Attractive City - an interactive city generator

Edyta Augustynowicz1, Stefanie Sixt2, Sofia Georgakopoulou3
[1,2,3]*Chair of Computer Aided Architectural Design, Department of Architecture, Institute for Technology in Architecture (ITA), ETH Zurich, Switzerland.*
[1]*edytaau@gmail.com,* [2]*sixts@student.ethz.ch,* [3]*sofiag@swissonline.ch*

Abstract. *The Attractive City Generator is an interactive installation with which users can create or rearrange virtual cities by placing and moving physical objects. Through the set of simple inputs that the users provide and which represent city areas, landscapes or landmarks, the ACG is capable of creating complex and growing cities. In addition, a plethora of factual feedback on these cities is calculated. Becoming physically involved enhances the learning process and increases the user's interest. The users are drawn to the colourful and simple interface. Due to the practical feedback that the city returns, the users can feel more attached to the city they have created. A dynamic dialogue is formed between the city and its creators, that involves the wishes of the users versus the needs of the virtual city, the practical interaction on the table versus the theoretical calculations taking place inside the computer, the physical input versus the digital output.*

Keywords. *Interactivity; city planning; human-computer interface; L-system; developmental models.*

Introduction

Cities are complex, living systems, enclosing a dynamic and a behaviour that is extensively investigated both from a sociological and a structural point of view. Modeling and visualization of cities using computer programming is becoming increasingly common (Parish and Müller, 2001). Although this has undoubtedly been stimulated by advances in computer technology, such developments are part of wider transformation in the methodology of the social scenes (Batty, 1971).

Up to date, city modeling has focused on creating realistic representations of cities, to be used as a platform for computer games and movies (Parish and Müller, 2001), as models of urban expansion and dynamics (Batty et al., 1999; Wu et al., 2010), or as sociological study tools (Hillier, 1997). A lot of research has gone into the visualization and rendering of cities as well as on the hierarchy and organization that leads to the creation of feasible city patterns (Davis et al., 1999; Decorel et al., 1999; Fuji et al., 1995; Rubin and Whitted, 1980). On the other hand, Hillier (1997) in his space syntax used computer modeling to explain social aspects such as human behaviour and activities from a spatial point of view. So far, little effort has been devoted to the study of human-computer interaction during the development of virtual cities. The creation of an interface through which people can interact with the computer and influence the evolution of a city is the goal of this project.

Project goals

The Attractive City Generator (ACG) [1] is designed to combine the physical and virtual worlds, with the users as mediators. The setup of the installation is made for open public spaces. The ACG serves mainly as a game, which enhances the physical interaction of a group of people using a digital medium. This ability to effectively combine direct physical movements with a virtual creation giving real time feedback is what makes the ACG unique. Only a few moves are needed for the users to create and rearrange big cities – a task, which would have been tedious and time consuming when using physical models [Figure 1] (Wejchert, 1984). The application is designed so that the user can understand the concept fast and in a playful manner. Moreover, many users can work simultaneously and complete each other, without inhibiting each other's choices. Every move, whatever effect it has in the virtual city, can very easily be undone.

Figure 1
Urban planners with physical model.

The ACG can also serve as an educational tool, capable of raising awareness on the complexity involved in the life of a city (Sala et al., 2000; Syphard et al., 2005). Users learn that decisions concerning specific city areas can on certain cases have larger effects in all aspects: structural, environmental (pollution), economic (employment) and sociologic (living standards) (Herold et al. 2003; Alberti 2005). Although the ACG is still far from providing a thorough understanding on city dynamics and urban rules (Lehnerer, 2009), citizens can profit by gaining an insight into the complexity of the urban systems, and the intricate problems urban planners are constantly facing and are asked to resolve.

In addition, the ACG is an indicator of how people understand a city, and in which environment and social surroundings they prefer to inhabit (Wright, 2002). The question comes up, "how do we want to live now and in the future". Users get to experience that sometimes their wishes and ideas of how a city should develop and function can be contradictory and therefore not viable. Thus, they are urged to think deeper and to prioritize, in order to build a city that will both fulfill their needs and satisfy their wishes.

Tool description

The ACG creates urban environments based on a hierarchical set of comprehensible rules. The process of the city creation is based on three steps. Initially, the user is asked to apply the input data, using the simple and relatively minimal interface described in the next paragraph. In the second step,

Figure 2
The ACG attractors on the light box. Users collaborating to build a virtual city.

the arrangement and colour values are translated by the colour recognition program and fed to the city pattern generation system. The latter evaluates the relations and dependencies among the physical inputs and drives a chain of reactions that leads to the creation of a city.

The interface

The user interface consists of a set of colourful objects, drawings or grayscale images called *attractors*, which the user can place on a bright, uniform surface, the *light box* [Figure 2]. A home-video camera records the arrangement of these attractors. The recorded information is translated into a growing, complex urban environment via an object-oriented program written in Processing [2], based on *Java*.

Inputs

The input data that the attractors represent and that is used to build up a virtual city can be categorized into three general classes:

- Landscape maps – elevation maps, represented by a grayscale image; water/vegetation scheme, drawn by the user [Figure 3]
- Sociostatistical maps – zone maps: residential, business or industry zones represented by physical objects (attractors)
- Important buildings and landmarks – churches and bell towers, city halls, monuments, represented by physical objects (attractors)

Outputs

The ACG provides the user with a very strong visual output, which comes in the form of various "city-scapes" [Figure 4]. Each city can be different in terms of:

- the overall city landscape – whether the city has, for example, a strong skyline or many high chimneys;
- the city's main building typology – what type of structures dominate the city? Are there more high-rise buildings or is the general impression

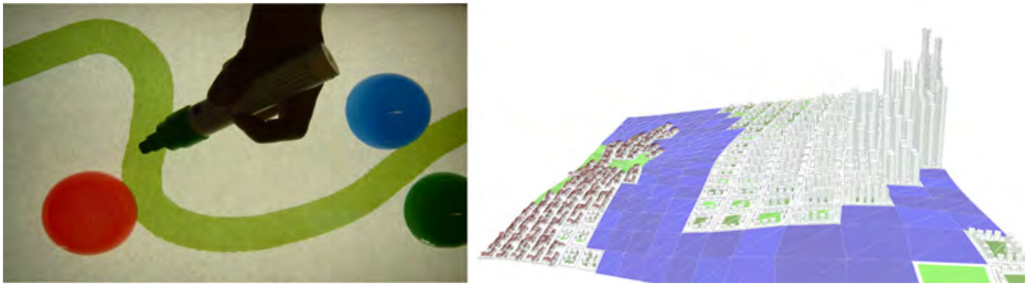

Figure 3
A user drawing a river on the light box and the resulting landscape.

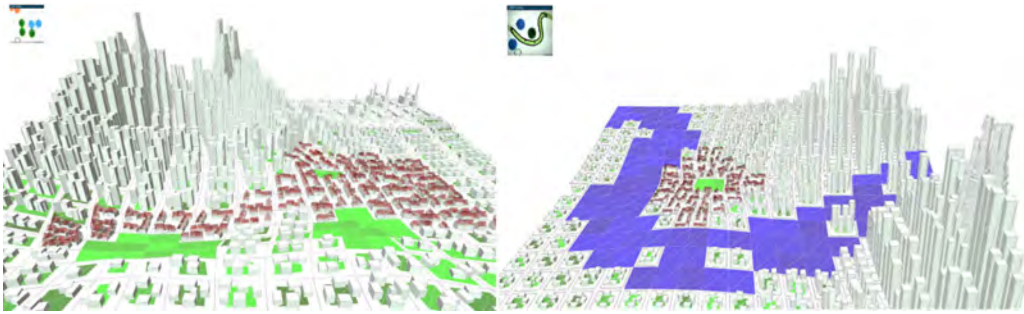

Figure 4
City-scapes arising from different arrangement of the attractors.

rather that of a village?

- the natural landscape features – the number of parks and rivers it contains and their arrangement throughout the urban tissue; and finally,
- the arrangement of all the city's neighbourhoods and features – how are the different areas distributed with respect to one another? Are there factories next to the residential areas? Are there parks in the middle of a high-rise area?

The user can constantly challenge and change this output by rearranging the attractors on the light box. This visual output is abundant in information on the city and its characteristics. On the right-hand-side of the screen, the user can find a table with facts about the created city that can be divided into 3 types of data:

- sociologic (population; density and permeability; the general quality of life)
- economic (city's main function and products; employment)
- environmental (pollution)

The choice of inputs and outputs of the ACG has been made after meticulously selecting and rejecting a wide variety of factors that comprise a city. The main goal has been the creation of a simple interface, where people can sit together and physically interact with a virtual generator in a playful manner. A lot of consideration has gone into making the interface minimal and easy, while keeping the output intriguing to the user, and this is reflected by the restricted amount of inputs, versus the plethora of outputs that the ACG has to offer.

Methods

The ACG system is built according to the dependencies scheme shown in the [Figure 5]. In the first step, the physical input is fed to the program with the use of the colour recognition application. The initial grid, created with L-system is distracted by the newly created attractors. The way the grid is distracted is not a direct reflection of the positions of the attractors. Therefore the resulting grid pattern is often unexpected and unpredictable. Each attractor sets a rule in their final area of influence. This rule extends the basic code of the grid. The regions within the grid and its edges gain properties that define the typology of the plots and the width of the roads. In the last step, buildings are created – their type, arrangement and density are determined by the function of each plot and the neighbouring areas.

Colour recognition

Connection of the physical input data with the virtual city is possible due to the colour recognition application, based on computer vision system (Nixon and Aguado, 2002). Once the program is launched an external camera starts to record the initial settings. The colour values of the recorded pixels are then passed to the corresponding grid of cells – these are the basic 2D elements with the capacity to store complex information. They are displayed in the top left corner of the screen and they constitute the starting point of the virtual city creation.

Initially, the user is asked perform the "camera

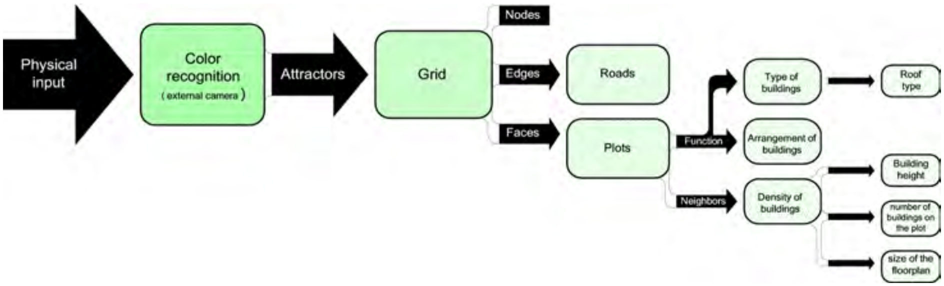

Figure 5
Diagram of program dependencies.

calibration", i.e. to assign functions to each coloured object. The program will then start following the selected colours. Depending on the data type, three methods of calibration can be distinguished. To set the function zones, one attractor of each colour should be placed in a specified spot on the light box. By pressing the "1", "2" or "3" key, different functions are assigned to the captured RGB values. The program detects areas of colour and finds the middle point of each area. This information of the centre point and the respective colour is subsequently passed on to an array of the so-called *function_attractors*. Their positions are updated within every capture, so the user is able to rearrange the objects freely during the game.

The second method of calibration refers to the water/vegetation scheme. The user is asked to place the image on the light box, within the specified area. By pressing the "g" key for vegetation or the "w" key for water, the user defines the land usage colours. In contrast to the *function_attractors*, the *land_usage* class objects are defined by the entire array of cells that constitute areas of specific colours, and not just by the centre position of these areas. The scope of influence of such attractors is limited to their boundaries. Similarly to the previous case, the user can change or redraw the land usage map at every stage of the city generation.

Finally, the last type of calibration is applied to the grayscale image representing the elevation map. The user is asked to place the picture on the light box and press the "e" key. Unlike in other methods, no attractor is created. This time, each cell object is filled with the value of grey colour from the corresponding place on the map. This value is stored within a variable called *elevation_value* and does not react to the displacement of the map. After calibration, this image can be taken from the light box. To change the hypsometry of the city the whole process should be repeated.

Values obtained by the camera recognition part are passed on to the city grid and deform it according to the applied rules.

Creation of the grid – L-systems

The idea of the city creation, in terms of grid arrangement and organization, was based on Lindenmayer Systems - an algorithmic model for the growth of fractal-like organic forms (Lindenmayer, 1968). This method allows to model complex patterns using a few simple rules. The topology of the city tissue can be developed using the formalism of map L-systems, which, in contrast to the typical L-system strings, allows the formation of areas in an enclosed structure. The starting point of segmentation is not a line, like in typical L-systems, but a closed 2D polygonal shape, representing the city boundaries.

Map L-systems is a mathematical graph that consists of a certain set of regions - 2D planar faces. Each region is surrounded by a boundary of lines, called edges. Two or more of those lines meet at one point - vertex [Figure 6]. A map corresponds to the city tissue, regions reflect city plots, while edges represent plot boundaries. A map-rewriting process generates new city grid divisions. There are several ways to determine this process. The one used in the ACG is a formalism of the "Binary Propagating Map OL-system with markers", also called mBMOL-system (Nakamura et al., 1986). The latter is a refinement of the basic concept of map L-systems introduced by Lindenmayer and Rozenberg (1979). It is based on the control of the edges and is a context-free mechanism (the regions are modified irrespective to their neighbours). While iterations are running, each edge divides itself according to its assigned string symbol (see paragraph below) and in a way that is specified by the code. New regions result from splitting the existing ones by inserting new edges between the markers. The system is binary, which means that each area can split into at most two parts in one iteration.

Each edge is described by a string of symbols – the so-called production rules – that determine the development of regions enclosed by the edges. Local variations of the grid may be obtained by sequential differentiation of the code in predefined areas of the structure. Those changes depend on the

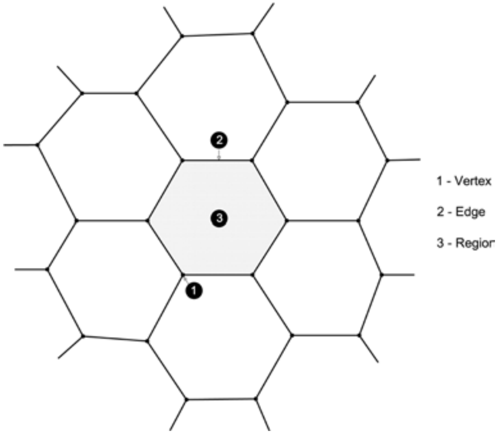

Figure 6
Diagram of program dependencies.

distance of each plot from the attractors and on the distances between the attractors. That means that in the areas of influence of attractors the initial code of the particular edges will be changed, which will lead to different grid divisions. The user affects the tessellation pattern of the city by moving the attractors.

City organization and typology

The arrangement and relative positions of the attractors give rise to a certain grid pattern. This pattern is responsible for the assignment of different city typologies. The created grid is transformed into a city map, where each grid region represents plots of a specified function and in consequence, a different city area. As a starting point we have used three city typologies, i.e. residential, commercial and industrial areas [Figure 7].

The array of plots forms an intelligent system that can rearrange according to their neighbour situation. Plots interact with each other, which means that each may change function depending on the functions of the adjacent plots: residential areas move away from industrial plots, and develop more towards a river. Neighbouring plots that share the same function may join and thus influence the surrounding street network. All this adds to the ACG an element of surprise. The result is no longer intuitive, as it is not a mere projection on the screen of plots according to the attractors' positions.

The newly formed city plots give rise to buildings. Each plot leads to the creation of different building types depending on the city typology that is assigned to it and the on the neighbouring situation. The type of buildings and their arrangement are called by the function of the plot, while the plot density is influenced by its neighbour's type. The buildings themselves are represented by simple, generic boxes. However, depending on the assigned typology, they vary in height and width, thus discriminating between i.e. warehouses, apartment buildings or skyscrapers. Other features, such as wedged roofs or large chimneys, may appear upon creation of a new area. As an example, residential area blocks give rise to medium-height, connected buildings, while commercial blocks call high-rise buildings and skyscrapers.

According to the typology assigned to each plot, the number and height of the buildings is calculated and an estimate of the city density and its population – permanent and commuting – is estimated. Various other city features, such as main services and functions, are derived from the density of the respective areas and the relation among them. A city with a dominant industrial environment might represent an exporter of goods; the size and position of its residential area will define the amount of commuters as well as the availability and variety of offered services (restaurants, cinemas, etc).

The created city is a chain reaction that the user initiates simply by placing a colourful object on the

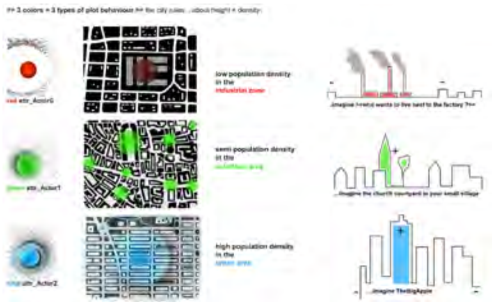

Figure 7
Example of function_attractors and the created city typologies.

light box. Any physical movement of these objects sparks the creation and transformation of a vast and complex virtual system.

The user experience

The ACG interface appears to be simple and minimal in terms of input and output. The complex calculations and scripts involved in the city creation are not visible thus allowing for an undisturbed and pleasant experience.

At start-up, the ACG gives the user – Mr. ACU (Attractive City User) –the feeling of complete freedom. By initialization, he is confronted with an empty screen. The light box in front of him and the set of attractors are his only clues. Mr. ACU quickly discovers the meaning of each attractor simply by placing them on the light box. Various city features and areas appear and grow. Every time Mr. ACU starts over with a new attractor, the city emerges with a different initial input, creating a new starting point of view. This alters the further development of the city: if Mr. ACU begins with an attractor that creates a river, the arrangement of the city will evolve around it and Mr. ACU's choices will depend strongly on the position and orientation of the river. If the first created feature is a church, he is more likely to try to arrange the rest of the city so that a residential area lies around it. Two very different cities will emerge as a result of choosing different initial attractors.

By moving the attractors around the light box Mr. ACU subsequently discovers that different combinations of the attractors also lead to new structures of the city. If he places a residential area close to large industrial areas he will not be successful in creating a living environment, as the two areas, in the given proportions, are not compatible with each other. On the other hand, he will also not be successful in building industry around the city hall. By discovering these type of dynamics, Mr. ACU learns that as a creator he is still limited by social and economical factors. He therefore gets an idea of the type of problems and puzzles urban planners are typically confronted with, and he gains an understanding of the meaning and importance of city dynamics.

Mr. ACU also starts noticing and using the list of factual data that appear and change with the city. He can therefore continue to move the attractors and re-arrange his city based not only on his aesthetics, but also on the practical feedback. He thus becomes more involved, as he tries to control the lifestyle and the overall quality of life of the citizens. The generated city will reflect the ACU's own desires, fears and needs.

Reactions and comments

The ACG is received by the public with interest and enthusiasm. People are eager to approach and try it out, as well as to give their opinion and feedback on the project.

One of the main requests we got from the ACG users is to incorporate more factual feedback in the program. People are eager to find out all the details concerning the city they have created, from the carbon emission levels, to the traffic system efficiency. The latter subject is in fact a matter that users wished to see developed further, so that the evolving city will eventually develop on its own an efficient transport system.

People have also been urging for an "upgrade" of the user interface, using touch-screens or digital tables. For the time being such an upgrade would defy the purpose of the ACG interface, which is meant to be of low-cost and of low-complexity so that it can be widely accessible. However, we can very well imagine a fancier version of the ACG, with a state-of-the-art user interface, which could be used in museums and universities.

The users of the ACG have been overall satisfied with the level of graphics involved, which were kept minimal. This was intended, so that people are not distracted by the details of the city, but can focus on the overall city structure and arrangement. The minimal graphics also allow the city to react very fast to all the input movements, thus increasing the users' stimulation and keeping their interest to a high level.

Outlook and conclusions

The ACG is a tool with unlimited possibilities for development and usage. We wish to incorporate the views that the users have expressed and enrich the calculations of factual data that are derived from the city. The output will be a visual effect on the city – i.e. increased pollution will lead to parks and rivers turning brown and to a grey sky.

Furthermore, it is our goal to start taking the ACG to a new direction, by implementing intricate calculations of urban rules and urban expansion simulations. In the long run it can serve as a powerful tool, able to investigate actual city dynamics. Although we simplify the underlying model of a virtual city, the ACG has the potential for extensions with the methodology applied. With the help of the L-system approach (Lindenmayer, 1968) and the shape grammar developed by Stiny (1975), reaching this ambitious purpose is feasible. For urban planners, the ACG can then be used as a quick method for visualizing their ideas, thus opening the way towards solving complex city planning questions and problems.

The ACG can also serve as an educational tool, enhancing public interest in the development of city neighbourhoods and districts. Historians or museums exploring the urban development of different areas can use it to visualize and share the results of their research. The ACG can also take the form of a web application, collecting landscape information and city plans from digital satellite maps and leading users towards a digital recreation of the existing world. The common theme underlying all these applications is the increase of public awareness in urban planning – a subject that is extremely important for the society, and yet very poorly recognized by the citizens.

The ACG is a tool that combines the physical and virtual worlds in a stimulating platform. It urges the users to get interested and get involved in the issues that concern them and the cities they live in. For the first time, people are allowed to sit together around a table and, by moving the objects on this table, to create a new world. They can give rise to cities where they would like to co-inhabit, or cities that achieve a certain common goal. This is the ultimate aim of the ACG.

Acknowledgements

We would like to thank B. Dillenburger and M. Hansmeyer for their invaluable comments and suggestions, as well as Prof. L. Hovestadt for his support of our work.

References

Alberti, M 2005, 'The effects of urban patterns on ecosystem function', International Regional Science Review, 28, pp. 168–192.

Batty, M, Xie, Y and Sun, Z 1971, 'Modelling cities as dynamic systems', Nature, vol. 231, pp.425-428.

Batty, M, Xie, Y and Sun, Z 1999, 'Modeling urban dynamics through GIS-based cellular automata', Computers, Environment and Urban Systems, 23, pp. 205–233.

Davis, D, Ribarsky, W, Jiang, TY, Faust, N and Ho, S 1999, 'Real Time Visualization of Scalably Large Collections of Heterogeneous Objects', IEEE Visualization '99, pp.437-440, 10.

Decoret, X, Schauffler, G, Sillion, F and Dorsey, J 1999, 'Multi-layered Impostors for Accelerated Rendering', Eurographics, 18(3).

Fujii, T, Imamura, K, Yasuda, T, Yokoi, S and Torikawi, J 1995, 'A Virtual Scene Simulation System for City Planning', Computer Graphics International.

Herold, M, Goldstein, CN and Clarke, CK 2003, 'The spatiotemporal form of urban growth: measurement, analysis and modeling', Remote Sensing of Environment, 86, pp. 286–302.

Hillier, B 1997, Space is the Machine: A Configurational Theory of Architecture, Cambridge University Press, Cambridge, UK.

Lehnerer, A (ed.) 2009, Grand Urban Rules, 010 publishers, Amsterdam.

Lindenmayer, A 1968, Mathematical models for cellular

interaction in development', Parts I and II, Journal of Theoretical Biology, 18, pp.280 – 315.
Lindenmayer, A and Rozenberg, G 1976, 'Parallel generation of maps: Developmental systems for cell layers' in V.Claus, H.Ehrig, and G.Rozenberg (eds), Graph grammars and their application to computer science; First International Workshop, Lecture Notes in Computer Science 73, Springer-Verlag, Berlin, pp. 301-316.
Nakamura, A, Linenmayer, A, and Aizawa, K 1986, 'Some systems for map generation' in G.Rozenberg and A.Salomaa (eds), The Book of L, Springer-Verlag, Berlin, pp. 323-332.
Nixon, M and Aguado, A (ed.) 2002, Feature Extraction and Image Processing, Newnes, Oxford.
Parish, YIH and Müller, P 2001, 'Procedural Modeling of Cities', International Conference on Computer Graphics and Interactive Techniques, ACM (ed.), New York, USA, pp.301 - 308.
Prusinkiewicz, P and Lindenmayer, A 1996, 'Modeling of cellular layers', The Algorithmic Beauty of Plants, Springler-Verlag, New York, pp. 145-174.
Rubin, SM and Whitted, T 1980, 'A 3-dimencional Representation for Fast Rendering of Complex Scenes', Computer Graphics, 14(3), pp.110-116.
Sala, OE, Chapin, FS, Armesto, JJ, Berlow, E, Bloomfield, J, Dirzo, R, Huber-Sanwald, E Huenneke, LF, Jackson, RB, Kinzig, A, Leemans, R, Lodge, DM, Mooney, HA, Oesterheld, M, Poff, NL, Sykes, MT, Walker, BH, Walker, M and Wall, DH 2000, 'Global biodiversity scenarios for the year 2100', Science, 287, pp. 1770–1774.
Stiny, G (ed) 1975, Pictorial and Formal Aspects of Shapes and Shape Grammar, Birkhauser, Basel, Switzerland.
Syphard, AD, Clarke, KC and Franklin, J 2005, 'Using a cellular automaton model to forecast the effects of urban growth on habitat pattern in southern California', Ecology Complex, 2, pp. 185–203.
Wejchert, K (ed.) 1984, Elementy kompozycji urbanistycznej, wyd. 2, Arkady, Warsaw.
Wu, D, Liu, J, Wang, S and Wang, R 2010, 'Simulating urban expansion by coupling a stochastic cellular automata model and socioeconomic indicators', Stochastic Environmental Research and Risk Assessment (2010) 24:235–245.
Wright, W 2002, 'Sims, BattleBots, Cellular Automata God and Go', The International Journal of Computer Game Research, vol. 2, issue 1.
[1] Attractive City Generator: www.youtube.com/watch?v=ei812Ze8y-w
[2] processing.org