

# The Art of Programming: Processing as an Artistic and Culture- -Forming Tool

**Anna Paprzycka**

Faculty of Polish and Classical Philology  
at Adam Mickiewicz University in Poznań

Even before the 21st century began, Friedrich Kittler, a German media theoretician and literary scholar, had said that in present times, students ought to know at least two programming languages to be able to talk about culture [Software 20]. In the early 21st century, Lev Manovich, a media researcher of Russian origin, attempted to define the notion of “new media” in his book *The Language of New Media*. He pointed out that the humanities needed a new line of research – one that would focus on the cultural meaning of software. In this publication, Manovich wrote:

*“To understand the logic of new media, we need to turn to computer science. It is there that we may expect to find new terms, categories, and operations that characterize media that became programmable. From media studies we move to something that can be called ‘software studies’—from media theory to software theory” [Language 48].*

In 2007, Manovich and Matthew Fuller jointly founded Software Studies Initiative, which currently has two branches: in New York and La Jolla, California. This research center connects researchers who aim to conduct a critical discourse about software – its meaning, possibilities, threats, and role in the digital culture. Obviously, software studies are no longer ignored by cultural researchers, and a scholarly trend that focuses on this very issue has successively grown within the humanities. This article is devoted to the programming language called Processing, which is only one element from a range of possibilities available to artists who want to combine programming with art. However, it is indeed an exceptional tool that may be an exit point to an analysis of the meaning of software studies in culture. This text aims to provide a survey of the extent of its influence, considering a variety of fields, such as popular culture, art, and science.

Processing is comprised of a programming language and its integrated programming environment (IDE).<sup>1</sup> It was

---

1 IDE (Integrated Development Environment) is an application or a complex of applications designed for writing and running source code.

created in 2001 by two MIT Media Lab students: Ben Fry and Casey Reas. This project built on the idea explored in the programming language Design By Numbers (DBN). Its author was John Meada, the supervisor of the initiative Aesthetic and Computation Research Group (operating under MIT Media Lab) whose members also included the subsequent authors of Processing. Design by Numbers was completed and made public in 1999. Its core assumption was to create a programming language that could be used by people without specialist skills. The DBN programming language is simple because it employs a clear system, based on manipulating variables and formulating conditional statements. John Meada believed that raising the quality of media art and design hinged on improving the programming curricula in art and technology schools. Hence, DBN was meant to be user-friendly and free, and by extension, widely available financially and intellectually.

Processing, created by Meada's students, was an ideological and technological continuation of the DBN project. Its IDE was based on a similar interface. The authors of Processing also designed it to be a publicly available and user-friendly tool for creating visuals. For this reason, the program is open-source based and available to anyone for download. Processing was based on the Java language and uses its simplified version. The authors wanted to ensure that anyone could potentially familiarize themselves with software, which typically appears to be complicated and

off-putting.<sup>2</sup> To use this tool, the user is not required to have advanced knowledge in mathematics, although, naturally, mathematical literacy is an additional asset. Processing was primarily addressed to artists and was meant to help them think like engineers, and to engineers, whom, in turn, it was supposed to inspire to artistic activity. It was a pioneer endeavor because it was one of the first creative tools designed for a universal application, based on a classic programming language syntax.

### **Creative communities**

Processing developed rapidly and gained popularity across the world. A number of textbooks have been written by the authors of Processing (*Processing: A Programming Handbook for Visual Designers*, *Make: Getting Started with Processing*, *Visualizing Data*) as well as its enthusiasts. The idea of creative coding by means of Processing has spread largely thanks to Daniel Shiffman, a mathematician, philosopher, and programmer, and the author of two textbooks for learning this language (*Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction*, and *The Nature of Code*). He is active online under the handle The Coding Train and has a YouTube channel, where he posts videos with interesting lessons. Their attractive and clear layout makes the lessons very popular. Daniel Shiffman is also the director of The Processing Foundation, whose goal is to develop competenc-

---

<sup>2</sup> An online course created by Daniel Shiffman (available on <https://hello.processing.org/>) shows that you only need 40 minutes to program your first interactive visualization using Processing.

es in this tool across the world, in all environments. More than improving the program and expanding its libraries, the Foundation is more invested in nurturing the community which organized around it across the world. Shiffman explained:

*“Processing, for me, has always been just the most wonderful thing ever. It’s given me a mission and a passion, to bring computation to everyone: artists, designers, musicians, biologists, doctors, dancers, animators, bankers, photographers, librarians, fashion designers, architects, psychologists, journalists, and writers, just to name a few. Writing code can be scary, something many mistakenly think is reserved for computer scientists and engineers. Processing has helped eliminate that fear, making programming accessible to a wider audience, particularly artists”*  
[“Introduction” IX].

The mission of the foundation is to spread awareness of coding, and to support those individuals for whom the software is unavailable for a variety of reasons (including gender, age, occupation, qualifications, social class, sexuality, etc.). Its members think about programming in terms of a medium, a communications tool and a method of expressing one’s way of thinking and creating. For this reason, they advocate for keeping it open, publicly available, and free.

Processing is distributed under an open-source license, which means that its users have the right to use it free of charge, including copying, modifying, and developing it. While initially, it was only the project's authors who worked on its expansion, now a large, continually evolving virtual community is invested in its development as well.

In the past, if you were interested in the program and how it worked, you could e-mail the authors your questions. Today, Fry's official website requests that you do not e-mail him directly as there is no shortage of websites where users can help one another. Since the tool was available for download free of charge, creative people from all over the world have also helped to develop it for free. Over time, Processing has been implemented in other programming languages, such as Javascript, Python or Ruby.

One example of a website where the enthusiasts of this language inspire each other and share their code is the Open Processing Forum.<sup>3</sup> Notably, the entire development of Processing has been propelled by the activity of its users and contributions that they have made to this language. A programming language develops along with the expansion of its creative users that gather around it and communicate with each other on the internet. Hence, for Processing to exist and evolve, it needs to be co-created by a community growing around it.

---

3 See: *Open Processing*. <https://www.openprocessing.org>.

An interesting example of communication between Processing's users is a video clip to the song *House of Cards*, created for the band Radiohead by Aaron Koblin, one of Casey Reas' students. The programmer made the animation entirely by visualizing the data obtained thanks to a 3D scanner that collects information about the shapes and distances of objects in space. In the description to the clip, the viewers were encouraged to create their own visualizations using the code and instructions provided by Koblin, and then to share the effect with others. The result was a number of alternative, fan-generated versions of the video clip.

Processing is also used with commercial projects available on the internet. In 2007, Jessica Rosenkratz, an MIT graduate, and Jesse Louis-Rosenberg, a mathematician, started a platform where you can design and order bionics-inspired jewelry.<sup>4</sup> Mary Huang, an artist from California, used Processing to make software for designing code-generated clothes with an eye towards the democratization of made-to-measure tailoring, which usually tends to be a luxurious service.<sup>5</sup>

The users of Processing exchange the software they have created via the internet, which in this case, may be treated as a speech act, a communique, and a way of transferring information to another person. Geoff Cox observed that each speech act conducted in the public space is political

---

4 See: *Nervous System*. <https://n-e-r-v-o-u-s.com>.

5 See: *Mary Huang*. <http://www.rhymeandreasoncreative.com/portfolio/index.php?project=continuum>.

in its nature because it happens in front of the society. This way, the researcher pointed to the political character of sharing code online [Cox and McLean 13–14]. Software written in Processing can also be described as a political act of speech which deals with the democratization of the code and its transference to anybody who wants to drive an independent, conscious narrative based on digital and creative instruments. Therefore, considering its political nature alone, Processing is an attractive reference point to the humanities. Computing code may be viewed as another language molded in the cybernetic culture. As Cox highlighted it, this also means that it is not a universal and invisible cement of the world but it is political and contains specific ideology according to which it shapes the reality. Even the selection of the programming language used for creative processes is a political decision because what you essentially choose is an ideology, in the presence of which you act.<sup>6</sup>

### **The impact on other creative tools**

Aside from the fact that Processing is evolving by itself, it also has an impact on other creative tools. This applies, for instance, to Arduino, which was launched in 2005. This name refers to microcontrollers based on easy-to-use hardware and software. Before the emergence of microcontrollers, building artistic installations required placing an entire computer in its vicinity while main-

---

<sup>6</sup> Interestingly, the dispute whether a system under which an artwork is created is objective or not had existed in art even before the popularization of computers. Most conceptualists (including Sol LeWitt) and systemic artists believed that it was universal and devoid of subjectivity. The art historian Grzegorz Dziamski, on the other hand, stressed that even the decision about which system to apply in your work is marked by subjectivity [71].

taining the functionality and aesthetics of the composition. Arduino makes it possible to build small subsets on boards and create portable and extremely practical microcontrollers. It allows for using the individual functions of a computer without the need for having the whole machine around the installation. First, you use Arduino IDE, which is based on the IDE developed for Processing, to write the software which controls the entire subset. Next, the software is transferred onto Arduino boards. Arduino microcontrollers can be combined rather easily with the code written in Processing thanks to libraries that have been created for that purpose. These phenomena are related – Arduino may be considered as an extension of the idea of Processing. The difference between them is that the creativity linked to Processing concentrates around writing the software, while Arduino's users are primarily focused on building physical subsets.

Geoff Cox believed that indeterminism was a feature of software. Invoking the speech act theory, introduced by John Langshaw Austin, Cox referred to the programming code as a self-fulfilling prophecy [Cox and McLean X]. This means that it determines reality according to the information contained within it. From this perspective, we might say that software conditions the shape of the future by containing the data about it within itself. As it is evidenced by Processing, programming languages also shape the subsequent digital tools, hence, they have a direct impact on how their users can express themselves.

### **Processing as an artistic tool**

In 2003, Ars Electronica, a worldwide festival of new art media, featured two conference talks by Casey Reas, one of the makers of Processing. Moreover, a special part of the exhibition was reserved only for the works created with Processing. At that time, this programming language had not been as popular yet, hence, initially, the works produced with the program had to be presented collectively to first draw attention to the phenomenon. In 2005, Ben Fry and Casey Reas were the laureates of The Prix Ars Electronica; they received the Golden Nica Award in recognition for their work. Thanks to these events, Processing gained popularity in the field of art. Both authors used the program to create works presented in galleries, thus, contributing to the dissemination of this tool among artists.

Fry's works are a series of experiments devoted to the presentation of various types of information. Typically, the artist tackles difficult data that is intangible due to its complexity or the enormous size of the database. One of his first projects was *Valence* (1999–2002), in which he illustrated the content of Mark Twain's novel *The Innocents Abroad* using dynamic, three-dimensional visualization. The program written by Fry reads Twain's novel linearly, sentence by sentence, and dynamically adds each successive word to the three-dimensional space, combining all the words in the novel into a relational web. A beam connects each word to others as they develop a relationship. The entire form resembles

a globe of data. It is a spatial structure that changes over time as subsequent elements of the novel emerge. The word-components of this network construction impact one another but also keep defining their own place in the whole composition. Fry assumed that the best way of presenting and analyzing enormous amounts of data was to use a method of visualization that will accentuate the tendencies for data organization (trends) and the anomalies occurring within the database. However, as the author of these works observed himself, this example of a data analysis, which is simply a collection of words comprising a book, is not very practical because this data continues to be a collection of words, and studying them does not contribute beyond the calculation of repeating words. Consequently, in later years, the artist improved on the Valence system and used it to illustrate various types of information, such as activity on websites, or to depict the human genome, which he achieved in his 2002 project *Genome Valence*. This piece was a visual representation of the algorithm (called BLAST) most commonly used for DNA analyses. Fry's project was exhibited in a gallery space as an artwork but it was also applied by scientists in their work. Moreover, it has entered the cultural circuit and appeared in the movie *Hulk* from 2003. One of the last scenes shows the protagonist, Betty Ross (played by Jennifer Connelly), analyzing samples on a computer screen in her laboratory. You can spot the name of the program *Valence* and the distinctive spherical visualization.

Fry's work is multidimensional and cannot be viewed unambiguously as an artistic endeavor. Instead, it functions simultaneously on various cultural planes, in the space of art, science, and public social reception. It is a perfect illustration of how deeply one programming language can be immersed in a culture.

Another example of using Processing for artistic purposes is the work *Unnamed Sound-Sculpture* (2012) by Daniel Franke and Cedric Kiefer. This sound sculpture is an expression of how visual dramaturgy can be built based on processed data. The visualization showed the data downloaded from three Kinect controllers that scanned the body of a dancer in motion. The program written in Processing transformed the analyzed silhouette into a moving, scattering cloud of points. The combination of Processing's programming language with Kinect controllers was possible only thanks to the library written by Daniel Schiffman, expanding the possibilities of the language.

All the design disciplines, including architecture, are facing a new challenge created by the possibility of generating infinitely dense information. Computing architecture is based on forwarding various design decisions to software. Michael Hansmeyer is an artist who uses Processing to create detailed architectural forms. He designs his patterns in Processing, prints them fragmentarily using 3D technology and assembles them. The artist is trying to ask if we can produce more than we are

able to design. Can design keep up with the possibilities offered by the rapidly advancing technology?

### **Processing and the humanities**

The basic problem, pointed out by the members of the Software Studies Initiative, was the lack of a suitable, universal language that could be used to talk about software and would allow for promoting its cultural meaning in a way that was comprehensible for any scientific field. The researchers involved in the initiative founded by Manovich have tried to develop this kind of language via the cultural analysis of the software included in their textbooks. Most of them have been published online, thanks to which their contents can quickly reach a wider readership than would have been the case if they had been printed in an analog form. The target audience of these books were the people connected to academia. The creators of Processing have also aimed at creating a language that would be clear for everyone. Fry and Reas have managed to attain their goal. Processing's programming language is a meta-narrative about software and its meaning in the culture while also being incredibly universal. Processing is, therefore, an example of a critical and educational discourse operating within the dominant consumerist era. It has developed rapidly, engaged a very wide and diverse spectrum of users, and additionally, turned out to be easily accessible. Considering the hundreds of thousands of views of Daniel Shiffman's clips on YouTube, it may be concluded that more people have heard about Processing than about the Software Studies Initiative. Therefore, from the perspec-

tive of the humanities, Processing's important feature stems from the fact that similarly to scientific theories, it was shaped in the academic environment. However, it has been much more effective at producing a universal, interdisciplinary way of speaking about the cultural significance of software. Processing has created a universal meta-language containing a hidden narrative about the meaning of the code which people have been eager to learn because it has allowed them to express themselves creatively. However, it is noteworthy that the culture generally ignores the presence of programming languages. Even in the area of science, in the description of artistic works that are meant to indicate the materials and technique used to produce a piece, such as "oil on canvas", it is extremely rare that programming languages are listed. In digital works, most often, a general phrase "custom software" is used, which means unspecified popular software. It clearly illustrates that culturally-speaking, we are not yet aware or sensitive to the variations of programming languages and their significance.

Humanists should commit themselves to a reflection on software in their research because they are responsible for generating the critical tools that allow for analyzing cultural phenomena which directly impact human beings existing in the cultural system. Today, the digital culture is not short on various strategies for influencing people that often reduce them to the role of passive consumers of mass-produced contents. This is why we need humanities researchers to formulate theories and intro-

duce a critical discourse on software and its significance. Regrettably, technological advancement is much more dynamic than the advancement of humanities. In the face of this crisis, open-source tools, such as Processing, prove useful. The turning point linked to rapid technological development and humanities lagging behind may lead to changes in the latter, making them more interdisciplinary and willing to collaborate with other fields. In a survey conducted for the magazine “Czas Kultury”, a linguist from the Jagiellonian University, Jan Rybicki, concluded that the opportunities afforded by digital humanities are expressed in the fact that this discipline will be more open to the atmosphere of partnership and the value of teamwork. A single researcher, even if they are a specialist, cannot face the challenges of digital humanities. It can only be done by a diverse team where each member will bring in different competences [Rybicki 5]. As in culture, where different spheres of life, such as art, society, and technology blend together, humanities, challenged by software, should also encourage an interfusion of qualifications by creating interdisciplinary research teams. In the meantime, Processing continues to be a medium combining all these disciplines and providing them with a space to enter into the creative conversation.

**Translated by Katarzyna Szuster-Tardi**

## Works cited

- Cox, Geoff, and Alex McLean. *Speaking Code: Coding as Aesthetic and Political Expression*. The MIT Press, 2013.
- Dziamski, Grzegorz. *Przełom konceptualny i jego wpływ na praktykę i teorię sztuki*. Wydawnictwo Naukowe UAM, 2010.
- Fry, Ben. *Visualizing Data*. O'Reilly Media, 2008.
- Manovich, Lev. *The Language of New Media*. The MIT Press Cambridge, 2001.
- . *Software takes command*. Bloomsbury Academic, 2016.
- Mary Huang. <http://www.rhymeandreasoncreative.com/portfolio/index.php?project=continuum>.
- Nervous System*. <https://n-e-r-v-o-u-s.com/cellCycle>.
- Open Processing*. <https://www.openprocessing.org>.
- Reas, Casey, and Ben Fry. *Make: Getting Started with Processing*. Maker Media, 2015.
- . *Processing: A Programming Handbook for Visual Designers*. The MIT Press, 2014.
- Rybicki, Jan. "Cyfrowa humanistyka w Polsce: działania i konteksty". *Questionnaire directed by Lucyna Marzec, and Marcin Wilkowski*, Czas Kultury, vol. 185, no. 2, 2015, p. 5.
- Shiffman, Daniel. "Introduction". Ira Greenberg. *The Essential Guide to Processing for Flash Developers*, Springer, 2009, p. IX.
- . *Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction*. Morgan Kaufman, 2008.
- . *The nature of Code*. Daniel Shiffman, 2012.