Evolutionary Art with an EEG Fitness Function

Ingrid Něměčková [0000-1111-2222-3333] and Carl James-Reynolds[1111-2222-3333-4444] and Edward Currie[1111-2222-3333-4444]

[1] Middlesex University, London NW4 4BT
[2] Middlesex University, London NW4 4BT
[3] Middlesex University, London NW4 4BT
C.James-Reynolds@mdx.ac.uk

Abstract. This project involved the use of an interactive Genetic Algorithm (iGA) with an electroencephalogram (EEG)-based fitness function to create paintings in the style of Piet Mondrian, a Dutch painter who used geometric elements in his later paintings. Primary data for the prototype was gathered by analysis of twenty-seven existing Mondrian paintings. An EEG gaming headset was used to read EEG signals, which were transmitted by Bluetooth to an Arduino running an iGA. These values were used as the iGA fitness function. The data was sent to a PC running Processing to display the artwork. The resultant displayed artwork evolves to favour higher attention and meditation levels, which are considered to represent greater mindfulness. The process ends when the observer identifies a piece of art they would like to keep. However, convergence of the algorithm is difficult to test as many parameters can affect the process. A number of issues arising from the research are discussed and further work is proposed.

Keywords: Interactive Genetic Algorithm, Generative Art, EEG.

1 Background

1.1 Context

Mondrian was a Dutch painter and theoretician who is viewed by many as one of the greatest artists and pioneers of the 20th century abstract art movement. His work inspired fashion designers such as Lola Prusac, who worked for Hermes in Paris and Yves Saint Laurent, who introduced The Mondrian Collection in 1965. He also inspired the serialist composer Pierre Boulez and the development of an esoteric programming language “Piet” in which programs resemble abstract art.” [1].

At the start of World War One Mondrian became acquainted with M.H.J. Schoenmaekers, a theosophical philosopher whose work on the symbolical meaning of lines and the mathematical construction of the universe had a decisive influence on Mondrian’s vision of painting. It is this that seems to have inspired Mondrian to start using just the three primary colours; he ”wrote in 1915 in ‘The New Image of the World’ that red, yellow and blue are the only true colours, because all others can be derived from these three” [2]. Mondrian and Theo van Doesburg, an artist and architect, founded the journal De Stijl [3] in 1917, in which Mondrian published his concepts of art and colour. De Stijl, (”the style”), also known as Neoplasticism was a movement among Dutch artists, architects, and designers that presented an ideal of total abstraction as a model for harmony and order across the arts. (4) The movement had a far-reaching effect on the development of both abstract art and modern architecture and design.

In 1919 Mondrian moved back to Paris, where he began creating the iconic abstract paintings for which he is most famous. He began using only the primary colours red, yellow and blue, together with the non-colours white, black and grey [2]. Mondrian wrote to his friend Van Doesburg: ”I am in search of the perfect harmony of rhythm... that’s really difficult.”[2] He became obsessed with perfect harmony and determined to reflect it in his paintings. Initially, Mondrian’s use of primary colours was experimental. He mixed them with white and black to experiment with different shades of the primary colours and sometimes gave his painting a grey or brown over-shade. He then moved away from the shades of grey and brown and also decided that all lines in his paintings should be equally black. At first the lines were thin, tending to fade at the edges. [1] On 4 December, Mondrian wrote to his friend Van Doesburg that he had finished ”a thing” that pleased him more than all previous works. This was possibly an early version of Composition A. [2] he adopted the use of a “rhythmic” pattern of squares in his paintings. During the period 1920 – 1939, he created his most iconic paintings, at first exhibiting them without any frames. Later, he edged them with thin white wooden strips and later moved on to building white shadow boxes.

1.2 Interactive Art Projects

Edmonds states that “Art becomes interactive when audience participation is an integral part of the artwork.”[5] It should draw the audience into becoming part of the project, by active participation or passive exploration**.** New technology, mainly microprocessor based, has empowered the rise of new interactive art.

According to Tempel “Generative art is created by a system that operates autonomously, or semi- autonomously, rather than directly by the artist, it is a sub-category of temporary art. Generative art systems are usually computer programs.”[6]. One of the early pioneers was the artist Harold Cohen who is the creator of AARON [7], a computer program written in the LISP programming language, designed to produce art autonomously. Cohen improved his program throughout his life and paintings created by the program were not only valued by the public, but also by art critics. Cohen traveled the world with AARON, exhibiting his art at numerous prestigious museums and galleries.

Since Harold Cohen, the generative art movement has produced acknowledged artists such as Sonia Landy Sheridan, who founded the first generative systems department at the Art Institute of Chicago in 1970 [8] and Lillian Schwartz, who set about demonstrating that Da Vinci himself was in all probability the model for the Mona Lisa by using computer software in 1984 [9]. Due to the increasing popularity of generative art, new computer programs were developed over the years. One such is Processing [10], which has become popular recently and is based on Java. Generative art has flourished since the start of the 21st century. Artists in this area have created a strong on-line community of open-source websites where ideas, code and results are shared. Low cost micro-controller platforms such as Arduino [11] are frequently used for digital installations and other physical computing projects, which has also contributed to this phenomenon.

1.3 Related work

Mondrian-style work has been explored by a number of computer artists. “ARTMOVEMENT”, created by Dominic Fee [12], was a ”web-based educational resource, with interactive elements, designed to assist in the education and appreciation of fine art.” This allows users to interactively create their own versions of abstract art by Mondrian, Albers, Malevich, Martin, Rothko, Riley, Kline and Pollock. The artwork is modified by mouse manipulation and the system was created using Processing. The system allows the user to move away from Mondrian's chosen colours and elements; for example the canvas can be changed to a single colour and lines can be colours other than black.

Shen and Gedeon [13] explored similarities in Mondrian’s paintings and focused on the repetitiveness of the lines, which were split into 8 different sections. They created one ultimate rule in their software: ”Each line in a Mondrian-like graph, must have one of its ends being nodal or online, another end being terminal, nodal or online.” They applied an Interactive Bacterial Evolution Algorithm where the first painting created was the parent generation, which went through evolution by each chromosome being assigned a value by the fitness function, which represented its quality. The chromosome absorbs good gene pieces from another one to replace its own. This results in the next generation, which was evaluated by a human. The researchers were however not interested in how the human liked the generated Mondrian graph, but the evaluation was specific to shape, colour, none or both. Depending on the answer of the human a new graph was generated. The conclusion of this research was that ”Mondrian paintings are more than random compositions of lines and rectangles by revealing the fact that even minor modifications lead to less subjective satisfaction of audiences.”

1.4 Genetic Algorithms

”Genetic algorithms (GA) were invented by John Holland in the 1960s and were developed by Holland and his students and colleagues at the University of Michigan in the 1960s and the 1970s.” [14] They adopt a heuristic approach in applying the principles of evolutionary biology to the solution of complex problems for which there is no usable exact algorithm [15] and are a subset of evolutionary computations in the field of artificial intelligence. Evolutionary computations use techniques that mimic known evolutionary processes such as selection, reproduction, mutation and crossover, as described by Charles Darwin in his book ”On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life” [16] Individuals in a population of candidate solutions undergo an evaluation of their fitness to survive. The natural process of selection will preserve the individuals with the best genetic characteristics to survive in their environment, which then become the next generation of candidate solutions. An individual can be represented as a binary string of ones and zeros or a matrix. The development of the population then proceeds by repeating the above mentioned operations [17].

The terminology used for GAs is derived from genetics. Individuals are referred to as genotypes, genomes, chromosomes which are structures representing their characteristics. These characteristics can be presented as binary values. Chromosomes in nature consist of deoxyribonucleic acid (DNA) forming strings of DNA. A chromosome is divided into genes, the order of which is important. Each gene encodes an individual trait such as eye colour, hair colour or height; these individual traits are called alleles. In a GA these traits can be modelled in binary to represent a candidate solution [14].

1	0	0	1	1	0	1

Fig. 1. Binary representation of chromosome

The principle of the GA process is the repeated creation of new populations of individual solutions to the problem. As the population evolves, solutions improve. Typically, the first generation of individuals is created randomly, although it may be seeded to ensure a wide range of differences in the initial parents. In each successive generation, a fitness function value is calculated for each individual, which expresses the quality of the solution represented by that individual. According to this quality, individuals are selected who are then modified (by mutation and crossover) to produce a new population. This process is continuously repeated so that the overall quality of the solutions is improved. The algorithm is usually stopped after a certain time or number of generations, or if a sufficient quality of solution is achieved [18].

1.5 Fitness function

To create a new population from the previous population it is necessary to define the criteria for selecting successful individuals (parents). In nature, individuals compete with each other in their natural environment, and the more successful individuals have a greater probability of surviving to produce more offspring than the less successful. In an artificial environment it is necessary to create similar competition for survival, to simulate the natural process and the fitness function plays a key role in this. [18] Its function is to evaluate each individual against the expected solution and individuals with scores closest to the expected solution are preferred in the production of the next generation.

1.6 Roulette wheel selection

This has been described as “a genetic operator used in genetic algorithms for selecting potentially useful solutions for recombination.” [19] Individuals of the population are assigned sectors of the simulated ‘wheel’, the size of each sector being directly proportional to the fitness value of the individual [14]. Insertion of the ball into the roulette wheel and spinning the wheel until the ball settles in one of the sectors is then simulated and the corresponding individual is selected for breeding. Several balls can be inserted at the same time, but no more than the number of individuals to be selected from [18] In this case all parents can be selected in one roulette spin. ”Once a pair of parents is selected, they cross over to form two offspring.” [14] If the same parent is chosen twice, another choice is made until the two parents differ.

1.7 Mutation and Crossover

Mutation serves to create genetic diversity in the population genome. It is a genetic operator that changes one or more gene values in the chromosome of an individual. A simple example of a mutation is changing number 0 to 1 or vice versa [18]. This operator is used based on a given probability value of the mutation occurrence, which may be set by the user [20]. The probability of mutation occurrence should be set with caution, as too high will result in degradation of the GA and too low may result in the GA converging on local maxima in the solution space.

Crossover is a reproduction operator that simulates the random exchange of information contained in parents when creating a new child. During a crossover a combination of genetic information from two parents occurs. It may be single point crossover, or multi point crossover. The crossover points are typically randomly selected [20].

2 Methodology

An initial prototype was developed based on the work described in [21] who published his code on GitHub. This raised issues including the distribution of colours, the generation of appropriate random data and the distinction between squares and rectangles in the initial analysis of ten Mondrian paintings using Google’s Vision API. [22]. The code was then written from scratch to address these issues.

A total number of 27 Mondrian paintings were chosen for analysis:

- Composition A, 1920
- Composition B, 1920
- Composition with Yellow, Red, Black, Blue and Gray,1920
- Composition with Yellow, Blue, Black, Re and Gray, 1921
- Composition with Large Blue Plane, Red, Black, Yellow and Gray, 1921
- Tableau I, with Black, Red, Yellow, Blue, and Light Blue, 1921
- Composition with Red, Yellow, Black, Blue and Gray, 1921

- Tableau II, with Red, Black, Yellow, Blue and Light Blue, 1921
- Composition with Red, Blue, Yellow, Black and Gray,1992
- Composition with Blue, Yellow, Red, Black and Gray,1992
- Composition with Blue, Red, Yellow and Black, 1922
- Composition with Blue, Yellow, Black and Red, 1992
- Tableau 2, with Yellow, Black, Blue, Red and Gray, 1992
- Tableau with Yellow, Black, Blue and Red and Gray, 1923
- Tableau No II. With Black and Gray, 1925
- Composition with Red, Yellow and Blue, 1927
- Composition with Black, Red and Gray, 1927
- Composition: No. III with Red, Yellow and Blue, 1927
- Composition I: with Black, Yellow and Blue, 1927
- Large composition with Red, Blue and Yellow, 1928
- Composition No III. With Red, Blue, Yellow and Black, 1929
- Composition No I. with Red, 1931
- Composition with Blue and Yellow, 1932
- Composition with Black and White, with Double Lines, 1934
- Composition with Double Lines and Blue, 1935
- Composition A with Double Lines and Yellow, 1935
- Composition White, Red and Yellow: A, 1936

2.1 Analysis of Mondrian's Paintings

Piet Mondrian chose to limit his paintings to the use of the three primary colours red, blue and yellow together with black, white and grey. Mondrian referred to this genre as 'Neo-Plasticism'. Because the artist was so specific in the colours he used it was important to represent those colours as accurately as possible. It was decided to use the Google Vision API for the analysis. The image analysis was able to identify the colours and represent them in RGB format. The pictures of paintings were taken from Piet Mondrian: Life and Work [2] , scanned and analysed. In order to use the Google vision API it was important to choose paintings of predominantly one colour in order to analyse the RGB content for each of the colours. From the selection of paintings 5 were chosen for each colour to analyse and the results are shown in fig. 2.

Colour Analysis						
	RGB Colours					FINAL
Yellow	241,199,21	246,209,22	253,216,13	252,216,14	251,237,68	249,215,28
Red	220,67,45	229,83,51	218,56,42	204,72,49	199,38,48	214,63,47
Blue	27,74,115	31,48,75	27,64,115	18,71,131	75,121,174	36,76,122
Gray	240,240,243	228,228,228	229,231,237	224,224,224	238,238,237	232,232,234

Fig. 2. Analysis of colour in Mondrian painting

2.2 Physical Construction

Processing [10] was used to create the display software. It is open source, has built in graphics libraries and works well with the Arduino Platform [11]. Arduino uses Atmel processors and has its own compiler with many libraries available. Arduino was used to run the iGA and also to create the random variables for the initial population; the code for this was an adaptation of the code used by James-Reynolds and Currie [23]. It was decided that the complexity of the paintings, including the number of columns and rows and the number of colours would be decided by the iGA, but the precise details of where the colours would be distributed would be random. The fitness function used EEG data from a NeuroSky Brainwave Starter Kit [24]. The code allows EEG signals for attention, meditation or mindfulness (a combination of both signals) to be used.

Using EEG signals as the basis for the fitness function enables the evaluation to take place in a subconscious way. The interaction is not about users performing an conscious action, but instead a more subtle interaction that can be clearly demonstrated.

The integration of the iGA and the EEG was based on previous work [23]. A gaming headset [24] was used to capture and transmit the EEG readings via Bluetooth. A Bluesmirf receiver using an RN-41 module [25] TX pin was connected as shown in figures 3 and 4 to the RX pin on the Arduino. The data from the headset was parsed and attention and meditation values extracted from the data stream. One Arduino managed this process and passed the data over an I2C connection to a second Arduino running the iGA code.

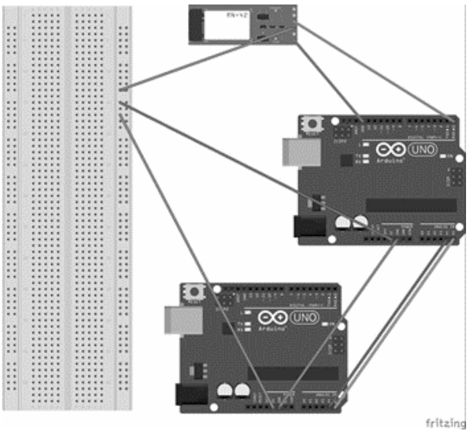


Fig. 3. Schematic of Arduino boards and Bluetooth

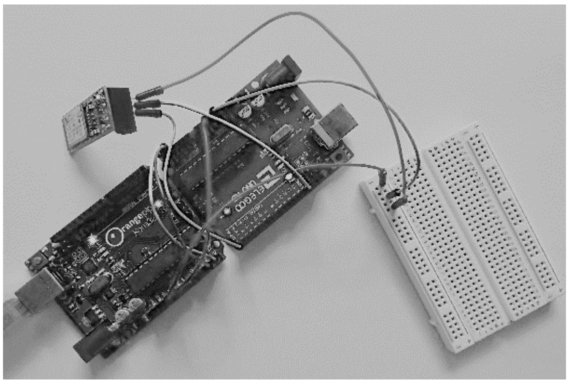


Fig. 4. The equipment: Arduino boards and BlueSmirf

The code firstly generates 8 parameters for each of 4 parent individuals. The selection of future parents in each generation is generated by the roulette approach based on the EEG-based fitness function.

Random numbers in Arduino are generated using randomSeed() which initialises the pseudo-random generator. To ensure that a different starting point in the pseudo-random sequence was obtained each time, the command analogRead() on an unconnected pin was used as an argument to randomSeed() The mutation rate for the iGA was set at 3%.

Table 1 shows the visual analysis of the paintings that provided data for setting the limitations of the solution space.

Table 1. Findings from analysis of paintings

Second Painting Analysis		
Item	Number	Average
Number of rows	3 - 9	4,85
Number of columns	2 - 7	4,11
Number of red objects	0 - 3	1
Number of yellow objects	0 - 4	1,18
Number of blue objects	0 - 4	1,11
Number of black objects	0 - 4	1
Number of grey objects	0 - 13	4,25
Number of white objects	0 - 17	2,88

During this analysis it was discovered that the numbers of coloured objects in Mondrian paintings appeared within certain limits. The mean value for each represented colour was calculated and used as starting values for a colors[] array.

Arduino serial monitor - output 3	
Columns:4 Rows:9 Cells:36 Gray:9 White:6 Yellow:4 Blue:9 Black:1 Red:4	Columns:3 Rows:6 Cells:18 Gray:6 White:4 Yellow:3 Blue:2 Black:2 Red:1
Columns:4 Rows:3 Cells:12 Gray:4 White:3 Yellow:4 Blue:1 Black:0 Red:0	Columns:4 Rows:8 Cells:32 Gray:9 White:7 Yellow:5 Blue:4 Black:1 Red:2
Columns:4 Rows:9 Cells:36 Gray:12 White:5 Yellow:4 Blue:6 Black:3 Red:2	Columns:3 Rows:5 Cells:15 Gray:4 White:3 Yellow:3 Blue:1 Black:1 Red:1
Columns:2 Rows:5 Cells:10 Gray:4 White:3 Yellow:1 Blue:2 Black:0 Red:0	Columns:6 Rows:5 Cells:30 Gray:9 White:6 Yellow:6 Blue:2 Black:3 Red:0
Columns:2 Rows:5 Cells:10 Gray:4 White:3 Yellow:1 Blue:2 Black:0 Red:0	Columns:6 Rows:5 Cells:30 Gray:9 White:6 Yellow:6 Blue:2 Black:3 Red:0
Columns:3 Rows:5 Cells:15 Gray:6 White:5 Yellow:2 Blue:1 Black:0 Red:0	Columns:2 Rows:6 Cells:12 Gray:4 White:3 Yellow:1 Blue:1 Black:1 Red:0
values of Columns:3 Rows:6 Cells:18 Gray:7 White:4 Yellow:1 Blue:2 Black:0 Red:2	Columns:6 Rows:8 Cells:48 Gray:10White:7Yellow:9Blue:9Black:2Red:5

Table 2. Serial monitor output of Arduino random code - Final Version

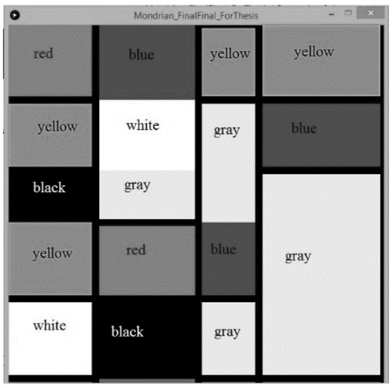
Table 2 shows examples of the initially generated parents. The Arduino was passing 8 values to Processing; 6 for colours, 1 for number of rows and 1 for number of columns. The distances between the lines were generated randomly and not decided by the algorithm.

3 Testing

Testing was performed with volunteers. The volunteers varied in gender and nationality, but because they were all students, the age range was limited; they were typically around 20 years of age. Each volunteer sat in front of a laptop and was given a brief explanation of the process. They were then instructed to sit back, relax and just look at the paintings. A new painting was generated every 5 seconds for a period of 2 minutes, so that in total 36 paintings were generated and presented to each volunteer.

The pattern of changes in the complexity of the paintings seemed to depend upon the individual volunteer; for some, complex paintings evolved into simpler ones and vice versa for other volunteers.

The original aim of testing was to see if the generated paintings would become more or less complex. However, this became hard to differentiate as there was not a consistent direction in the paintings' development, and no 'final' state where the subject's ideal painting had emerged. Furthermore, different subjects had different preferences. There were no sudden changes from complex to simple paintings or vice versa - the transitions seemed to be smooth and gradual. However, there was a slight distinction between different volunteers on how much the paintings simplified. Generally, initial populations contained more complex images and tended to stay complex (as in fig. 5) or simplify (as in fig.6) depending on the user. With some users the images did not evolve towards simplicity, although those that did seemed to stay in this area of the solution space.



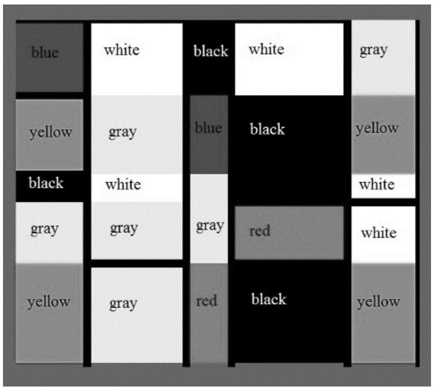


Fig. 5. More complex paintings produced by subjects

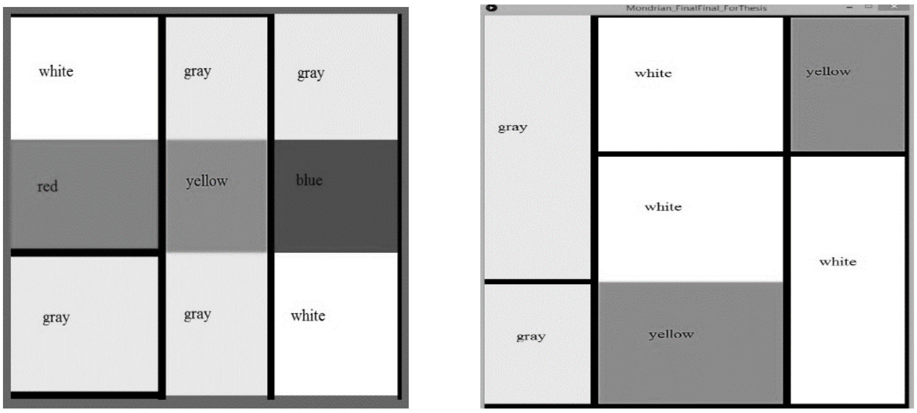


Fig. 6. Simpler paintings produced by subjects

4 Discussion

Whilst we can see a general pattern of convergence, the results are difficult to quantify. The validity of judging a piece of art by a subconscious response as opposed to a conscious one can also be questioned. Some pieces of art need to be lived with before a deeper appreciation grows and many pieces of art are created not to induce a calm, mindful state of mind but to provoke and challenge. It is unfortunate that we cannot ask Mondrian these questions as we appear to be exploring the concept of "harmony of rhythm" by directly looking at the effect of art on brain activity. Although convergence was observed, the test environment itself may have been a distraction as could the novelty of the approach and it is possible that a test subject may try to consciously effect choices. A better strategy would be the use of a Virtual Environment with less distractions in which the images could be more three dimensional, allowing for more realism in the texture of the canvas and its hanging in a space. Whilst more sophisticated EEG devices might have provided better data for these experiments, a gaming headset was used as this does not pose the ethical considerations that this would entail, due to the potential for intrusive monitoring of likes and dislikes**.**

Planned future work includes exploration of the differences between using a conscious and subconscious response. Removing some of the random elements of the image presentation and making all parameters controlled by the iGA would make it possible to explore some art theory and see if, for example, golden section values emerged. In this work the solution space was confined to values that were derived from the analysis of Mondrian's paintings but it would be possible to employ a less constrained solution space.

5 Conclusion

This paper has presented an innovative approach to art generation using EEG waves as a fitness function. While the project has not produced easily quantifiable results, it has nevertheless raised interesting new questions about how we can interact with art through software.

While the current research has limitations, it has enabled the exploration of new approaches to generative art through the application of evolutionary systems and has brought some insights in this area and pointers to future work. The developed platform is viable for some of this future research.

References

1. Blotkamp, Carel (2001). *Mondrian: The Art of Destruction*. Reaktion Books. isbn: 9781861891006.
2. Jong, Cees De (2015). *Piet Mondrian: Life and Work*. Abrams. isbn: 9781419714085. Knuth,
3. Bris-Marino, P., 2014. La influencia de la teosofía sobre la obra neoplástica de Mondrian. Arte, Individuo y Sociedad 26. https://doi.org/10.5209/rev_ARIS.2014.v26.n3.42960
4. Foundation, The Art Story (2019). *De Stijl*. [Online; accessed 12-November-2018]. url: https://www.theartstory.org/movement-de-stijl.htm.
5. Edmonds, Ernest (n.d.). *Interactive Art*. online. [Online; accessed 15-November- 2018]. https://pdfs.semanticscholar.org/e245/971a.pdf.
6. Tempel, M., 2017. Generative art for all. Journal of Innovation and Entrepreneurship 6. https://doi.org/10.1186/s13731-017-0072-1
7. Cohen, H., n.d. 1988-How to Draw Three People in a Botanical Garden 10.
8. Sheridan, S.L., 1990. Mind/Senses/Hand: The Generative Systems Program at the Art Institute of Chicago 1970-1980. Leonardo 23, 175. https://doi.org/10.2307/1578602
9. Schwartz, Lillian F. (1984). *ArtAnalysis-1984THEHIDDENMONALISA*. [On- line; accessed 20-November-2018]. url: http://lillian.com/art- analysis/.
10. Reas, Casey and Ben Fry (2015). *Processing: A Programming Handbook for Visual Designers and Artists*. The MIT Press. isbn:9780262321853.
11. Arduino Platform https://www.arduino.cc/ [On-line; accessed 20-June-2019]
12. Fee, D., n.d. Supplementing Fine Art Education with Digital Interactivity 49.

13. J. Y. Shen and T. Gedeon, "Cyber-Genetic Neo-Plasticism – An AI program creating Mondrian-like paintings by using interactive bacterial evolution algorithm," 2007. http://cs.anu.edu.au/escience/project/06S2/report/JianShen_report.pdf
14. Melanie, M., n.d. An Introduction to Genetic Algorithms 162.
15. Banzhaf, Wolfgang et al. (1999). *Genetic Programming* Arduino (2019). *Arduino*. [Online; accessed 02-December-2018]. url: https:// www.arduino.cc/.
16. Darwin, Charles, 1809-1882. On The Origin of Species by Means of Natural Selection, or Preservation of Favoured Races in the Struggle for Life. London :John Murray, 1859
17. Hudson, D.L., Cohen, M.E., 2000. Neural networks and artificial intelligence for biomedical engineering, IEEE Press series in biomedical engineering. Institute of Electrical and Electronics Engineers, New York.
18. Haupt, R.L., Haupt, S.E., 2004. Practical genetic algorithms, 2nd ed. ed. John Wiley, Hoboken, N.J.
19. Thierens, D., Goldberg, D., 1994. Convergence models of genetic algorithm selection schemes, in: Davidor, Y., Schwefel, H.-P., Männer, R. (Eds.), Parallel Problem Solving from Nature — PPSN III. Springer Berlin Heidelberg, pp. 119–129.
20. Srinivas, Mandavilli and Lalit M Patnaik (1994). "Adaptive probabilities of crossover and mutation in genetic algorithms". In: *IEEE Transactions on Systems, Man, and Cybernetics* 24.4, pp. 656–667
21. WillPhelps1. https://gist.github.com/WillPhelps1/4da7f9b2eb718340bedf885b63c9f729 [Online; accessed 02-July-2019
22. Google Cloud (2018). *Vision API*. [Online; accessed 27-December-2018]. url: https://cloud.google.com/vision/.
23. James-Reynolds, C., Currie, E., 2016. EEuGene: Employing Electroencephalograph Signals in the Rating Strategy of a Hardware-Based Interactive Genetic Algorithm, in: Bramer, M., Petridis, M. (Eds.), Research and Development in Intelligent Systems XXXIII. Springer International Publishing, Cham, pp. 343–353. https://doi.org/10.1007/978-3-319-47175-4_25
24. NeuroSky(2018).*NeuroSkyBrainwaveStarterKit.*[Online;accessed01-February- 2019].
25. Microchip RN-41 https://www.microchip.com/wwwproducts/en/RN41[Online; accessed 02-July-2019