Adaptive Modular Housing Design for Crisis Situations

Anastasiya Pechko[1][0009−0000−8464−3385], Katarzyna Grzesiak-Kopeć[1][0000−0001−5736−7661], Barbara Strug[1][0000−0002−2204−507X], and Grażyna Ślusarczyk[1][0000−0003−1032−1644]

Institute of Applied Computer Science, Jagiellonian University,
Łojasiewicza 11, 30-059 Kraków, Poland
anastasiya.pechko@doctoral.uj.edu.pl, katarzyna.grzesiak-kopec@uj.edu.pl,
barbara.strug@uj.edu.pl, grazyna.slusarczyk@uj.edu.pl

Abstract. This paper presents an innovative approach to using computational methods in designing modular housing estates with the Wave Function Collapse (WFC) algorithm. Currently, as a result of fast changing humanitarian situations in various parts of the world, like the one caused by the war in Ukraine or climate changes, many people have to leave their homes. It results in complex problem of providing large groups of people with conditions that will allow them to function with dignity for a long time. Therefore the adaptation of the WFC algorithm to design modular settlements is proposed. The applied heuristics allow the solutions to adapt to specific project requirements, generating various modular settlement designs that consider functionality and social aspects. The proposed approach is illustrated by examples of generated arrangements of housing modules for family-type floor plans.

Keywords: modular design · heuristics · WFC algorithm · constraint programming.

1 Introduction

Nowadays the growing number of natural and human-made crisis situations requires immediate and effective solutions to rebuild and improve the living conditions of people affected by conflicts or other emergencies. Modular construction is increasingly recognized as an effective, fast, and ecological method for building homes and offers a promising approach to address these needs.

The current armed conflict on the territory of Ukraine has led to significant damage to urban infrastructure, a housing crisis, and a massive refugee migration. Ukrainian architects from the Balbek studio have designed a modular system of rooms with an area of $21m^2$, accommodating living rooms, kitchens, bathrooms, and meeting rooms. These modules can be used to create both small settlements for hundreds of people and larger towns for thousands [1]. This innovative solution forms the basis for the proposed tool that should support architects in designing process. The created WFC-Ukraine Housing software uses

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_32

the Wave Function Collapse (WFC) algorithm [2] to generate solutions in accordance with the above-mentioned project. The WFC algorithm has been modified and enhanced with specific heuristics to meet the design requirements of functionality, aesthetics, and social factors.

2 Related work

Recently great deal of research has been conducted in the field of computer-aided architectural design particularly in the automatic generation of room arrangements for floor plans [9, 11]. In most existing approaches, the graph-based representation of floor layouts, where nodes represent specific modules, is used [10, 17]. Many techniques also rely on artificial intelligence and machine learning methods [14, 15]. However, most of these methods are not suitable for complex layouts design, because of their high time complexity and a large amount of data needed to attain good accuracy.

Generating plans of modular settlements can be considered as a problem of the procedural content generation in architectural and urban design. Procedural content generation techniques, like Cellular Automata, enable efficient content generation, automating the process and reducing the need to manually create individual elements [3]. Another well-known approach is based on shape grammars [12]. In [13] an agent system combined with shape grammars was used to support floor layout designs. However, this approach does not give much possibility for designer interaction.

The WFC algorithm can also be used to design architectural projects. It is derived from an algorithm originally known as Model Synthesis [18]. However, its limitations include considering only local dependencies between grid tiles, poor scalability, and a lack of parameterization. As a result, many modifications of the WFC algorithm have been proposed, enabling its effective use as a tool for procedural content generation [8].

3 Generating modular housing estates using a modified WFC algorithm

The modular housing system used as the basis of the research was proposed by architects from the Ukrainian studio Balbek. The proposed system allows for quick organization of space while minimizing costs and ensuring comfortable conditions. The design concept emphasizes the importance of functionality and social factors. Four types of modules (residential, public, kitchen and sanitary) were proposed as the basic construction elements of the estate, from which the final solution is assembled and six different variants of module layouts (Fig. 1). The generated layouts of housing estates are easily modifiable, which allows them to be quickly adapted to various environmental conditions.

The WFC algorithm operates on a grid of basic units, called cells, sequentially determining cell values and propagating changes based on specified rules and

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_32

Fig. 1. The RE:Ukraine Housing project: the family arrangement type and 1-3 identified types of standard layouts.

constraints. The algorithm fills the grid cells with the specified tiles. Initially, each cell in the grid can potentially be any tile from the tile set, representing a superposition of all possible states. The entropy measures the uncertainty of a state of a cell (i, j) and is the number of possible tiles for this cell. The cell in the grid is collapsed when its state has been determined, reducing its possible values to a single tile. The neighborhood of a tile (i, j) defines its local surroundings that may affect the evolution of a grid. Adjacency rules define how tiles can be placed next to each other. The adjacency list for a tile specifies the allowed neighboring tiles in the grid and their possible position.

The WFC algorithm generates housing layouts in the following steps:

1. Initialization:
   - The tile set is established and augmented with rotated and mirrored variants to increase diversity.
   - The adjacency rules are defined based on tile compatibility.
   - The grid size and initial cell states are defined.
2. Generation:
   - **Observation**: identify the cell with the lowest entropy, collapse the cell to a single tile based on heuristic selection methods; if the entropy value is equal to zero a contradiction has been reached and there is no information about available moves; if the maximum number of iterations is reached, the algorithm will not be able to find a solution. This may mean that the constraints are poorly defined or there are too many of them.
   - **Propagation**: update the neighboring cells' possible tile sets based on the collapsed cell, ensuring all adjacency rules are maintained.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_32

– **Iteration**: continue the observation and propagation steps until all cells are collapsed, resulting in a complete and coherent layout.
– **Render Observations**: generate the output image.

The modifications and enhancements introduced to WFC algorithm to effectively apply it to the design of modular housing estates are as follows: (1) heuristic functions, (2) enhanced tile encoding, (3) establishing excluded areas, (4) adjacency rules and constraints and (5) customizable parameters. These enhancements ensure that the algorithm not only generates functional and aesthetically pleasing layouts but also adheres to the specific requirements of modular construction and the unique needs of housing projects for displaced people.

Several *heuristic functions* were introduced, namely the bath placement, the symmetry patterns, the door and window placement and the residential standard one. They incorporate domain-specific knowledge and design principles to influence the selection and placement of tiles. The elaborated *enhanced tile encoding* includes additional metadata to provide more detailed information about each tile, such as: functional attributes (space type), structural attributes (doors and windows) and service attributes. Thanks to the use of *establishing excluded areas* enhancement, in which designated paths or areas remain free from any building modules, the algorithm gains greater flexibility and adaptability to various user requirements, existing infrastructure and local urban conditions.

The *adjacency rules* have been extended to incorporate complex relationships between tiles, which respects functional and aesthetic criteria. For instance, kitchens should be adjacent to dining areas, and bathrooms should not open directly into living rooms. *Constraints* related to the structural integrity and safety of the modular units have been integrated, ensuring that generated layouts are compliant with building codes and standards. *Customizable parameters* specify the density of modules needed to accommodate different population sizes, and set priorities for different functional areas (e.g., more space for communal areas or more bedrooms) based on the specific needs of the project.

4 Modular Housing Generation System Implementation

The implementation of a modular housing generation system was carried out using Visual Studio Code. Key libraries and languages used in the project include JavaScript, p5.js and jQuery.

The Re:Ukraine Housing modular housing system assumes four types of modules: residential, public, kitchen, and sanitary marked on the design with different colors (Fig. 1) [1]. It also includes six different variants of module layouts and the first of them (Type I a family-type section) was taken as the starting point in our project. Based on the different possible arrangement of modules, three types of standard layouts were distinguished:

1. Standard 1: includes one double bedroom, a kitchen and a bathroom.
2. Standard 2: includes one double bedroom, one triple bedroom, a kitchen and a bathroom.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_32

3. Standard 3: includes two double bedrooms, a kitchen and a bathroom.

Bitmaps representing the appropriate standards were divided into regular tiles of square shape (Fig. 2) and subsequently, the set of tiles was extended by using rotation and reflection transformations. Each tile represents a specific functional or structural component of the housing units, such as living spaces, kitchens, and bathrooms and is encoded with metadata that provides detailed information about its attributes and relationships with other tiles. This metadata includes:

1. Functional attributes: descriptions of the tile's purpose and usage, ensuring that the generated layout meets the functional requirements.
2. Structural attributes: information about walls, doors, windows, and other structural elements, ensuring the structural integrity of the housing units.
3. Service attributes: indications of the presence of plumbing, electrical wiring, and other utilities, ensuring that the necessary infrastructure is in place.

Tile encoding should include information describing possible relations between them. Symbols of a set $C = \{h, f, m, g, v, p, l, o, e, c, s, q, t, n, j, u, r, z\}$ represent connections of tiles, while four special symbols represent an outer wall (a), door (x), window (w), and empty space (d). The primary relations defined between tiles are: connector relation: $\{\{i_C, i_C\} : i_C \in C\}$, window relation: $\{\{w, d\}\}$, door relation:$\{\{x, d\}\}$, outer wall relation:$\{\{a, d\}, \{a, a\}\}$ and empty relation:$\{\{d, i_S\} : i_S \in S\}$. Having relations between tiles defined explicitly, the algorithm can efficiently determine the compatibility of adjacent tiles. In Fig. 2 a set of tiles with the assigned codes for all three standard layouts is presented.

Fig. 2. Set of tiles with the assigned codes.

Heuristics are essential for guiding the WFC algorithm to produce coherent, functional, and aesthetically pleasing layouts. **Residential Standard Heuristic** ensures that the generated layouts meet the predefined residential standards and functional requirements. It involves assigning weights to tiles based on the desired number of occurrences of each standard. **Bath Placement Heuristic** ensures that bathrooms are placed logically within the housing layout. It considers grouping bathrooms with kitchens to optimize plumbing infrastructure, and ensures that bathrooms are placed away from communal areas. **Symmetry**

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_32

Patterns Heuristic is crucial for ensuring that the generated housing layouts maintain a visually pleasing and structurally balanced design. Balanced layouts can reduce the complexity of plumbing, electrical, and HVAC systems, as these systems can be designed more efficiently in symmetrical spaces. **Establishing Excluded Areas Heuristic** allows specific paths or regions within the layout to be designated as non-buildable zones. This is particularly useful for maintaining open spaces, pathways, or respecting existing urban infrastructure. The algorithm accepts user-defined input for excluded areas. **Accessibility Heuristic** ensures that the layout is free from illogical connections and maintains logical flow. For example, the algorithm can generate layouts of modular housing estates where doors lead to inaccessible or closed areas. In order to eliminate the above anomalies, an additional step was introduced into the algorithm, which analyzes the available doors and checks whether there is at least one coherent path connecting them.

5 Experimental results

The experiments were conducted on a system running macOS Ventura 13.2.1, equipped with 16 GB of RAM, a 12-core CPU, and a 19-core GPU, with a 512 GB SSD. Over 100 iterations were performed to ensure the stability and consistency of the results. Different grid sizes (e.g., 9x9, 10x10, 11x11, 12x12) were tested to evaluate the scalability of the algorithm. Various heuristic functions, including Symmetry V, Symmetry H, Bath Placement, and their combinations, were applied to assess their impact on the layout generation process. Four predefined initial configurations of excluded areas were specified to simulate real-world constraints and test the algorithm's flexibility and adaptability.

The **residential standard heuristic** was verified by defining two levels of comfort and generating layouts corresponding to the expected density of people in a given area. The **bath placement heuristic** reduced the chaotic nature of generated designs and minimized the distance between bathroom modules (Fig. 3.A). Two **symmetry patterns heuristic**, symmetry H and symmetry V, were tested to maintain visual appeal and structural balance. Obtained symmetrical designs were consistently rated higher in terms of aesthetic quality without compromising functionality (Fig. 3.B). The algorithm effectively handled excluded areas by the **excluded area heuristic** without compromising the functionality of the housing layouts. Also, this heuristics with the one taking into account appropriate symmetry can improve the quality of the generated modular estate layouts and increase the number of successful generations (Fig. 4).

The algorithm demonstrated efficiency in generating layouts across different grid sizes and heuristic applications. The average generation time increased with grid size, which was expected due to the larger search space and complexity. The conducted experiments systematically compared various configurations by modifying parametric variables. The results were analyzed both qualitatively and quantitatively, providing a comprehensive understanding of the impact of different variables on the efficiency of generating modular layouts. The modified

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_32

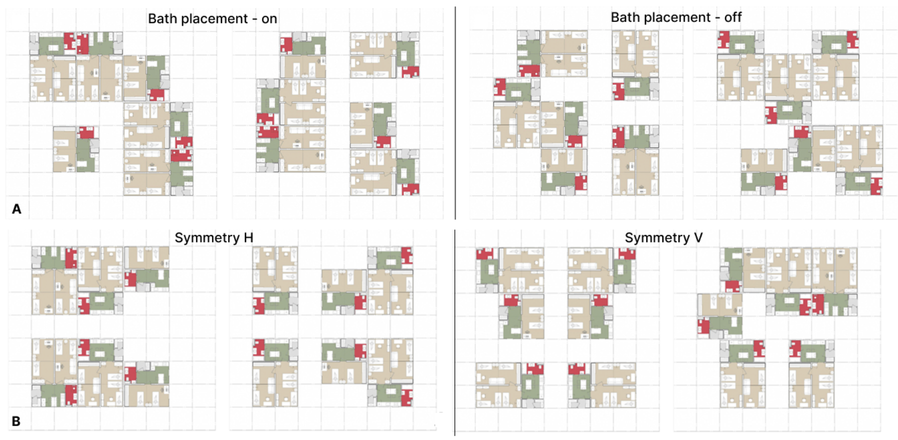

Fig. 3. The bath placement (A) and symmetry (B) heuristics.

Fig. 4. The excluded area heuristic along with the symmetry patterns heuristic.

WFC algorithm offers a high degree of *customizability*, allowing users to define specific standards, comfort levels, and environmental constraints.

6 Conclusion

This study explored the feasibility and effectiveness of using the WFC algorithm in designing modular housing estates. The implementation of the generation tool demonstrated the algorithm's ability to generate modular housing layouts that are functional, scalable, and aesthetically pleasing. The heuristic functions significantly enhanced the quality of the layouts, with the Bath Placement and Symmetry Patterns heuristics proving particularly effective. The algorithm maintained performance and quality across different grid sizes.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_32

Future research directions include exploring additional heuristic functions, integrating real-world data, further developing hierarchical approaches, and extending the algorithm's application to other domains such as urban planning and emergency housing.

References

1. RE:UKRAINE — balbek bureau. https://www.balbek.com/reukraine-eng (accessed Jul. 23, 2024).
2. Javadi, H. Wave function collapse and CPH Theory, Jan. 2021.
3. Temuçin, M. B., Kocabaş, I., Oğuz, K. Using cellular automata as a basis for procedural generation of organic cities, European Journal of Engineering and Technology Research, vol. 5, no. 12, pp. 116–120, 2020.
4. Parish, Y. I., Müller, P. Procedural modeling of cities, Proc. 28th annual conference on Computer graphics and interactive techniques, 2001, pp. 301–308.
5. Radies, C. Procedural random generation of building models based Geobasis data and of the urban development with the software CityEngine, Bernburg, Germany, vol. 175, p. 184, 2013.
6. Beirão, J., Duarte, J. Urban grammars: towards flexible urban design, in Proc. 23rd Int. eCAADe Conf, 2005, pp. 491–500.
7. Hansmeyer, M., Michael Hansmeyer - L-Systems in Architecture, https://www.michael-hansmeyer.com/l-systems (accessed Jul. 02, 2024).
8. Kleineberg, M. Infinite procedurally generated city with the Wave Function Collapse algorithm, https://marian42.de/ article/wfc/ (accessed Jun. 28, 2024).
9. Upasani, N., Shekhawat, K., Sachdeva, G. Automated generation of dimensioned rectangular floorplans, Autom. Constr. 2020, 113, 103-149.
10. Ślusarczyk, G. Graph-based representation of design properties in creating building floorplans. Comput. Des. 2018, 95, 24–39.
11. Strug, B., Grabska, E., Ślusarczyk, G. Supporting the design process with hypergraph genetic operators. ´ Adv. Eng. Inform. 2014, 28, 11–27.
12. Grzesiak-Kopeć, K., Ogorzałek, M.J. Intelligent 3D layout design with shape grammars. Proc. 6th International Conference on Human System Interactions (HSI), Sopot, Poland, 6–8 June 2013; pp. 265–270.
13. Grabska, E. Grzesiak-Kopeć, K., Ślusarczyk, G. Designing Floor-Layouts with the Assistance of Curious Agents. Proc. 6th International Conference on Computational Science (ICCS), Reading, UK, Springer, LNCS 3993, pp. 883–886, 2006
14. Wu, W., Fu, X.M., Tang, R., Wang, Y., Qi, Y.H., Liu, L. Data-driven interior plan generation for residential buildings. ACM Trans. Graph. 2019, 38, 1–12.
15. Hu, R., Huang, Z., Tang, Y., van Kaick, O., Zhang, H. Huang, H. Graph2Plan: Learning Floorplan Generation from Layout Graphs. ACM Trans. Graph. 2020, 39, 1–14.
16. Flemming, U., Coyone, R., Gavin, T., Rychter, M. A Generative Expert System for the Design of Building Layouts—Version 2; Topping, B., Ed.; Artificial Intelligence in Engineering Design; Computational Mechanics: UK, 1999; pp. 445–464.
17. Grzesiak-Kopeć, K., Strug, B., Ślusarczyk, G. Evolutionary Methods in House Floor Plan Design, Applied Sciences, 11, 2021.
18. Merrell, P. Model Synthesis. Ph.D. Dissertation, University of North Carolina at Chapel Hill, 2009.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_32