

Plan Layout Generator (PLG)

A Rule-Based Plan Layout Generator for Mardin Houses

Belinda Torus¹, Birgül Çolakoğlu²

¹Bahçeşehir University, Turkey, ²Yıldız Technical University, Turkey

¹belindatorus@gmail.com, ²bigi@alum.mit.edu

Abstract: *The work presented in this paper illustrates automation of specific architecture language rules. Medieval city of Mardin architecture is used as case for developing the program named “Plan Layout Generator – PLG” which generates new abstract layout alternatives of Mardin house aimed to be utilized in the preliminary design phase.*

Keywords: *Generative design; Mardin house typologies; rule-based design.*

Introduction

Mardin is a city with unique design language. The principles of this language have been a research interest of many scholars (Alioğlu, 2000). Design principles of Mardin house are formalized by utilizing the shape grammar method (Özbek, 2004). These design rules are used in the generation of new typologies coherent with the city's existing architectural language. (Özkaraduman, 2007)

The work presented in this paper utilizes these rules in the development of an automated program named “Plan Layout Generator – PLG”. The program generates abstract layout alternatives of Mardin House. It is developed in Java platform in collaboration with a programmer.

Related works

There have been many generative design tool applications developed to explore generative approaches in design processes such as, 3DShaper (Wang and Duarte, 2002), Shaper2D (McGill, 2002), Con-Gen (Çolakoğlu and Dionyan, 2005) and QShaper (Yazar and Çolakoğlu, 2007). Most of these applications were aimed to be used for abstract form generations. Some of them developed as independent systems such as 3D shaper and some as plug-ins under CAD softwares.

PLG is a generative application developed for specific use utilizing existing rules of Mardin architecture. Therefore it differs from above mentioned works that generate abstract form compositions from scratch. The program is built up on an

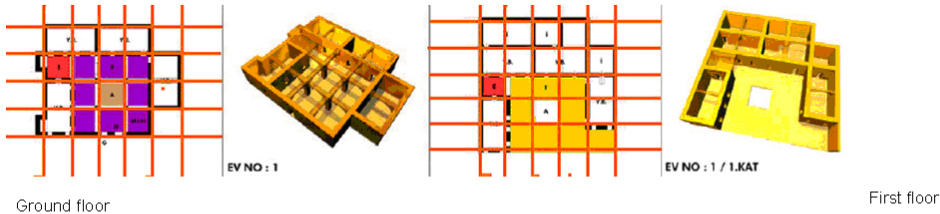
Figure 1a (left)
Medieval city of Mardin

Figure 1b (middle)
Topography of Mardin

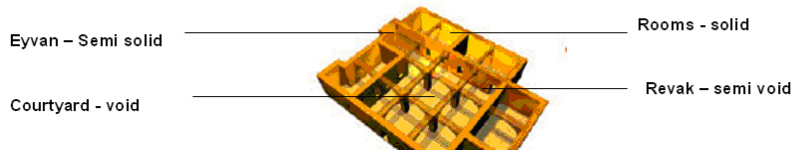
Figure 1c (right)
Typical summer house in
Mardin



Figure 2
Mardin House modular vocabulary (Özbek, 2004)



The vocabulary of Mardin house derives from this module.



independent platform (JAVA). It allows generated abstract plan layouts to be saved in 3dsMax format for further 3D explorations of a new house form.

Mardin

The medieval city Mardin located in south-eastern Anatolia known with its unique stone architecture is built through centuries. (Figure 1a, 1b, 1c). The stone houses of Mardin surround the narrow, steep and winding streets of the city. They are inward looking courtyard buildings developed on a 4x4 grid (Figure 2).

Mardin house is composed of open spaces in the form of courtyard and terraces, semi open spaces -eyvan- closed on three sides and the top- arcades -revak-, and closed places-rooms. Courtyard and terraces are private quarters, surrounded by rooms, eyvans and/or high walls. The vocabulary of Mardin house is modular. The smallest vocabulary is a room with the size of approximately 4m x 4m which derives from the structural system and can be expanded (Figure 2).

Rules of Mardin

The design principles of Mardin House are classified

in ten groups: 1- House expansion, 2- Living space; 3-Work-living space relation; 4- Eyvan; 5-Revak; 6- Balcony; 7- Terrace; 8-Stair; 9- Entrance; 10- First floor formation rules. (Çolakoğlu, 2007)

In this work, the basic rules that are used to develop PLG can be specified in three groups:

- Space relations: The open, closed and semi-open spaces have been placed according to their relationships.
- Orientation and terrace relations: All the houses are oriented toward south overlooking the valley and they do not close one another's scenery.
- Functional relations: The public spaces are on the ground floor while the private spaces are on the upper floor(s).

The modular vocabulary of Mardin House consists of three types of spaces: open, semi-open and closed spaces as illustrated in figure 2. There are three types of semi-open spaces (eyvan, revak, kosk) named according to their relations with other spaces. Semi-open spaces are the main vocabulary of Mardin architecture. Mostly they are open on the south site overlooking the valley.

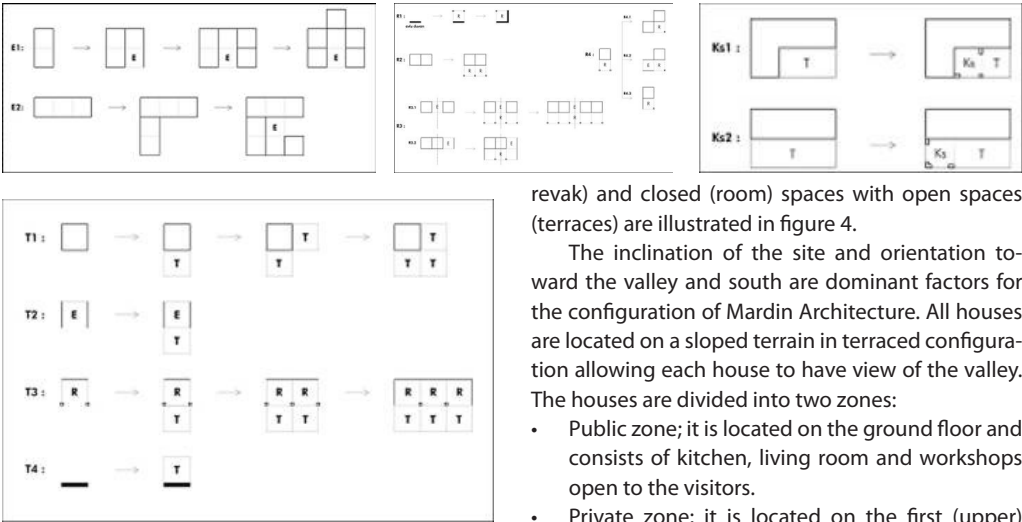
- Eyvan (iwan) is a semi-open space with a vaulted roof, usually surrounded on three sides by closed spaces, and the fourth side can be open or semi-

Figure 3a (left)
Formation of eyvan (E)
(Özbek, 2004)

Figure 3b (middle)
Formation of revak (R)
(Özbek, 2004)

Figure 3c (right)
Formation of kosk (K)
(Özbek, 2004)

Figure 4
Relationship with terraces
(Özbek, 2004)



- open. It is the main vocabulary used for expanding the house as shown in figure 3a.
- Revak (arcade) is a semi-open space surrounded by closed space (or wall), semi-open spaces of different kind such as Eyvan or kiosk and an open space as illustrated in Figure 3b.
 - Kosk (kiosk) is the third type of a semi-open space which is rarely preferred (Figure 3c). The relations between semi-open (eyvan and

revak) and closed (room) spaces with open spaces (terraces) are illustrated in figure 4.

The inclination of the site and orientation toward the valley and south are dominant factors for the configuration of Mardin Architecture. All houses are located on a sloped terrain in terraced configuration allowing each house to have view of the valley. The houses are divided into two zones:

- Public zone; it is located on the ground floor and consists of kitchen, living room and workshops open to the visitors.
- Private zone; it is located on the first (upper) floor and composed of bedrooms. In order to assure privacy in the private zone, the first floor is never aligned with the ground floor as illustrated on the figure 5.

Plan Layout Generator-PLG-

The program is developed using Java programming language. MAXScript is added to the program to generate 3d representation of house configurations that can be transported into 3dsMAX environment for further elaborations.

First, four basic parameters of a house x-y (the dimensions of a house), z (the number of floor), a (the number of room); are defined than by pressing the “Generate” button the house layout is produced. The program first controls the values whether they are in the defined limits and whether the stated values are adequate for the required space. If they are not, it displays an error message. Some limitations such as “the size of the grid” and “the number of the rooms” constrain the formation of a house as illustrated in the flow chart given in figure 6.

The program generates floor plan layouts based on design principles of a Mardin house. The

Figure 5
Situation of the upper floor
(Özbek, 2004)

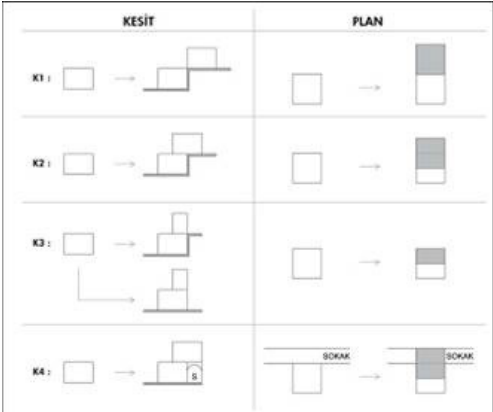


Figure 6
Flowchart of the program

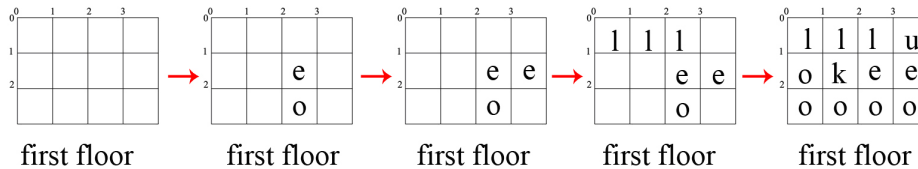
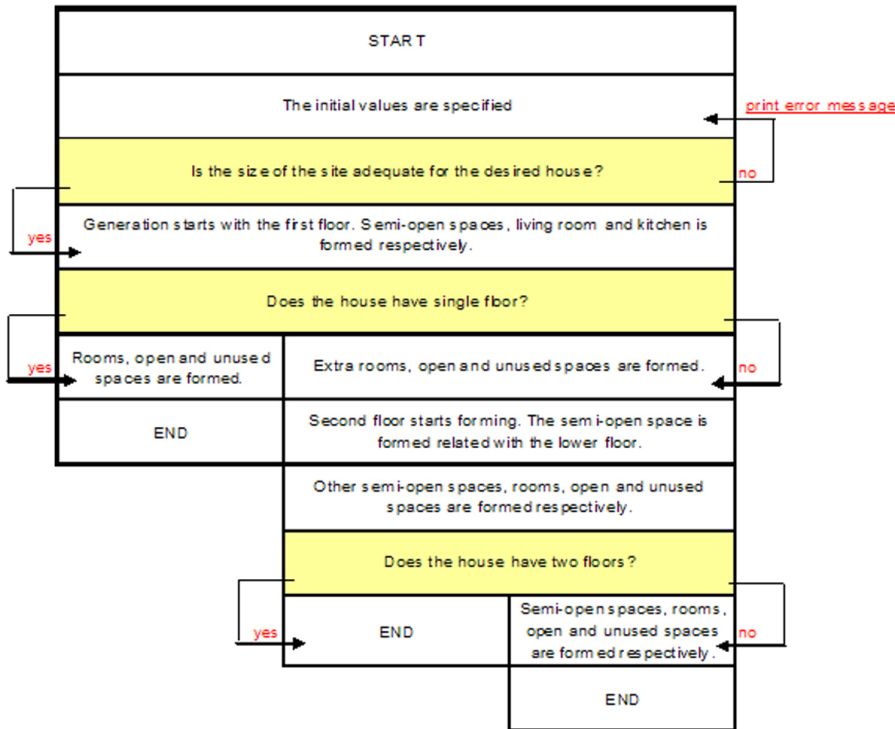


Figure 7
Formation of the plan layout
ground floor

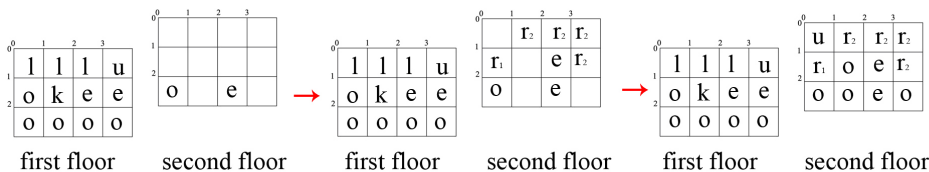
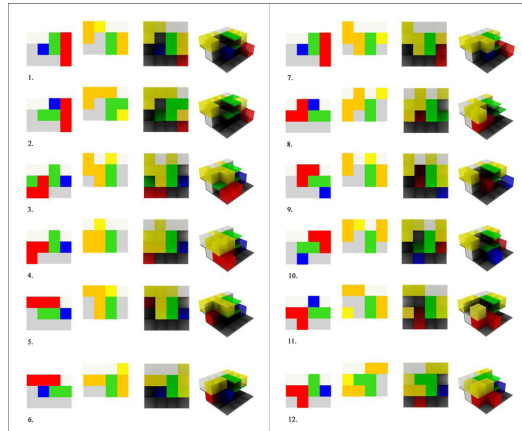


Figure 8
Formation of the plan layout
-first floor

Figure 9
Some alternatives generated
by PLG with the same initial
values



vocabulary of the house is distinguished with colored cubes. In the center of the house there is always a semi-open space –eyvan– which can be counted as the core of the house. The floors are related to each other through continuation of the open spaces such that the open space on the ground floors is reflected as open space on the first floor.

First, the numbers of semi-open spaces are selected randomly and the grids are drawn according to the given data. The first semi-open space (eyvan) indicated with letter “e” is placed in the center of the grid. Then living spaces (l - 3 units), kitchen (k - 1 unit), open spaces (o) and unused spaces (u) are placed respectively (Figure 7). If there is more than one floor, after the ground floor is generated, other floors generation is executed. On the upper floor(s) semi open spaces, single and double rooms (r_1 , r_2) and open and unused spaces are placed respectively as shown in figure 8.

If all specified parameters are acceptable for the program, the generation starts as described above. With the same initial values, alternatives of new house configurations can be obtained as shown in figure 9. In the example the parameters are:

- x: 4 (the dimension of one side is $4*4=16m$.)
- y: 3 (the dimension of the other side is $4*3=12m$.)
- z: 2 (the number of floor is two)
- a: 3 (the number of room is three)

Different house configurations can be generated

Figure 10
Three different house configurations generated with given parameters; x: 4, y: 3, z: 3, a: 7; x: 4, y: 3, z: 2, a: 3 and x: 3, y: 4, z: 3, a: 6

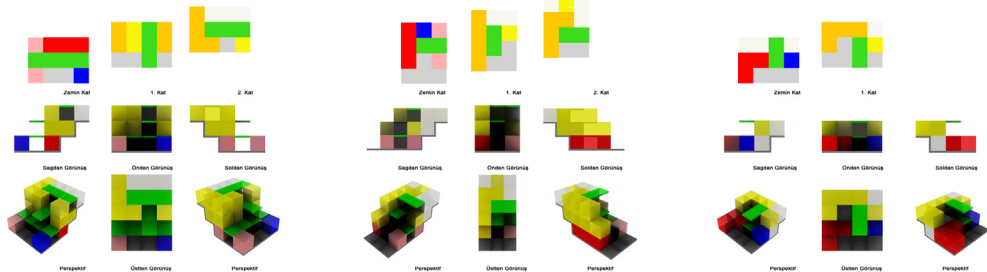
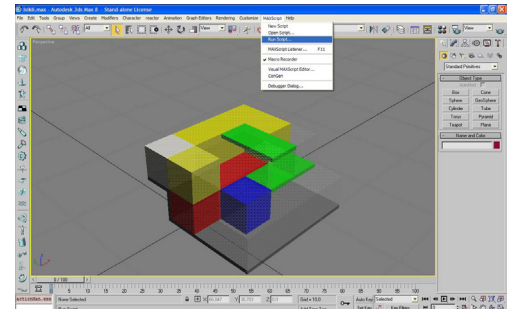
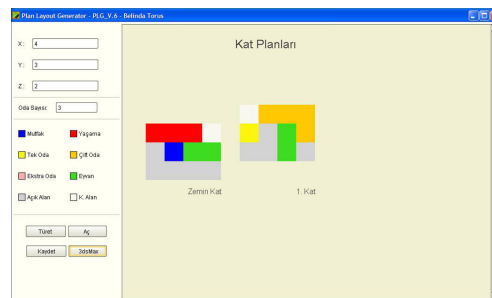


Figure 11 (left)
Interface of the program

Figure 12 (right)
Generating the 3d of the abstract house



rapidly with specified values as shown in figure 10. Then these new house form compositions can be transported into 3dsMax environment for further architectural articulations.

Program interface

The simple program interface illustrated in figure 11 allows user to specify the size of the site (x, y), number of floors (z) and number of rooms (a) through a dialog box located on the left side of the user interface. The color legend distinguishes the vocabulary of the spaces. The program operates with four commands "Generate", "Open", "Save" and "3dsMax" executed by pressing their related buttons.

The "3dsMax" button transforms 2d layout of the house into 3d format and saves it with a .ms extension in a script format. This file can be transported into 3dsMax for further architectural elaborations as illustrated in figure 12.

Conclusion

PLG is a specific program that generates Mardin style new house layouts aimed to assist the designer in the preliminary design phase. It can produce a family of abstract house configurations in a relatively short period of time allowing the designer to spend more time on architectural articulation of the designs.

The house configurations generated in PLG are abstract forms which emphasize functional and spatial properties. These abstract forms are open for different interpretations by different designers allowing wide spectrum of house designs to be derived.

The program has an open structure that allows a designer (or a programmer) to add his/her rules in order to make a personalized tool.

Acknowledgement

We want to thank Viki Handeli for her support and enthusiasm for developing the program.

References

- Alioğlu, F.: 2000, Mardin Şehir Dokusu ve Evleri, TE ve TT Vakfı.
- Çolakoğlu B., Dionyan S.: 2005, A Parametric Form Generator: ConGen, eCAADe 23 Conference, Portugal, pp. 623-628.
- Çolakoğlu B.: 2007, Design by Grammar: Hybrid Applications of Grammar in Design Studio Projects, eCAADe 25 Conference, Frankfurt, pp. 919-924.
- Duarte, J. P.: 2001 Customizing Mass Housing: A Discursive Grammar for Siza's Malagueta Houses, PhD Thesis, MIT.
- McGill, M. C.: 2002, Shaper2D: visual software for learning shape grammars, in Proceedings of the 20th eCAADe Conference, Warsaw, pp. 148-151.
- Özbek, H.: 2004, Gelenekselden Türeyen Çağdaş Mardin Konut Yerleşimi, Masters thesis, YTÜ.
- Özkaraduman, Y.: 2007, Geleneksel mimari dil için geliştirilen tasarım grameri: Mardin, Masters Thesis, YTÜ.
- Wang, Y., Duarte, J. P.: 2002, Automatic generation and fabrication of designs, Automation in Construction, 11, pp. 291-302.
- Yazar, T., Çolakoğlu B.: 2007, "QShaper: A CAD Utility for Shape Grammars", eCAADe 25 Conference, Frankfurt, pp. 941-946.
- [1] Grammar Based Design: http://www.bot.yildiz.edu.tr/_sites/mardin/English/indexe.htm