

# Exploring Randomness in Digital Art

Lali Barrière

Departament de Matemàtiques, Univ. Politècnica de Catalunya, Catalonia; lali.barriere@upc.edu

## Abstract

Since the beginning of the twentieth century, randomness is one of the key pieces that constitute the language of art. The use of randomness is an essential characteristic of computer art, although other artists, as the Dadaists and the conceptual artists, also made use of chance in their creative processes. In this workshop we will explore some ways of using randomness in digital visual art, by means of provided examples. As a result of our exploration, we will use random numbers to generate a still image or an animation in Processing, a programming language for artists and designers.

## Some practical issues

In this workshop we will explore the use of randomness in digital art. Its final goal is to create an image or an animation involving random numbers using the Processing programming language. The workshop is open to anyone interested in creative programming and the applications of randomness to art, however we think it is worth mentioning some practical questions:

- It might create a good dynamics to work in pairs in order to share ideas and exchange points of view. Working in pairs will allow people with no programming skills to attend and contribute to the workshop. To make easy to find a partner with a complementary background, we will provide a list to sign in before the workshop.
- As it may seem obvious, a computer with Processing installed is needed to do the practice in this workshop. Only one computer per pair is needed. Processing is available for Mac OS, Linux, and Windows platforms and freely downloadable from <https://www.processing.org/download/>.
- For those who don't know how Processing works, the video [12] is a good, very short introduction.

## Randomness, Unpredictability, and Creative Processes

Randomness plays an important role in art since the beginning of the twentieth century. The Dadaists, with their cut-up technique, are the most known example. Conceptual artists also used chance in one way or another. For example, Sol LeWitt wrote “lines not short, not straight, crossing and touching, drawn at random, using four colors, uniformly disposed with maximum density, covering the entire surface of the wall” as the set of instructions for his *Wall Drawing #69* (1971). Another example is Jackson Pollock's painting process, which involves randomness in a very different way. Indeed, as he said, his dripping technique is controlled and far from accidental, although not totally deterministic.

Nevertheless, it is in digital art that randomness is essential both conceptually and technically. Max Bense, a philosopher that influenced the first generation of computer artists, linked the creative process to unpredictable intuitive decisions and proposed a generative aesthetics that is a combination of mathematical rules or structures with the use of probability [2]. Frieder Nake, a famous computer art pioneer, wrote what he called a “late manifesto on computer art” [8]. Among other interesting paragraphs, number 26 is devoted to randomness as a fundamental aspect of computer art. In [9], Nake analyses several artworks, showing how diverse can be the use of randomness in computer art. Other interesting papers discuss different meanings

of chance relatively to other concepts such as probability, combination, chaos, disorder, control, and also the capacity of randomness to generate structure [3, 4], and address the importance of randomness in generative art [7].

The main ideas about how randomness affects the artistic practices are:

- The use of randomness changes the production process from totally planned to a more accidental one. Randomness allows the artists to delegate some of their intuitive, subjective decisions to a process that they cannot control. The degree of accidentality is a decision of the artist.
- In terms of artwork production, randomness is related with seriality and repetition. A generative process together with the huge possibilities of combinatorics result in a class of work, instead of an individual artwork. The need of choosing among all the possible results involves randomness in many diverse ways.
- Chance, unpredictability, variation, repetition, chaos, noise, the tension between order and disorder, are concepts related with randomness. Generative art incorporates these concepts through a large set of mathematical techniques and methods.

## The Workshop

In this workshop we will use Processing, a programming language and environment for artists, to explore a variety of elementary ways of using randomness in digital visual art. The workshop will be carried out through examples, which will help us to understand several different uses of random numbers in digital art, together with the need for control when using randomness in an artwork. Due to obvious time limitations, we will focus only on the exploration of several different uses of the `random()` function, a Processing function that comes from the pseudo-random number generator available in the Math Java library.

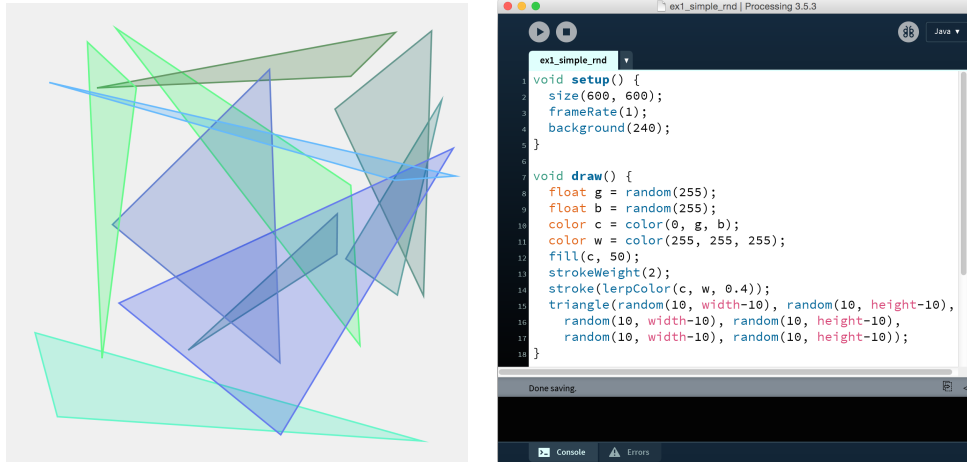
The participants will learn four different strategies that involve the use of random numbers. Those strategies will be introduced through examples in increasing order of conceptual complexity and technical difficulty. In the conceptual aspect, we will turn our attention to how randomness is used, which are the advantages, and how could pure randomness be improved or changed to obtain other results. After the presentation of the examples, the attendees will choose one of the examples or come out with their own idea in order to build a new Processing image or animation.

The examples and further codes will be built in Processing [10], a free and open source programming language and development environment for creative coding of stills, animations, and interactions. It runs as a library of the Java programming language, and both the language and the environment allow the user to avoid the complexity of the usual programming environments. Processing has been used in Bridges workshops in two past editions [1, 11]. The Processing web page [10] is a great source of information, including some introductory tutorials. For those interested in a deeper knowledge of Processing, the handbook [5] by Reas and Fry and the book [13] by Shiffmann, which devotes one of the chapters to the use of randomness, are both good choices. All the needed codes will be available before and during the conference at <http://www.lalibARRIERE.net/bridges2019/bridges2019.zip>. Beside the code examples, a Processing *cheat sheet* will also be provided, including the basic Processing instructions together with some remarks about the use of the `random()` function.

## Examples

### *Simple Randomness*

In our first example we will play with totally random figures. Indeed, Figure 1 shows a composition of random triangles, i.e., triangles with vertices in random position, with random colors chosen in a range between



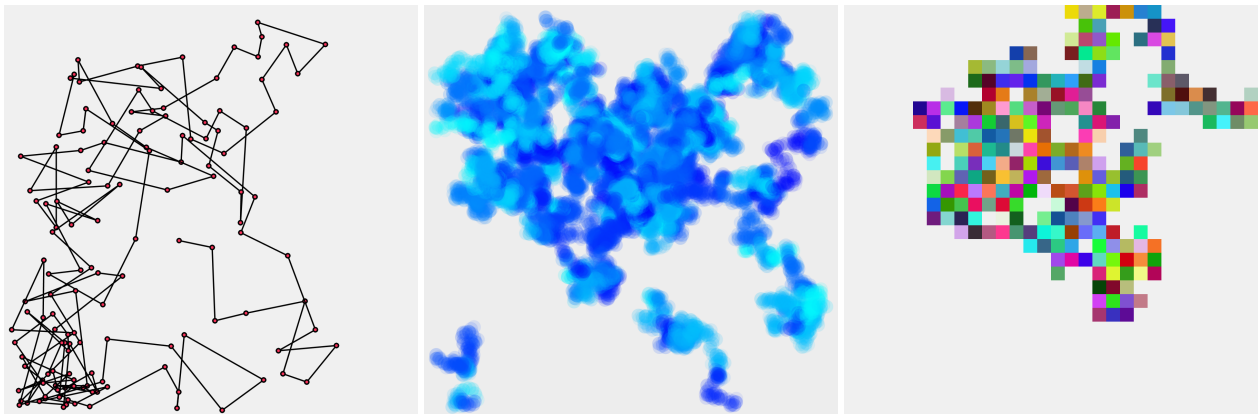
**Figure 1:** *Some random triangles.*

green and blue. It also shows the corresponding Processing code. Although the code that accompanies the drawing is short and simple, we will dedicate some time to explain it and to show different ways to modify it to get different results: how to computationally deal with color; which other random figures, such as lines, polygons, circles, can give aesthetically appealing results; and also which parameters, such as the color or the weight of the stroke, can be randomized.

Because of its simplicity, this example is a good starting point and it is worth investigating. However we will notice that totally random shapes are difficult to control.

### ***Random Walk***

A random walk is a dynamic process easily defined as a walk with random steps. Random walks constitute one of the random processes or structures that have been well-studied from both scientific and artistic points of view. In their still or animated versions, there is a really wide range of ways to generate them and also to visually represent them. Figure 2 shows three versions of a random walk. Their distinct visual appearance depends on some graphic parameters, but also on how the steps are selected. We will compare them and

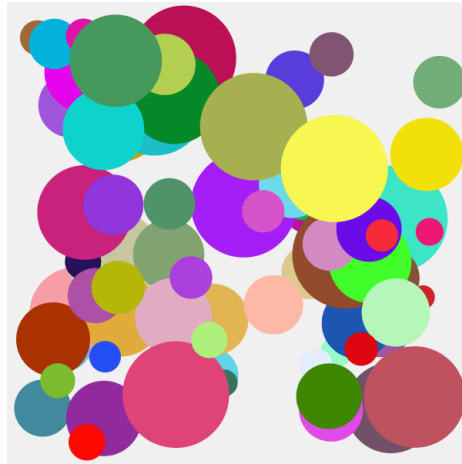


**Figure 2:** *Three different ways to generate and visualize a random walk.*

explain which parameters can we tweak or change in order to get different versions. We will also show some more abstract ways of representing random walks.

### ***Random Initialization***

The example shown in Figure 3 is a frame from an animation with many balls, each of them with random size, color, initial position, and velocity. This example creates an opportunity to discuss how to use the `random()` function to control some parameters of the animation. For instance, in this case, the initial positions are selected uniformly at random in a square, the animation frame, while the velocities are selected uniformly at random in a circle, i.e., a random speed in some range, with random direction. There are



**Figure 3:** *Many balls with random position, velocity, size and color.*

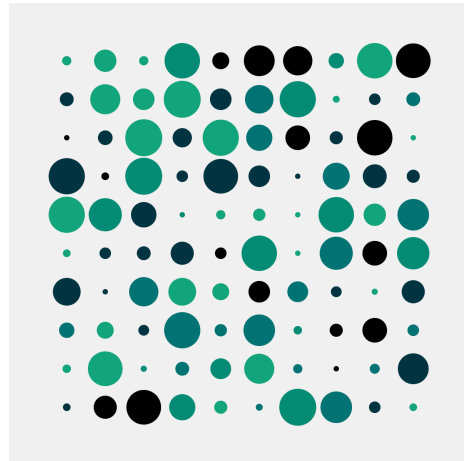
many other distributions to work with: discrete probabilities; random points in a rectangle with probability increasing with the distance to one of the sides; random points in a circle with probability increasing with the proximity to the center or, inversely, with probability increasing with the proximity to the border; the Gaussian distribution.

### ***Interplay Between Order and Disorder***

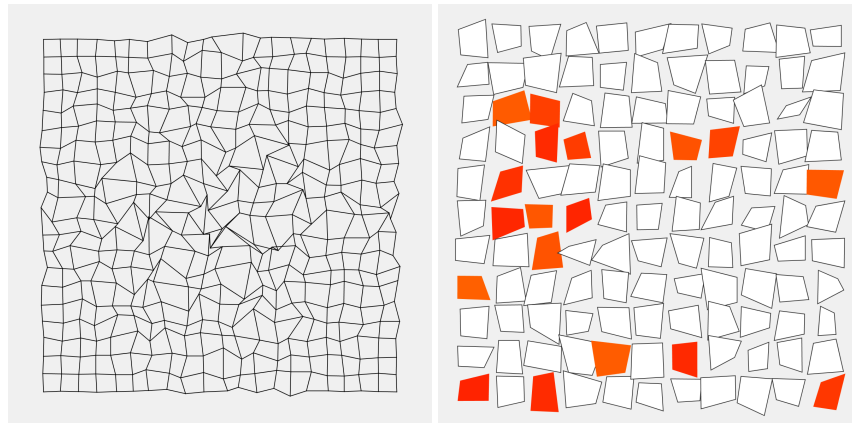
The last four examples are built accordingly to a simple rule: to merge the grid structure with some randomness. Among all the possible structures, the grid seems to be the most suitable to our objectives, because it is very robust, it gives a strong feeling of order, which will be broken by the application of random distortion, and it is not very difficult to program. During the workshop, we will show some examples and give some ideas for working with other combinatorial or geometric structures. Figure 4 shows a set of circles centered at the points of a regular grid, each of which has been assigned a random green color and random size. The code produces an animation, and we will explore how we can use the `random()` function in the parameters that define the motion. For example, we can use random period and phase.

The same strategy is applied in the examples shown in Figures 5 and 6, with an important difference: here randomness affects the structure, introducing disorder in it. In Figure 5, the underlying grid structure has been distorted or destroyed by means of some random displacement of points. For instance, in the right example, at every step of the animation, each point of the grid is translated by a vector. The magnitude of this vector is inversely proportional to the distance of the point to the center of the grid.

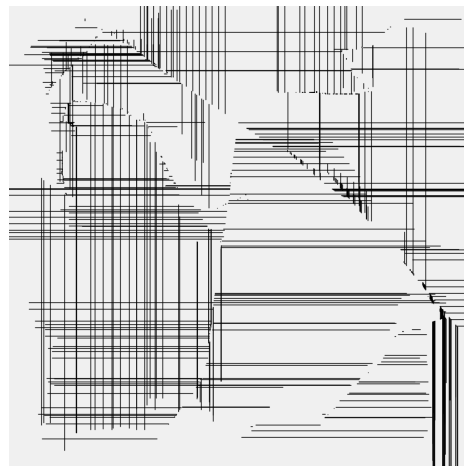
In Figure 6, the vertical and horizontal lines are broken following interaction with the user via the mouse. Then, some small random displacement is applied to its parts. This is the most involved code, and it is suitable for those participants with a more advanced programming background.



**Figure 4:** *Introducing some random values in a grid structure.*



**Figure 5:** *Two different ways to distort a grid, using randomness.*



**Figure 6:** *Distorting a grid with some randomness and some mouse interaction. (Thanks to Anders Hoff, a.k.a. inconvergent [6] for the initial code of this example.)*

## Summary and Conclusions

The use of randomness is a key feature of generative art. We propose a workshop to explore at a basic level different ways of using random numbers in digital art. The workshop is carried out through some provided code examples, in Processing. The participants will choose one of the examples or come out with a new idea in order to built their own Processing animation.

Beside the code examples, a *cheat sheet* will also be provided to help to program in Processing.

Our proposal is open to anyone interested in digital art and the use of randomness. Participants can work in complementary pairs, so that those with no programming knowledge can contribute with their ideas to the development of the workshop.

Due to time limitation, some interesting issues and techniques regarding the use of randomness in generative art will not be explored. Some of them are: Perlin noise, a deterministic function that appears to generate organic randomness; complex structures involving computational geometry, combinatorics or algebra; advanced probabilistic techniques, such as Markov chains, stochastic processes; randomness as part of evolutionary systems. All of these concepts and techniques have sometime been used by artists.

## References

- [1] L. Barrière. “Lissajous Curves: an Experiment in Creative Coding.” *Bridges Conference Proceedings*, Baltimore, Maryland, USA, Jul. 29 – August 1, 2015, pp. 549–554. <http://archive.bridgesmathart.org/2015/bridges2015-549.html>
- [2] M. Bense. “The Projects of Generative Aesthetics”. In *Cybernetics, Art and Ideas*, Jasia Reichard, ed., 1971.
- [3] M.-P. Corcuff. “Chance and Generativity.” *Proceedings of the 11th Generative Art Conference*, pp. 189–199, 2008.
- [4] M.-P. Corcuff, R. Munch, and G. Rannou. “Randomness, Dis(order), and Generativity.” *Proceedings of the 17th Generative Art Conference*, pp. 230–248, 2014.
- [5] B. Fry and C. Reas. *Processing: A Programming Handbook for Visual Designers*, 2nd ed. (2014), The MIT Press.
- [6] Inconvergent web site. <http://inconvergent.net> (as of February 27, 2019).
- [7] M. Kostner, F. Schubert, and C.-B. Schönlieb. “Chaos, Noise, Randomness and Coincidence as Constitutional for Generative Art.” *Bridges Conference Proceedings*, Pecs, Hungary, Jul. 24–28, 2010, pp. 467–470. <http://archive.bridgesmathart.org/2010/bridges2010-467.html>
- [8] F. Nake. “Paragraphs on Computer Art, Past and Present.” In *Cat 2010: Ideas Before Their Time: Connecting the Past and Present in Computer Art*, 2010, pp. 55–63.
- [9] F. Nake. “The Disappearing Masterpiece. Digital Image & Algorithmic Revolution.” *Proceedings of the Fourth Conference on Computation, Communication, Aesthetics & X*, 2016, pp. 12–27.
- [10] Processing web site. <http://www.processing.org> (as of February 27, 2019).
- [11] Z. Ruttkay, “Programming for Artists with Processing.” *Bridges Conference Proceedings*, Pecs, Hungary, Jul. 24–28, 2010, pp. 555–558. <http://archive.bridgesmathart.org/2010/bridges2010-555.html>
- [12] D. Shiffman. Web tutorial “Hello Processing”. <https://hello.processing.org> (as of February 27, 2019).
- [13] D. Shiffman. *The Nature of Code: Simulating Natural Systems with Processing*. The Magic Book Project, Shannon Fry, ed., 2012. <https://natureofcode.com/book/>