Adventure of a Simple Circle (in the jungle of my mind)

Evangelina Sousa
Arte & Design Department
Madeira University
Campus da Penteada, Funchal
Madeira - Portugal
E-mail: angelwht@gmail.com , angelss@uma.pt

Abstract

Although the use of computers and graphic software is no longer contested as a valid tool in the creative process of art making, the use of code can still be seen as obstructive of creativity, considered to be the ultimate expression of subjectivity and intuition. This paper is an attempt to demonstrate that code writing is an option for art making, as pliant and expressive as paint and brush. The creative steps, aesthetic choices and experimental processes involved in the development of a piece of code art will be discussed, taking into account that the final product is intended to be a work of visual expression in its own right, representative of the author's aesthetic vision even if in a simplified way. The code delivers a simple geometric pattern, obtained through the progressive modification of the 2D coordinates of each of the points that makes up a circle, creating organic forms that populates the space in a simultaneously ordered and chaotic way. Since the connection with natural environments and the colors of nature play an important role on the author's personal visual language, the improbable connections between natural landscape's light and color and the geometric pattern delivered by this code will be analyzed.

"I would like to paint as the bird sings" *Monet*

It is not natural for the visual artist to analyze the ongoing creative process while developing a piece of art work. Actually, one may think that this analysis would prevent the normal development of the creative action. But it cannot be denied that, due to his/her privileged point of view in relation to an art work in the making, the art practitioner is the ideal observer of the creative process (specially his own).

The creative process has been researched before, in relation with every area of human activity, including art making, but this research is normally undergone by "outsiders", and/or non artists related to the field of art: psychologists, art historians, art critiques. There is sound research done in this area that describes the creative process in general terms, in relation to more conventional fields of art making: painting, sculpture, photography, film, and so on. [1] The model can easily and appropriately be applied to art making using a computer and graphic software. Since most graphic software provides tools and features that are electronic versions of the conventional ones, there is no objection and no doubts about its use for creative production. These tools and means (the material ones and their electronic counterparts) are used and manipulated by every artist in his own way, and this individual manipulation and use of tools, materials and means is one of the factors that gives rise to every artist's personal visual language, his unique visual "words" and "sentences" with which he communicates aesthetic ideas.

But what happens when, instead of these well known and well accepted tools and means, the practitioner uses a piece of code for developing and delivering his/her visual work? The creative process involved is definitely different. The actual code can be seen as a tool and/or a means. But is it as flexible, intuitive and as suitable for this end as other more conventional tools? Can visual artists actually develop a unique and distinctive language for visual expression when using code to create their work? By analyzing

the creative process followed in the development of a piece of work, I will try and come up with some ideas that might contribute to possible answers to these questions.

The starting point for this piece is a little Processing programming exercise found in a book [3]. The code (original code from the authors of the book) delivers the x and y coordinates of successive points in a circle, by calculating the sine and cosine values of an angle that is incremented by small amounts from 0 to 360 degrees. Since the sine and cosine values delivered by the functions **sin ( )** and **cos ( )** are situated between -1.0 and 1.0, a value for the radius of the circle is added to these values, as well as the coordinates of the center of the circle.

The running code produces an animated image. On **figure 1**, we can see the result of this initial code, as well as one of the many results of my final code developed from this initial one, after a journey that I will now share with you.

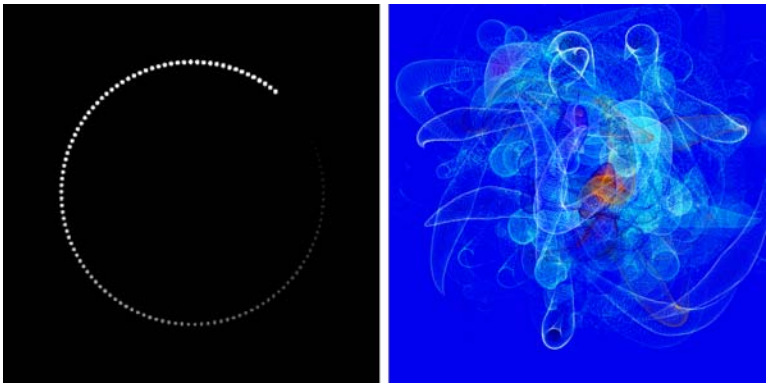

Figure 1: *The beginning and possible end of an aesthetic adventure*

This exercise was part of (another…) learning session on Processing programming language, and as a continuation, some variation to the radius of the circle was introduced. It was at this point that the potentialities of the resultant shape start showing up. From this point onwards, my own code developed from the original one, in a natural and smooth way. It was as when we see the potentialities of a scribble on a wall and follow the impulse to develop the ideas it rises on our mind. (**Figure 2**) The fact that I was using code instead of a pencil or a brush didn't come up as an obstacle.

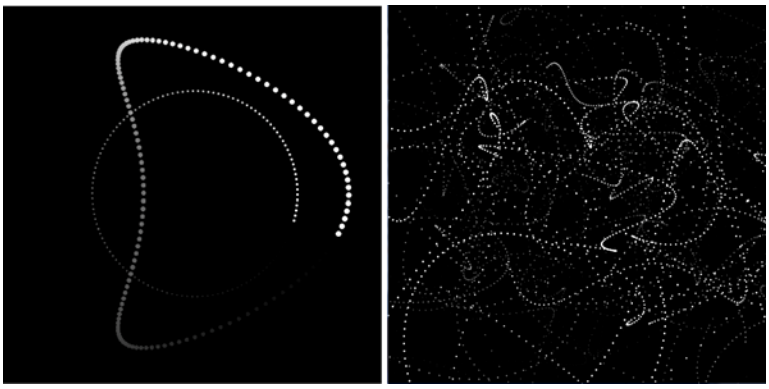

Figure 2: *A scribble, and what came from it*

The potential range of possibilities for the next step was almost infinite. So, how could I know what to do next? What were the criteria leading me on my aesthetic choices? In my opinion, it is our own personal aesthetic vision that leads us in a certain direction. Different people with different aesthetic

pursuits would take different decisions leading to the concretization of their own visions. Before sunlit landscapes, Monet took decisions about how to approach composition, the use of color, the interpretation of light, the actual brush stroke, according to an intimate and very personal aesthetic vision. Before the same landscape, Cézanne would take a different, but most certainly, equally valid approach. Our particular aesthetic reasons and choices play an important part on the development of our personal visual language.

The next step was to introduce interactivity, allowing people to take an active part on the making of the animated and ever changing image. The x and y coordinates of the many partial and randomly deformed circles were now determined by the position of the mouse pointer. Also, the radiuses of the little dots that make up each circle were increased by a random amount, and color was added. Different possible ways were experimented, namely, keeping the dots as circles with no filling, or making them solid with some transparency. **Figure 3** shows two of the most successful attempts.

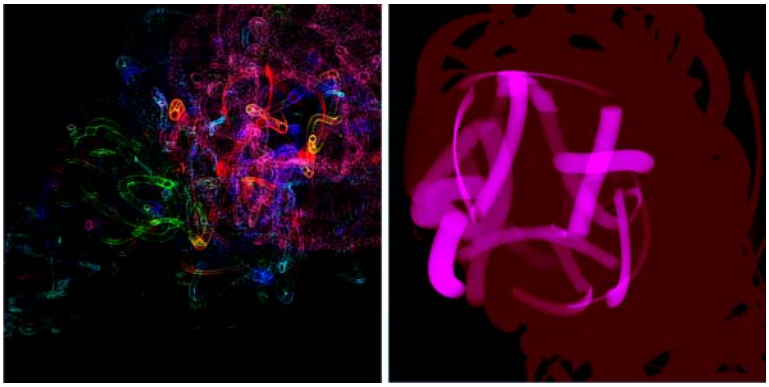

Figure 3: *Not all the roads lead to Rome*

I was reaching a point where pictorial qualities were showing up on the generated pictures. I took the decision of exploring this way a bit further. Many attempts were made, following many “subways”. Once again, the code proved to be pliant and responsive. Were there any specific criteria leading my choices? Yes, but they were very subjective and intuitive. I would go in the direction that makes my moving pictures to look “right” to me, in terms of form, color, and composition over time. Now, I am aware that the actual elements that make a picture look right to me can have the opposite effect on other people. This means that other practitioners at the same point of the art work would take different decisions, and therefore, produce different results. This reinforces my idea that this subjective, intuitive and very personal criterion of choice is an important component on the construction of one’s personal and unique language for artistic expression.

Now my work was at the level of a potentially finished work, even if I rarely consider a piece of work definitely finished. But there were still two non satisfactory points that I had to address: each deformed circle had a constant thickness from start to end. This thickness would vary randomly only from one circle to another, and this would make the resultant pictures somewhat rigid and mechanical. I felt the need for some organic qualities that would bring up the feeling of nature and natural environments, so important in all my art projects. Also, the color randomly defined for each element of the composition was giving it a very decorative look that I definitely wanted to avoid.

The answer to the first problem was simple. I changed the thickness of the beginning and end for each element, giving it a very organic, worm like shape. The color problem was a bit more complicated to solve. After some attempts at manipulating the RGB values, trying to condition the generation of random colors, I ended up finding out a very effective solution: the code picks up the color of a random pixel of a previously prepared image of a natural landscape (or a suitable detail of it). This ensures that, if I am happy with the colors in the landscape picture, I am happy with the colors delivered by my code. Also, by changing the landscape picture we can change the color, light and general feeling of the resultant work. In Figure 4 we can see the very pictorial but (too much) decorative and mechanical result of the code before

the changes, and the result of using the color palette of a landscape image and a more organic construction. The rightmost picture is the color source.

The final piece is animated and interactive, running continuously on a computer screen, controlled at some extent by the observer. The different aspects that can be assumed by the ever changing image can be controlled by the right and left mouse buttons, as well as by moving the mouse without pressing any button. The final result presented here is one of the many possible aesthetic solutions, and this is one of the most fascinating features of a piece of code art: it is never really finished…

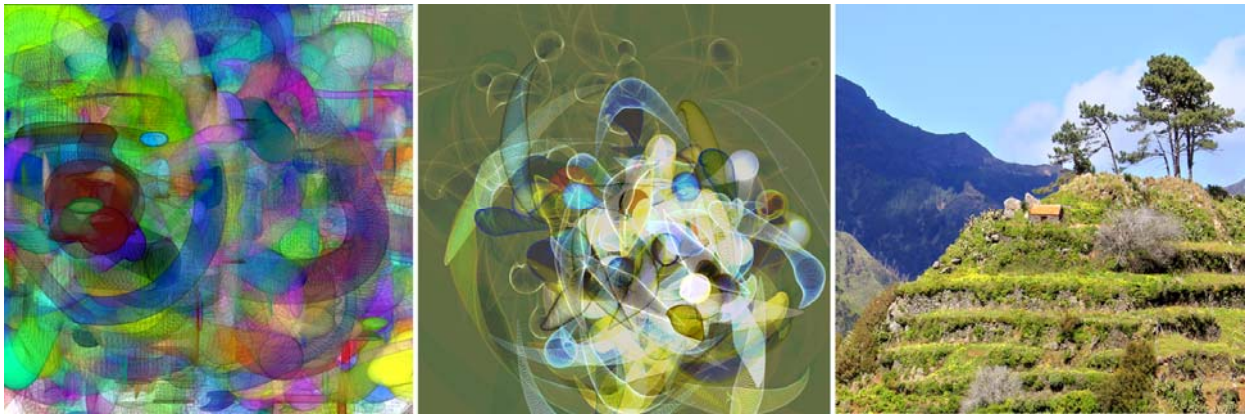

Figure 4: *The end is just a new beginning…*

In his 1974 paper "Computer film as Film Art", [2] Malcolm Le Grice states that "The need to produce a program as a means to achieving a work of art ***imposes one very significant process on the artist, that of some kind of analysis of the component factors of his image (or output)***, plus an analysis of the kind of principles by which these components are brought together." (the emphasis is mine).
Since 1974, both software and hardware evolved enormously. Commercial image processing software is now readily available and used by visual artists in a daily basis. This means that the use of code is not a necessary "need" anymore, "imposing" a specific process of work to the artist, as suggested by Le Grice. It is a choice, a very determined and conscious choice, aiming at the exploration of specific visual results. It is, definitely, my prime choice to conduct my aesthetic research project, even if I continue to develop visual work with commercial software. The two are like the two wings of a bird: none can be discarded if the experimentation is to be complete. It is true that the methods of art making with code are different and unique, maybe more rational, and implies a systematic learning process of the logics and techniques of code writing. But doesn't the use of a pencil or a brush imply an also difficult and systematic learning process? Please ask any art student…

I would like to write code as Monet used to paint. My aim is to write code, not as a computer scientist or an engineer, but as a visual artist in love with the sensitive logics of nature.

Reference

[1] M. A. Mace, *Modeling the Creative Process: A Grounded Theory Analysis of Creativity in the Domain of Art Making,* University of Canterbury, Psychology, 1998.

[2] M. Le Grice, *Computer Film as Film Art,* in *Computer Animation,* ed. John Hallas, Focal Press, 1974

[3] C. Reas, B. Fry, *Processing: a programming handbook for visual designers and artists,* The MIT Press, Cambridge, Massachusetts; London, England, 2007