: https://doi.org/10.5281/zenodo.2619613
42. Stella Boess, David Durling, Cherie Lebbon and Christopher Maggs. 2022. Participative Image-based Research as a Basis for New Product Development. In Green, W.S., Jordan, P.W. (Eds.), Pleasure with Product, 221-245. Taylor & Francis.

43. Marvin Harris. 1979. Cultural Materialism: The Struggle for a Science of Culture. AltaMira Press.
44. Genevieve Bell. 2001. Looking Across the Atlantic: Using Ethnographic Methods to Make Sense of Europe. Intel Technology Journal, 3rd Quarter, 2001.
45. Jane Fulton Suri. 2003. Empathic Design: Informed and Inspired by Other People's Experience. In Koskinen, I., Battarbee K., Mattelmäki T. (Eds.), Empathic Design – User Experience in Product Design, 51-58. IT Press, Helsinki.
46. Elizabeth B.-N. Sanders and Uday Dandavate. 1999. Design for experiencing: New tools. In Overbeeke, C. J.; Hekkert, P. (Eds.), Proceedings of the First International Conference on Design and Emotion, 87-92. DOI: https://doi.org/10.5281/zenodo.2635711
47. Leon D. Segal and Jane Fulton Suri. 1997. The Empathic Practitioner: Measurement and Interpretation of User Experience. In Proceedings of the 41st Annual Meeting of Human Factors and Ergonomics Society, September 22-26, 1997, Albuquerque, New Mexico, USA. DOI: https://doi.org/10.1177/107118139704100199
48. Donald A. Norman. 2005. Emotional design: Why we love (or hate) everyday things. Basic Books.
49. Anders Warell. 2008. Multi-modal visual experience of brand-specific automobile design. The TQM Journal, 20(4), 356–371. DOI:10.1108/17542730810881348
50. Haim Mano and Richard L. Oliver. 1993. Assessing the Dimensionality and Structure of the Consumption Experience: Evaluation, Feeling, and Satisfaction. Journal of Consumer Research, 20, 451-466. DOI: http://dx.doi.org/10.1086/209361
51. Hendrik N. J. Schifferstein and Pieter M. A. Desmet. 2008. Tools Facilitating Multi-sensory Product Design. The Design Journal, 11(2), 137–158. DOI:10.2752/175630608X329226.
52. Monica Bordegoni, Umberto Cugini, Giandomenico Caruso and Samuele Polistina. 2009. The Role of Mixed Prototyping in Product Design Assessment. In Proceedings of the 2nd International Conference on Research into Design (ICORD 09), Bangalore, India, 07.-09.01.2009, 427–434.
53. John Dewey. 1934. Art as an Experience. New York: Perigree.
54. Peter Wright, John McCarthy and Lisa Meekison. 2008. Making sense of experience. Making Sense of Experience. In Blythe, M., Monk, A. (Eds.), Funology 2. Human–Computer Interaction Series. Springer, Cham. DOI: https://doi.org/10.1007/978-3-319-68213-6_20
55. James J. Gibson. 1979. The Ecological Approach to Visual Perception. Psychology Press.
56. Harold T.Nefs. 2008. 1 - ON THE VISUAL APPEARANCE OF OBJECTS. In Hendrik N.J. Schifferstein, Paul Hekkert (Eds.), Product Experience, 11-39. Elsevier. DOI: https://doi.org/10.1016/B978-008045089-6.50004-6
57. Paul Hekkert and Hendrik J. Schifferstein. 2008. INTRODUCING PRODUCT EXPERIENCE. In Hendrik N.J. Schifferstein, Paul Hekkert (Eds.), Product Experience, 1-8. Elsevier. DOI: https://doi.org/10.1016/B978-008045089-6.50003-4
58. Lars-Erik Janlert and Erik Stolterman. 1997. The Character of Things. Design Studies, 18, 297-314. DOI: http://dx.doi.org/10.1016/S0142-694X(97)00004-5
59. Rune Monö. 1997. Design for Product Understanding. Liber AB, Stockholm.
60. Marc Hassenzahl. 2003. The Thing and I: Understanding the Relationship Between User and Product. In Blythe, M.A., Overbeeke, K., Monk, A.F., Wright, P.C. (Eds.), Funology.

Human-Computer Interaction Series, 3. Springer, Dordrecht. DOI: https://doi.org/10.1007/1-4020-2967-5_4
61. William Curran and Alan Johnston. 1996. The effect of Illuminant position on perceived curvature. Vision Research, 36(10), 1399–1410. DOI: https://doi.org/10.1016/0042-6989(95)00213-8
62. Harold T. Nefs, Jan J. Koenderink and Astrid M. L. Kappers. 2005. The Influence of Illumination Direction on the Pictorial Reliefs of Lambertian Surfaces. Perception, 34(3), 275–287. DOI: https://doi.org/10.1068/p5179
63. Harold T. Nefs, Jan J. Koenderink and Astrid M. L. Kappers. 2006. Shape-from-shading for matte and glossy objects. Journal of Vision 2004; 4(8):473, 473a. DOI: https://doi.org/10.1167/4.8.473
64. James T. Todd and Ennio Mingolla. 1983. Perception of surface curvature and direction of illumination from patterns of shading. Journal of Experimental Psychology: Human Perception and Performance, 9(4), 583–595. DOI: https://doi.org/10.1037/0096-1523.9.4.583
65. James T. Todd, Farley J. Norman, Jan J. Koenderink and Astrid M. L. Kappers. 1997. Effects of Texture, Illumination, and Surface Reflectance on Stereoscopic Shape Perception. Perception, 26(7):807-822. DOI: https://doi.org/10.1068/p260807
66. Farley J. Norman, James T. Todd and Guy A. Orban. 2004. Perception of Three-Dimensional Shape From Specular Highlights, Deformations of Shading, and Other Types of Visual Information. Psychological Science,15(8):565-570. DOI: https://doi.org/10.1111/j.0956-7976.2004.00720.x
67. Andrew Blake and Heinrich Bulthoff. 1991. Shape from specularities: computation and psychophysics. Philosophical Transactions of the Royal Society of London: Series B: Biological Sciences, 331(1260), 237–252.
68. J. P. Claessen. 1996. Shaped by colour. A study on the effect of colour on the perceived shape of objects. Doctoral thesis, University of Delft, The Netherlands.
69. Byung-Geun Khang, Jan J. Koenderink and Astrid M. L. Kappers. 2003. Perception of Surface Reflectance of 3-D Geometrical Shapes: Influence of the Lighting Mode. Perception, 32(11):1311-1324. DOI: https://doi.org/10.1068/p5070
70. Kurt Koffka. 1935. Principles of Gestalt Psychology. Harcourt, Brace.
71. Tiffany Field. 2003. Touch. London: MIT Press.
72. Ashley Montagu. 1971. Touching. New York: Columbia University Press.
73. Marieke H. Sonneveld and Hendrik N. J. Schifferstein. 2008. 2 - THE TACTUAL EXPERIENCE OF OBJECTS. In Hendrik N.J. Schifferstein, Paul Hekkert (Eds.), Product Experience, 41-67. Elsevier. DOI: https://doi.org/10.1016/B978-008045089-6.50005-8
74. James J. Gibson. 1962. Observations on active touch. Psychological Review, 69(6), 477–491.
75. Gregory Burton. 1993. Non-Neural Extensions of haptic Sensitivity. Ecological Psychology, 5(2), 105–124. DOI: 10.1207/s15326969eco0502_1
76. Roberta L. Klatzky, Susan J. Lederman and Catherine Reed. 1987. There’s more to touch than meets the eye: the salience of object attributes for haptics with and without vision. Journal of Experimental Psychology: General, 116(4), 356–369. DOI: https://doi.org/10.1037/0096-3445.116.4.356

77. Michael F. Ashby and Kara Johnson. 2002. Materials and Design: The Art and Science of Material Selection in Product Design. Oxford: Butterworth-Heinemann. DOI: https://doi.org/10.1016/S0261-3069(03)00043-8
78. Mark Hollins and S. Ryan Risner. 2000. Evidence for the duplex theory of tactile texture perception. Perception and Psychophysics, 62, 695–705. DOI: https://doi.org/10.3758/BF03206916
79. Susan J. Lederman. 1981. The perception of surface roughness by active and passive touch. Bulletin of the Psychonomic Society, 18, 253–255. DOI: https://doi.org/10.3758/BF03333619
80. Mark Hollins, Sliman Bensmaïa, Kristie Karlof and Forrest Young. 2000. Individual differences in perceptual space for tactile textures: Evidence from multidimensional scaling. Perception & Psychophysics, 62, 1534–1544. DOI: https://doi.org/10.3758/BF03212154
81. Delphine Picard, Catherine Dacremont, Dominique Valentin and Agnès Giboreau. 2003. Perceptual dimensions of tactile textures. Acta Psychologica, 114(2), 165–184. DOI: https://doi.org/10.1016/j.actpsy.2003.08.001
82. Susan J. Lederman and Roberta L. Klatzky. 1987. Hand movements: A window into haptic object recognition. Cognitive Psychology, 19(3), 342–368. DOI: https://doi.org/10.1016/0010-0285(87)90008-9
83. Ingrid M. L. C. Vogels, Astrid M. L. Kappers and Jan J. Koenderink. 2001. Haptic after-effect of successively touched curved surfaces. Acta Psychologica, 106(3), 247–263. DOI: https://doi.org/10.1016/S0001-6918(00)00052-4
84. Jean-Louis Juan de Mendoza. 1979. Demonstration of an aftereffect occurring in the tactile-kinesthetic domain. Psychological Research, 40, 415–422. DOI: https://doi.org/10.1007/BF00309421
85. Randall J. Flanagan, Alan M. Wing, Sue Allison and Anna Spenceley. 1995. Effects of surface texture on weight perception when lifting objects with a precision grip. Perception and Psychophysics, 57(3), 282–290. DOI: https://doi.org/10.3758/BF03213054
86. Kenneth Saladin. 2001. Anatomy & Physiology: The Unity of Form and Function. New York: McGraw-Hill Education.
87. Arthur Vander, James Sherman and Dorothy Luciano. 2001. Human Physiology. The mechanisms of body functions. New York: McGraw-Hill.
88. James J. Gibson. 1963. The useful dimensions of sensitivity. American Psychologist, 18(1), 1–15. DOI: https://doi.org/10.1037/h0046033
89. Fritz Heider and Marianne L. Simmel. 1944. An Experimental Study of Apparent Behavior. The American Journal of Psychology, 57(2), 243–259. DOI: https://doi.org/10.2307/1416950
90. Albert Michotte. 1963. The Perception of Causality. London: Methuen.
91. René van Egmond. 2008. 3 - THE EXPERIENCE OF PRODUCT SOUNDS. In Hendrik N.J. Schifferstein, Paul Hekkert (Eds.), Product Experience, 69-89. Elsevier. DOI: https://doi.org/10.1016/B978-008045089-6.50006-X
92. Claudia Carello, Krista L. Anderson and Andrew J. Kunkler-Peck. 1998. Perception of Object Length by Sound. Psychological Science, 9(3), 211–214. DOI: https://doi.org/10.1111/1467-9280.00040
93. Massimo Grassi. 2005. Do we hear size or sound? Balls dropped on plates. Perception and Psychophysics, 67(2), 274–284. DOI: https://doi.org/10.3758/BF03206491

94. Andrew J. Kunkler-Peck and Michael Turvey. 2000. Hearing shape. Journal of Experimental Psychology – Human Perception and Performance, 26(1), 279–294. DOI: https://doi.org/10.1037/0096-1523.26.1.279
95. Mark M. J. Houben, Armin Kohlrausch and Dik J. Hermes. 2004. Perception of the size and speed of rolling balls by sound. Speech Communication, 43(4), 331–345. DOI: https://doi.org/10.1016/j.specom.2004.03.004
96. Mark M.J. Houben, Armin Kohlrausch and Dik J. Hermes. 2005. The contribution of spectral and temporal information to the auditory perception of the size and speed of rolling balls. Acta Acustica United with Acustica, 91(6), 1007–1015.
97. Bruno L. Giordano and Stephen McAdams. 2006. Material identification of real impact sounds: Effects of size variation in steel, glass, wood, and Plexiglas plates. Journal of the Acoustical Society of America, 119, 1171–1181.
98. Dik J. Hermes. 1998. Auditory material perception. IPO Annual Progress Report 33, 95–102.
99. Daniel Västfjäll, M. A. Gulbol, Mendel Kleiner and Tommy Garling. 2002. AFFECTIVE EVALUATIONS OF AND REACTIONS TO EXTERIOR AND INTERIOR VEHICLE AUDITORY QUALITY. Journal of Sound and Vibration, 255(3), 501–518. DOI: https://doi.org/10.1006/jsvi.2001.4166
100. Jont B. Allen. 1999. Psychoacoustics. In J. G. Webster (Ed.), Wiley Encyclopedia of Electrical and Electronics Engineering, 422–437. New York: John Wiley.
101. Eberhard Zwicker and Hugo Fastl. 2007. Psychoacoustics. Facts and models (3rd Ed.). Berlin: Springer Verlag.
102. Ira J. Hirsh and Charles S. Watson. 1996. Auditory Psychophysics and Perception. Annual Review of Psychology, 47, 461–484. DOI: https://doi.org/10.1146/annurev.psych.47.1.461
103. Brian Moore. 2003. An Introduction into the Psychology of Hearing (5th Ed.). Amsterdam: Academic Press.
104. William A. Yost. 2007. Fundamentals of Hearing: An Introduction (5th Ed.). New York: Academic Press.
105. Robert Plutchik. 1980. Emotion: A Psychoevolutionary Synthesis. New York: Harper and Row.
106. Andrew Ortony, Gerald Clore and Allan Collins. 1988. The Cognitive Structure of Emotions. Cambridge: Cambridge University Press.
107. René Van Egmond, Pieter Desmet and Aadjan Van Der Helm. 2004. Basic and Cognitive Emotions in Sound Perception. In A. Kurtgözü (Ed.), Proceedings of the Fourth International Conference on Design and Emotion, 1–11. Ankara: METU.
108. Armand V. Cardello and Paul M. Wise. 2008. 4 - TASTE, SMELL AND CHEMESTHESIS IN PRODUCT EXPERIENCE. In Hendrik N.J. Schifferstein, Paul Hekkert (Eds.), Product Experience, 91-131. Elsevier. DOI: https://doi.org/10.1016/B978-008045089-6.50007-1
109. Rachel S. Herz. 1997. Emotion Experienced during Encoding Enhances Odor Retrieval Cue Effectiveness. American Journal of Psychology, 110(4), 489–505. DOI: https://doi.org/10.2307/1423407
110. Enrique J. Cometto-Muñiz and William S. Cain. 1995. Olfactory adaptation. In R. L. Doty (Ed.), Handbook of Olfaction and Gestation, 257–282. New York: Marcel Dekker.

111. Harry T. Lawless. 1997. Olfactory Psychophysics. In G. Beauchamp and L. Bartoshuk (Eds.), Tasting and Smelling, 125–168. San Diego, CA: Academic Press. DOI: https://doi.org/10.1016/B978-012161958-9/50005-1
112. William S. Cain. 1974. Perception of Odor Intensity and the Time-Course of Olfactory Adaptation. ASHRAE Transactions, 80, 53–75.
113. Hans Distel and Robyn Hudson. 2001. Judgement of Odor Intensity is Influenced by Subjects' Knowledge of the Odor Source. Chemical Senses, 26, 247–251. DOI: https://doi.org/10.1093/chemse/26.3.247
114. Trygg Engen. 1982. The Perception of Odors. New York: Academic.
115. Trygg Engen. 1988. The acquisition of odour hedonics. In S. Van Toller and G. H. Dodd (Eds.), Perfumery, 79–90. Springer, Dordrecht. DOI: https://doi.org/10.1007/978-94-017-2558-3_4
116. Margret Schleidt, Barbara Hold and Grazia Attili. 1981. A cross-cultural study on the attitude towards personal odors. Journal of Chemical Ecology, 7, 19–31. DOI: https://doi.org/10.1007/BF00988632
117. Charles J. Wysocki and Avery N. Gilbert. 1989. National Geographic Smell Survey: Effects of Age Are Heterogenous. In C. L. Murphy, W. S. Cain, and D. M. Hegsted (Eds.), Chemical Senses and Nutrition in Aging, 12–28. New York: New York Academy of Sciences. DOI: https://doi.org/10.1111/j.1749-6632.1989.tb20966.x
118. Saho Ayabe-Kanamura, Ina Schicker, Matthias Laska, Robyn Hudson, Hans Distel, Tatsu Kobayakawa and Sachiko Saito. 1998. Differences in Perception of Everyday Odors: a Japanese-German Cross-cultural Study. Chemical Senses, 23(1), 31–38. DOI: https://doi.org/10.1093/chemse/23.1.31
119. Hendrik J. Schifferstein and Charles Spence. 2008. 5 - MULTISENSORY PRODUCT EXPERIENCE. In Hendrik N.J. Schifferstein, Paul Hekkert (Eds.), Product Experience, 133-161. Elsevier. DOI: https://doi.org/10.1016/B978-008045089-6.50008-3
120. Roberta L. Klatzky and Susan J. Lederman. 2010. Multisensory Texture Perception. In Kaiser, J., Naumer, M. (Eds.), Multisensory Object Perception in the Primate Brain. Springer, New York, NY. DOI: https://doi.org/10.1007/978-1-4419-5615-6_12
121. Lawrence L. Garber Jr., Eva M. Hyatt and Richard G. Starr Jr. 2001. Placing Food Color Experimentation into a Valid Consumer Context. Journal of Food Products Marketing, 7(3), 3–24. DOI: https://doi.org/10.1300/J038v07n03_02
122. Hendrik N. J. Schifferstein. 2001. Effects of Product Beliefs on Product Perception and Liking. In Frewer, L.J., Risvik, E., Schifferstein, H. (Eds.), Food, People and Society. Springer, Berlin, Heidelberg. DOI: https://doi.org/10.1007/978-3-662-04601-2_6
123. Peter G. Grossenbacher and Christopher T. Lovelace. 2001. Mechanisms of synesthesia: cognitive and physiological constraints. Trends in Cognitive Sciences, 5(1), 36–41. DOI: https://doi.org/10.1016/S1364-6613(00)01571-0
124. Lorraine E. Bahrick and Robert Lickliter. 2000. Intersensory redundancy guides attentional selectivity and perceptual learning in infancy. Developmental Psychology, 36(2), 190–201. DOI: https://doi.org/10.1037/0012-1649.36.2.190
125. Charles Spence. 2002. The ICI report on the secret of the senses. London: The Communication Group.
126. Barry E. Stein and M. Alex Meredith. 1993. The Merging of the Senses. Cambridge MA: MIT Press.

127. Martin Lindstrom. 2005. Brand Sense: Build Powerful Brands Through Touch, Taste, Smell, Sight and Sound. New York: Free Press.
128. Gerald C Cupchik. 2019. Emotion and industrial design: Reconciling meanings and feelings. In Proceedings of the first international conference on Design & Emotion. DOI: https://doi.org/10.5281/zenodo.2635690
129. Fenko Anna, Hendrik N.J. Schifferstein and Paul Hekkert. 2008. Which senses dominate at different stages of product experience? In Durling, D., Rust, C., Chen, L., Ashton, P. and Friedman, K. (Eds.), Undisciplined! - DRS International Conference 2008, 16-19 July, Sheffield, United Kingdom. DOI: https://dl.designresearchsociety.org/drs-conference-papers/drs2008/researchpapers/19
130. Irvin Rock and Jack Victor. 1964. Vision and Touch: An Experimentally Created Conflict between the Two Senses. Science, 143(3606), 594-596. DOI: 10.1126/science.143.3606.594
131. Epp A. Miller. 1972. Interaction of vision and touch in conflict and nonconflict form perception tasks. Journal of Experimental Psychology, 96, 114-123. DOI: https://doi.org/10.1037/h0033468
132. Robert Teghtsoonian and Martha Teghtsoonian. 1970. Two varieties of perceived length. Perception and Psychophysics, 8, 389-392. DOI: https://doi.org/10.3758/BF03207030
133. Randolph Easton and Peter W. Moran. 1978. A Quantitative Confirmation of Visual Capture of Curvature. Journal of General Psychology, 98, 105-112. DOI: https://doi.org/10.1080/00221309.1978.9920861
134. G. Singer and R. N. Day. 1969. Visual capture of haptually judged depth. Perception & Psychophysics, 6, 203-205.
135. John C. Hay, Herbert L. Pick Jr. and Karren Ikeda. 1965. Visual capture produced by prism spectacles. Psychonomic Science, 2, 215-216. DOI: https://doi.org/10.3758/BF03343413
136. Paul Bertelson. 1999. Ventriloquism: A case of cross-modal perceptual grouping. In G. Aschersleben, T. Bachmann and J. Müsseler (Eds.), Cognitive contributions to the perception of spatial and temporal events. Elsevier, Amsterdam, 347-362. DOI: https://doi.org/10.1016/S0166-4115(99)80034-X
137. Hendrik N. J. Schifferstein and Pieter M. A. Desmet. 2007. The effect of sensory impairments on product experience and personal well-being. Ergonomics, 50, 2026-2048. DOI: https://doi.org/10.1080/00140130701524056
138. Hendrik N. J. Schifferstein. 2006. The relative importance of sensory modalities in product usage: A study of self-reports. Acta Psychologica, 121, 41-64. DOI: https://doi.org/10.1016/j.actpsy.2005.06.004
139. Anna Fenko, Hendrik N.J. Schifferstein and Paul Hekkert. 2007. Shifts in sensory dominance between various stages of user–product interactions. Applied Ergonomics, 41(1), 34-40, ISSN 0003-6870. DOI: https://doi.org/10.1016/j.apergo.2009.03.007
140. William W. Gaver. 1999. Irrational aspects of technology: Anecdotal evidence. In C. J. Overbeeke & P. Hekkert (Eds.), Proceedings of the 1st International Conference on Design and Emotion, 47-54. Delft: Delft University of Technology.
141. C. J. Overbeeke and Paul Hekkert. 1999. Editorial. In Proceedings of the 1st International Conference on Design and Emotion. Delft: Delft University of Technology.

142. Patrick W. Jordan. 2003. Designing great stuff that people love: Foreword by Patrick W. Jordan. In Blythe, M.A., Overbeeke, K., Monk, A.F., Wright, P.C. (Eds.), Funology. Human-Computer Interaction Series 3, xi-xii. Springer, Dordrecht.
143. Caroline Hummels. 2019. Engaging contexts to evoke experiences. In Proceedings of the First International Conference on Design and Emotion. DOI: https://doi.org/10.5281/zenodo.2635707
144. Marc Hassenzahl. 2010. Experience Design: Technology for All the Right Reasons. Springer Cham.
145. Patrick W. Jordan. 1998. Human factors for pleasure in product use. Applied Ergonomics, 29(1), 25–33. DOI: https://doi.org/10.1016/S0003-6870(97)00022-7
146. Pieter Desmet, Paul Hekkert and J.J. Jacobs. 2000. When a Car Makes You Smile: Development and Application of an Instrument to Measure Product Emotions. In Stephen J. Hoch and Robert J. Meyer (Eds.), Advances in Consumer Research, 27, 111-117. Provo, UT: Association for Consumer Research.
147. Marc Hassenzahl and Noam Tractinsky. 2006. User experience - a research agenda. Behaviour and Information Technology, 25(2), 91-97. DOI: 10.1080/01449290500330331
148. John McCarthy and Peter Wright. 2004. Technology as Experience. Cambridge, MA: MIT Press.
149. Pieter Desmet and Paul Hekkert. 2007. Framework of Product Experience. International Journal of Design, 1, 57-66.
150. Nico H. Frijda. 1986. The emotions. Cambridge: Cambridge University Press.
151. Pieter M. A. Desmet. 2002. Designing emotions. Delft: Delft University of Technology.
152. Richard S. Lazarus. 1991. Emotion and Adaptation. Oxford: Oxford University Press.
153. Charlie Ranscombe, Ben Hicks, Glen Mullineux and Baljinder Singh. 2011. Characterizing and Evaluating Aesthetic Features in Vehicle Design. In Proceedings of the 3rd International Conference on Research into Design Engineering (ICORD 11), Bangalore, India.
154. Peter Coughlan and Robert Mashman. 1999. Once is not enough: repeated exposure to and aesthetic evaluation of an automobile design prototype. Design Studies, 20(6), 553–563. DOI:10.1016/S0142-694X(99)00007-1
155. Mitsuo Nagamachi. 1999. Kansei engineering: the implication and applications to product development. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 6, 273-278.
156. Marianna Obrist, Manfred Tscheligi, Boris de Ruyter and Albrecht Schmidt. 2010. Contextual user experience: how to reflect it in interaction designs? In CHI '10 Extended Abstracts on Human Factors in Computing Systems (CHI EA '10). Association for Computing Machinery, New York, NY, USA, 3197–3200. DOI: https://doi.org/10.1145/1753846.1753956
157. Virpi Roto. 2006. COST294-MAUSE Workshop on User Experience—Towards a Unified View in conjunction with NordiCHI’06 Conference, 2006.
158. Gary G. Wang. 2002. Definition and Review of Virtual Prototyping. J. Comput. Inf. Sci. Eng. September 2002, 2(3), 232–236. DOI: https://doi.org/10.1115/1.1526508
159. William R. Sherman and Alan B. Craig. 2003. Understanding virtual reality: Interface, application, and design / William R. Sherman, Alan B. Craig (Morgan Kaufmann series in computer graphics and geometric modeling). San Francisco, CA: Morgan Kaufmann.

160. Stephen R. Ellis. 1995. Human Engineering in Virtual Environments. In Proceedings of Virtual Reality World Conference ’95 (Stuttgart, February 21-23), IDG, Munich, 295–301.
161. Dustin B. Chertoff, Brian Goldiez and Joseph J. LaViola. 2010. Virtual Experience Test: A virtual environment evaluation questionnaire. In Proceedings of the 2010 IEEE Virtual Reality Conference (VR), 103-110. DOI: 10.1109/VR.2010.5444804
162. Dustin B. Chertoff, Sae Lynne Schatz, Rudy McDaniel and Clint A. Bowers. 2008. Improving Presence Theory Through Experiential Design. Presence: Teleoperators and Virtual Environments, 17(4), 405-413. DOI: https://doi.org/10.1162/pres.17.4.405
163. Narcis Parés and Roc Parés. 2006. Towards a Model for a Virtual Reality Experience: The Virtual Subjectiveness. Presence: Teleoperators and Virtual Environments, 15(5), 524–538. DOI: https://doi.org/10.1162/pres.15.5.524
164. Stephen R. Ellis. 1991. Nature and Origins of Virtual Environments: A Bibliographical Essay. Computing Systems in Engineering, 2(4), 321-347. DOI: https://doi.org/10.1016/0956-0521(91)90001-L
165. J.N. Latta and David J. Oberg. 1994. A Conceptual Virtual Reality Model. IEEE Computer Graphics & Applications, 14, 23-29.
166. Eva L. Waterworth and John A. Waterworth. 2001. Focus, Locus, and Sensus: The Three Dimensions of Virtual Experience. Cyberpsychology & Behavior: the impact of the Internet, multimedia and virtual reality on behavior and society, 4(2), 203–213. DOI: https://doi.org/10.1089/109493101300117893
167. Wijnand IJsselsteijn and Giuseppe Riva. 2003. Being there: The experience of presence in mediated environments. In G. Riva, F. Davide, & W. A. IJsselsteijn (Eds.), Being there: Concepts, effects and measurements of user presence in synthetic environments, 3–16. IOS Press.
168. Matthew Lombard and Theresa Ditton. 1997. At the heart of it all: The concept of presence. Journal of Computer Mediated-Communication, 3(2). DOI: https://doi.org/10.1111/j.1083-6101.1997.tb00072.x
169. Thomas B. Sheridan. 1992. Musings on telepresence and virtual presence. Presence: Teleoperators and Virtual Environments, 1(1), 120-125.
170. Jorge L. Borges. 1972. Labyrinths: Selected Stories and Other Writings. Harmondsworth: Penguin.
171. Gordon Calleja. 2007. Digital Game Involvement. Games and Culture, 2(3), 236–260. DOI: 10.1177/1555412007306206.
172. Vero Vanden Abeele, Katta Spiel, Lennart Nacke, Daniel Johnson and Kathrin Gerling. 2020. Development and validation of the player experience inventory: A scale to measure player experiences at the level of functional and psychosocial consequences. International Journal of Human-Computer Studies, 135, 102370. DOI: https://doi.org/10.1016/j.ijhcs.2019.102370
173. Samuel Almeida, Ana Veloso, Licinio Roque and Óscar Mealha. 2013. The Video Game and Player in a Gameplay Experience Model Proposal. In Proceedings of Videojogos 2013 - 6th Annual Conference in the Science and Art of Video Games, Coimbra, Portugal: University of Coimbra, Portugal.
174. Samuel Almeida, Ana Veloso, Óscar Mealha, Licínio Roque and Arnaldo Moura. 2012. A Video Game Level Analysis Model Proposal. In Proceedings of the 16th International Conference on Information Visualisation, 474-479. DOI: 10.1109/IV.2012.82

175. Regina Bernhaupt. 2010. Evaluating User Experience in Games. Concepts and Methods. London: Springer-Verlag London (Human-Computer Interaction Series).
176. Andrew Evans. 2001. This Virtual Life: Escapism in the Media. London: Vision.
177. Csikszentmihalyi Mihaly. 2009. Flow: The psychology of optimal experience. HarperCollins.
178. Laura Ermi and Frans Mäyrä. 2005. Fundamental Components of the Gameplay Experience: Analysing Immersion. In Proceedings of the DiGRA 2005 Conference: Changing Views – Worlds in Play, 3.
179. J. L. González Sánchez, N. Padilla Zea, and Francisco L. Gutiérrez. 2009. Playability: How to Identify the Player Experience in a Video Game. In Tom Gross, Jan Gulliksen, Paula Kotzé, Lars Oestreicher, Philippe Palanque, Raquel Oliveira Prates and Marco Winckler (Eds.): Human-Computer Interaction – INTERACT 2009, Lecture Notes in Computer Science, 5726. Springer, Berlin, Heidelberg. DOI: https://doi.org/10.1007/978-3-642-03655-2_39
180. ISO 9241-11: Guidance on Usability, also issued by the International Organization for Standardization. 1998.
181. Christopher A. Oswald, Chris Prorock and Shane M. Murphy. 2014. The perceived meaning of the video game experience: An exploratory study. Psychology of Popular Media Culture, 3(2), 110–126. DOI: https://doi.org/10.1037/a0033828
182. Karolien Poels, Yvonne de Kort and Wijnand Ijsselsteijn. 2007. "It is always a lot of fun!": exploring dimensions of digital game experience using focus group methodology. In Proceedings of the 2007 conference on Future Play (Future Play '07). Association for Computing Machinery, New York, NY, USA, 83–89. DOI: https://doi.org/10.1145/1328202.1328218
183. Elizabeth Gerber and Maureen Carroll. 2012. The psychological experience of prototyping. Design Studies, 33(1), 64–84. DOI: https://doi.org/10.1016/j.destud.2011.06.005
184. Stephanie Houde and Charles Hill. 1997. What do Prototypes Prototype. In Helander, M., Landauer, T., and Prabh, P. (Eds.), Handbook of Human-Computer Interaction (2nd Ed.), 367– 381. Elsevier Science, Amsterdam, The Netherlands, 2nd edition.
185. Yvonne Rogers, Helen Sharp and Jenny Preece. 2002. Interaction Design: Beyond Human-Computer Interaction. Wiley John + Sons.
186. Yin Yin Wong. 1992. Rough and ready prototypes: lessons from graphic design. In Posters and Short Talks of the 1992 SIGCHI Conference on Human Factors in Computing Systems (CHI '92). Association for Computing Machinery, New York, NY, USA, 83–84. DOI: https://doi.org/10.1145/1125021.1125094
187. Elbert J. Hennipman, Evert J. R. G. Oppelaar, Gerrit C. van der Veer and Bert Bongers. 2008. Rapid and rich prototyping: proof of concepts for experience. In Proceedings of the 15th European Conference on Cognitive Ergonomics - ECCE '08, Madeira, Portugal, ACM, Jan. 2008, 6 pages.
188. V. Scott Gordon and James M. Bieman. 1995. Rapid Prototyping: Lessons Learned. IEEE Software, 12(1), 85-95. DOI: 10.1109/52.363162
189. Roberta L. Klatzky and Susan J. Lederman 2003. Touch. In Handbook of Psychology: Experimental Psychology, edited by I. B. Weiner, pp. 147–176. New York: John Wiley & Sons.

190. Youn-Kyung Lim, Erik Stolterman and Josh Tenenberg. 2008. The anatomy of prototypes: Prototypes as filters, prototypes as manifestations of design ideas. ACM Trans. Comput.-Hum. Interact. 15, 2, Article 7 (July 2008), 27 pages. DOI: https://doi.org/10.1145/1375761.1375762
191. Hans-Peter Krueger, Martin Grein, Armin Kaussner and Christian Mark. 2005. SILAB – A Task Oriented Driving Simulation. In Proceedings of the Driving Simulation Conference (DSC'05), Orlando FL, United States, 323–331.
192. Arthur Barz, Jan Conrad and Dieter Wallach. 2020. Advantages of Using Runtime Procedural Generation of Virtual Environments Based on Real World Data for Conducting Empirical Automotive Research. In Stephanidis, C., Duffy, V.G., Streitz, N., Konomi, S., Krömker, H. (Eds.), HCI International 2020 – Late Breaking Papers: Digital Human Modeling and Ergonomics, Mobility and Intelligent Environments. HCII 2020. Lecture Notes in Computer Science, 12429. Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-59987-4_2
193. Mark Colley, Marcel Walch and Enrico Rukzio. 2019. For a better (simulated) world: considerations for VR in external communication research. In Proceedings of the 11th International Conference on Automotive User Interfaces and Interactive Vehicular Applications: Adjunct Proceedings (AutomotiveUI '19). Association for Computing Machinery, New York, NY, USA, 442–449. DOI: https://doi.org/10.1145/3349263.3351523
194. Mark McGill and Stephen Brewster. 2019. Virtual reality passenger experiences. In Proceedings of the 11th International Conference on Automotive User Interfaces and Interactive Vehicular Applications: Adjunct Proceedings (AutomotiveUI '19). Association for Computing Machinery, New York, NY, USA, 434–441. DOI: https://doi.org/10.1145/3349263.3351330
195. Jingyi Li, Ceenu George, Andrea Ngao, Kai Holländer, Stefan Mayer and Andreas Butz. 2020. An Exploration of Users' Thoughts on Rear-Seat Productivity in Virtual Reality. In Proceedings of the 12th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '20). Association for Computing Machinery, New York, NY, USA, 92–95. DOI: https://doi.org/10.1145/3409251.3411732
196. Giandomenico Caruso. 2011. Mixed Reality System for Ergonomic Assessment of Driver's Seat. Int. J. Virtual Real. 10, 2, 69–79. DOI: https://doi.org/10.20870/ijvr.2011.10.2.2813
197. Matthias de Clerk, Manfred Dangelmaier, Gernot Schmierer and Dieter Spath. 2019. User centered design of interaction techniques for VR-based automotive design reviews. Front. Robot. AI 6. DOI: https://doi.org/10.3389/frobt.2019.00013
198. The Procedural Nature of the Horizon Zero Dawn (80.lv), The Procedural Nature of the Horizon Zero Dawn (80.lv) [last accessed 1 July 2022].
199. Procedural Technology in Ghost Recon: Wildlands (80.lv), Procedural Technology in Ghost Recon: Wildlands (80.lv) [last accessed 1 July 2022].
200. Guoning Chen, Gregory Esch, Peter Wonka, Pascal Müller and Eugene Zhang. 2008. Interactive procedural street modeling. ACM Trans. Graph. 27, 3 (August 2008), 1–10. DOI: https://doi.org/10.1145/1360612.1360702
201. Yoav I. H. Parish and Pascal Müller. 2001. Procedural modeling of cities. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH

'01). Association for Computing Machinery, New York, NY, USA, 301–308. DOI: https://doi.org/10.1145/383259.383292

202. Eric Bruneton and Fabrice Neyret. 2008. Real-time rendering and editing of vector-based terrains. Computer Graphics Forum, Wiley, 2008, Special Issue: Eurographics ’08, 27(2), 311-320. DOI: ff10.1111/j.1467-8659.2008.01128.xff. ffinria-00207679f
203. Jing Sun, Xiaobo Yu, George Baciu and Mark Green. 2002. Template-based generation of road networks for virtual city modeling. In Proceedings of the ACM symposium on Virtual reality software and technology (VRST '02). Association for Computing Machinery, New York, NY, USA, 33–40. DOI: https://doi.org/10.1145/585740.585747
204. Ruben Smelik, Tim Tutenel, Klaas Jan de Kraker and Rafael Bidarra. 2010. Integrating procedural generation and manual editing of virtual worlds. In Proceedings of the 2010 Workshop on Procedural Content Generation in Games (PCGames '10). Association for Computing Machinery, New York, NY, USA, Article 2, 1–8. DOI: https://doi.org/10.1145/1814256.1814258
205. Ruben M. Smelik, Tim Tutenel, Klaas Jan de Kraker and Rafael Bidarra. 2011. Semantic 3D Media and Content: A declarative approach to procedural modeling of virtual worlds. Comput. Graph. 35, 2 (April, 2011), 352–363. DOI: https://doi.org/10.1016/j.cag.2010.11.011
206. George Kelly and Hugh McCabe. 2007. Citygen: An interactive system for procedural city generation. In Proceedings of the GDTW 2007: The Fifth Annual Conference in Computer Game Design and Technology, 8–16.
207. Kevin R. Glass, Chantelle Morkel and Shaun D. Bangay. 2006. Duplicating road patterns in south african informal settlements using procedural techniques. In Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa (AFRIGRAPH '06). Association for Computing Machinery, New York, NY, USA, 161–169. DOI: https://doi.org/10.1145/1108590.1108616
208. Zhanfeng Jia and P. Varaiya. 2004. Grid discretization based method for anisotropic shortest path problem over continuous regions. In Proceedings of the 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601), 2004, 4, 3830-3837. DOI: 10.1109/CDC.2004.1429335
209. Lyudmil Aleksandrov, Anil Maheshwari and Jörg-Rüdiger Sack. 2005. Determining approximate shortest paths on weighted polyhedral surfaces. J. ACM 52, 1 (January 2005), 25–53. DOI: https://doi.org/10.1145/1044731.1044733
210. Gustavo Alomía, Diego Loaiza, Claudia Zúñiga, Xun Luo and Rafael Asorey-Cacheda. 2021. Procedural modeling applied to the 3D city model of Bogota: a case study. In Virtual Reality & Intelligent Hardware, 3(5), 2021, 423-433, ISSN 2096-5796. DOI: https://doi.org/10.1016/j.vrih.2021.06.002
211. Peter Dam, Fernanda Duarte and Alberto Raposo. 2019. Terrain Generation Based on Real World Locations for Military Training and Simulation. In Proceedings of the 18th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames), 2019, 173-181, DOI: 10.1109/SBGames.2019.00031
212. CityGen3D, https://www.citygen3d.com/, [last accessed 1 July 2022].
213. Tom Kelly. 2021. CityEngine: An Introduction to Rule-Based Modeling. Urban Informatics. The Urban Book Series. Springer, Singapore. DOI: https://doi.org/10.1007/978-981-15-8983-6_35

214. https://maps.probabletrain.com/#/, [last accessed 1 July 2022].
215. http://www.cgchan.com, [last accessed 1 July 2022].
216. https://docs.mapbox.com/unity/maps/examples/city-simulator/, [last accessed 1 July 2022].
217. https://cesium.com/learn/unreal/unreal-quickstart/, [last accessed 1 July 2022].
218. Przemyslaw Prusinkiewicz. 1986. Graphical applications of L-systems. In Proceedings of Graphics Interface '86/Vision Interface '86, 247–253. Canadian Information Processing Society, CAN.
219. Jean-Eudes Marvie, Julien Perret and Kadi Bouatouch. 2005. FL-system: A Functional L-system for procedural geometric modeling. The Visual Computer, 21, 5 (June 2005), 329-339. DOI: 10.1007/s00371-005-0289-z
220. Yoav I. H. Parish and Pascal Müller. 2001. Procedural modeling of cities. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH '01). Association for Computing Machinery, New York, NY, USA, 301–308. DOI: https://doi.org/10.1145/383259.383292
221. Andrew Rau-Chaplin, Bryan MacKay-Lyons and Peter F.Spieremburg. 1996. The LaHave House Project: Towards an Automated Architectural Design Service. Technical University of N.S., School of Computer Science, Architecture + Urban Design, Halifax, Canada.
222. Mathieu Larive and Veronique Gaildrat. 2006. Wall grammar for building generation. In Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia (GRAPHITE '06). Association for Computing Machinery, New York, NY, USA, 429–437. DOI: https://doi.org/10.1145/1174429.1174501
223. Michael Schwarz and Pascal Müller. 2015. Advanced procedural modeling of architecture. ACM Trans. Graph. 34, 4, Article 107 (August 2015), 12 pages. DOI: https://doi.org/10.1145/2766956
224. Diego Jesus, António Coelho and António A. Sousa. 2016. Layered shape grammars for procedural modelling of buildings. Springer-Verlag Berlin Heidelberg.
225. Peter Wonka, Michael Wimmer, François Sillion and William Ribarsky. 2003. Instant architecture. ACM Trans. Graph. 22, 3 (July 2003), 669–677. DOI: https://doi.org/10.1145/882262.882324
226. Tim Tutenel, Ruben M. Smelik, Ricardo Lopes, Klaas Jan de Kraker and Rafael Bidarra. 2011. Generating Consistent Buildings: A Semantic Approach for Integrating Procedural Techniques. IEEE Trans. Comput. Intell. AI Games 3(3), 274-288.
227. Ivan Silveira, Daniel Camozzato, Fernando Marson, Leandro Dihl and Soraia R. Musse. 2015. Real-time procedural generation of personalized facade and interior appearances based on semantics. In Proceedings of the 14th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames), 89-98. DOI: 10.1109/SBGames.2015.32
228. Paul Merrell, Eric Schkufza and Vladlen Koltun. 2010. Computer-generated residential building layouts. ACM Trans. Graph. 29, 6, Article 181 (December 2010), 12 pages. DOI: https://doi.org/10.1145/1882261.1866203
229. Filip Biljecki, Hugo Ledoux and Jantien Stoter. 2016. Generation of multi-LOD 3D city models in CityGML with the procedural modelling engine Random3Dcity. In ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 4(4W1), 51-59. DOI: https://doi.org/10.5194/isprs-annals-IV-4-W1-51-2016

230. Peter Dam, Fernanda Duarte, and Alberto Raposo. 2019. Terrain Generation Based on Real World Locations for Military Training and Simulation. In Brazilian Symposium on Games and Digital Entertainment (SBGAMES), IEEE Computer Society, Rio de Janeiro, Brazil, 173–181. DOI: https://doi.org/10.1109/SBGames.2019.00031
231. Werner Gaisbauer, William L. Raffe, Jaime A. Garcia and Helmut Hlavacs. 2019. Procedural generation of video game cities for specific video game genres using WaveFunctionCollapse (WFC). In Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play (CHI PLAY ’19). Association for Computing Machinery, Inc, 397–404. DOI: https://doi.org/10.1145/3341215.3356255
232. Pascal Müller, Gang Zeng, Peter Wonka and Luc Van Gool. 2007. Image-based procedural modeling of facades. ACM Trans. Graph. 26, 3 (July 2007), 85–es. DOI: https://doi.org/10.1145/1276377.1276484
233. Paul Merrell. 2007. Example-based model synthesis. In Proceedings of the 2007 symposium on Interactive 3D graphics and games (I3D '07). Association for Computing Machinery, New York, NY, USA, 105–112. DOI: https://doi.org/10.1145/1230100.1230119
234. Paul Merrell and Dinesh Manocha. 2008. Continuous model synthesis. ACM Trans. Graph. 27, 5, Article 158 (December 2008), 7 pages. DOI: https://doi.org/10.1145/1409060.1409111
235. Paul Merrell and Dinesh Manocha. 2009. Constraint-based model synthesis. In 2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling (SPM '09). Association for Computing Machinery, New York, NY, USA, 101–111. DOI: https://doi.org/10.1145/1629255.1629269
236. Ruben M. Smelik, Tim Tutenel, Rafael Bidarra and Bedrich Benes. 2014. A survey on procedural modelling for virtual worlds. Comput. Graph. Forum, 33(6), 31–50.
237. Ruben M. Smelik, Klaas Jan De Kraker, Saskia A. Groenewegen, Tim Tutenel and Rafael Bidarra. 2009. A survey of procedural methods for terrain modelling. In Proceedings of the CASA Workshop on 3D Advanced Media In Gaming And Simulation (3AMIGAS), 25–34.
238. Ian Parberry. 2014. Designer Worlds: Procedural Generation of Infinite Terrain from Real-World Elevation Data. J. Comput. Graph. Tech., 3(1), 74–85.
239. Houssam Hnaidi, Eric Guérin, Samir Akkouche, Adrien Peytavie and Eric Galin. 2010. Feature based terrain generation using diffusion equation. Comput. Graph. Forum, 29(7), 2179–2186.
240. Jean-David Génevaux, Éric Galin, Eric Guérin, Adrien Peytavie and Bedrich Benes. 2013. Terrain generation using procedural models based on hydrology. ACM Trans. Graph. 32, 4, Article 143 (July 2013), 13 pages. DOI: https://doi.org/10.1145/2461912.2461996
241. Alex D. Kelley, Michael C. Malin and Gregory M. Nielson. 1988. Terrain simulation using a model of stream erosion. In Proceedings of the 15th annual conference on Computer graphics and interactive techniques (SIGGRAPH '88). Association for Computing Machinery, New York, NY, USA, 263–268. DOI: https://doi.org/10.1145/54852.378519
242. Przemyslaw Prusinkiewicz and Mark Hammel. 1993. A Fractal Model of Mountains with Rivers. In Proceeding of Graphics Interface ’93, 174–180.
243. Farès Belhadj and Pierre Audibert. 2005. Modeling landscapes with ridges and rivers: bottom up approach. In Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia (GRAPHITE '05).

Association for Computing Machinery, New York, NY, USA, 447–450. DOI: https://doi.org/10.1145/1101389.1101479
244. Przemyslaw Prusinkiewicz and Aristid Lindenmayer. 1990. The Algorithmic Beauty of Plants. Springer-Verlag, New York, NY, USA.
245. Bernd Lintermann and Oliver Deussen. 1999. Interactive Modeling of Plants. IEEE Computer Graphics and Applications, 19(1), 56-65. DOI: 10.1109/38.736469
246. Miłosz Makowski, Torsten Hädrich, Jan Scheffczyk, Dominik L. Michels, Sören Pirk and Wojtek Pałubicki. 2019. Synthetic silviculture: multi-scale modeling of plant ecosystems. ACM Trans. Graph. 38, 4, Article 131 (August 2019), 14 pages. DOI: https://doi.org/10.1145/3306346.3323039
247. Takashi Ijiri, Shigeru Owada and Takeo Igarashi. 2006. Seamless Integration of Initial Sketching and Subsequent Detail Editing in Flower Modeling. Comp. Graph. Forum, 25(3), 617–624.
248. Makoto Okabe, Shigeru Owada and Takeo Igarashi. 2006. Interactive design of botanical trees using freehand sketches and example-based editing. In ACM SIGGRAPH 2006 Courses (SIGGRAPH '06). Association for Computing Machinery, New York, NY, USA, 18–es. DOI: https://doi.org/10.1145/1185657.1185779
249. Jamie Wither, Frédéric Boudon, Marie-Paule Cani and Christophe Godin. 2009. Structure from silhouettes: a new paradigm for fast sketch-based design of trees. Computer Graphics Forum 28, 2 (2009), 541–550. DOI: https://doi.org/10.1111/j.1467-8659.2009.01394.x
250. Derek Bradley, Derek Nowrouzezahrai and Paul Beardsley. 2013. Image-based reconstruction and synthesis of dense foliage. ACM Trans. Graph. 32, 4, Article 74 (July 2013), 10 pages. DOI: https://doi.org/10.1145/2461912.2461952
251. Boris Neubert, Thomas Franken and Oliver Deussen. 2007. Approximate image-based tree-modeling using particle flows. In ACM SIGGRAPH 2007 papers (SIGGRAPH '07). Association for Computing Machinery, New York, NY, USA, 88–es. DOI: https://doi.org/10.1145/1275808.1276487
252. Ping Tan, Tian Fang, Jianxiong Xiao, Peng Zhao, and Long Quan. 2008. Single image tree modeling. In ACM SIGGRAPH Asia 2008 papers (SIGGRAPH Asia '08). Association for Computing Machinery, New York, NY, USA, Article 108, 1–7. DOI: https://doi.org/10.1145/1457515.1409061
253. Ping Tan, Gang Zeng, Jingdong Wang, Sing Bing Kang and Long Quan. 2007. Image-based tree modeling. ACM Trans. Graph. 26, 3 (July 2007), 87–es. DOI: https://doi.org/10.1145/1276377.1276486
254. Chuan Li, Oliver Deussen, Yi-Zhe Song, Phil Willis and Peter Hall. 2011. Modeling and generating moving trees from video. ACM Trans. Graph. 30, 6 (December 2011), 1–12. DOI: https://doi.org/10.1145/2070781.2024161
255. Shaojun Hu, Zhengrong Li, Zhang Zhiyi, Dongjian He and Michael Wimmer. 2017. Efficient tree modeling from airborne LiDAR point clouds. Comput. Graph. 67, C (October 2017), 1–13. DOI: https://doi.org/10.1016/j.cag.2017.04.004
256. Yotam Livny, Soeren Pirk, Zhanglin Cheng, Feilong Yan, Oliver Deussen, Daniel Cohen-Or, and Baoquan Chen. 2011. Texture-lobes for tree modelling. In ACM SIGGRAPH 2011 papers (SIGGRAPH '11). Association for Computing Machinery, New York, NY, USA, Article 53, 1–10. DOI: https://doi.org/10.1145/1964921.1964948

257. Ke Xie, Feilong Yan, Andrei Sharf, Oliver Deussen, Hui Huang and Baoquan Chen. 2016. Tree Modeling with Real Tree-Parts Examples. IEEE Transactions on Visualization and Computer Graphics, 22(12), 2608-2618. DOI: 10.1109/TVCG.2015.2513409
258. Neil A. Campbell, Jane B. Reece, Lisa A. Urry, Michael L. Cain, Steven A. Wasserman, Peter V. Minorsky and Robert B. Jackson. 2009. Biology (8th Edition). Pearson Benjamin Cummings.
259. Oliver Deussen, Pat Hanrahan, Bernd Lintermann, Radomír Měch, Matt Pharr and Przemyslaw Prusinkiewicz. 1998. Realistic modeling and rendering of plant ecosystems. In Proceedings of the 25th annual conference on Computer graphics and interactive techniques (SIGGRAPH '98). Association for Computing Machinery, New York, NY, USA, 275–286. DOI: https://doi.org/10.1145/280814.280898
260. Johan Hammes. 2001. Modeling of Ecosystems as a Data Source for Real-Time Terrain Rendering. In Proceedings of the First International Symposium on Digital Earth Moving (DEM '01). Springer-Verlag, Berlin, Heidelberg, 98–111.
261. Roland Fischer, Philipp Dittmann, René Weller and Gabriel Zachmann. 2020. AutoBiomes: procedural generation of multi-biome landscapes. Vis. Comput. 36, 10–12 (Oct 2020), 2263–2272. DOI: https://doi.org/10.1007/s00371-020-01920-7
262. Robert H. Whittaker. 1975. Communities and ecosystems. Macmillan USA.
263. Richard H. Waring and Steven W. Running. 2007. Forest Ecosystem Analysis at Multiple Time and Space Scales. Forest Ecosystems (Third Edition), 1 – 16. Academic Press.
264. James I. Drever. 2005. Surface and Ground Water, Weathering, and Soils. Elsevier Science.
265. Mehdi Heydari and Ali Mahdavi. 2009. The Survey of Plant Species Diversity and Richness Between Ecological Species Groups (Zagros Ecosystem, Ilam). Journal of Applied Sciences, 9(4), 745-751. DOI: 10.3923/jas.2009.745.751
266. F. L. Zhang, J. J. Wang, S. H. Liu, and S. M. Zhang. 2016. Development of economic and environmental metrics for forest-based biomass harvesting. IOP Conference Series: Earth and Environmental Science 40, 1. DOI: https://doi.org/10.1088/1755-1315/40/1/012052
267. Rodney J. Keenan. 2015. Climate change impacts and adaptation in forest management: a review. Annals of Forest Science 72, 145–167. DOI: https://doi.org/10.1007/s13595-014-0446-5
268. Greg Guest, Arwen Bunce and Laura Johnson. 2006. How many interviews are enough? An experiment with data saturation and variability. Field Methods, 18(1), 59.82. DOI: https://doi.org/10.1177/1525822X05279903
269. Despoina Salpisti, Matthias de Clerk, Sebastian Hinz, Frank Henkies and Gudrun Klinker. 2022. Procedural Street Generation to Support Automotive Designers in the Creation of Context for Mixed Reality Design Experiences. In Proceedings of the European Transport Conference 2022, Milan, Italy.
270. Despoina Salpisti, Matthias de Clerk, Sebastian Hinz, Frank Henkies and Gudrun Klinker. 2022. Requirements for 3D Environments as the Context for Mixed Reality Automotive Design Experiences. In Proceedings of the Driving Simulation Conference 2022, Strasbourg, France.
271. Despoina Salpisti, Matthias de Clerk, Sebastian Hinz, Frank Henkies and Gudrun Klinker. 2022. A Procedural Building Generator based on Real-World Data Enabling

Designers to Create Context for XR Automotive Design Experiences. In Proceedings of the EuroXR Conference 2022, Stuttgart, Germany.
272. Houdini Software Official Website: https://www.sidefx.com/ [last accessed 1 July 2022].
273. Gavin Ambrose, Paul Harris and Sally Stone. 2008. The Visual Dictionary of Architecture. Switzerland: AVA Publishing SA, 233. ISBN 978-2-940373-54-3
274. GPA Consulting. 2016. Architecture and Engineering, 1850-1980: Beaux Arts Classicism, Neoclassical, and Italian Renaissance Revival Architecture, 1895-1940. Los Angeles Citywide Historic Context Statement, City of Los Angeles Office of Historic Resources.
275. Roger Dixon and Stefan Muthesius. 1978. Victorian Architecture: With a Short Dictionary of Architects and 251 Illustrations, ISBN 978-0-500-18163-8. Thames and Hudson.
276. GPA Consulting. 2016. Architecture and Engineering, 1850-1980: Arts and Crafts Movement, 1895-1930. Los Angeles Citywide Historic Context Statement, City of Los Angeles Office of Historic Resources.
277. GPA Consulting. 2016. Architecture and Engineering, 1850-1980: American Colonial Revival, 1895-1960. Los Angeles Citywide Historic Context Statement, City of Los Angeles Office of Historic Resources.
278. John London. 2002. A review of The Cambridge Companion to Modern German Culture, edited by Eva Kolinsky and Wilfried van der Will. Contemporary Theatre Review, 12, 1-2, 269-271. DOI: 10.1080/10486800208568668
279. Qing Chang. 2019. Architectural Models and Their Contexts in China’s 20th-Century Architectural Heritage: An Overview. Built Heritage 3, 1–13. DOI: https://doi.org/10.1186/BF03545715
280. Nicolas Davies and Erkki Jokiniemi. 2012. Architect's Illustrated Pocket Dictionary. Taylor & Francis Ltd.
281. Quixel Megascans Official Website: https://quixel.com/megascans/home [last accessed 1 July 2022]
282. Hylke E. Beck, Niklaus E. Zimmermann, Tim R. McVicar, Noemi Vergopolan, Alexis Berg and Eric F. Wood. 2018. Present and future Köppen-Geiger climate classification maps at 1-km resolution. Sci Data 5, 180214. DOI: https://doi.org/10.1038/sdata.2018.214
283. https://proj.org/operations/projections/eqc.html, [last accessed 1 July 2022].
284. Tom L. McKnight and Darrel Hess. 2000. Climate Zones and Types. Physical Geography: A Landscape Appreciation, ISBN 978-0-13-020263-5. Upper Saddle River, NJ: Prentice Hall.
285. https://www.britannica.com/science/tropical-rainforest/Biota [last accessed 1 July 2022].
286. tropical dry forest | Description, Biome, Ecosystem, Plants, Animals, & Facts | Britannica, https://www.britannica.com/science/tropical-dry-forest [last accessed 1 July 2022].
287. savanna - Flora | Britannica, https://www.britannica.com/science/savanna/Flora [last accessed 1 July 2022].
288. desert - Biota | Britannica, https://www.britannica.com/science/desert/Biota [last accessed 1 July 2022].

289. Péter Török, Iwona Dembicz, Zora Dajić-Stevanović and Anna Kuzemko. 2020. Grasslands of Eastern Europe. Encyclopedia of the World's Biomes, 703-713, ISBN 9780128160978. Elsevier. DOI: https://doi.org/10.1016/B978-0-12-409548-9.12042-1
290. Juan Gaitan, Donaldo E. Bran and Gabriel Oliva. 2019. Patagonian Desert. Encyclopedia of the World's Biomes, 163-180, ISBN 9780128160978. Elsevier. DOI: https://doi.org/10.1016/B978-0-12-409548-9.11929-3
291. https://www.britannica.com/science/maquis-vegetation [last accessed 1 July 2022].
292. https://www.britannica.com/plant/chaparral [last accessed 1 July 2022].
293. deciduous forest | Definition, Climate, & Characteristics | Britannica, https://www.britannica.com/science/deciduous-forest [last accessed 1 July 2022]
294. coniferous forest | Definition & Facts | Britannica, https://www.britannica.com/science/coniferous-forest [last accessed 1 July 2022].
295. taiga | Plants, Animals, Climate, Location, & Facts | Britannica, https://www.britannica.com/science/taiga [last accessed 1 July 2022].
296. tundra - The biota and its adaptations | Britannica, https://www.britannica.com/science/tundra/The-biota-and-its-adaptations [last accessed 1 July 2022].
297. Level Streaming Overview | Unreal Engine Documentation, https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/LevelStreaming/Overview/ [last accessed 1 July 2022].
298. Geographically Accurate Sun Positioning | Unreal Engine Documentation, https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/LightingAndShadows/SunPositioner/ [last accessed 1 July 2022].
299. MetaHuman Creator - Unreal Engine, https://www.unrealengine.com/en-US/metahuman-creator [last accessed 1 July 2022].
300. Nancy Carter, Denise Bryant-Lukosius, Alba DiCenso, Jennifer Blythe and Alan J Neville. 2014. The use of triangulation in qualitative research. Oncol Nurs Forum. 2014 Sep;41(5), 545-7. DOI: 10.1188/14.ONF.545-547.
301. Martin Schrepp, Jörg Thomaschewski and Andreas Hinderks. 2017. Design and Evaluation of a Short Version of the User Experience Questionnaire (UEQ-S). International Journal of Interactive Multimedia and Artificial Intelligence, 4(6), 103-108. DOI: 10.9781/ijimai.2017.09.001
302. Lisa Mlekus, Dominik Bentler, Agnieszka Paruzel, Anna-Lena Kato-Beiderwieden and Günter W. Maier. 2020. How to raise technology acceptance: user experience characteristics as technology-inherent determinants. Gr Interakt Org 51, 273–283. DOI: https://doi.org/10.1007/s11612-020-00529-7
303. Joseph F. Hair, William C. Black, Barry J. Babin and Rolph E. Anderson. 2010. Multivariate Data Analysis (Seventh Edition). Prentice Hall, Upper Saddle River, New Jersey.
304. Mark Mason. 2010. Sample Size and Saturation in PhD Studies Using Qualitative Interviews. Forum Qualitative Sozialforschung / Forum: Qualitative Social Research, 11(3). DOI: https://doi.org/10.17169/fqs-11.3.1428
305. Sarah E. Baker and Rosalind Edwards. 2012. How many qualitative interviews is enough. In Baker, S.E. and Edwards, R. (Eds.), National Centre for Research Methods Review Paper.

306. Satu Elo and Helvi Kyngäs. 2008. The qualitative content analysis process. Journal of Advanced Nursing, 62(1), 107-115. DOI: 10.1111/j.1365-2648.2007.04569.x
307. Marc Hassenzahl, Michael Burmester and Franz Koller. 2003. AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität [AttrakDiff: A questionnaire for measuring perceived hedonic and pragmatic quality]. Mensch & Computer 2003: Interaktion in Bewegung, 187–196. Stuttgart: B.G. Teubner.
308. Stephan R. Richter, Hassan Abu AlHaija and Vladlen Koltun. 2021. Enhancing Photorealism Enhancement. IEEE Transactions on Pattern Analysis and Machine Intelligence. DOI: https://doi.org/10.48550/arXiv.2105.04619