Computer Program for Artists: *ART I*

Katherine Nash, Richard H. Williams

Leonardo, Volume 3, Number 4, October 1970, pp. 439-442 (Article)

Published by The MIT Press

Leonardo, Vol. 3, pp. 439–442. Pergamon Press 1970. Printed in Great Britain

COMPUTER PROGRAM FOR ARTISTS: *ART I**

Katherine Nash and Richard H. Williams*****

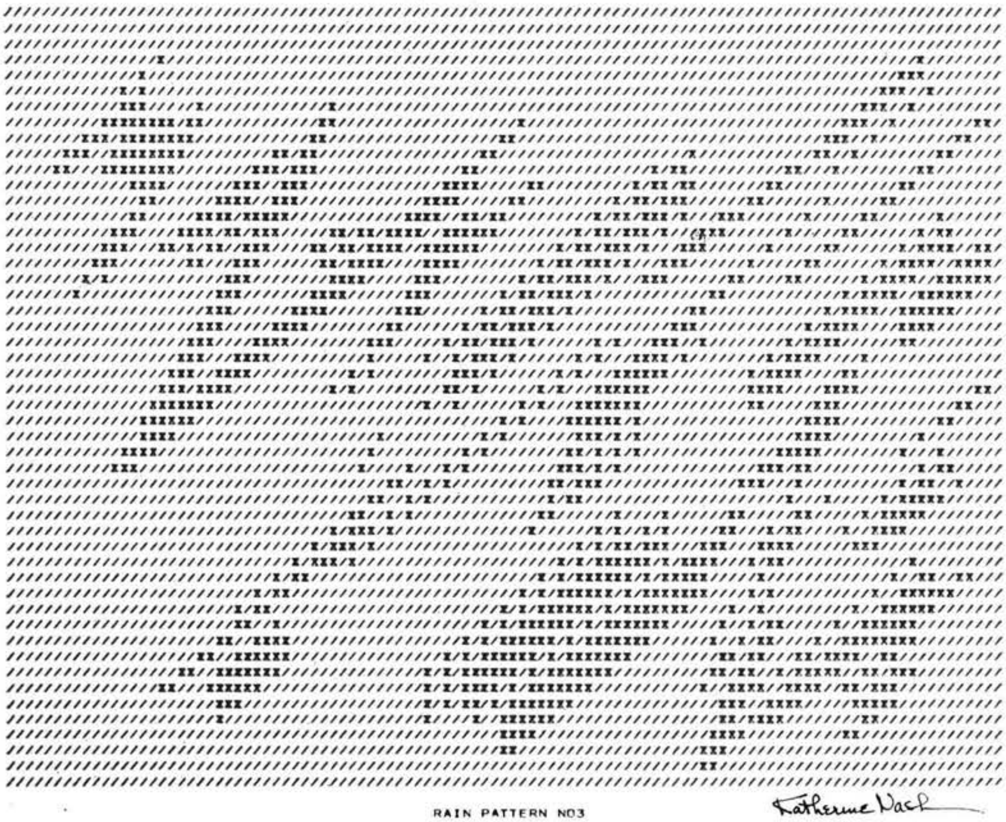

Fig. 1. *'Rain Pattern, No. 3', digital computer drawing on paper by K. Nash*, 9 × 12 in., 1969.

Twenty years of a computerized society make it apparent that twenty years hence no artist can ignore the computer. He will have to adjust to it, cope with it or use it. He cannot reject it. It will influence his creative thinking, as all aspects of society have always influenced the artist. The computer will be another tool for creativity.

There are three ways in which an artist can creatively approach a computer.

1. He can set aside a portion of his time to learn a computer language and then write a program for the making of a work of art.
2. He can collaborate with a computer engineer in the making of a work of art.
3. He can work with a computer that already has within its memory bank an art language upon which he can draw to make a work of art.

The average artist finds difficulty even contemplating the first. He doesn't want to take his time to be either a mathematician, or an engineer, or a scientist and logic is not always to his liking. But if he is willing to assume the responsibility of learning a language to write a program, there is still the almost insurmountable problem at present of finding a computer that he can afford and a situation that will accommodate him.

Some artists attempt the second solution. An imaginative man (the artist) attempts to verbally describe the requirements to a man of logic (the computer programmer). Too often a compromise results and the artist, in attempting to make a personal statement, finds the statement no longer is personal.

The third alternative has been answered by a program called *ART 1*, which was developed at the Department of Electrical Engineering and Com-

* This note is based on a text published in *Page 7, Bulletin of the Computer Arts Society*, March 1970.

** Artist, Department of Studio Arts, University of Minnesota, Minneapolis, Minnesota 55455, U.S.A.

*** Engineer, Department of Electrical Engineering and Computer Science, University of New Mexico, Albuquerque, New Mexico 87106, U.S.A.

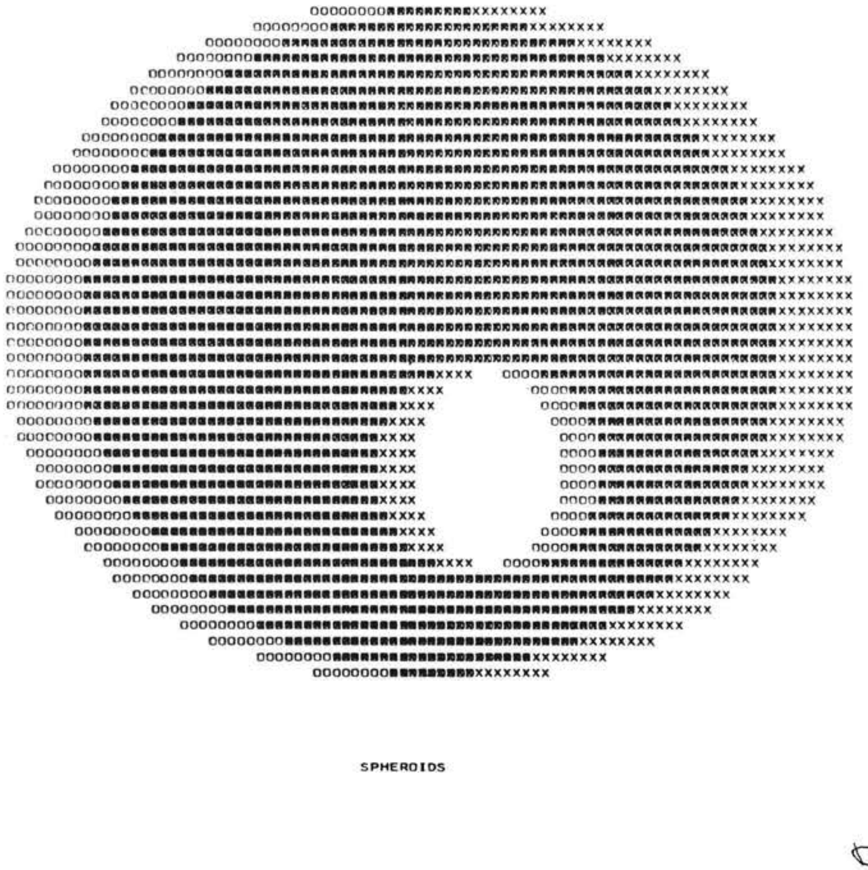

Fig. 2. 'Spheroids', digital computer drawing on paper by K. Nash, 9 × 12 in., 1969.

puter Science at the University of New Mexico. It does allow the artist to use his imagination to make a personal statement. It also permits him to gain a nodding acquaintance with the digital computer. *ART 1* results from the collaboration of an artist and an engineer. It is a language stored in a computer memory bank upon which an artist can draw. It is just a beginning. It does not require a computer programmer to hold the artist's hand. It is written to use the simplest and least expensive equipment. It is meant as a bridge between the artist's world and the world of technology. For the artist this language presents a simplified introduction to the computer, as well as presenting new horizons for creativity. This language makes it possible for an artist to work with a computer without a primary need of computer knowledge. The designs are rendered by a high-speed printer controlled by a digital computer which, of course, is controlled by the artist. Examples of designs produced by this technique are shown in Figs. 1 to 3.

The computers presently being used are the following: An IBM 360/40, with an IBM 1403 printer that is capable of printing a line of 120 characters at the rate of 1100 lines per minute (located at the Computer Center of the University of New Mexico, Albuquerque, New Mexico) and a Control Data 6600, with a 501 printer (located at the University of Minnesota, Minneapolis, Minnesota).

ART 1 in all its detail is somewhat involved. It contains approximately 350 separate statements written in FORTRAN IV, a programming language. An artist, however, need not concern himself with programming complexities, since the entire program is stored in the computer's memory bank. The program may be obtained by simply using one card that had been key-punched:

/INCLUDE ART1

ART 1 causes the computer to reserve in its memory two arrays of storage spaces. It is these storage spaces into which the artist will put characters that the printer will ultimately print to make the design. Each array is considered to be a rectangle of 50 rows by 105 columns. Since the printer places the columns closer together than the rows, the proportion of rows to columns produces a printed field of approximately 4 by 5.

The contents of the two arrays are controlled by the artist as described below. When the artist completes his design, he signals the computer and this causes the printer to print the contents of one array over the other. The value range available to the artist by one over-printing of any character over another is significantly greater than that produced by the characters themselves. The value range, however, is not significantly extended by additional overprinting. *ART 1* contains only one overprinting.

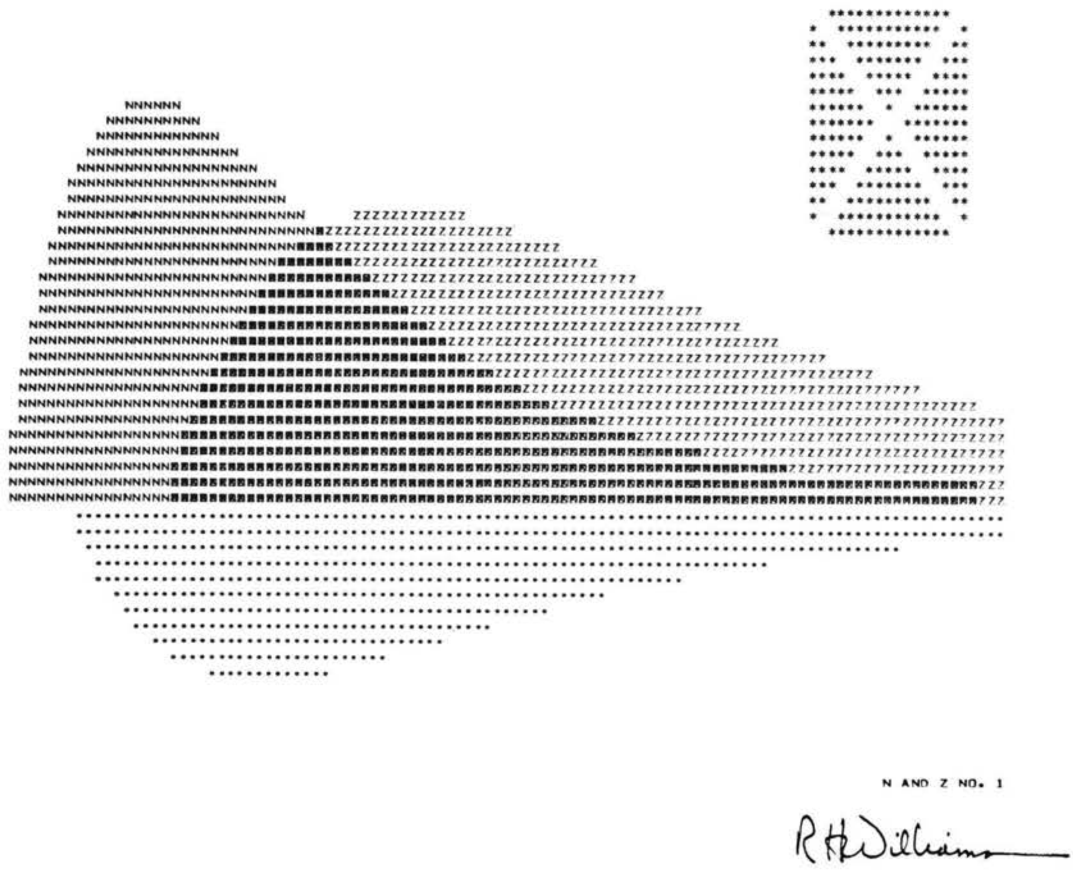

Fig. 3. '*N and Z, No.* 1', *digital computer drawing on paper by R. H. Williams*, 9 × 12 in., 1969.

After the artist obtains the program by the /INCLUDE ART1 statement, he lists several *data cards*. The first data card pertains to *initializing* the arrays in the computer's memory. Much of this initializing involves details that need not be discussed here. Suffice it to say this data card is used to enter the design's title into the computer (see Figs. 1 to 3) and to state how many copies of the design are desired (1 to a maximum of 6 copies).

All data cards, after the initializing data card, begin with 01, 02, 03, . . . or 09. If an artist begins a data card with 02, the computer enters a LINE into one of the arrays. The proportion of the LINE and the location of the LINE in the design's field are specified to the computer by a relatively short code word (14 data card spaces) on the data card, following the 02. The code word also specifies into which of the two arrays the LINE will be entered, as well as which specific character is to be used. Given a starting point in the design's field, the artist can control the LINE's placement horizontally, vertically or in any intermediate direction. One data card is required to place one LINE in one array. However, if the artist desires to repeat at a different location a LINE already specified on a data card, then a labor-saving feature is available. The same LINE may be repeated in a different location by stating a new starting row and column on the same data card, using spaces 21 through 25. A third repeat may be specified in data card spaces 26 through 30 and so on, to a 10th repeat specified in data card spaces 66 through 70. This same system of repeating a specified shape is available with other shapes in *ART 1* besides LINE. If the data card spaces reserved for repeating figures are left blank, then no repetitions are made. Extensive use was made of the repetition technique in Fig. 1, a design composed of LINE elements. The background of this design was specified on the initializing card; only the superimposed design was made using LINE.

If the artist begins a data card with 03 or 04, the computer is instructed to generate rectangles, 03 specifies SOLID RECTANGLES and 04 specifies OPEN RECTANGLES. As before, the character used to make the rectangle, as well as its location and proportions, are determined by a code word following either the 03 or 04 as the case may be. For example:

03 called SOLID RECTANGLE;
0 specifies the character (symbol) to be used;
2 specifies that the character is to be put into second array;
14 specifies the row of the upper left starting point;
019 specifies the column of the upper left starting point;
010 specifies the number of rows in the SOLID RECTANGLE and

0026 specifies the number of columns in the SOLID RECTANGLE.

Thus, the complete code word in this example is 0302140190100026 and it is placed in the first 16 data card spaces. This is all that would be required to place a SOLID RECTANGLE in a drawing. One data card would be required for each different rectangle in each array.

A TRIANGLE can be placed in a design by starting a data card with 05. Again, the character proportion and location of each TRIANGLE is specified by a code word which follows 05. The kind of TRIANGLE that can be generated by *ART 1* is limited. Typical kinds are illustrated in Fig. 3. Each TRIANGLE is an isosceles one and the diagonal edges are limited to slopes of one column by one row. This latter restriction is imposed in *ART 1* because other slopes may present a stairstep appearance, which tends to detract from the TRIANGLE's form.

An ELLIPSE may be specified to the computer by starting a data card with 06. The design in Fig. 2 was made in this way. In addition to specifying the printed character and the location of each ELLIPSE, the code word following 05 determines whether either the vertical axis is longer or the horizontal axis is longer, or whether the two axes are equal, in which case a circle is generated. Figure 2 shows examples of the latter two cases. Here two large ELLIPSES are overlapped and slightly displaced one from the other. A smaller ELLIPSE and a circle of 'blanks' is placed in each of the larger ELLIPSES to make the effect of a hole. This is an example of the use of the 'blank' as a character in its own right.

Data cards beginning with 07 and 09 generate what in *ART 1* is called QUADRANTS LEFT and QUADRANTS RIGHT. 07 takes whatever design is in the upper left quadrant of the arrays and generates a mirror image about the horizontal and vertical axes of symmetry. 09 does the same thing as 07 but starts with the upper right quadrant. An artist may find that QUADRANTS LEFT will produce an effect entirely different than QUADRANTS RIGHT. The reader can visualize this by imagining the drastically different effects on Fig. 3 that would result when QUADRANTS LEFT or QUADRANTS RIGHT was used.

A different kind of shape called EXPONENTIAL is generated by a data card starting with 08 (cf. Fig. 3). The reader may recognize that EXPONENTIAL is defined by the mathematical function

$$y = Ax\exp(Bx).$$

This curve has many pleasing aspects to the second author. EXPONENTIAL is used as an asymmetrical component that an artist may include in a design.

When the computer finally comes to a data card beginning with 01, the design assembled in the arrays is automatically printed by the high-speed printer and the program is terminated.

The reader should note that *ART 1* is the result of a study effort that extended over 18 months. It began in rather simple form when the engineer–author began to work under the aegis of the kinetic sculptor, Charles Mattox. A number of professional artists and art students influenced the program's development. Some of the results of using *ART 1* have already been reported in the literature [1, 2]. The program was given the name *ART 1* not only to signify that there may be an *ART 2* but also to symbolize that fruitful interactions of the arts and technology are legion. There may be, some day, an *ART 1000*.

REFERENCES

1. F. Hammersley, My First Experience with Computer Drawings, *Leonardo* **2**, 407 (1969).
2. J. Hill, My Plexiglas and Light Sculptures, *Leonardo* **3**, 9 (1970).