TECHNICAL TRANSACTIONS | CZASOPISMO TECHNICZNE
ARCHITECTURE | ARCHITEKTURA
4-A/2015

FARID NASSERY,* PAWEŁ SIKORSKI**

NEW POSIBILITIES OF USING PROCESSING AND MODERN METHODS OF THE "GENERATIVE ART" GRAPHICS IN ARCHITECTURE

WYKORZYSTANIE PROCESSINGU I NOWOCZESNYCH METOD TWORZENIA GRAFIKI „GENERATIVE ART" DLA POTRZEB ARCHITEKTONICZNYCH

Abstract

This paper presents the process of creating graphics belonging to generative art and possibilities of its application in architectural works. Main focus is on different ways of creating pictures that fall under Generative Art movement, based on Processing programing. Differences and similarities between classical (analog) and digital creation are discussed, as well as successive phase of creating scripts and its application to architects' drawing technique. Processing allows to easily create almost infinite numbers of variants for once designed and programed graphic. Generative Art can be used in vivid and artistic arrangement of elevation.

Keywords: *Generative Art.*, *Processing*

Streszczenie

Artykuł prezentuje proces tworzenia grafik zaliczanych do sztuki generatywnej oraz możliwości jej zastosowania w twórczości architektonicznej. Skupiono się na sposobach tworzenia rysunków z nurtu Generative Art opartych na uproszczonym języku programowaniu Processing. Pozwala w szybki sposób tworzyć wręcz nieskończoną ilość wariantów raz zaprojektowanej i zaprogramowanej grafiki. Omówiono w zarysie kolejne fazy tworzenia skryptów oraz możliwości ich wykorzystania w warsztacie rysunkowym architekta. Zaproponowano także zastosowanie Generative Art do plastycznej aranżacji elewacji.

Słowa kluczowe: *Generative Art*., *Processing*

* Ph.D. Arch. Farid Nassery, Division of Descriptive Geometry, Technical Drawing & Engineering Graphics, Faculty of Architecture, Cracow University of Technology.
** Student Paweł Sikorski, SKN IMAGO, Faculty of Architecture, Cracow University of Technology.

1. Introduction

In the first part of our review we shall present the process of creating graphics which fall within the new branch of artistic creation i.e. generative art. We shall also suggest some possibilities of its application in other artistic fields as well as economy. In this publication we shall acquaint the reader with the issues of generative systems and generative art both of which are widely discussed in the literature on the subject: Daniel Shiffman [16], Wiliam Mitchell [10], Yehuda Kalay and Wiliam Mitchell [7], Krystyna Januszkiewicz [6] dealing with the issue of generative systems; Hartmut Bohnacker, Benedikt Groß, Jula Laub and Claudius Lazzeroni [1] and Philip Galanter [3, 4] in which generative art is discussed – to name but a few.

In the second part of the paper we shall describe Processing, a programming language used in generative art. It is discussed in detail in the books of Matt Pearson [12], Casey Reas and Benjamin Fry [13], Daniel Shiffman [15] to mention a few.

In the next part we shall focus only on the graphic artwork itself and its visual features. Moreover, we shall give a few primary examples of approaches and methods used for graphics' generation. The question of relationship between generative art, design and architecture is widely debated in the books of Casey Reas and Chandler McWilliams [14], Lisa Iwamoto [5].

The last part of the review includes authors' own concepts of using generative art in the architect's drawing technique and its application in artistic arrangement of the elevation.

2. Generative Art

The definition of the term generative art is ambiguous. There are at least two reasons for that – firstly, it is a fairly innovative approach to art [4], secondly, it refers to a wide variety of artistic creations. Contemporary authors often use definitions adjusted to the features of their own artworks. These include pieces which fall within a diversity of various areas, such as electronic and algorithmic music composition, computer graphics and animation, spatial installation art, industrial design and architecture [3]. In the enquiry for the correct definition one should be looking for a criterion broad enough to include an artistic activity within all the mentioned fields and any new yet to be discovered. But since any act of art creating is also a process of art generation and not all art is generative art, the sought after definition should be restrictive enough to exclude art pieces created not by the means and methods of generative art. Thus we are looking for their specific feature constituting the distinction between generative art and other trends. Therefore generative process refers to a process generating a piece of art which is autonomous and independent of the will of the author who is only a designer of a certain system composed of the strict rules managing the generation. During the design, the artist decides on the complexity and order of its constituents, initial parameters and possible states of the system [6]. Note that in this language a system is at the same time a collection of possibilities representing an artwork as such and a complex set of rules managing their evolution. Usually a system operates to some extent randomly and thus a distinct and unique result is obtained each time the generating process is evaluated.

One can observe that the term generative describes primarily how piece of art is created. It is simultaneously characterised by the influence of the artist on the way the generative process is designed and by the generation itself which is independent of any influence. The whole process is finalised with the selection of the generated results, however it may be partially implemented already in the algorithm.

Generative Art is not only about artistic creation, its focus lies mainly on the abstract process itself and the corresponding system. Although the nature of the process favours the use of technological solutions employed in computerisation and informatics [3], nothing prevents us from applying elaborated solutions in traditional tasks [4, 6]. Therefore generative art is strongly connected with natural science (especially physics and biology) along with mathematics and cutting-edge technologies often serve as its subject. To elucidate our consideration one can adduce Roman Verostko's series of ink pen drawings entitled Cyber Flowers [18]. The artist modified a plotter and translated a digital code into a curvilinear motion of a drawing device obtaining pure, minimalistic pieces. The influence of other artists such as Piet Mondrian and Kazimierz Malewicz can easily be seen.

Although new systems and algorithms are in demand, essentially only the following are available: Voronoi diagrams, L-Systems, Cellular Automata, fractals, genetic algorithms [6].

3. Processing

A user friendly programming language Processing is a convenient tool for a generative art creation. It was developed in 2001 as a project of Benjamin Frey and Casey Reas – two graduate students at the MIT (Massachusetts Institute of Technology) Media Lab. The very inspiration and the key ideas of Processing can be traced back to earlier Visible Language Workshop released in 1973 and held by Muriel Cooper, a pioneering researcher, designer and digital media educator [19]. The avant-garde class broadened horizons of attending students. Employing a wide variety of experimental activities it linked cutting-edge technologies of that time, design trends along with art and related them to the meanings to human communication. In 1985 the research group was incorporated into MIT Media Lab which was established by that time [19]. However it was the innovative classes Design by Numbers by John Maeda which influenced Processing and gave to it its final shape [2]. His main ambition was to acquaint graphic designers and artists with the basics of programming language. For this aim the creators of Processing concentrated on simplicity and wide accessibility even for those without any technical background and moreover decided to make it open-source. Rapid and common uptake stimulated its further development and emerging of new versions. In order to secure high quality of the project, founding was essential and thus the Processing Foundation was established. Its board of directors consists of Ben Fry, Casey Reas and Daniel Shiffman[1] who significantly contributed to the development of Processing. He created the basics of the syntax of this language along with teaching materials [15, 16] with examples and exercises.

[1] Professor at the Interactive Telecommunications Program at NYU's Tish School of the Arts.

To clarify the notion – Processing is not only a programming language but also an integrated development environment (IDE) [15]. Its versatility and simplified syntax are the most specific features setting this language apart. From the deep technical viewpoint Processing is based on Java. To generate the application which displays graphical instructions of the code , three steps needs to be carried out: firstly, one writes a script; secondly, the code is transformed into the Java syntax; thirdly, the compilation and displaying of the results take place [12].

The primary capability of Processing is to give a graphical interpretation to the digital code. This is the main advantage which makes it a highly useful and handy tool for data visualisation and broadly defined graphic design [13]. From another point of view, the flexibility and virtually infinite possibilities of script creation allow it to be thought of as the artistic output of audiovisual art and other various digital media activities [1]. It is compatible with the need to manipulate and work on sound, graphic, video, 3D models, other data and moreover communication technologies. The interaction between human and machine may occur by a numerous distinct ways such as cameras, tablets, microphones, joysticks and high-tech devices similar to Leap Motion (hand gestures reader) [9], Kinect (Xbox motion sensor 360) [8] and Oculus Rift (glasses for viewing virtual reality) [11].

4. The variety of the methods used in generative graphics

Roughly speaking, several major methods used to create generative graphics can be distinguished [3]. Probably the most intuitive and straightforward approach is by creation of intelligent and dynamic digital brushes. In the case of this tool the artist determines only the composition and arrangement of brush marks on virtual canvas while the final graphic result is solely generated. The displacement of the brush marks can be guided with the use of the mouse, tablet, gamepad etc...

The use of data visualisation gives rise to another method of graphics' creating. From the artistic point of view, the data is only a pretext and an inspiration which should not be presented as such but rather artistically processed as a material to create a graphic composition. The accent here lies on an aesthetic form which is to be obtained as a final result[2]. Due to the ease with which computer fonts may be processed, many graphic compositions are often dealing with typography.

Another method is image analysis which, roughly speaking, involves the processing of any existing visual material. The key aspect here is that the processing regulated by the generative system of determining rules is fully specified by the content of the processed graphic.

Probably a creative activity which consists in processing of images influenced by visual simulation modelling illustrates the best what generative art is about. The result here is one of the least predictable and has the most experimental character.

---

[2] As an example of this approach may serve the installation "What does the river hear?" made by the students of Cracow University of Technology during the workshop „in[formational] infrastructures" under the tutorship of D.Eng. in Architecture Farid Nassery held by Prof. Gregory Spaw, Tennessee University in Knoxville [20].

Diversity of puzzles and patterns gives rise to another possibility. The graphic as a whole can be composed from the individually generated constituents. Compositions which make use of fractal algorithms and recursions have to be singled out here for their tremendous consistency and harmony of the self-similar forms.

Yet another result comes from the generating of geometric graphics with the use of three--dimensional shapes. Technically it is printing a two-dimensional surface submerged in the space of higher dimension although now it is possible to create any three-dimensional shape with the use of a 3D printer.

Finally, quite understandably, one can mix and combine all the mentioned methods which results in almost endless possibilities of new aesthetic values restricted only by the boundaries of our imagination [3].

5. The craft of generative art

One can easily see that technological progress and ubiquitous digitalisation enforce the enhancement of use of computers not only in the architect's craftsmanship but almost in virtually any professional activity. To comprehend how script is written enables bringing the principles of computers' operation closer to theirs users and reminds them that, leaving aside the interface, they are just calculating machines.

"The conceptual distinction between conventional form-making and form-finding underlies nowadays the division of design methodology. Generative systems and instruments caused architect's imagination and sensibility to be pushed into the domains of predictions of the results of a given processes"[3].

The artist as such becomes rather a supervisor of the process [6]. A better understanding of the instruments used in our professions provides us with a rigorous and effective oversight over the act of creation and encourages to open up to new possibilities. Thus generative art makes the artistic results even more diversified [3]. The single artistic act in which creation takes place at the level of code scripting indicates the generation of the whole visual system consisting of a spectrum of possibilities enclosed in an innumerable series. Description given here refers at the same time to a conceptual creation and a virtual reality wherein they are executed. It is the author who decides which of the generated works shall be preserved, if he so wishes, with the use of paper and pigment.

Creating a code requires it to be expressed in an unambiguous, clear and precise way. To achieve such discipline one necessarily needs to plan and make a sketch of one's concepts on the paper before one can rephrase them in the code. This conceptual stage is essential as it serves as an opportunity to figure out and analyse the possible results. Therefore a decision if our concept is good enough to be carried out (i.e. written into the code) should and can be taken at this point.

Depending on the specific features of the concept, scripts require at least basic knowledge of mathematics and geometry. Program is composed of data subjected to operations which are executed in a given order and of additional instructions which could trigger and regulate

[3] K. Januszkiewicz, *Systemy i narzędzia generatywne*, Archivolta**,** No. 4, 2012, pp. 45.

them. In Processing one has at one's disposal more or less one hundred thematically organised commands. By making use of them and those crafted by oneself, one can script a code evaluating one's graphical concept.

An example of the code which is continuously animating semi-translucent azure and light grey squares:

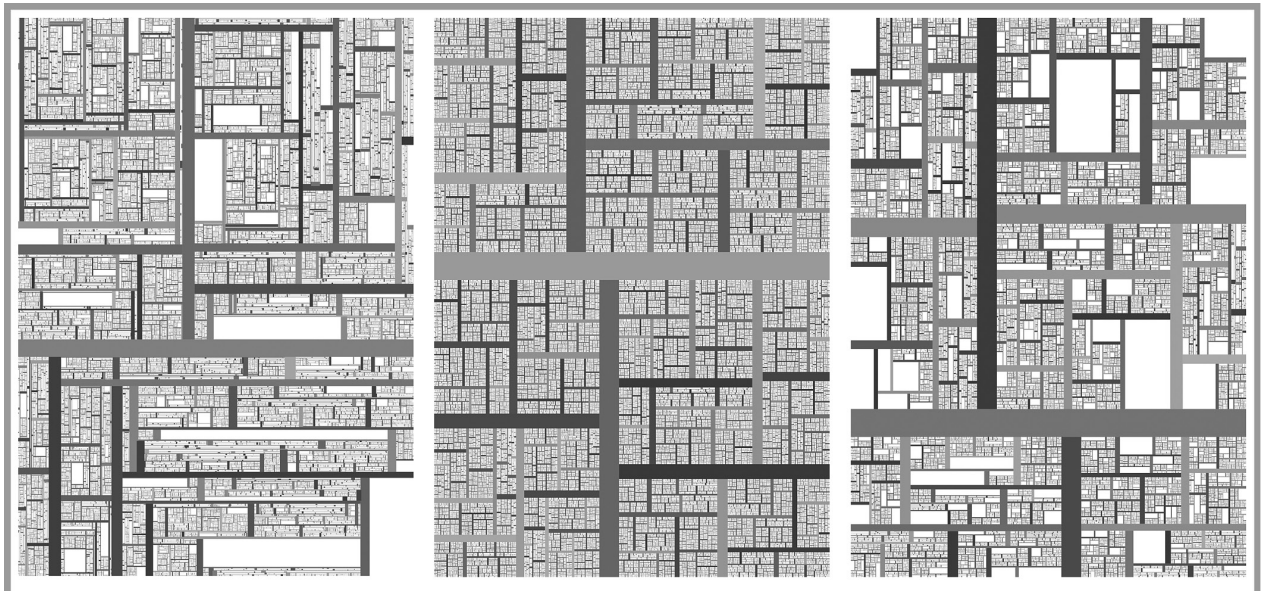
```
void setup() //initial instructions
{
 size(500, 500); //sets the magnitude of the window (in pixels)
}
void draw()//countinous drawing
{
 noStroke();  //turn off the contour
 fill(color(100, 0, random(100,255), 50));   //sets the filling
 rect(random(500-40), random(500-40), 40, 40);  //draw a square
 fill(167,152,123,101);
 rect(random(500-40), random(500-40), 40, 40);
}
```

The features of the scripts described above make them the ideal solution for artistic arrangement of buildings' elevations. Nowadays new and inexpensive methods of individualised elements production are invented. This makes generative art practically applicable in design of diverse and simultaneously consistent elevation panels. A series of patterns generated in this manner is a cross-section of a certain visual system. Their features may become apparent only when observed from a specific distances and therefore would give a building an unignorable and identifiable appearance and would make it a dominant point in the surrounding area – if not due to its scenery at least due to its aesthetic values. We shall present an exemplary graphical project (Ill. 1) which could serve as a starting point for a creation of similar panels. Its final details would depend mostly on the used material and expected visual effect, for example: in the case of panels made of glass or metal sheet one could use printing, sandblasting and acid etching or laser cutting; for stone – claddings sandblasting or milling; printing or using unique moulds would be adequate for concrete. In the case of concrete or stone panels a designed graphic must be transformed into a relief if printing is not an option. A similar process must also occur in the case of pattern cutting or sandblasting.

6. Summary

The adaptation of Processing to graphics' creation was a milestone which reformed the very design process along with the role of the creator who became rather a supervisor of the processes generating the artwork on their own. The use of this technology gives rise to the creation of a great many versions of a given graphic but it is the author who singles out the final variant or chooses the entire collection. However, programming is still preceded by the conceptual stage carried out with conventional artistic techniques.

Changing the methodology may trigger new areas of graphics application (the example of individualised elevation panels was discussed above) due to rapid creation of consistent collection of distinct artworks. The use of cutting-edge devices such as CNC router or 3D printer reduces the production costs even for highly individualised elements.



```
void makeSRD(float X, float Y, int k, PGraphics gg) {
  if (k < 32 && X > 1*saveScale && Y > 1*saveScale) {
    k++;
    gg.pushMatrix();
    gg.translate(X, 0);
    gg.rotate(HALF_PI);
    float A = random(Y*startP, Y*endP);
    float W = divisionWidth*Y;
    float halfW = W/2;
    gg.rect(A, X/2, W, X);
      makeSRD(A -halfW, X, k, gg);
    gg.translate(A +halfW, 0);
      makeSRD(Y-A -halfW, X, k, gg);
    gg.popMatrix();
  }
```

```
float saveScale = 10;
float startPosition = 0.25;
float endPosition = 0.75;
float divisionWidth = 0.05;
color colorFunction()
(...)
{
  if (isColored) {
    float a = random(80, 240);
    color c =
 color(a, 80 + a/2, random(156, 222));
    return c;
  } else
    return color(55);
}
```

Ill. 1. Works from series “Simple Rectangle Division” with programming code fragments (Processing generative graphics by Paweł Sikorski, 2014)

References

[1] Bohnacker H., Groß B., Laub J., *Lazzeroni C., Generative Design: Visualize, Program, and Create with Processing*, Princeton Architectural Press, New York 2012.

[2] *Design By Numbers* (online) Massachusetts Institute of Technology. homepage: http://dbn.media.mit.edu/ (date of access: 15-01-2015).

[3] Galanter P., *Generative Art after Computers*, [in:] *GENERATIVE ART – Proceedings GA2012 XV Generative Art Conference*, red. Soddu C., Domus Argenia Publisher, Milan 2012, p. 271-382.

[4] Galanter P., *What is Generative Art? Complexity Theory as a Context for Art Theory*, [w:] *GENERATIVE ART 2003 – Proceedings of the 6th International Conference Generative Art*, red. Soddu C., AleaDesigne Publisher, Milan 2003, p. 225-345.

[5] Iwamoto L., *Digital Fabrications: Architectural and Material Techniques*, Princeton Architectural Press, New York 2009.

[6] Januszkiewicz K., *Systemy i narzędzia generatywne*, Archivolta, nr 4, 2012, s. 44-51.

[7] Kalay Y.E., Mitchell W.J., *Architecture's New Media: Principles, Theories, And Methods of Computer-aided Design*, MIT Press, Cambridge 2004.

[8] *Kinect for Windows* (online) Microsoft. homepage: http://www.microsoft.com/en-us/kinectforwindows/ (access: 10-12-2014).

[9] *Leap Motion* (online) Leap Motion. homepage: https://www.leapmotion.com/product (access: 12-12-2014).

[10] Mitchell W.J., *The Logic of Architecture*, MIT Press, London 1990.

[11] *Next-Gen Virtual Reality* (online) Oculus. homepage: https://www.oculus.com/rift/ (access: 15-12-2014).

[12] Pearson M., *Generative Art: A Practical Guide Using Processing*, Manning Publications Co., New York 2011.

[13] Reas C., Fry B., *Processing: A Programming Handbook for Visual Designers*, MIT Press, Cambridge 2015.

[14] Reas C., McWilliams C., *FORM+CODE in Design, Art, and Architecture*, Princeton Architectural Press, New York 2012.

[15] Shiffman D., *Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction*, Morgan Kaufmann, Burlington 2008

[16] Shiffman D., *The Nature of Code* (online) Daniel Shiffman. homepage: http://natureofcode.com/book/ (access: 03-11-2014).

[17] *Uczestnicy warsztatów na PK wykonali instalację*, [in:] Nasza Politechnika red. Peters L., nr 5/2014 (online) Politechnika Krakowska. homepage: http://nasza.pk.edu.pl/images/stories/NP/np-2014-05.pdf (access: 15-01-2015).

[18] Verostko R., *Cyber Flowers* (online) compart – center of excellence digital art. homepage: http://dada.compart-bremen.de/item/artwork/916 (access: 03-12-2014).

[19] *Visible Language Workshop, Muriel Cooper, 1975* (online) MIT Museum. homepage: http://museum.mit.edu/150/115 (access: 15-12-2014)