ESTOA

Procedural architectural settlement generator for container housing: A study on Marmara and Mediterranean Regions

Generador de asentamientos arquitectónicos procedurales para contenedores: un estudio en las regiones de Mármara y Mediterráneo

Authors:
Asena Kumsal Şen-Bayram*
asenakumsalsenbayram@maltepe.edu.tr
Belinda Torus**
belindatorus@gmail.com
Oğuz Orkun Doma***
doma@itu.edu.tr
Sinan Mert Şener ***
mert@itu.edu.tr

*Maltepe University
**Bahçeşehir University
***Istanbul Technical University

Turkey

Submitted: 29/11/2022
Accepted: 07/06/2023
Published: 19/07/2023

Abstract:

There is a growing dwelling problem on a global scale. The crisis is getting more severe each day with incidents like war and pandemics, which resulted in some groups seeking alternative life scenarios. Therefore, expanding the research related to housing and answering these needs become obligatory. Game engines and procedural generation have been used to rapidly represent solutions with their high-quality renderings for these kinds of situations. This paper aims to present a user-friendly container settlement generation tool using procedural generation, developed using the Unity 3D game engine, focusing on various real-life scenarios based on contextual and financial parameters. Along with the tool's development and functionalities, the paper presents a case study with ecological and post-disaster scenario presets in the Marmara and Mediterranean Regions of Türkiye to demonstrate its applicability in the settlement generation process.

Keywords: procedural generation; game engines; container housing; post-disaster housing; Unity 3D.

Resumen:

Existe un creciente problema de vivienda a nivel mundial. La crisis se está agravando cada día con incidentes como la guerra y las pandemias, lo que ha resultado en que algunos grupos busquen escenarios de vida alternativos. Por lo tanto, ampliar la investigación relacionada con la vivienda y responder a estas necesidades se vuelve obligatorio. Los motores de juegos y la generación procedural se han utilizado para representar rápidamente soluciones con sus renderizaciones de alta calidad para estos tipos de situaciones. Este artículo tiene como objetivo presentar una herramienta de generación de asentamientos de contenedores fácil de usar mediante la generación procedural, desarrollada usando el motor de juegos Unity 3D, centrándose en varios escenarios de la vida real basados en parámetros contextuales y financieros. Junto con el desarrollo y las funcionalidades de la herramienta, el artículo también presenta un estudio de caso con presets de escenario ecológico y postdesastre en las regiones turcas de Mármara y Mediterráneo para demostrar su aplicabilidad en el proceso de generación de asentamientos.

Palabras clave: generación procedural; motores de juegos; viviendas de contenedores; viviendas postdesastre; Unity 3D.

1. Introduction

Persecution, violence, and human rights violations had driven 84 million people from their homes by mid-2021 (UNHCR, 2021). Aside from ongoing issues and disagreements, an unanticipated global public health emergency in 2020 posed a new threat to people's safety and livelihoods. One hundred sixty-eight countries blocked their borders entirely or partially during the initial wave of the pandemic in April, with ninety countries making no exceptions for asylum seekers (UNHCR, 2020). Besides refugees, COVID-19 also changed daily life in big cities. Some people started to search for alternative accommodations out of the city, changing their lifestyles to avoid quarantine restrictions. Designing various types of dwellings to answer numerous living needs has always been one of the primary roles of architects. However, the abovementioned situations made rapidly designing dwelling settlements on different scales more critical and urgent for architects and urban designers.

Settlements are complex systems with lots of variables. Many constraints (such as architectural, structural, and social) and their interactions with subsystems raise difficulties in making accurate decisions while designing a settlement manually. Accordingly, analyzing the situation and development of a settlement design requires a long time, high cost, and interdisciplinary research. This multi-dimensioned practice includes many actors, from the first decision-maker to the last user. The decision-makers are responsible for arranging and organizing all these subjects on a bigger scale. Since their decisions will outline and affect the formations of the settlements, they need to understand and make appropriate decisions about these settlements concerning the user's needs and requirements. Therefore, it is essential to include the users in this process to close the gap between decision-makers ideas and the real-life needs of various user profiles. Architects and designers have started using algorithms, developing various tools, and building digital prototypes to compute such complex design problems for the last few decades. In the settlement case, the analysis and estimations can also be a part of these virtual generations (Akgül et al., 2017). These tools help the developer calculate, design, and model rapidly for optimized settlements (Alaçam et al., 2022). Apart from creating a rapid and effective solution, 3D modelling and realistic representation became prominent for different actors (Aydin et al., 2017). Game engines became one of the most popular platforms for developing these tools because of their 3D visualizations, photo-realistic and real-time rendering abilities. Representing a complex settlement generation system in a game-like environment also helps to fill the gap between users and various actors.

This paper presents a rapid, user-friendly settlement generation tool called bBox, focusing on real-life scenarios. One of the main characteristics of this tool is that it is being developed by architects to understand and implement these real-life scenarios that can accommodate multiple households with architectural touch. Secondly, a game engine is used to develop bBox, to generate high-quality, real-time renders for users to comprehend the generated models more realistically. Furthermore, since bBox is designed by various actors and users related to the generation, gamified production creates a user-friendly environment.

Following the introduction, this study continues with a detailed definition of the designed procedural settlement generator. The second chapter starts with a literature review focusing on procedural settlement generators that also point out from which previous research the study was influenced and, with additional features, how it aims to fill a gap in the literature. This definition of the main framework continues with details about the scenarios, container configurations, interface, outputs and the generation process.

The in-depth explanation of the tool continues with a case study focusing on the Marmara and Mediterranean Regions of Türkiye, which have different climate parameters. Comparing these two regions was also strengthened with generations using ecological and post-disaster settlement presets to illustrate the tool's capabilities and potential.

The study ends with a conclusion in which the outcomes of the case study and the generator's overall design are discussed with future predictions about the tool's potential.

2. A procedural settlement generator

Procedural modelling has been used to generate designs for complex urban structures using a set of parameters and rules (Gürbüz et al., 2010). Several production systems, such as attributed grammars (Knuth, 1968), shape grammars (Stiny, 1975), L-systems (Lindenmayer & Prusinkiewicz, 1991), Semi-Thue processes (Davis et al., 1994), Chomsky grammars (Sipser, 1996), graph grammars (Ehrig et al., 1999), and set grammars (Wonka et al., 2003) can be used for procedural architectural modelling. At first, these techniques were applied to generate vegetation and textures. In time, hybrid methods have extended to 3D modelling. Various research and stand-alone programs have recently been developed to generate 3D settlements (Güzelci, 2019; Aydın, 2020; Lacroix, 2022). Cities can be generated using various procedural modelling techniques, including roadways, building lots, exteriors, and complete interiors. These models consider various factors, including population, environment, vegetation, architecture, elevation, geology, and culture (Berwaldt et al., 2020).

Among all of these studies, the procedural modelling examples listed below stood out during the research conducted to determine the initial design decisions for bBox that will define the main outlines of the project and the gap it would fill in the literature:

- *CityEngine* may start from zero and develop an urban environment based on a set of rules that can be expanded according to user requirements (Parish & Müller, 2001). The system employs a variety of image maps to generate a city. CityEngine uses two different types of L-Systems to produce streets and buildings. Users can create various designs by altering the L-Systems' rules.

- Simple and adaptable population templates are used to generate a virtual city in the *Template-based development of road networks for city modelling* (Sun et al., 2002). The system requires three inputs: an image map with geographic information (land/water/vegetation), an elevation map with height information, and a population density map of a location. It is only concerned with the creation of a city's road network.

- Users can utilize *The Cities* to create a terrain and environment where a group of builders can construct a city (Lechner et al., 2003, 2004). During a simulation, users can alter the building process and surroundings. Roads, parcels, and buildings make up the city. The user may draw parameters on the landscape and so have an indirect influence over the city-building process. The user may pause the simulation and add new parameters to the landscape.

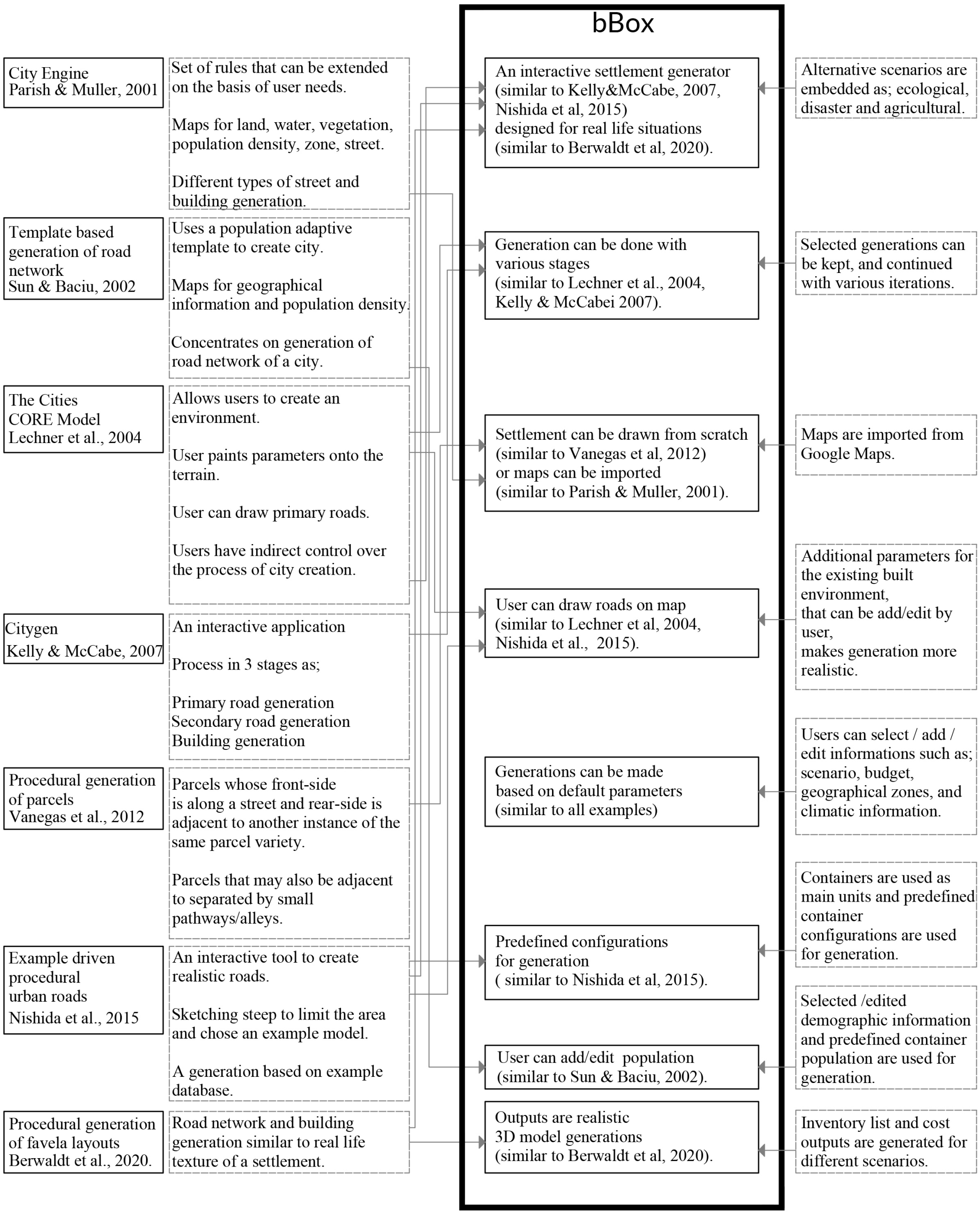


Figure 1: General structure and features of bBox
Source: Authors (2023)

- *CityGen* is an interactive program that provides a fully integrated workspace for city formation, dividing the process into three stages: primary road generation, secondary road generation, and building generation (Kelly & McCabe, 2006, 2007).

- Vanegas et al. presented a system for dividing construction lots within city blocks into two subdivisions: one that assures that all front sides are on the street and the other that splits the lots into quadrilateral forms with or without street access (2012).

- Nishida et al. showed an interactive tool that included a sketching step for limiting the region and selecting from a pre-designed model library (2016).

- Berwaldt et al. proposed utilizing the Unity Engine to create procedural favelas based on their unique road system, building lot partition, and lanes (2020).

The derived features of the abovementioned studies-such as scale, generation platform, generation method, and representation- are evaluated with additional features in bBox (Figure 1).

In the finished version, bBox became an interactive settlement generator (similar to Kelly & McCabe, 2007; Nishida et al., 2015) designed to respond to real-life scenarios (similar to Berwaldt et al., 2020). These real-life scenarios are embedded as; ecological, post-disaster, and agricultural. Generations can be done with various stages (similar to Lechner et al., 2004; Kelly & McCabe, 2007). Selected generations can be kept, and generations can be continued with various iterations. Settlement can be drawn from scratch (similar to Vanegas et al., 2012) or imported maps (similar to Parish & Muller, 2001). Users can draw roads on a map (similar to Lechner et al., 2004; Nishida et al., 2015). Also, additional parameters for the existing built environment can be added/edited by users. With this addition, the generations became more flexible and easily edited. Generations can be made based on default parameters (similar to all examples). Besides, users can select/add/edit information such as; scenarios, budget, geographical zones, and climatic data. Configurations are predefined (similar to Nishida et al., 2015), and population parameters can be added/edited (similar to Sun et al., 2002). Outputs are realistic 3D models (similar to Berwaldt et al., 2020). In addition, inventory lists and cost outputs are generated for all scenarios.

2.1. Scenarios and container configurations

bBox is a procedural settlement generator that creates, changes and evaluates different settlement alternatives for real-life scenarios using containers as the main unit (Authors, 2020). Since containers are structurally stable, standard, and modular, they are often used for different architectural purposes. Besides, recycled shipping containers are sustainable, easier to find if needed, and cost-effective.

Two types of containers (20 and 40 feet) are selected to predefine container configurations of bBox. Architects evaluate several design decisions based on the units, and their calculations are embedded in the study. These container configurations are defined with relations that will affect the number of households, functions of the spaces, and cost.

There are three scenario options predefined; ecological, post-disaster, and agriculture:

- In the post-disaster scenario, the main objective is to provide as much living space as possible, given a specific land area. Therefore, the algorithm prioritizes placing the shipping containers as close to each other as possible while maintaining a safe and functional living space.

- In the agriculture scenario, the focus is on creating enough farming areas around each living unit. The algorithm places the living units at the centre of farming sites, leaving enough space for farming around them to achieve this.

- The ecological scenario aims to minimize the carbon footprint of the settlement. Therefore, while the container units are already sustainable building materials, the algorithm makes design decisions prioritizing sustainability, including minimizing waste and pollution, using renewable energy sources, and incorporating green spaces. Regarding the placement of the shipping containers, the ecological scenario is less dense than the disaster scenario, with more space left between each unit. However, all options within the ecological scenario are designed to be environmentally friendly, so there is no new ecological approach in this mode beyond the already eco-friendly use of shipping containers. These features directly affect the population and budget for the overall generation.

On the other hand, population and budget parameters can be defined as additional parameters. If population and budget are added as input, bBox prioritizes these inputs during the evaluation and error control phase. If the region or the city is selected from climate and geography parameters, demographic data is automatically derived from the database showing the differentiation of users, such as family structure and adult/youth ratio. Population data can be modified based on the user’s preferences. Climate (data gathered from MGM, 2020) and geography parameters also define the configurations and relations of predefined containers.

According to the scenario, population, and climate parameters, the most suitable selections are made among predefined configurations seen in Figure 2. 40 ft. and 20 ft. container types are aligned in various positions to create configurations with open or semi-open areas. The number of containers is limited to three, but additional configurations can be designed and added to bBox. On the ground floor, there can be one (as in group

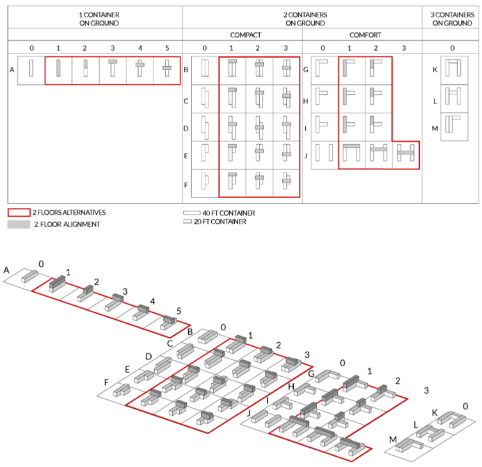


Figure 2: Parameters on the bBox interface
Source: Authors (2023)

A0), two (as in 0 groups B to J), or three containers (as in groups K0 to M0) placed.

If parameters (such as population and demographics) require more complex forms with more extensive areas, alternatives with two floors can be selected (as in groups 1 to 5 of A to J). According to the region, selections are predefined in percentages suitable to the climate data. In general, generations tend to choose compact forms for colder climates, and for hotter climates, generations are made with comfort configurations supported by semi-open areas.

2.2. Interface and inputs

Unity 3D game engine is used to develop bBox, because of its strengths in modelling and representation. The bBox graphical user interface consists of four panels: Toolbar, Screen Inputs, Parameters, and Outputs (Authors, 2020). The user experience is mainly based on interacting with this graphical interface by moving from top to bottom and left to right (Figure 3). There are various parameters in bBox, such as road, infrastructure, geographical and climatic data, contextual data, population data, number of users, and budget to create different settlement alternatives and data related to these alternatives are presented to the users. The bBox user interface allows users to open new projects, load existing projects, or get information about the software. The draft land is created after selecting the study area dimensions, unit grid size (1.2 m or 2.4 m), and interface language (Turkish or English). The grid size unit presets was determined based on the dimensions of containers and roads, and grid dimensions come from the smallest standard floor of the container and average road width dimensions. Using 1.2-meter grid units allows users to create more detailed defined environments, while 2.4-meter units are more suitable for large terrains.

Users can import a map by clicking the Map button and entering the coordinates copied from Google Maps. The main aim is to utilize accurate maps at the beginning of the generation. The selected map can be used as a guide by projecting it as a base on the field (Figure 4a). Since the slope will change and add various rules and constraints, areas with 0-5% slope are selected.

Land details in 3D space can be entered using the tools under the categories in the screen inputs panel. Edit site inputs that help data evaluate buildable areas are hardscape, tree (movable-fixed), pollution, obstacle, electric pole, infrastructure, and water. Road inputs can be defined with various values: primary road, secondary road, tertiary road, and pedestrian road (Figure 4b). These definitions are helpful, especially in the cost estimations, since they are accurate, and it is expensive to change the existing features.

Set North input determines the sun's direction, and container layouts are optimized according to the direction

Figure 3: Interface of bBox
Source: Authors (2023)

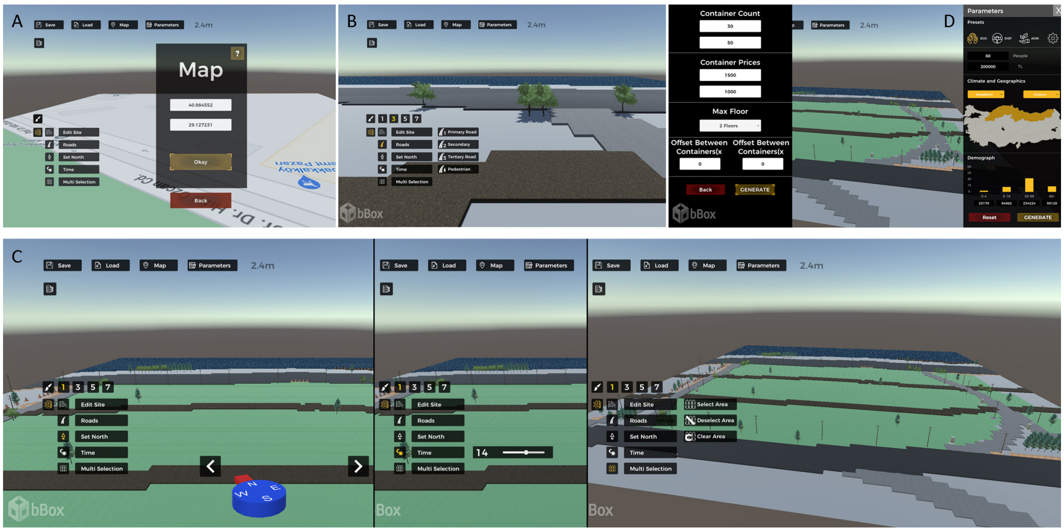


Figure 4: (a) Importing Google Maps with coordinates in bBox, (b) Road inputs, (c) Set north, time, and area selections, (d) Parameters on the bBox interface
Source: Authors (2023)

and angle of sunlight. Time of Day input shows day or night and adds shadows to make more realistic models and evaluate the sun effect. Multiple Selection input determines the buildable area, allowing one to choose, deselect and clear the desired area (Figure 4c).

After the buildable area definition, users can select, add, or edit additional parameters from the Parameters panel. First, Presets (Scenarios) are selected from predefined real-life scenarios: ecological, post-disaster, and agriculture. The Population parameter is optional to define the number of people in a settlement. Another optional parameter is budget, which allows users to prioritize budgeting to accommodate the entire population during generation. From Climate and Geography parameters, users can select the region and city where the settlement will be built in Turkey. Demographic parameters are collected from TUIK data automatically according to the selected region to calculate the number of family members (adult/youth ratio). Additional Parameters panel contains parameters for containers, such as; the number of containers and the cost of containers (Figure 4d).

Generation can be done when all parameters are entered or accepted. A printout screen shows the capacity, container usage, and cost in the Outputs panel. Container placement and output screen change with each generation, and the design can be manipulated manually if needed.

2.3. The generation process

The general process of bBox consists of several steps. First, the user sets various inputs, or default values are used. After the values are set, a general evaluation and error control are made. Generation operations are done using predefined procedural models. Finally, 3D models and calculations are produced as outputs (Figure 5).

In the first phase of the generation, the user defines and models the area for the settlement. Since one of the aims of bBox is to create realistic architectural solutions for real-life scenarios, accurate maps can be used as a base for generating abstract or drawn maps. As mentioned above, maps from Google Maps can be imported using coordinates. These maps are imported as scaled images on which the site information can be modelled to define inputs for the site. In this step, hardscape, tree (movable-fixed), pollution, obstacles, electric poles, infrastructure, and water can be selected and modelled using the embedded library. These parameters can be edited after their first implementation.

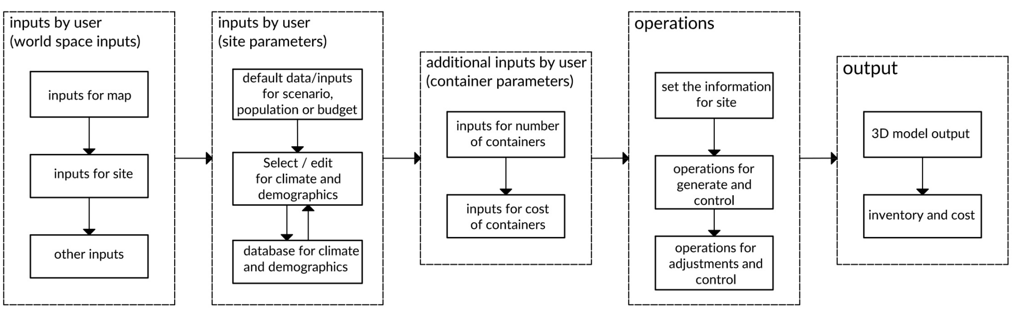


Figure 5: General flowchart of bBox
Source: Authors (2023)

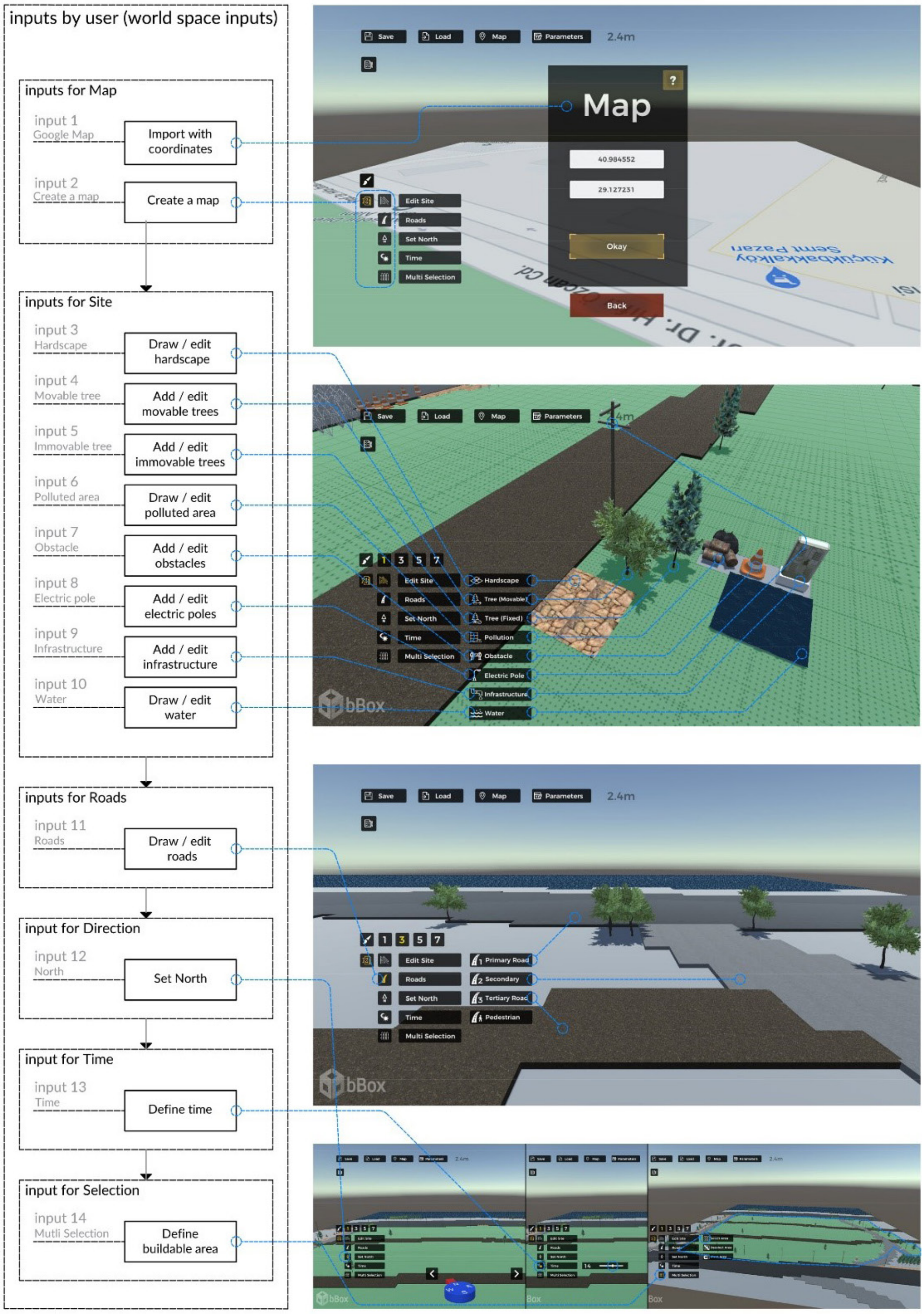


Figure 6: First inputs (3D modelling)
Source: Authors (2023)

The user can model the site' with the actual data for the parameters such as infrastructure and fixed obstacles. Also, users can define the parameters according to their design decisions for the abstract or revised versions. Roads are differentiated and used as a custom set of parameters since the settlements are generated concerning different types of roads based on the scenarios. Finally, north and time can be assigned, or default values can be used (Figure 6).

In the second phase of the inputs, parameters from panels can be assigned. bBox is capable of generating alternatives with the default selections that are embedded in this panel. However, these parameters can be modified and altered to create more realistic solutions according to various possibilities (Figure 7a).

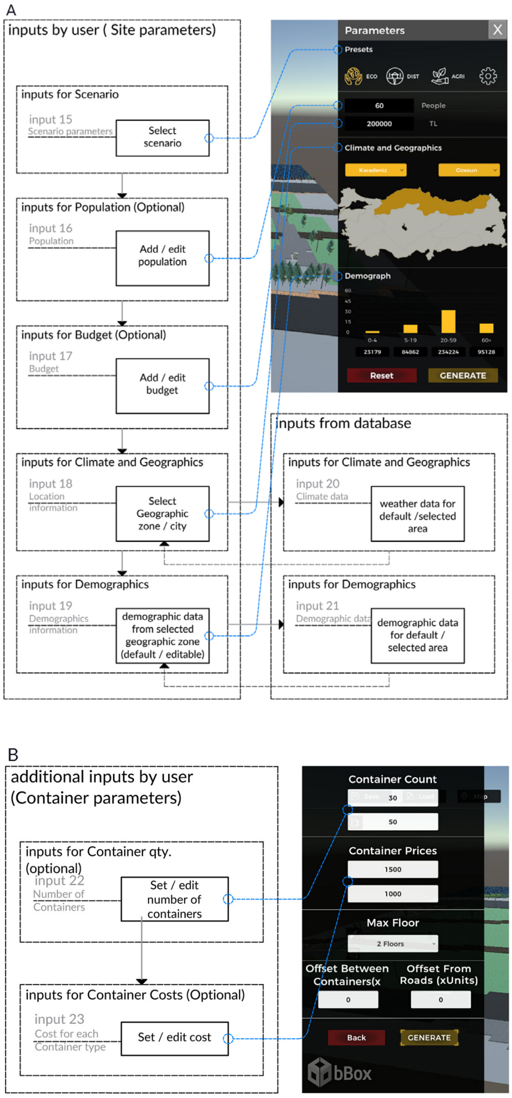


Figure 7: (a) Parameters panel, (b) Additional parameters
Source: Authors (2023)

There are additional parameters for container quantity. If the user has a specific number and type of containers in a generation, the amount can be defined to check whether that amount is enough for the defined population or scenario. Also, it is possible to define the cost for each container if needed. In this case, during the calculation phase, the tool takes this data as a budget input while generating the output for the cost (Figure 7-b).

After the inputs are set, or default parameters are accepted, generation can be initialized by pressing the generate button. In this phase, a series of operations are executed in the background for control, calculations, and production. The first operations are setting and controlling the buildable area and finalizing selecting the first site to place the container units. The container selection and area control processes start with the selection of the site and container units. Container units are randomly selected from the library, and another control took place for the placement. If the area for the site is suitable for the selected container, generation continues. If the area is unsuitable, other units are selected randomly (five times) and controlled for the placement fit. After five trials, a new site is selected to start the container selection and area control processes. After fifty trials, if the area is still unsuitable, the process stops with an error message stating the inadequacy of the site area. When the first container units are placed on the site, one of the adjacent sites is selected, and the generation continues until the units are placed on all the available buildable adjustments and final modifications are made according to the scenarios (Figure 8).

The final phase of the generation is calculating and generating 3D models. Population, various container types, and the final cost are calculated and printed as the final numeric values, while containers and adjustments are placed as 3D models (Figure 9).

Additional changes can be made to the model and parameters during generation. Since the process has a gradually expanding structure, these manipulations become new parameters for the generations. All or some parts of the generations can be kept based on user

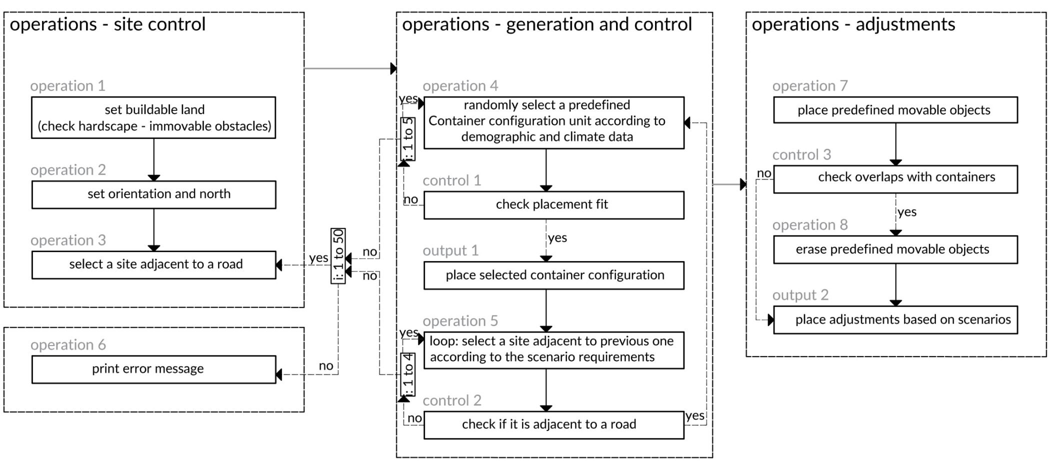


Figure 8: Operations and controls
Source: Authors (2023)

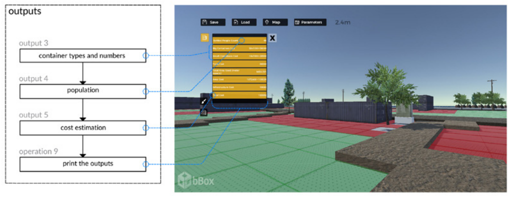


Figure 9: Outputs
Source: Authors (2023)

preferences. The areas that are selected to be generated again are emphasized with the red planes. New and additional generations can be done as it is described above (Figure 10).

3. Case Study

The case study of bBox is presented in two regions (Marmara and Mediterranean regions of Türkiye) with ecological and post-disaster settlement presets in both regions to illustrate the tool's capabilities and potential. Marmara and Mediterranean regions were selected due to their different climatic and demographic characteristics.

Marmara region is located in the northwestern part of Turkey and has a population of 24.9 million (TÜİK, 2020). Marmara has a humid continental climate with an oceanic climate on the northern coast, with an average of 12.88-14.69°C throughout the year (MGM, 2020).

The Mediterranean region is located in southern Turkey and has 10.6 million people (TÜİK, 2020). The region has a warm Mediterranean climate, with warm, dry summers and mild, rainy winters. The Mediterranean region has temperatures between 16.52-20.14°C yearly (MGM, 2020).

Ecological and post-disaster scenarios are selected to illustrate the prioritization differences in the settlement generation algorithm, showing its capability to produce alternative solutions for different scenarios and providing options to generate settlements meeting the contextual requirements of the local population and decision-makers.

The settlements were generated on the same site with identical land features (i.e., roads, plots, and environmental elements) but in different regions. Eighty settlement generations were made, corresponding to forty generations for every varying factor and ten generations for each specific scenario (preset, region, and plot type) in both regions. Table 1 shows the screenshots of all generations, classified by specific scenarios (e.g., ecological generations in Marmara on one-piece plots and post-disaster generations in the Mediterranean on quartered plots). The descriptive results generated from the case study are shown in Table 2. The table gives the mean values of population, number of containers, total area, area per capita, and cost for each scenario.

The statistics indicate that changing the scenario preset from ecological to post-disaster in the same region noticeably increases the total population on the same site.

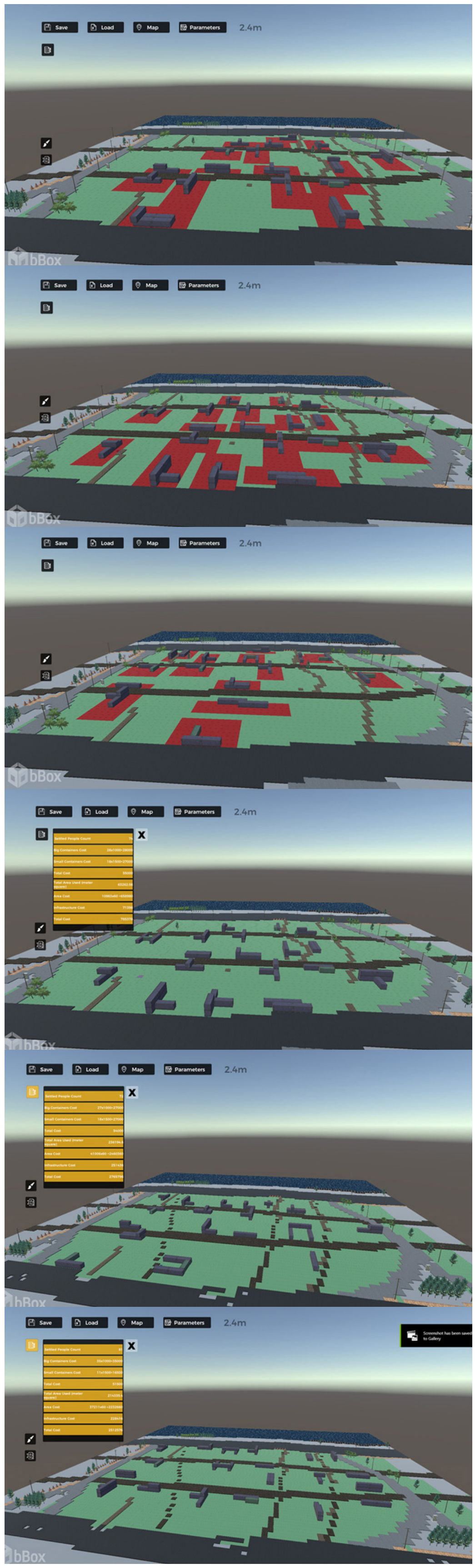


Figure 10: Area selection and generation
Source: Authors (2023)

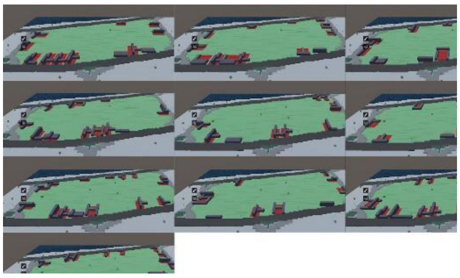
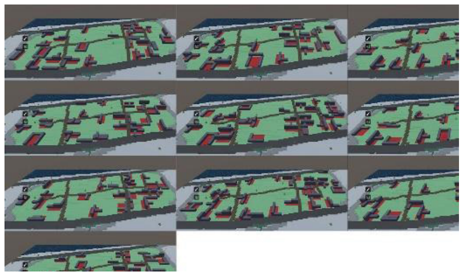
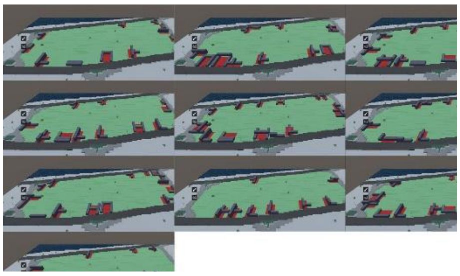
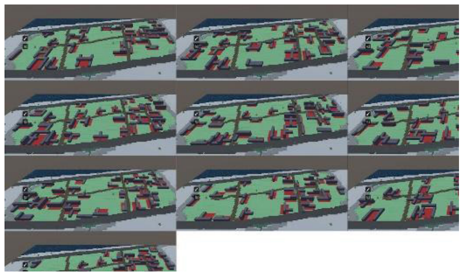
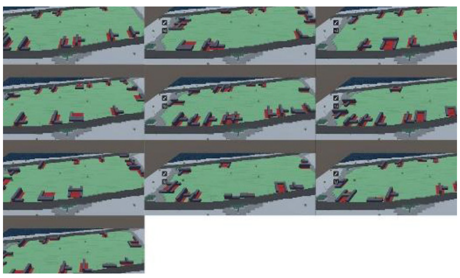
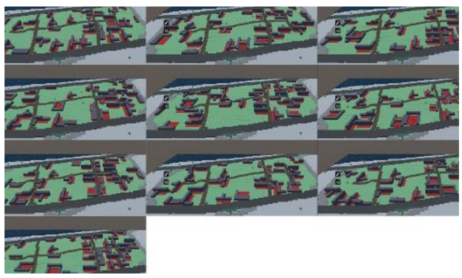
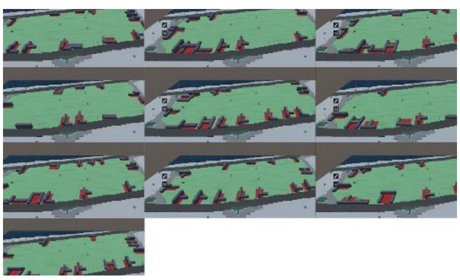
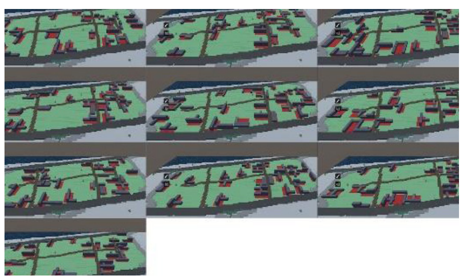
Region	Ecological Preset	
	One-piece plot	Quartered plots
Marmara		
Mediterranean		
Region	**Post-Disaster Preset**	
	One-piece plot	Quartered plots
Marmara		
Mediterranean		

Table 1: Screenshots of all settlement generation outputs for each specific scenario
Source: Authors (2023)

While the ecological preset prioritizes decreasing the carbon footprint and increasing the green areas, the post-disaster preset prioritizes accommodating the maximum population on the site. Changing only the region while keeping the other parameters constant causes minor differences in the statistics of ecological presets. However, changing the region with the same parameters does not show a notable difference in the statistics of post-disaster presets. Nevertheless, the placement of the container patterns changes noticeably between the regions, which can be attributed to the different climate and geography characteristics of the reasons. Again, the settlement generation algorithm prioritizes the most crucial factor in these scenarios: the maximum population capacity.

The generation results show that the increasing plot division of land with tertiary and pedestrian roads also increases the number of containers placed and the population accommodated. This change is expected because the unit placement algorithm tries to place the containers near the road, thus accessibility for pedestrians.

Region	Parameters (mean)	Ecological Preset		Post-Disaster Preset	
		One-piece plot	Quartered plots	One-piece plot	Quartered plots
Marmara	Population	55.4	107.8	58.4	139.2
	40' Container qty.	19	39.2	21	51.5
	20' Container qty.	14.5	29.4	16.4	36.2
	Area (sqm)	740	1.520	823	1.963
	Container area per capita (sqm)	13.36	14.10	14.09	14.10
	Cost (US Dollars $)	$ 48,920	$ 51,031	$ 48,686	$ 54,091
Mediterranean	Population	50.3	111	58.1	136.3
	40' Container qty.	18	42.4	21	49.6
	20' Container qty.	14.3	30.2	16.4	37.1
	Area (sqm)	709	1.621	823	1.921
	Container area per capita (sqm)	14.09	14.61	14.09	14.10
	Cost (US Dollars $)	$ 49,241	$ 55,188	$ 50,068	$ 54,768

Table 2: Descriptive results of the settlement generations for each specific scenario
Source: Authors (2023)

Another important observation is that no matter how much the population increases (decreasing the land area per user), the container area per capita stays almost constant in different scenarios. When the population increases, bBox also increases the number of containers to accommodate them.

The case study results show that the tool can generate upcycled shipping container settlements for different real-life scenarios in a short period. The procedurally generated settlements can accommodate a varying number of people depending on the context of the specific scenario. The tool is easy to use and provides the flexibility to change the parameters to meet users' specific needs.

4. Discussion and conclusions

In recent years, the use of digital tools in architectural design has grown in popularity. These tools have several advantages, such as quick calculations, ease of use, and high-quality visualization. Because they incorporate various variables and relationships, complex problems like settlement design lend themselves particularly well to digital tools. Using procedural generation, these tools can develop realistic solutions rapidly that would be difficult or impossible to generate manually. bBox is one such tool, which provides quick 3D modeling and estimating outputs using procedural generations.

Settlement design is a multifaceted subject with far-reaching social and environmental ramifications. Millions of people globally require homes, and it is more important than ever to design settlements on various scales quickly and efficiently. The capacity of bBox to develop realistic answers to real-world difficulties in settlement design could be extremely useful in meeting these demands with a wide range of potential uses. For these uses, quick calculations and estimations will be beneficial.

The ease of use of bBox is equally crucial. While these tools can be difficult and need specialized knowledge, they must be accessible to a wide range of users, including architects, urban planners, policymakers, community members, and other stakeholders involved in settlement planning. bBox's user interface and parameter structure are simple and intuitive, allowing for greater flexibility and investigation of alternate scenarios.

Incorporating different data and scenarios into bBox is a promising promise for settlement design. Several solutions can be developed by combining data from multiple cities or nations, modifying the existing rules, and changing the land options. This means that the tool can be used in various scenarios to generate solutions tailored to local conditions and demands. Furthermore, the capacity to incorporate other characteristics related to settlement production possibilities and sustainability might transform bBox into a powerful instrument for addressing more significant social and environmental challenges.

bBox's potential applications extend beyond settlement design and urban planning. Other applications for the tool could include disaster assistance, refugee settlements, and temporary housing options. bBox could be beneficial in solving many social and environmental concerns by enabling for speedy and efficient design solutions that reflect local demands and conditions.

The requirement for predetermined container configurations is one potential constraint of bBox. This provides for more efficient design solutions, but it also limits the tool's adaptability. Configurations, on the other hand, can be changed, updated, or new configurations can be introduced in future studies. Furthermore, using bBox should be complemented by a number of other design considerations.
Another limitation can be defined as a lack of creativity for over-reliance on technology. However, as long as designers are aware of these restrictions and employ digital tools on

purpose, the benefits might outweigh the risks. In the case of bBox, its procedural generation approach ensures that each design is unique and tailored to the demands of the community.

Overall, bBox is an innovative digital tool that utilizes procedural architectural generations to produce realistic solutions for settlement design problems. Its easy-to-use interface, real-time updates, and capacity to create solution alternatives for real-world challenges make it a powerful tool for architects and urban planners. The tool's adaptability and capacity to accommodate various data and scenarios make it a promising candidate for future applications in settlement planning.

Cómo citar este artículo/How to cite this article:

Şen-Bayram, A. K., Torus, B., Doma, O. O. & Şener, S. M. (2023). Procedural architectural settlement generator for container housing: A study on Marmara and Mediterranean Regions. *Estoa. Revista de la Facultad de Arquitectura y Urbanismo de la Universidad de Cuenca, 12*(24), 161-173. https://doi.org/10.18537/est.v012.n024.a13

5. References

Akgül, C. B., Sönmez, O. N., & Alaçam, S. (2017, May). Understanding a city from its visuals: An interdisciplinary program proposal. In *2017 25th Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). IEEE.

Alaçam, S., Karadag, I., & Güzelci, O. Z. (2022). Reciprocal style and information transfer between historical Istanbul Pervititch Maps and satellite views using machine learning. *Estoa. Revista de la Facultad de Arquitectura y Urbanismo de la Universidad de Cuenca, 11*(22), 97-113.

Aydin, S., Schnabel, M. A., & Sayah, I. (2017). Association Rule Mining to Assess User-Generated Content in Digital Heritage: Participatory Content Making in 'The Museum of Gamers'. In *Computer-Aided Architectural Design. Future Trajectories: 17th International Conference, CAAD Futures 2017, Istanbul, Turkey, July 12-14, 2017, Selected Papers 17* (pp. 231-251). Springer Singapore. https://doi.org/10.1007/978-981-10-5197-5_13

Aydin, S., & ŞIK, B.(2020). An Optimisation Study of Spatial Ad-Hoc Encounters with Quad-Tree Algorithms in the Context of Gldani Microrayons. In *IDU SPAD'20 International Spatial Planning and Design Symposium* (p. 208). Izmir Democracy University.

Berwaldt, N. L. P., Bettker, R. V., & Pozzer, C. T. (2020, November). Procedural generation of favela layouts on arbitrary terrains. In *2020 19th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)* (pp. 136-144). IEEE. https://doi.org/10.1109/SBGames51465.2020.00027

Davis, M., Sigal, R., & Weyuker, E. J. (1994). *Computability, complexity, and languages: fundamentals of theoretical computer science*. Elsevier. https://doi.org/10.5860/CHOICE.32-0356

Ehrig, H., Rozenberg, G., & rg Kreowski, H. J. (1999). *Handbook of graph grammars and computing by graph transformation* (Vol. 3). world Scientific. https://doi.org/10.1142/4181

Gürbüz, E., Çağdaş, G., & Alaçam, S. (2010, September). A generative design model for Gaziantep's traditional pattern. In *Proceedings of the 28th Conference on Education of Computer Aided Architectural Design in Europe* (pp. 841-849). ETH Zurich.

Güzelci, O. Z., & Şener, S. M. (2019). An Entropy-Based Design Evaluation Model for architectural competitions through multiple factors. *Entropy*, *21*(11), 1064.

Kelly, G., & McCabe, H. (2006). A survey of procedural techniques for city generation. *ITB Journal*, *14*(3), 342-351. https://doi.org/10.21427/D76M9P

Kelly, G., & McCabe, H. (2007, November). Citygen: An interactive system for procedural city generation. In *Fifth International Conference on Game Design and Technology* (pp. 8-16). http://www.citygen.net/files/citygen_gdtw07.pdf

Lacroix, I., Güzelci, O. Z., Lopes, G. F., & Sousa, J. P. (2022). Connecting the Portuguese system of evolutive housing with building information modeling: From analogical to digital methods. *International Journal of Architectural Computing*, *20*(4), 801-816.

Lechner, T., Ren, P., Watson, B., Brozefski, C., & Wilenski, U. (2006). Procedural modeling of urban land use. In ACM SIGGRAPH 2006 Research posters (pp. 135-es). http://www.cs.northwestern.edu/publications/techreports/2004_TR/NWU-CS-04-38.pdf

Lechner, T., Watson, B. A., Wilensky, U., & Felsen, M. (2003). Proceduring city modeling. In *1st Midwestern Graphics Conference*, St. Louis, MO, USA.

Meteoroloji Genel Müdürlüğü. (2020). Türkiye Ortalama Sıcaklık Değerleri. In *Resmi İstatistikler*.

Nishida, G., Garcia-Dorado, I., & Aliaga, D. G. (2016). Example-Driven Procedural Urban Roads. *Computer Graphics Forum*, *35*(6), 5–17. https://doi.org/10.1111/cgf.12728

Parish, Y. I. H., & Müller, P. (2001). Procedural modeling of cities. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '01*, *August*, 301–308. https://doi.org/10.1145/383259.383292

Prusinkiewicz, P., & Lindenmayer, A. (2012). *The algorithmic beauty of plants*. Springer Science & Business Media.

Sipser, M. (1996). Introduction to the Theory of Computation. *ACM SIGACT News*, *27*(1), 27–29. https://doi.org/10.1145/230514.571645

Stiny, G. (1975). Pictorial and Formal Aspects of Shape and Shape Grammars. In *Pictorial and Formal Aspects of Shape and Shape Grammars*. Birkhäuser Basel. https://doi.org/10.1007/978-3-0348-6879-2

Sun, J., Yu, X., Baciu, G., & Green, M. (2002, November). Template-based generation of road networks for virtual city modeling. In *Proceedings of the ACM symposium on Virtual reality software and technology* (pp. 33-40). https://doi.org/10.1145/585746.585747

Torus, B., Şen Bayram, A., Doma, O., & Şener, S. (2020). bBox: A Framework for Container Settlements. In N. Narlı, B. Torus, & N. Aydın Yönet (Eds.), *The Paradigmatic City (IV): Transforming Cities Selected Papers* (pp. 15–24). Mentora.

TÜİK. (2020). *Türkiye İstatistik Kurumu - Coğrafi İstatistik Portalı*. https://cip.tuik.gov.tr/

UNHCR. (2020, December 8). *UNHCR - Refugee Statistics*. https://www.unhcr.org/refugee-statistics/

UNHCR. (2021). *UNHCR - Refugee Statistics*. Refugee Data Finder. https://www.unhcr.org/refugee-statistics/

Vanegas, C. A., Kelly, T., Weber, B., Halatsch, J., Aliaga, D. G., & Müller, P. (2012). Procedural Generation of Parcels in Urban Modeling. *Computer Graphics Forum*, *31*(2pt3), 681–690. https://doi.org/10.1111/j.1467-8659.2012.03047.x

Wonka, P., Wimmer, M., Sillion, F., & Ribarsky, W. (2003). Instant architecture. *ACM Transactions on Graphics*, *22*(3), 669–677. https://doi.org/10.1145/882262.882324