Creative Programming Design Research Based on Generative Art with Processing

Jian Rao and Chuqi Qiu(✉)

Art and Design College, Hubei University of Technology, Wuhan, China
604704202@qq.com

Abstract. In the 1960s, in the context of computer new media, the creative form of "generating art" is similar to the "technological singularity" referred to today, which refers to works of art created by computer software algorithms or by random autonomous process algorithms. This paper adopts the methods of literature analysis and case study. The main innovation lies in the use of creative programming methods to deal with visualized picture language, explore new forms of computer-aided design visualization, improve the understanding of generative art by non-computer science majors. The implementation path uses Processing to complete instructions from conceptual artist Sol Lewitt"s wall-painting work at the Boston Museum, quantifying elements of points and lines to match the structure of the algorithm, comparing code with traditional manual work in the process. Experimental results show the complexity of the problem, the autonomy of the machine comes from the limitation of artificial rules, and the generative restoration depends on the preset processing of natural language compilation. Having the ability to reuse code does not mean thinking creatively programming at the same time, and the ever-changing challenges of digital survival need to be addressed in the diverse expansion of business scenarios.

Keywords: Generative Art · Creative Programming · Computer-Aided Design · Computer Graphics And Multimedia

1 Introduction

The synonyms of generative art include algorithm art, instruction art, etc. Although the emphasis is different, it mainly refers to the product of technological development in the computer environment and gradually gives birth to the change of design concept. Research in this area requires an approach to creative programming, that is, understanding the basic building blocks of computer science [1]. The compiler Processing has reduced the difficulty for non-computer majors to get in touch with assembly language. Experimental ideas and in-depth study can establish large-scale complex projects in

K. Subramanian et al. (Eds.): CTMCD 2022, ACSR 99, pp. 892–901, 2023.
https://doi.org/10.2991/978-94-6463-046-6_102

the future. In this context, cad can be divided into three parts: DATA collection (variables, arrays and other storage of their own information and review), CONTROL (conditions, loops and other instructions) setting, and ORGANIZATION (function, object programming) writing [2].

2 The Kernel and Extension of Generative Art

Essentially, all generative art focuses on the process of making art, which requires autonomy from the artist. Philip Galanter once explained generative art "From the Bottom Up" and "From the Top Down". The key element is that any practice using a somewhat autonomous system (a set of natural language rules, computer programs, machines or other programs) can contribute to the work of art [3].

The concept originated from computer graphics, which is a form of contemporary art creation. It's not necessarily about the artwork or the final product, but about the creative process and underlying logic. An artist or program follow the rules such as fractals, L systems, even randomization to produce unreplicable and beautiful results. In contrast to traditional artists, who might spend days or even months exploring for inspiration, generative code artists use computers to generate thousands of ideas in milliseconds. Generative artists use modern processing power to invent new aesthetics -- directing programs to run within a set of "artistic constraints" and guiding the process to the desired result.

Generative art, as a dimension of commercial, can be defined as a design process (emphasizing its evolutionary process, primarily involving evolutionary techniques in the creation and production of design solutions) or a design method (treating generative design as a design method, that is, as an algorithm-based or rules-based process based on which rules and processes can generate multiple and complex solutions that can include parameters or variables to form the process of design modeling, emphasizing its utility). Generative design has been widely used in the field of products, because design can be as low as daily necessities, and it can be as high as art. Whether Zhang Zhoujie's Sensor Chair should be in a bedroom or in a Design Miami exhibition depends a lot on what level of "perception" a cognitive collection interprets [4]. The semantic changes also result in the diverse and rich expressive forms of generative art design, and the underlying thinking framework is larger than the visual linear communication.

3 Cognitive Change in Creative Programming

Generative art has been controversial, with the existence of programming languages often overlooked in culture contexts. Traditional art creators are only interested in aesthetic issues work because they have difficulty accepting entirely computer-generated creations. This has led to resistance to learning digital programming tools. At the same

time, beauty in contemporary art is reduced to a hollow resource, and it is difficult to have a unified voice in the perspectives of practitioners in different professional backgrounds. The author tries to explain the relationship between computer art represented by generative art and art design as a whole by referring to the "cognitive development theory" of Jean Piaget, a Swiss child psychologist. Cognitive development is the process of professional construction and contains four core concepts. SCHEMA is consistent with the visual sensory interpretation of information to construct the object's cognition, using canvas and paint to construct the creation. ASSIMILATION compares traditional art to computer art in the information age and relates it to learners who assimilate new stimuli into existing cognitive structures using existing schemas, where code programming changes traditional tools but also carries the transmission significance of image language. ACCOMMODATION means that people interact with their surroundings on a daily basis, constantly creating, modifying and testing theories about the creative practices of art design to achieve EQUILIBRIUM.

The American artist John F. Simon Jr. Described the code as a form of creative writing [5]. Code is also considered a medium for digital artists. Under these circumstances, image language has provided the basis for learners to understand professional programming language, and the core ideas are similar to text languages. The essence of programming is sequence, the order in which instructions are completed, and hence the art of creation is also known as "Illustrative/Instruction Based Art" in 1960s. Condition clarifies the concept of "if so, then what," and the cod's language for trigger also includes the simultaneous parallelization of multiple units. Terms like command fit the context of the environment in which computers operate, but in the natural world commands are more attuned to the nonhuman "rules" by which natural things grow. So Galanter argues that even though matter is generally presented in digital form, the art of creation is uncoupled from any particular technology, and its broad cognition is far beyond the era dominated by the internet. Programming languages happen to be more valuable and desirable right now as systems with learning, adaptation, and mutation.

The development of programming Thinking and skills reaches younger generations. Computer scientist Jeannette Wing has introduced the concept Computational Thinking, which means that programming languages teach you how to think -- for example, you learn how to break down complex problems into simple ones, Learn how to find problems and debug them, and learn how to iterate and improve the design over time. This "computational thinking" applies not only to computers, but may also help us solve other types of problems. In the future, programming is going to be a language that makes "voice" under the trend of programming for young children: expressing ideas digitally, sharing ideas and communicating with people, a process that always involves digital states because it's an interesting way to create semi-autonomous systems.

4 Creative Programming Design Driven by Generative Art

Creative programming constructs an implementation path for computer graphics and multimedia interaction. It conveys creative ideas through the power of code, combined with visual, auditory and other sensory pathways. The essence of creative programming is to replace graphic expression tools, which breaks through the limitations of human thinking and traditional creation. It describes and extends the thought process in programming language. The exploration of "possibility" makes it more interesting than rational.

4.1 Tools for Creative Programming and Design

Generative art as a specific art form is determined by the tools used to provide a common syntax for the implementation of creative programming. Programming languages are simply a way for computers to respond to specific instructions and do not have the natural attributes to follow commands voluntarily.

The role of the computer in the visual realm of creative programming has been liberated from the user's direct control of tool cognition and transformed into collaborator or creative partner that may become an autonomous creative entity with an upgraded iteration in the future. When faced with two paths to perform actions or express ideas, a simpler implementation means it is closer to the essence of the tool.

One of the purposes of creative programming is to analyze graphics from the point of view of numbers and explore more possibilities of graphics with code rules. The second is to measure the human mind in order (O/C rule), thus developing training tools such as quantitative painting. It goes from tool use to high dimension, which is to discuss the natural language processing of human interaction with computers. According to the difficulty of technology implementation, these systems can be divided into three types: simple matching, fuzzy matching and paragraph understanding. The technical key is to make computers "understand" natural language, so natural language processing is also called natural language understanding, also known as computational linguistics. On the one hand it is a branch of natural information processing, and on the other it is one of the core subjects of artificial intelligence (AI).

Processing, developed by Casey Reas and Benjamin Fry, is the firstly used as artist/designer friendly tool in a wide range of programming languages [6]. It has a large family of contextual and programming environments, and its graphic elements are a branch of the Java language. Their language syntax rules are almost uniform, while Processing adds specific graphics and interaction features. Its graphics elements are associated with PostScript (the foundation of PDF) and OpenGL (a 3D graphics library).

The advantage of creative programming for graphic design using the Processing tool is its unique Tweak and REPL modes. Tweak is essentially a package of OSC code that allows any changes to the values in the DRAW function to be passed to the running program via OSC. Graphical programming often involves a lot of numerical tuning, and this feature is one of the reasons Processing users have not been able to switch completely to other frameworks and game engines. REPL can compile and update the code updated in draw in real time while the program is running, simplifying the four steps of closing, writing, saving, and running into two parts of writing and saving, thus saving time. Duplication is an important programming concept that can improve creative efficiency by writing classes or plug-ins that package the most common commands.

The purpose of using creative programming tools is not just to expand our capabilities. It should also enhance our creativity from the underlying logic. The latest software gives us the ability to shape the world in ways unimaginable 50 years ago, but there is still a long way to go in terms of the naturalness of its use. Almost all of the more powerful tools have a steep learning curve that requires commitment and constant use to master.

4.2 Presented Examples of Creative Programming Design

American conceptual artist Sol Lewitt has set the following instructions: 'On a wall surface, any continuous stretch of wall, using a hard pencil, place fifty points at random. The points should be evenly distributed over the area of the wall. All of the points should be connected by straight lines' [7].

This Wall Drawing was done over several days by the team at the Museum of Boston in 2012. The problem model, which was relatively simple in early exploration, belongs to the high ordered system. This process relinquishes the original author's handmade, that is, features that are not created directly by the original author and can be handed over to other teams for execution according to specific instructions. Paintings can be reproduced even though they have many variables (depth, thickness of hand painting), and the output is diverse and uniform.

Apparently, handmade wall painting has more uncontrollable elements than code-based works. It is more analog that means relating to, or behaving a mechanism in which data is presented by continuous variable-physical properties.

The creative programming code is as follows:

Algorithm 1 Creative Programming of texture

```
1   int pointX[] = new int[50];
2   int pointY[] = new int[50];
3   int xdir[] = new int[50];
4   int ydir[] = new int[50];
5
6   void setup()
7   {
8   size(displayWidth,displayHeight,P3D);
9     frameRate(60);
10    background(249);
11    strokeWeight(0.6);
12    stroke(100);
13    smooth();
14    for(int n = 0; n <50; n++){
15      pointX[n] = int(random(50, width-50));
16      pointY[n] = int(random(50, height-
    50));
17    }
18    for(int n = 0; n <50; n++){
19      xdir[n] = int(random(2))*2 -1;
20      ydir[n] = int(random(2))*2 -1;
21    }
```

```
22          }
23
24          void draw()
25          {
26           background(249);
27           for(int n = 0; n <50; n++){
28             for(int j = 0; j<n; j++){
29               line(pointX[n], pointY[n], pointX[j],
        pointY[j]);
30             }
31           }
32          }
```

Analysis of the results implemented by Processing:

- Fast running time without calculating the cost of learning for non-computer oriented practitioners (changes from almost a week to less than 20 min).
- Derivation from graphic language to algorithmic code is still limited, using only effects for transformation and lacking systematic understanding.
- Generative restoration relies on the developer's preset processing of natural language compilations (which can be done with any functionality available)
- It loses its pseudo-artistic texture, but lowers the threshold for the availability of results, while stimulating the democratization of procedural programming
- The art of computer-aided creation also creates the poetry of entropy (from the randomness of the instructions themselves) through "systems."

Creative programming research represented by Processing shows the specific stage in CAD life cycle [8]. So far, generative art has struggled to catch up outside the realm of math and computer science, which have not traditionally been seduced by more adventurous aesthetic obsessions [9]. Processing, as a tool for art in technology to form in an academic environment, takes on the task of talking about the cultural importance of software in an interdisciplinary way.

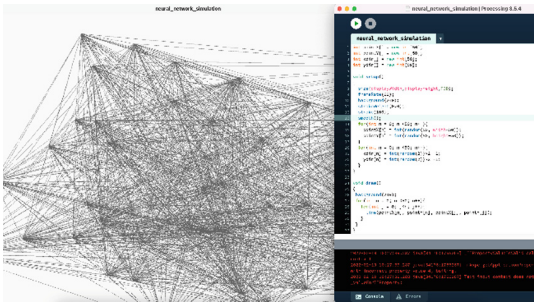

Fig. 1. Realization of the effect of the visual texture in Processing.

Table 1. The application distribution of generative art.

Hot Sector	Application Scenario	Amount of Generative Art
Display and Demonstration	festival activities	225
	performance space	301
	community	29
	park	10
	bars/nightclubs	44
	dining space	12
	theme park	60
	tourist attraction	77
	office	28
	shopping streets	93
	gallery pavilion	342
	homestay hotel	13
	public transport	12
	square	57

4.3 Presented Examples of Creative Programming Design

The author collected data from 20 popular industries across 15 popular scenes on the global art platform MANA. As of February 2022, the number of works with a generative art tag was concentrated in the field of display interaction (342, 68%), and as the number of technology R & D entrants grew, the construction of digital humanities in the metaverse was laid out across many extended scenes.

From the point of view of human's own senses, the amount of information received by vision is much higher than that of hearing. In terms of the form of content information, Graphical User Interface (GUI) mainly consists of pictures and text, relying on vision,

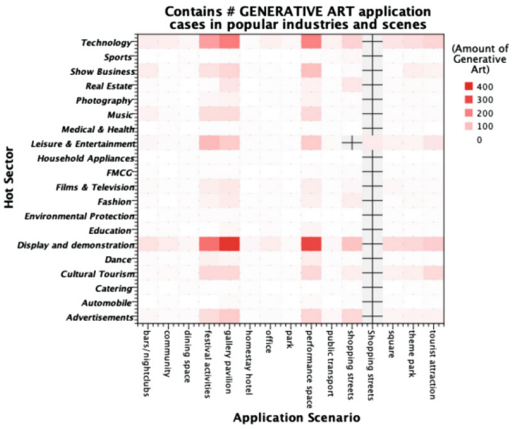

Fig. 2. Application chart of "generative art" tag in popular industries and scenes (data source: MANA as of 2022.2.20, drawn using SPSS)

while Voice User Interface (VUI) mainly consists of Voice and text, relying on hearing. This new creative path is not just about static and interactive visualization, but also exploring more interesting interactive ideas in sensory interactions such as H5.

Currently, the application of generative art design mainly focuses on graphic texture and visual dynamics in commercial projects, but actual generative art can be extended into a possibility in terms of interactive creativity with users. Processing, as a purely software platform, has no direct control over hardware, so with the help of Arduino, it can read the values of various sensors, as well as control a variety of electromechanical devices. It can be extended to a variety of hardware entities, from smart homes to drones to robots, transforming random and regular artistic ideas into commercial design through application [10]. The APP of the intelligent era must be able to enter data in multiple dimensions, realizing the functions of voice recognition, gesture recognition, image recognition and physical environment perception. It can display information from auditory, visual, tactile and holographic multidimensional, making the form of interaction "like human" more perceptual color.

5 Conclusion

Even in the narrow sense of generative art, special research is a subject of both theoretical and practical innovation for people with a single-disciplinary background. Having the ability to reuse code does not mean having a creative programming mindset at the same time, and the ever-changing challenges of digital survival need to be addressed in order to cope with the expansion of business scenarios.

Processing creates a universal meta-language containing a hidden narrative about what code means that people are eager to learn because it enables them to express creatively. At the same time, the boom in technology and the mismatched humanities will lead to changes in this field, which will make it more open to interdisciplinary and other cooperation.

Generative art brings new possibilities for design and new solutions for the rational use of design resources and the operation of the entire design system. This paper is only a basic exploration of generative art today. The relationship between technology and people is not one of dominance and domination. We can use technology tools to bring new breakthroughs to design and innovate the boundaries of design.

References

1. Bailey. J, DOING THINGS WITH COMPUTERS: AN ILLUSTRATED QUARTERLY MAGAZINE. Art in America, 108, 34. [J], 2020
2. Daniel Shiffman.Learning Processing: A Beginner"s Guide to Programming Images, Animation, and Interaction (Morgan Kaufmann Series in Computer Graphics)[M].Morgan Kaufmann.
3. Galanter. P, What is Generative Art? Complexity Theory as a Context for Art Theory[J], 2003
4. Spence, R. (2017). V & A announces plans for shenzhen show. FT.Com, Retrieved from
5. Christian Paul. [Li Zhen, Yan Feng trans. Digital Art: Exploration of Digital Technology and Artistic Concept, original book 3rd edition [M]. Beijing: China Machine Press.2021.
6. Reas. C and Fry. B, Getting Started with Processing[M], 2014.
7. Getting to Know Generative Art and Design, [online] Available: https://open.163.com/newview/movie/free?pid=MFJCOOCRT&mid=MFJCQCA4R
8. Pearson. M, Generative Art:A Practical Guide Using Processing[M], 2011.
9. Kang Z and Zhenbao F, Generative Art and Aesthetic Measurement in Computational Aesthetic[J], 2018.
10. Paprzycka A, The art of programming - Processing as an artistic and culture-forming tool. Stowarzyszenie Czasu Kultury, 2, 36–42[J], 2019.