SONGS IN THE LANGUAGE OF INFORMATION: USING PERSONAL COMPUTERS TO CREATE SOUNDS AND GRAPHICS FOR A LARGE SCALE INSTALLATION

by

SARAH GEITZ

Bachelor of Arts
Scripps College
Claremont, California
1977

Submitted to the Department of Architecture
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE IN VISUAL STUDIES
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June, 1985

Signature of the author____________________________

Sarah Geitz
Department of Architecture
May 9, 1985

Certified by____________________________

Otto Piene, Professor
of Visual Design
Thesis Supervisor

Accepted by____________________________

Nicholas Negroponte, Chairman
Departmental Committee on Graduate Students

TABLE OF CONTENTS

Songs in the Language of Information: Using Personal Computers to Create Sounds and Graphics For a Large Scale Installation

by
Sarah Geitz

Submitted to the Department of Architecture on May 17, 1985 in partial fulfillment of the requirements for the Degree of Master of Science in Visual Studies.

ABSTRACT

Personal computers are easy to program, inexpensive and portable. With two Commodore 64 computers, I created an artistic installation entitled "Songs in the Language of Information". It was composed of elements reflecting time, space, light, action and predictability. The two computers controlled projected light patterns, complementary synthesized sound and interactive relays triggered by viewers crossing light beams.

The following thesis documents "Songs in the Language of Information". The "Introduction" presents background development leading up to this work. All components of the installation are discussed in "Physical Description" and "Analysis of the Work". A survey of selected recent and past work that has been influential to me is outlined in "Historical Precedents". "The Program" illustrates graphic elements, presents flow charts and lists programs of all subroutines used in the installation. The "Conclusion" offers what for me is the next step in my artistic work with computers.

Thesis Supervisor: Otto Piene
Title: Professor of Visual Design

ACKNOWLEDGEMENTS

I would like to thank Paul Earls of the Center for Advanced Visual Studies, M.I.T. for helping to make this thesis possible through his faith, advice, inspiration and many hours of consultation. I am also indebted to Otto Piene, Director of C.A.V.S. / M.I.T., for his advice and support, and for guiding me into working on an environmental scale.

Additional thanks also go to my brother, Kurt Geitz, for building the relays and acting as copy editor, to my parents for their loving support, to Jessica Goldring for kindly lending me her computer, to Marek Holynski for being a reader, and most of all to my husband, Vin Grabill who truly made this thesis possible through his patience and support.

INTRODUCTION

I wanted to find a
way to express myself
in order to cope with
a world I do not hope
to understand. not
understand. the
world. I wanted to
express both my
frustration with the
mechanization of my
life with negative
ideas about the
mechanization
negative to I wanted
to find a way to
express myself in
order to cope with a
world I do not hope
to understand . not
understand. the world
I wanted to express
both my frustration
with the
mechanization of my
life with negative
ideas about the
mechanization
negative to I wanted
to find a way to
express myself In
order to cope with a
world I do not hope
to understand. no

Computers are extremely powerful and versatile tools for artistic expression. They are capable of manipulating vast amounts of information. The work described in this thesis started in 1978 when I began learning how to program computers. I felt that by learning how to manipulate this machine I would discover my own way of creating artworks of beauty .

I began by programming static images of geometric forms and patterns on mainframe computers. However, I found the slow step-by-step unfolding of lines and dots generated by the computer on the video monitor was always more interesting than the final image. To preserve the beauty of this process I made videotapes of it. Adding external soundtracks emphasized the movements of the patterns.

Besides using external soundtracks it is possible for computers to generate their own sounds to accompany the movements of the patterns. I began using this capability on an Apple II+ personal computer.

Personal computers have the added advantage of being easy to program, inexpensive and portable. Unlike mainframe computer systems, personal computers free artists to work and exhibit whenever and wherever they want.

With the advantages of the Apple computer and its speaker which is often used for video games, I began programming audio/visual works for live use. Each time the Apple plotted a point it emitted a noise-like sound, a series of sounds or no sound at all. The speed at which the points were plotted controlled the rhythms and tonal qualities of the sound.

The Apple programs I wrote use random numbers to create slightly different sounds and drawings each time the programs were run. The element of chance makes the computer an ideal instrument for use in live audio/visual performances. Its unpredictability makes each performance unique and exciting.

Realizing the potential of unpredictability, I conceived of and performed in two events using the Apple. In these events I created exhibitions of drawings. I programmed the computer to draw lines and emit sounds at varying speeds. The graphics were displayed by a video projector onto a large piece of paper tacked to the wall and the accompanying sounds were amplified. As the lines and patterns emerged at varying speeds, I traced them with large felt-tipped markers. Five different programs were run. Each was projected onto a different piece of paper. Sometimes the computer projected the images faster than I was able to draw them. This demonstrated the programmer being programmed by the machine. The final drawings reflected my inability to keep up with the machine.

The sounds the Apple programs produced were reminiscent of percussion instruments. For a greater variety of sounds I purchased a Commodore 64 computer because it has a built in sound synthesizer.

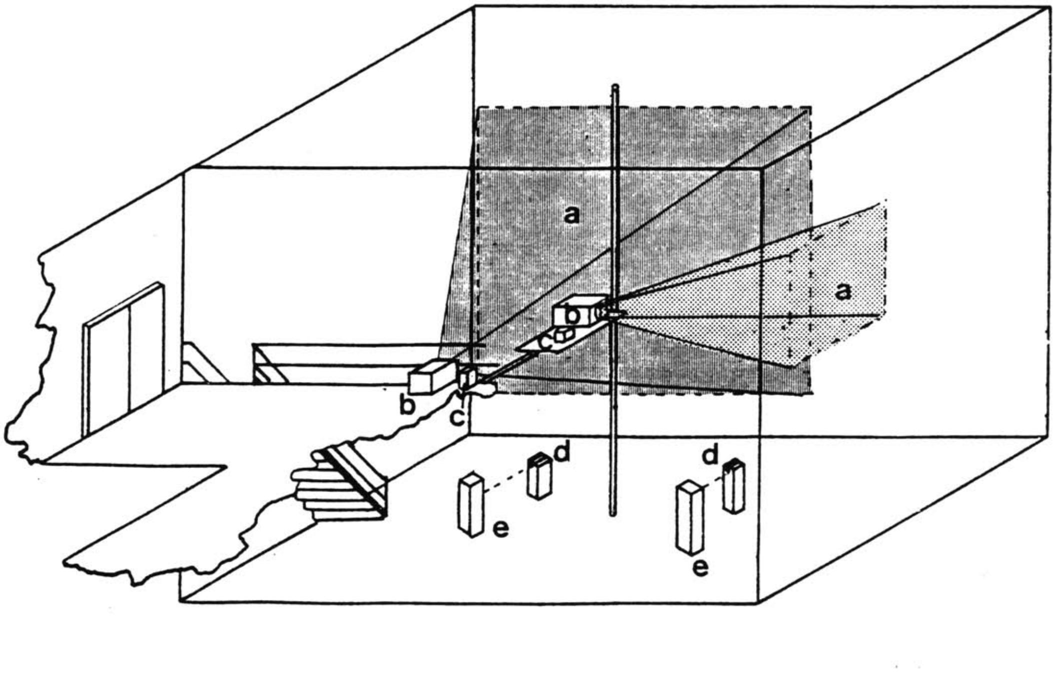
In addition, the graphic capabilities of the Commodore differ from those of the Apple. Commodore animations can be created through the manipulation of an alternate set of sixty-six keyboard graphic symbols. The keyboard graphics were developed by Commodore. Through programming, these graphic symbols can be moved about on the screen either alone or in combination with one another.

It was possible to complement the limited graphic vocabulary of the Commodore 64 by emphasizing the pattern transitions through the Commodore's greater variety of sounds. I expanded upon the random number capability of computers by simultaneously using more than one computer to compose visual/musical compositions. I put this work into an environmental scale by projecting the video onto two walls of a large room and by inviting the participation of the viewers. The viewers crossed beams of light which triggered the computers to randomly select different audio/visual compositions from a pre-programmed base. I considered the elements of time, space, light, action, and predictability to compose a work of art entitled, "Songs in the Language of Information".

PHYSICAL DESCRIPTION OF THE INSTALLATION

The exhibition entitled "Songs in the Language of Information" combined computer generated music/noise and graphics in a large scale interactive installation. It was set up for the first time at the Center for Advanced Visual Studies at the Massachusetts Institute of Technology from March 18-23, 1985. The physical components were two Commodore 64 computers, two

1.1



large screen black and white video projectors, two light activated relay switches, two flashlights, two audio speakers, one audio amplifier, one Commodore 1541 disk drive and four pedestals. Figure 1.1 is a diagram showing the position of the physical components of the installation in the large exhibition space at the M.I.T. Center for Advanced Visual Studies. The letter (a) on the diagram refers to the areas projected upon, (b) refers to the projectors, (c) refers to the speakers, (d) refers to the pedestals to which the flashlights are attached, and (e) refers to the pedestals to which the photocells are attached.

The two video projectors were installed overhead in a 40 x 40 x 30 foot room. The large screen video images were projected onto walls 90 degrees from one another. One video projector was on a loft forty feet from the wall. It projected a 20 by 26.8 foot image. The other projector was on a platform 12 feet high, 18 feet from the wall. It projected an image measuring 9 by 12 feet.

The video projectors were rented from a local video production house owned by a collector of black and white projectors. Because these projectors had only one tube, the image size increased with distance at a ratio of two to one. The color of light from these projectors was an old-fashioned television, electric-blue/white. This helped make the low resolution computer graphics being projected appear less like the video games for which they were originally designed.

The floor of the room and pedestals were painted gloss white as were the containers holding the flashlights. The walls of the room were also white. The video projectors and flashlights were the only sources of illumination in the room. Several visitors described the atmosphere as emitting a "radioactive" glow.

Both computers produced music. One computer was connected to a single channel of an amplifier. The other computer was connnected to another channel of the same amplifier. An AR speaker was placed next to the projector on the loft. It transmitted the audio for the video signal projected from the loft. A Minimus speaker next to the other projector transmitted its audio. I balanced the sounds from the two computers by ear to be of equal value from most points of the room.

Under normal conditions the acoustics in this room make it difficult for two people to carry on a conversation more than 10 or 12 feet apart. The echo of the sounds bouncing off the bare walls, cement floor and ceiling made it more difficult to appreciate the sight/sound synchronism in the piece. But the resonance produced by this echoing effect made the music become more vibrant. A number of people told me they thought the sounds were very peaceful.

The two sets of pedestals in the room were used as gates for visitors to pass through in order to change what they were seeing and hearing. One of the pedestals in each set was four feet tall by one foot square. To each of these was attached a photocell which was in turn connected to a computer interface. The other pedestal in each set was three feet eight inches tall by one foot square. To each of these was attached a plasic tube, painted white with a flashlight inside. The tubes camouflaged the flashlights and narrowed the light beams. The two pedestals in each gate were approximately four feet apart.

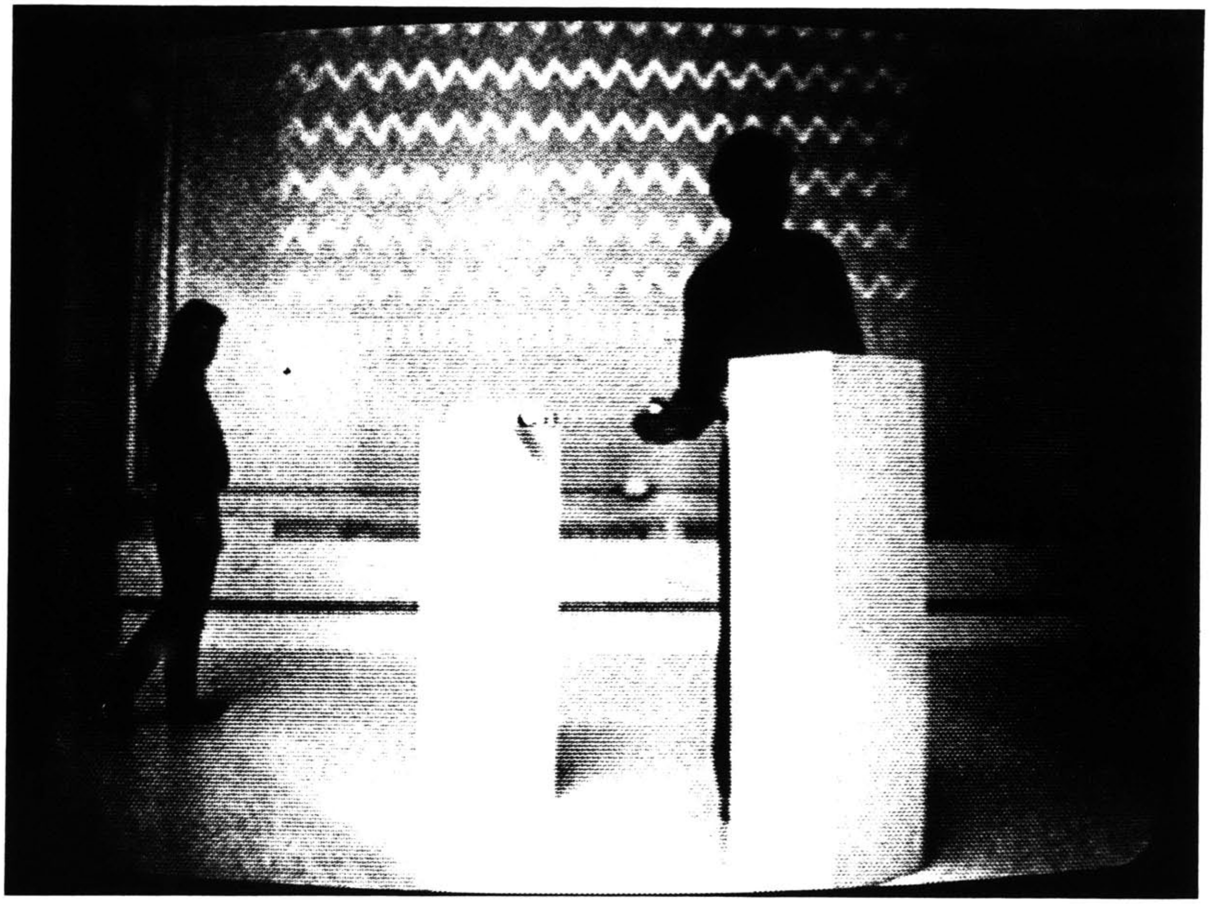
The gates acted as light activated switches. If a visitor passed through a gate, the photocell would detect the absence of light and trigger a relay attached to the computer interface. This information was then used to tell the computer to go back to the main program. When the relay switch was triggered, a random number generator in the main program was used to choose the next subroutine.

The components for the light activated relay circuit were all common, off-the-shelf products purchased at Radio Shack. The relay circuit was connected to an interface board in the Commodore's expansion port. The interface board was manufactured by Proteus Electronics, Belleville, Ohio.

The following photographs show people walking around the exhibition space with the graphic projections behind them. They are also interacting with the light beams by using their hands and bodies. Notice in photographs (a) and (d) the sudden change in graphic patterns in the projection. This is caused by the recent interruption of the light beams. Photographs (a), (b) and (d) are of the large screen and photograph (c) is of the small screen.

(a)

(b)



(c)

(d)

THE PROGRAM

One program controlled the entire exhibition. It created the music, the graphics and responded to the light activated relay switches. Written in Basic, the program consisted of a short main program and twelve subroutines. Flowcharts of the main program and the subroutines follow this section. These flowcharts are unconventional in an effort to present the information more creatively than traditional flowcharts.

The main program assigned variables to addresses in the computer's memory. A random number generator then supplied a number between one and twelve. The number was assigned to a variable which was passed through a series of if/then and goto statements. The goto statements directed the computer to the appropriate subroutine. Each subroutine created a different sequence of graphics and musical sounds.

The program was purposely slow running. This is how certain optical effects and sounds were achieved. The graphics consisted entirely of geometric symbols chosen from the Commodore keyboard graphic character set. By creating loops which contained print statements, then programming musical tones to manipulate the timing between the print statements, optical effects such as after-image and convergence were achieved.

The musical tones were made by 'poking' various numbers to addresses in the computers' memory. These addresses control different aspects of sound. When a number is 'poked' to a specified address a value is written into that memory location. The memory locations 'poked' for this purpose controlled the volume, envelope generator (attack and decay rate, the sustain and release rate), pitch (the high and low frequencies) and waveforms of each note. The Commodore 64 music synthesizer and sound effects generator made it possible to create new musical and noise sounds by simply 'poking' different numbers into the proper memory locations. The sounds I chose for the most part do not correspond with traditional western musical notation. Because the sounds were composed to be played only on the Commodore combined with the graphics, the program serves as the score.

Although the volume was consistently set at the highest level, variations did occur. The envelope generator controlled this with the attack, decay, sustain and release rates. The attack rate is the time it takes a note to rise from zero to its peak volume. The decay rate is the time it takes for a note to fall from peak volume to mid-range or sustain level. The release rate is the time it takes a note to fall from sustain to zero. Figure 2.1 is an illustration of the way in which this worked for the first sound in subroutine 1000. In Voice One the attack rate was zero. This translates to two milliseconds. The decay rate was ten, or one and a half seconds. The sustain was thirteen. This would be nine seconds without the timing loop in line 1185. The timing loop however extends this time. Finally the release rate is two. This translates into forty-eight milliseconds. The attack / decay /

sustain / release rates for Voices Two and Three are also provided.

The sounds and the graphics evolved together. I entered graphic symbols and numbers controlling the sounds into loops on a trial and error basis. If an interesting visual was achieved I found a sound to match it. More often the sounds were programmed first.

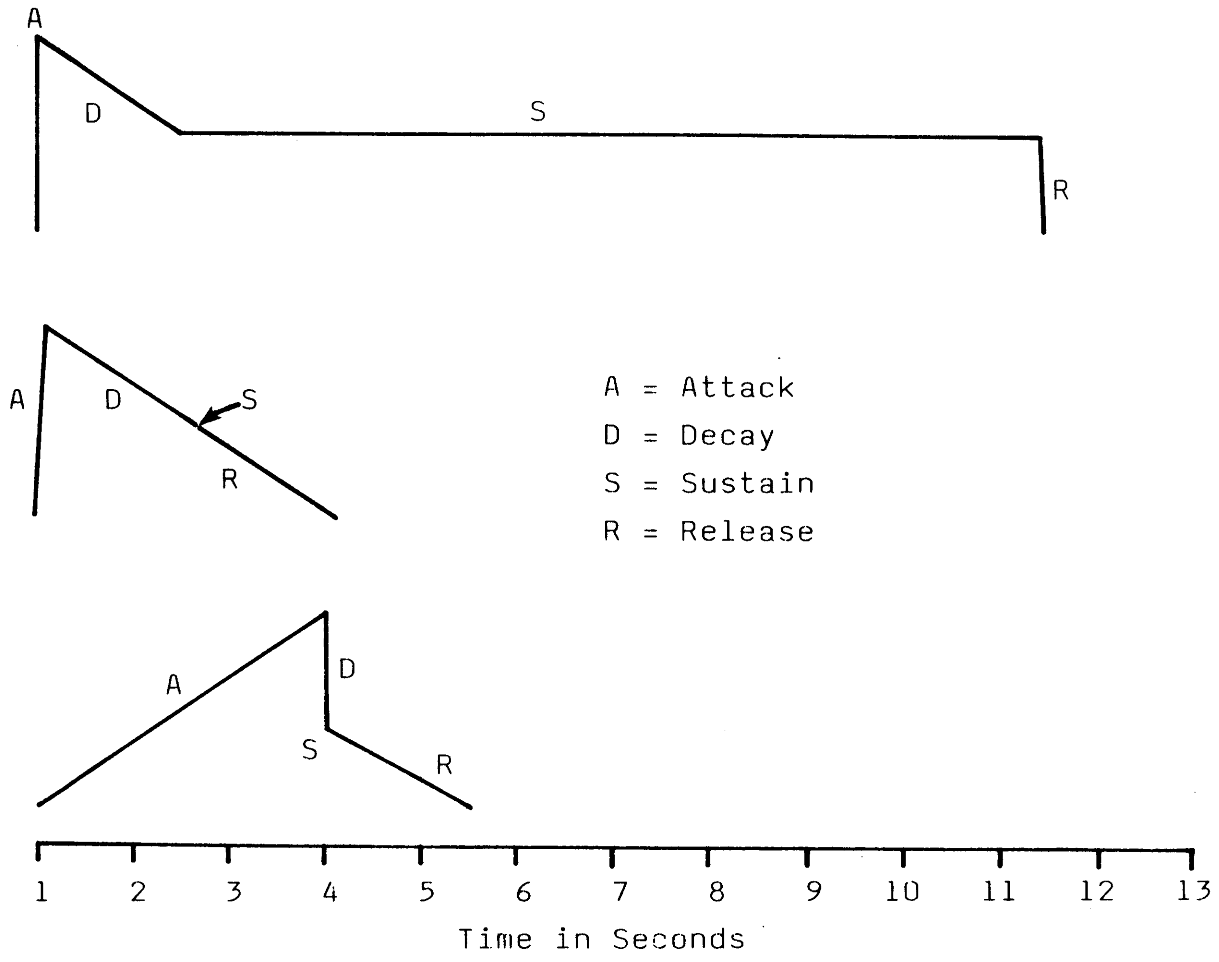
John Cage is perhaps best known for using random elements and chance in his musical compositions. Of chance operations Cage says,

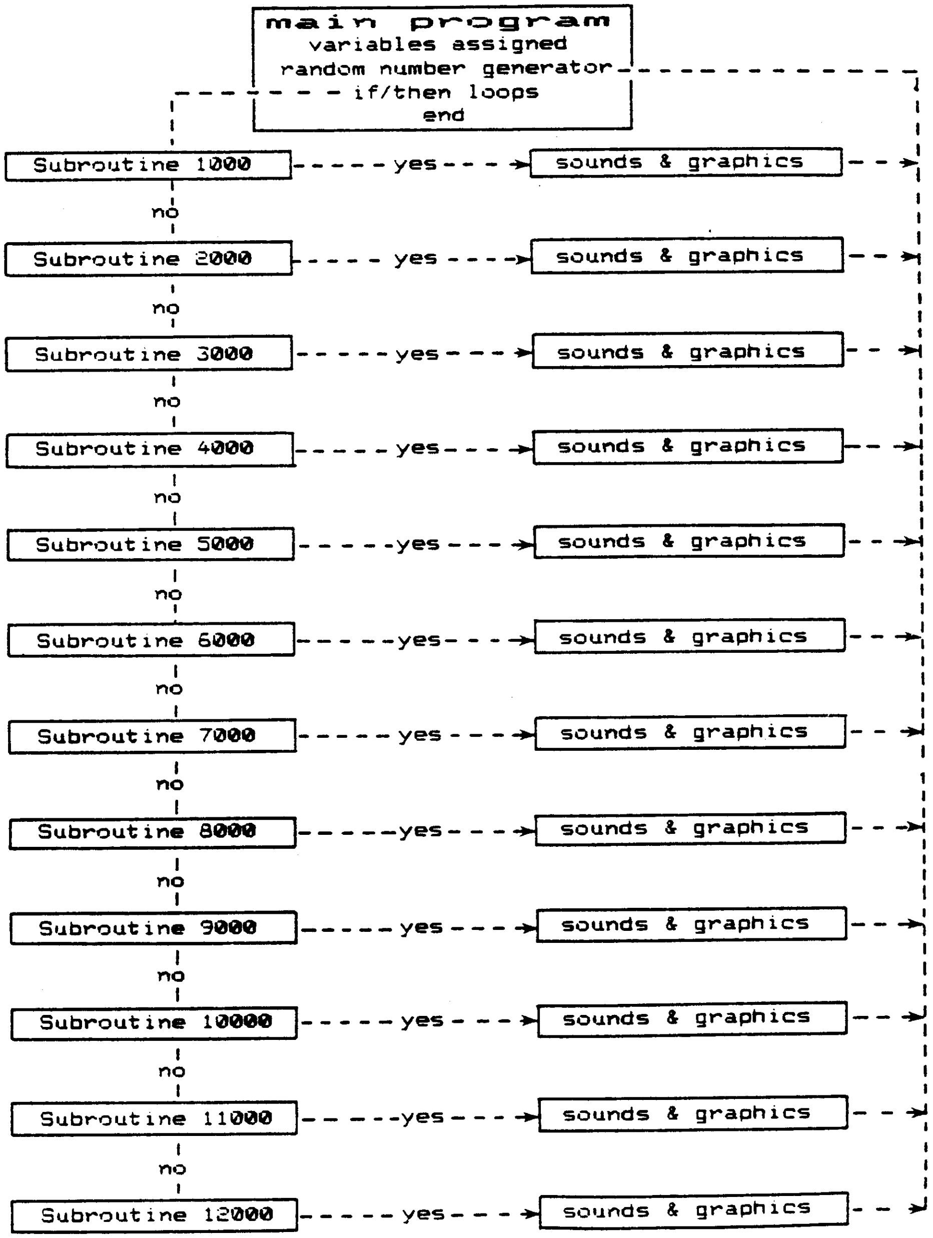
> Chance operations are not the mysterious source of the "right answers". They are a means of locating a single one among a multiplicity of right answers, and at the same time freeing the ego from its taste and memory, its concern for profit and power, of silencing the ego so that the rest of the world has a chance to enter into the ego's own experience whether that be outside or inside. (1)

Influenced by the writing and music of John Cage, I incorporated chance operations into this work. My use of chance operations consisted of using the Random function as part of my programming. This function produces a number between 0.0 and 1.0. The computer generates a sequence of random numbers by performing calculations on a starting number called a 'seed'. The random function is 'seeded' when the computer is turned on. The same 'pseudorandom' sequence of numbers is returned, starting from a given seed value. Different number sequences result from different seeds. Any sequence is repeatable by starting from the same seed number. I used the random function, made it into an integer (whole number, without decimal point) and multiplied the integer to come up with pseudo random variables for the arguments in many of the subroutines.

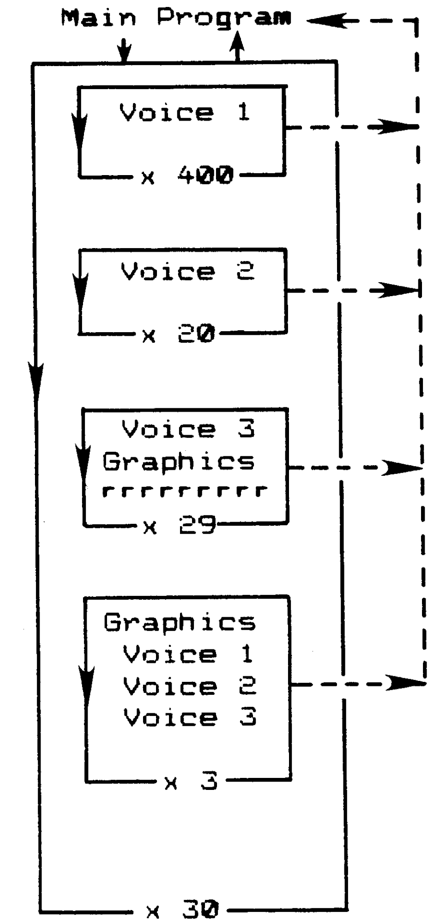
Interaction with the light beams was detected by 'poking' values for the input device in the beginning of each subroutine. These memory locations were then 'peeked', or read at various places throughout each subroutine. If the value had changed the program was instructed to return to the main program.

Figure 2.1 Graph of Sound Envelopes

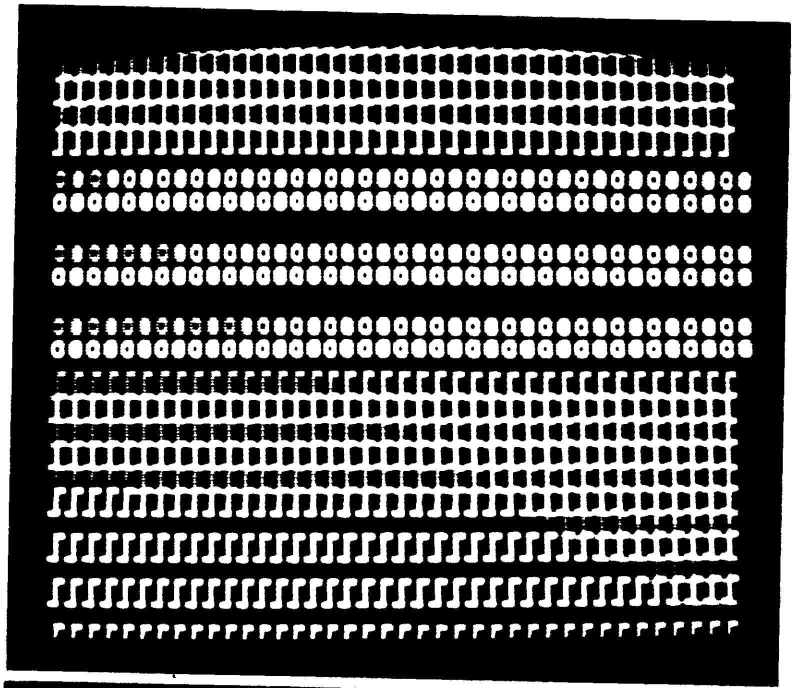


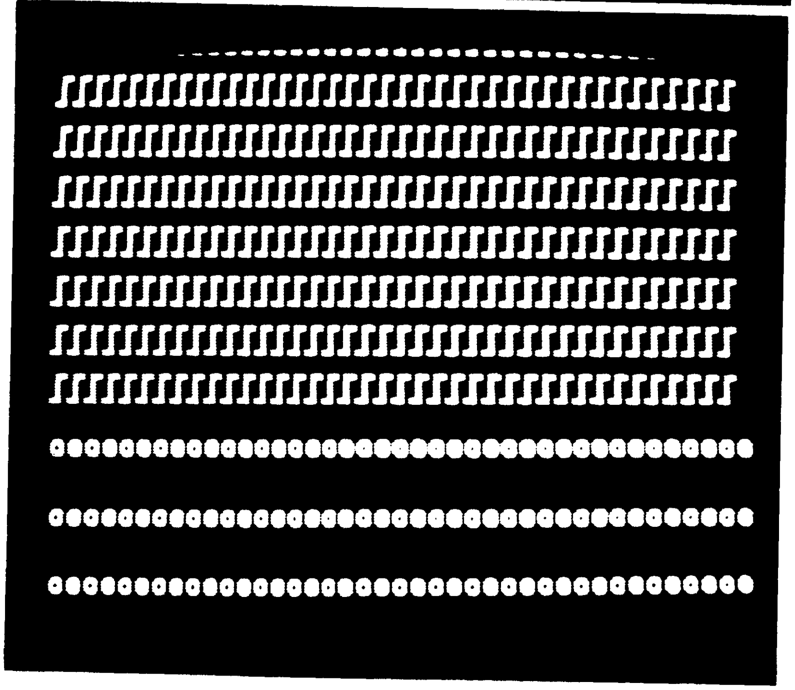

main program
variables assigned
random number generator
if/then loops
end
Subroutine 1000
yes
sounds & graphics
no
Subroutine 2000
yes
sounds & graphics
no
Subroutine 3000
yes
sounds & graphics
no
Subroutine 4000
yes
sounds & graphics
no
Subroutine 5000
yes
sounds & graphics
no
Subroutine 6000
yes
sounds & graphics
no
Subroutine 7000
yes
sounds & graphics
no
Subroutine 8000
yes
sounds & graphics
no
Subroutine 9000
yes
sounds & graphics
no
Subroutine 10000
yes
sounds & graphics
no
Subroutine 11000
yes
sounds & graphics
no
Subroutine 12000
yes
sounds & graphics

SUBROUTINE 1000

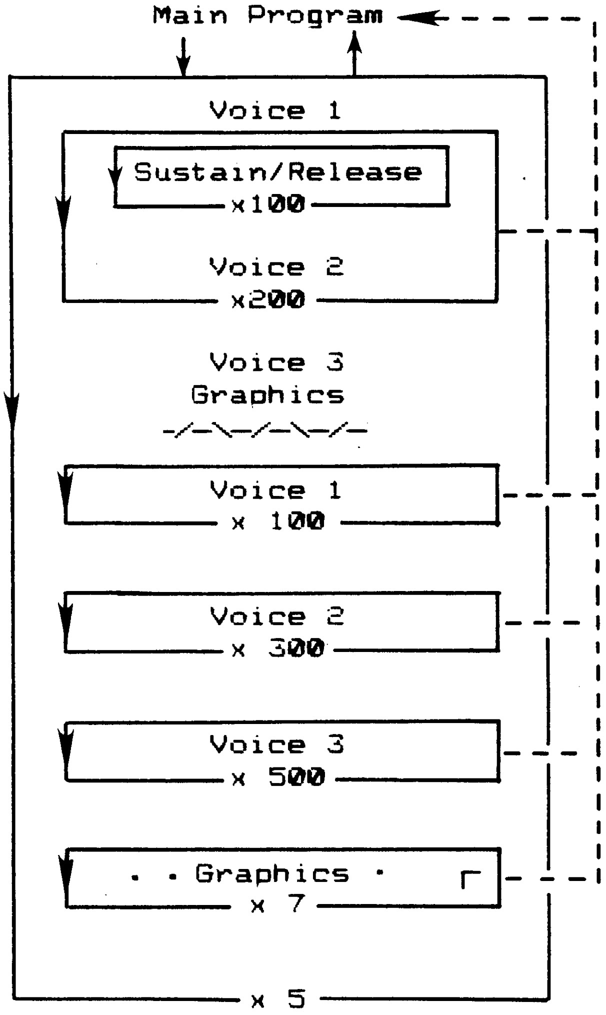


SUBROUTINE 1000

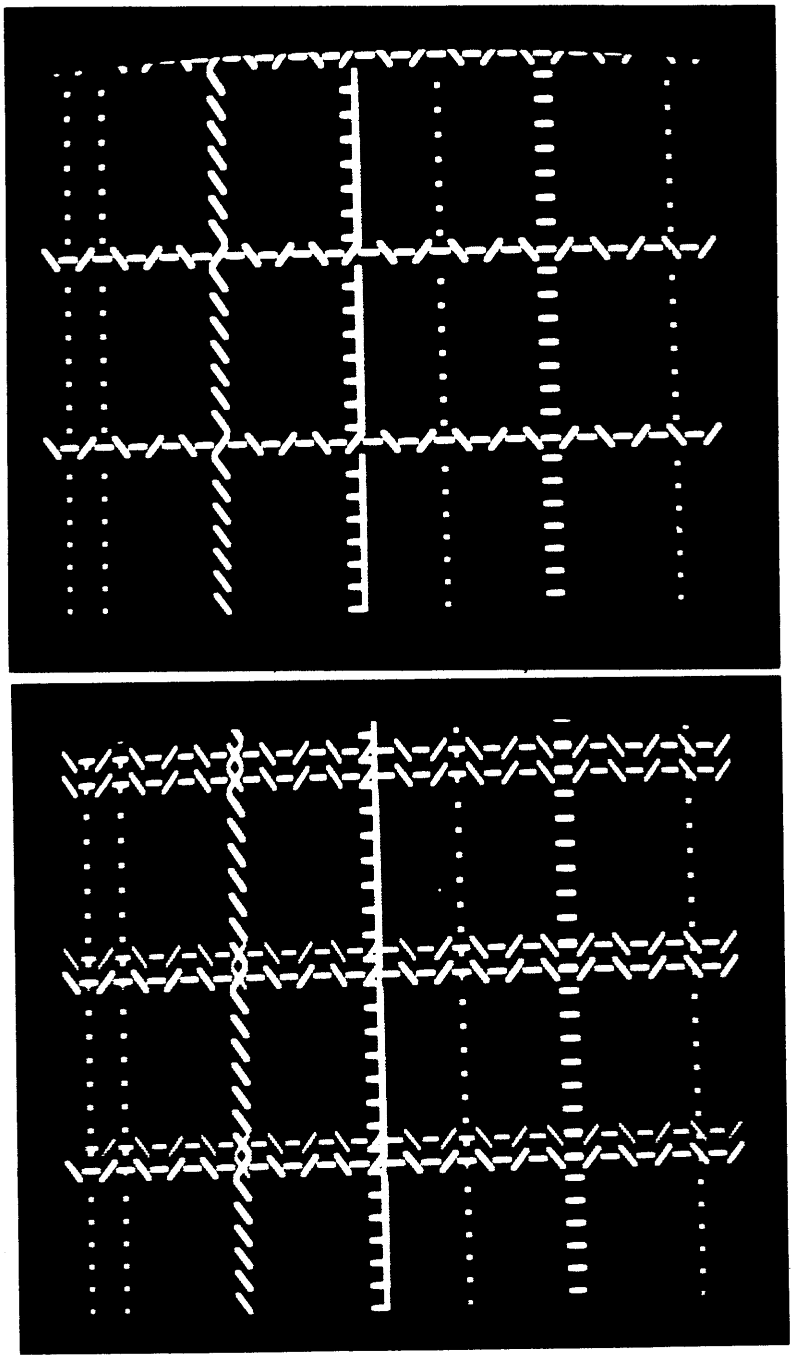




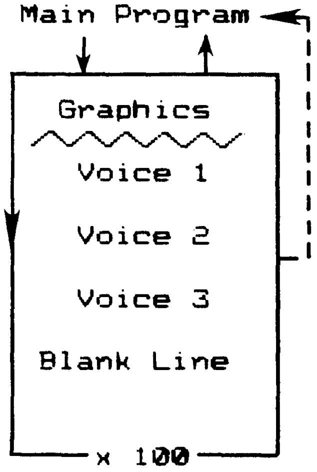
SUBROUTINE 2000



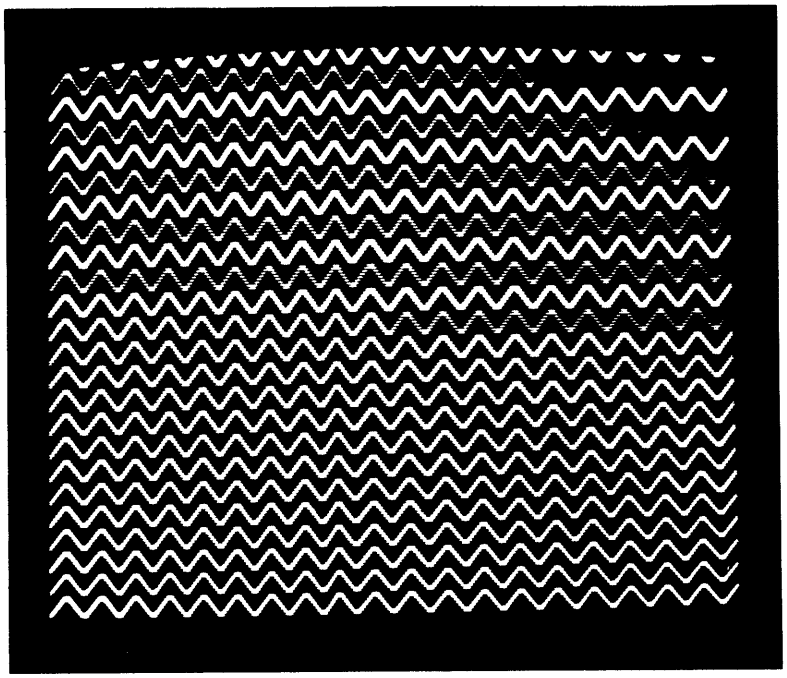
SUBROUTINE 2000

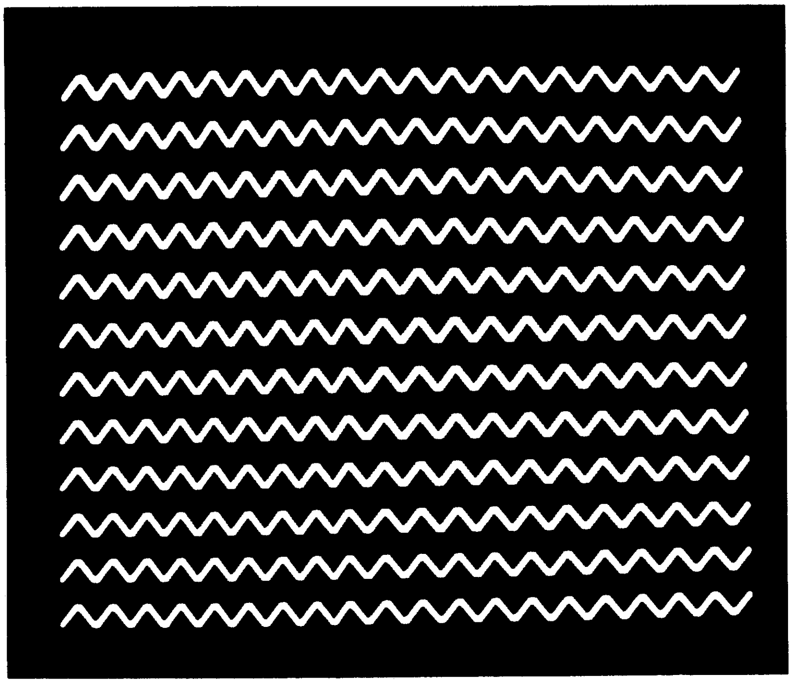


SUBROUTINE 3000

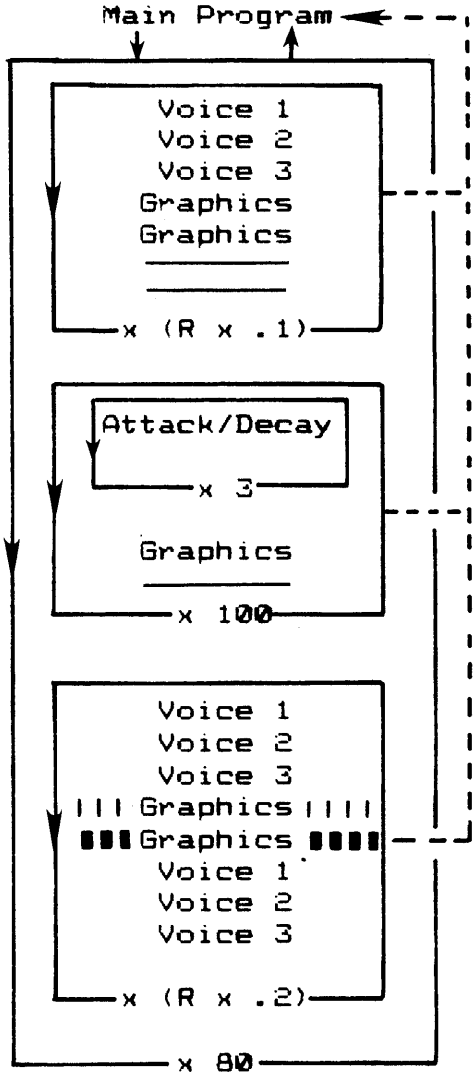


SUBROUTINE 3000



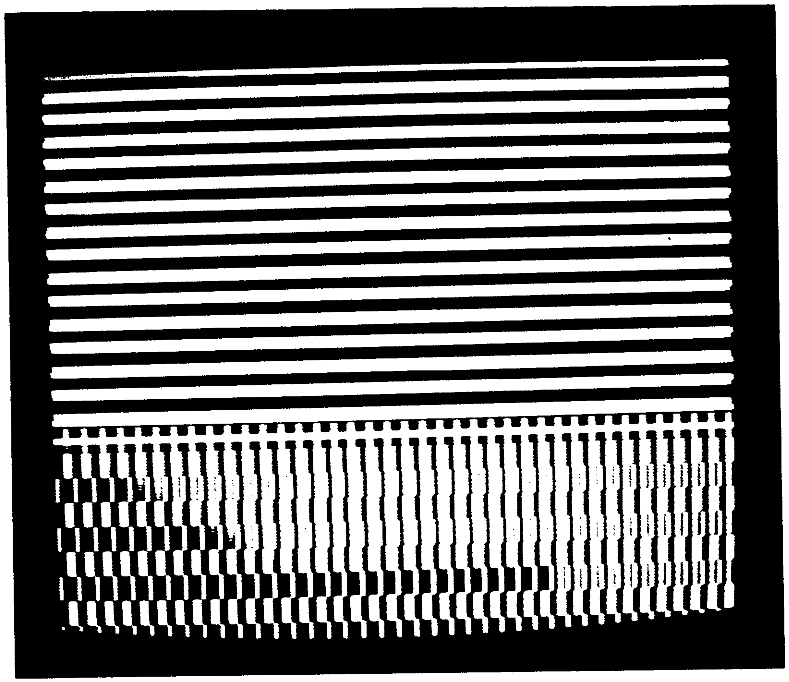


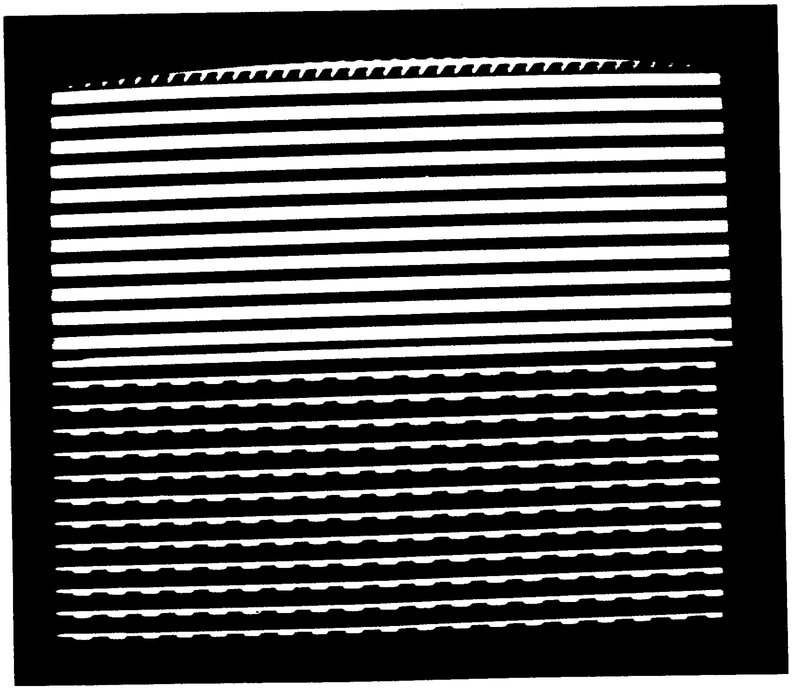
SUBROUTINE 4000



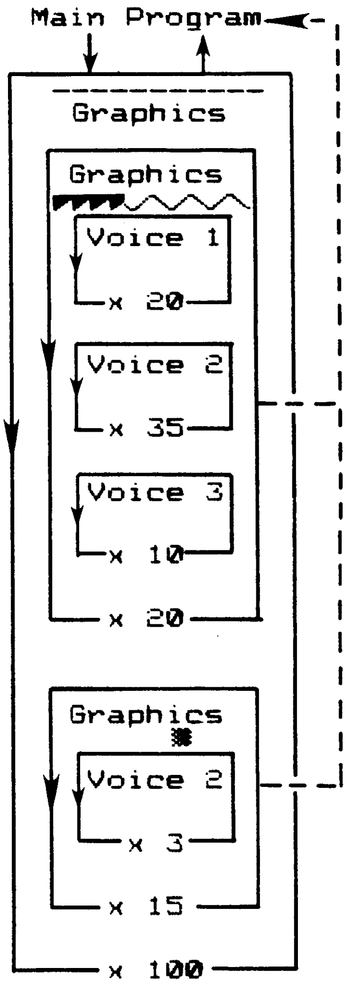
R = a random integer between 1 and 200

SUBROUTINE 4000

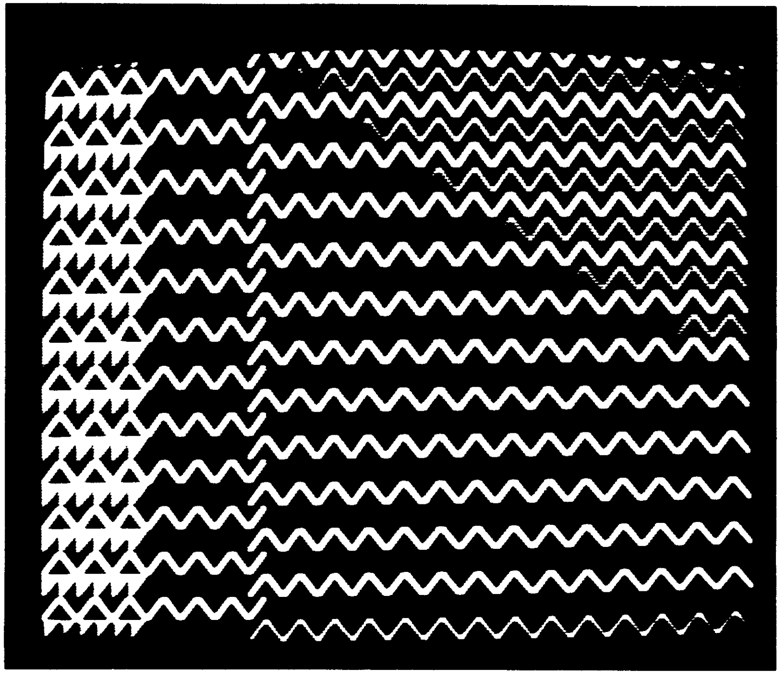


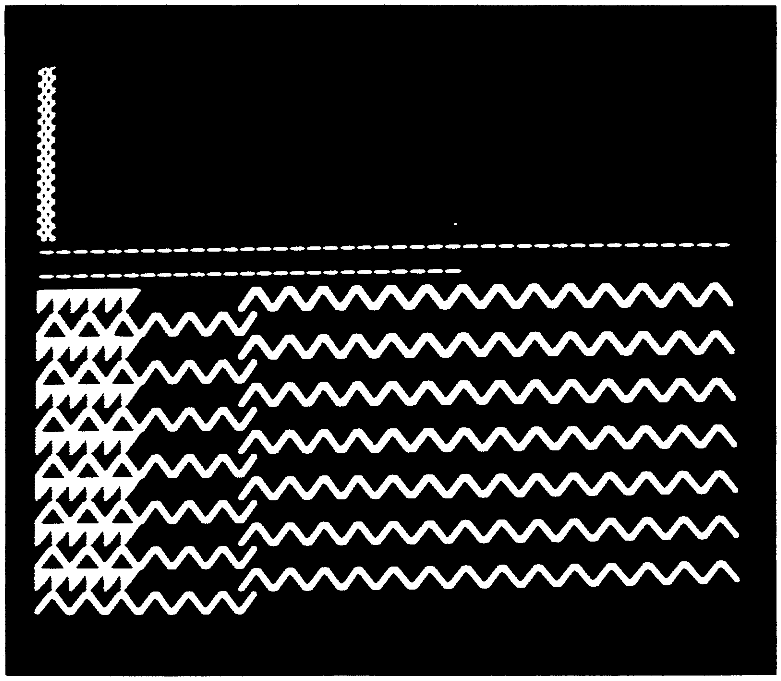


SUBROUTINE 5000

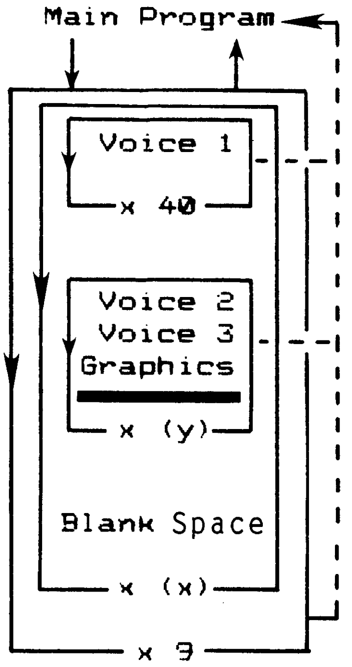


SUBROUTINE 5000



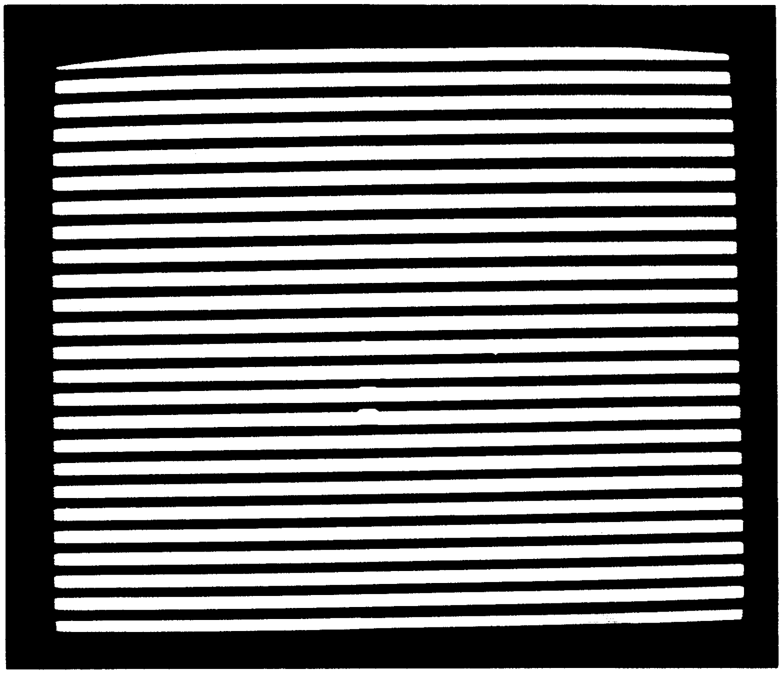


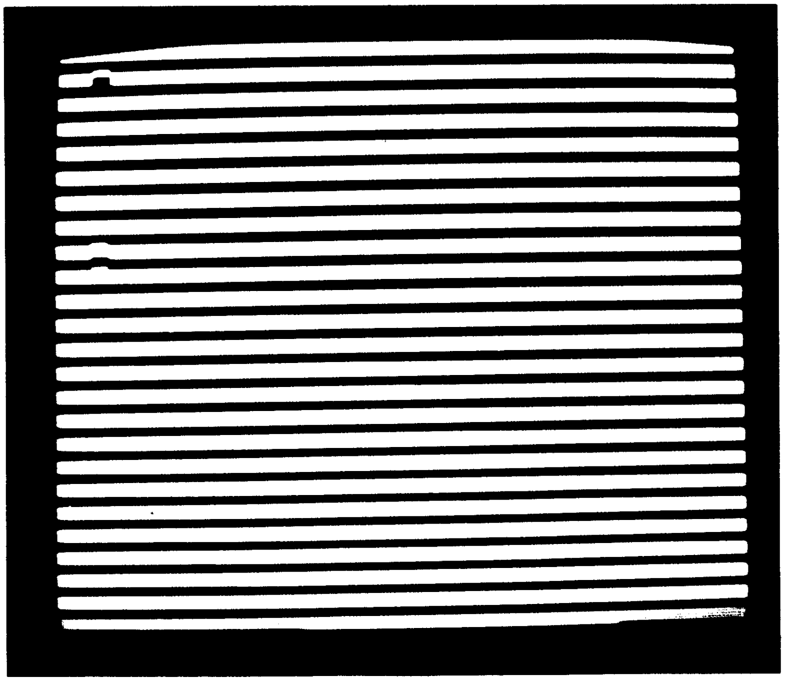
SUBROUTINE
6000



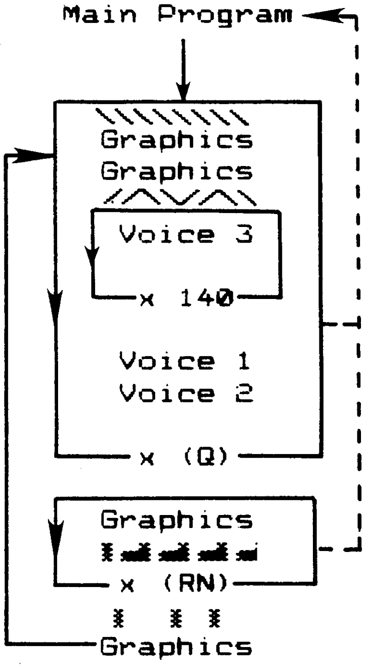
x = a random number between 1 and 39
y = a random number between 1 and 24

SUBROUTINE 6000



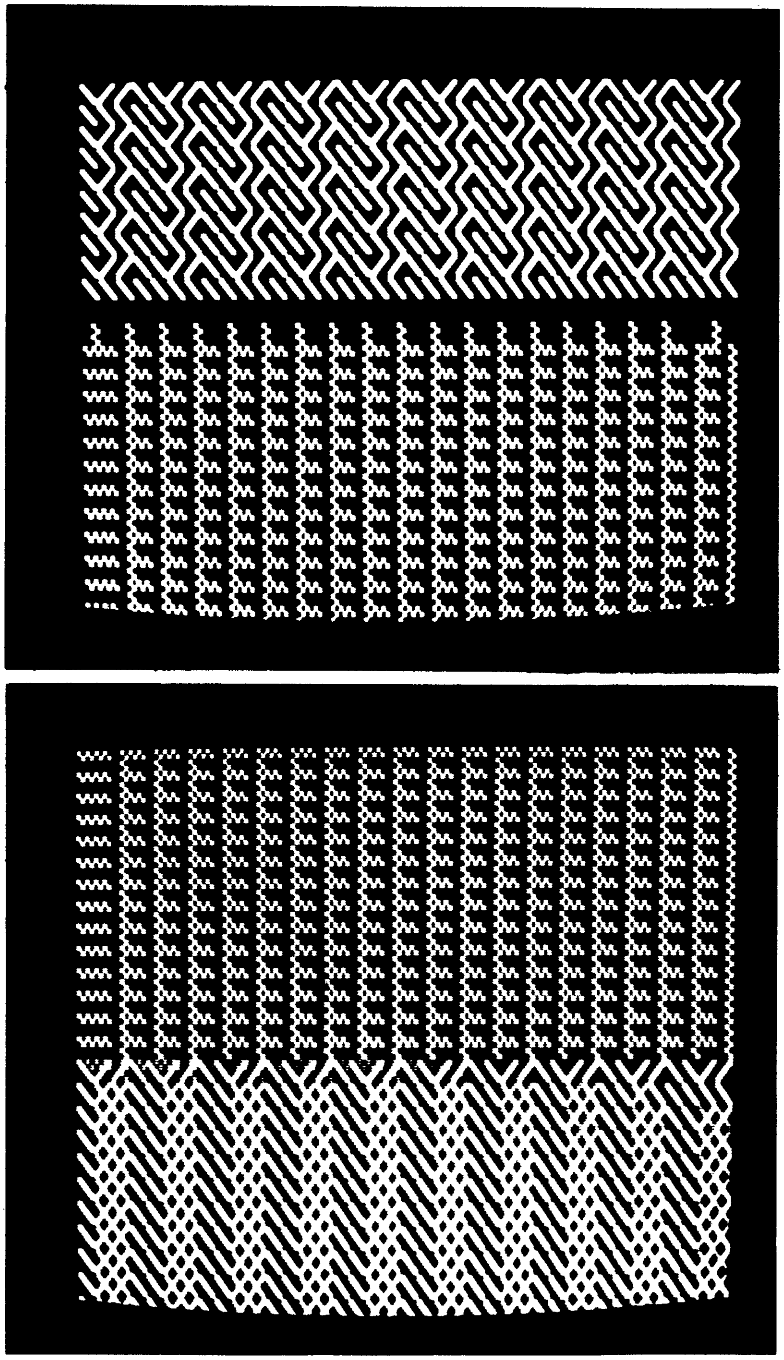


SUBROUTINE 7000

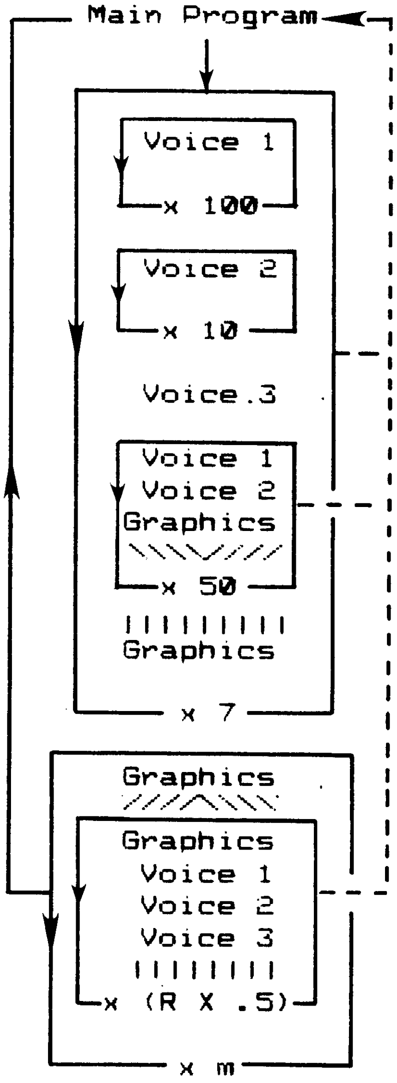


Q = a random number between 1 and 29
RN = a random number between 30 and 100

SUBROUTINE 7000

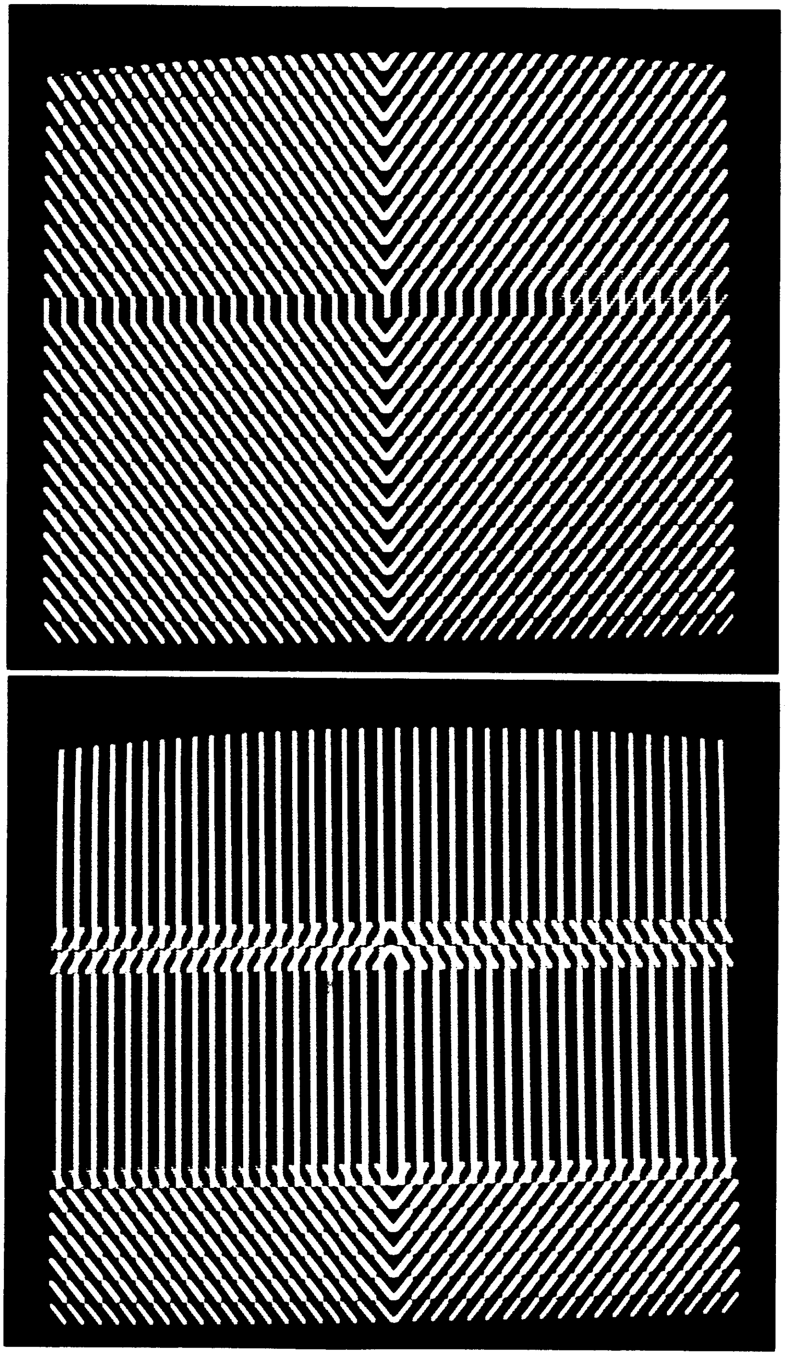


SUBROUTINE 8000

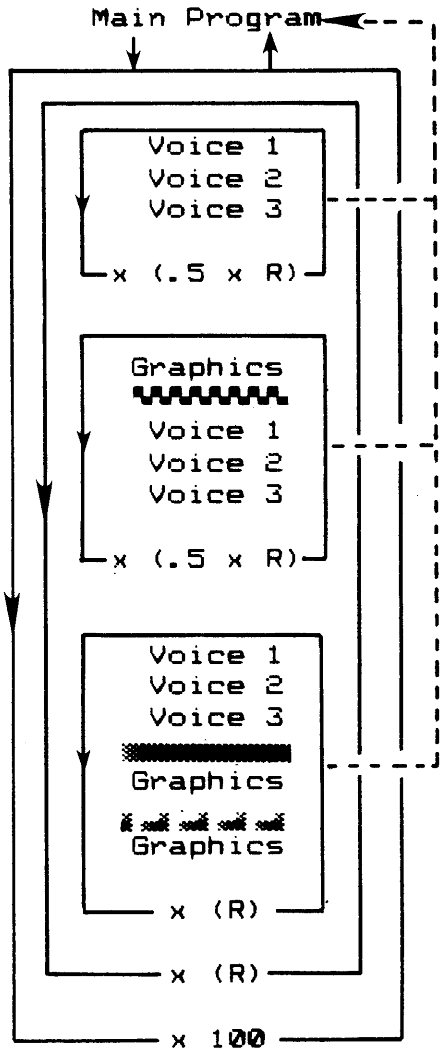


m = a random number between 1 and 12
R = a random number between 1 and 120

SUBROUTINE 8000

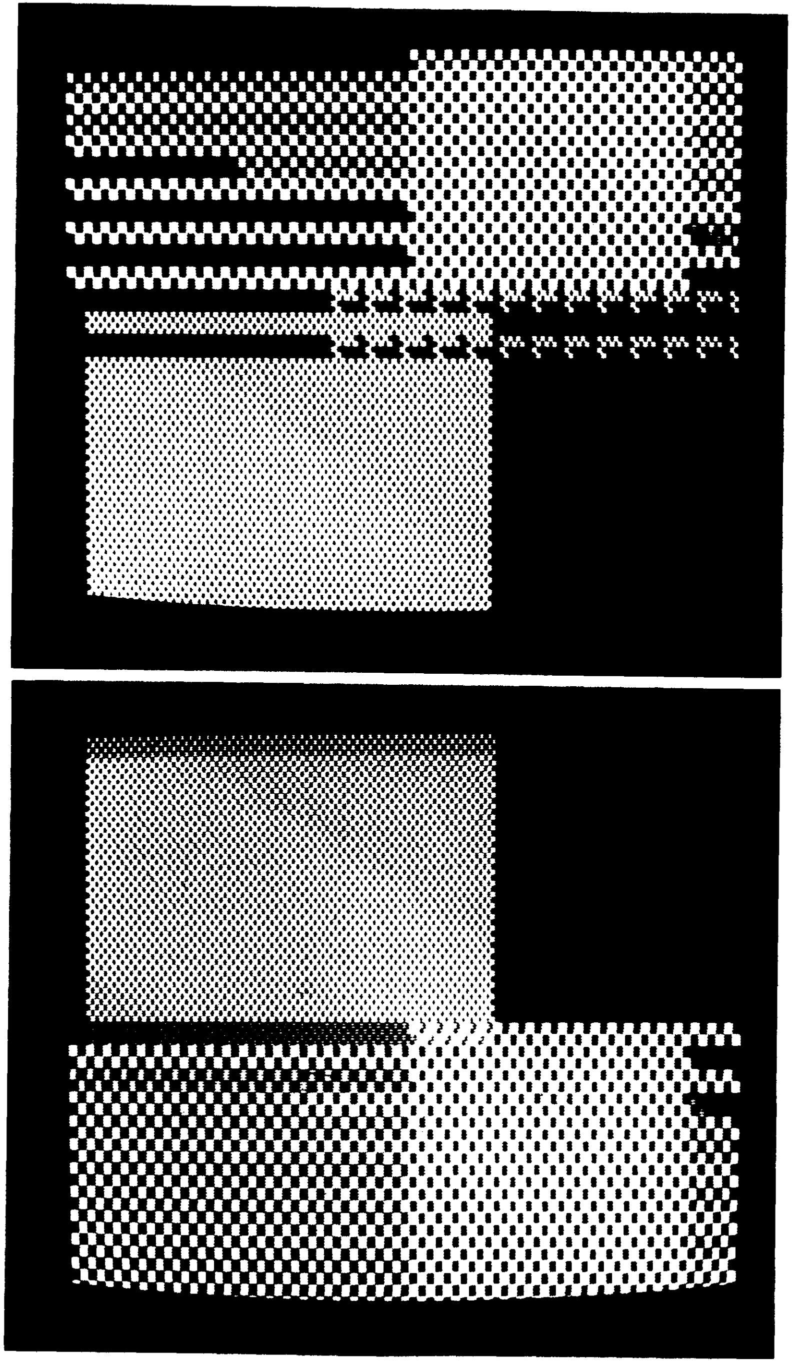


SUBROUTINE 9000

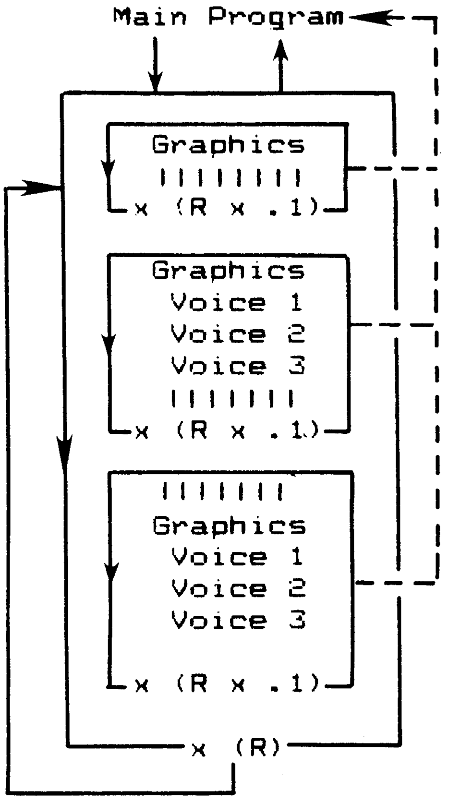


R = a random number between 1 and 260

SUBROUTINE 9000

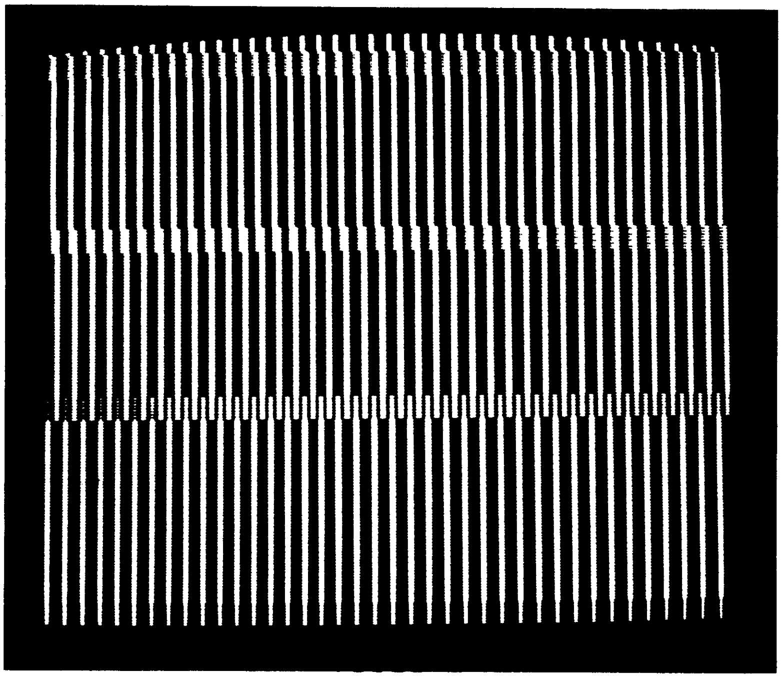


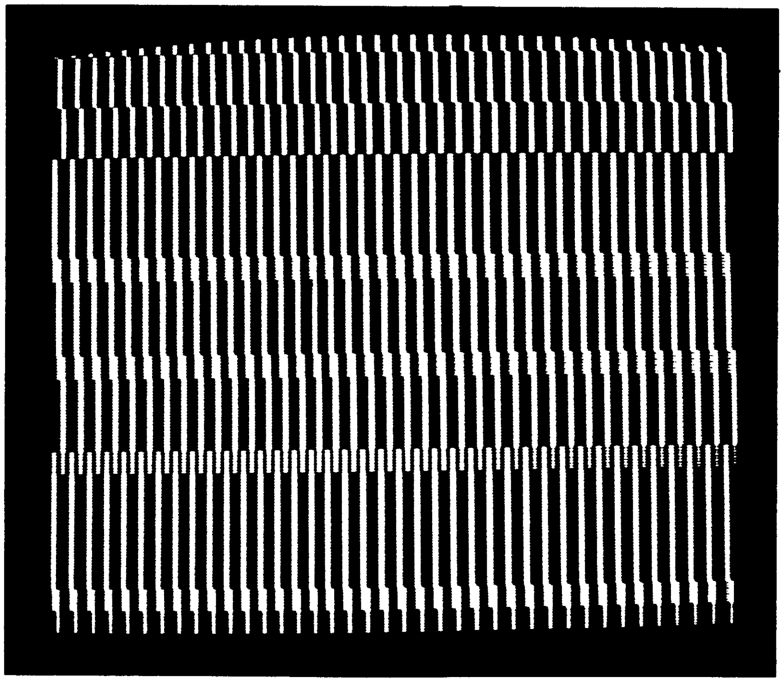
SUBROUTINE 10000



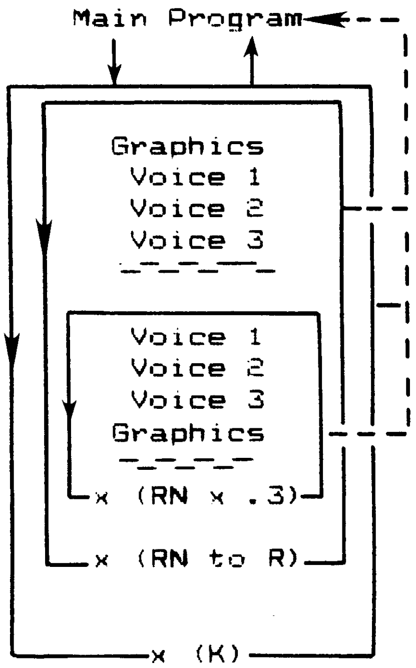
R = a random number between 1 and 220

SUBROUTINE 1ØØØØ



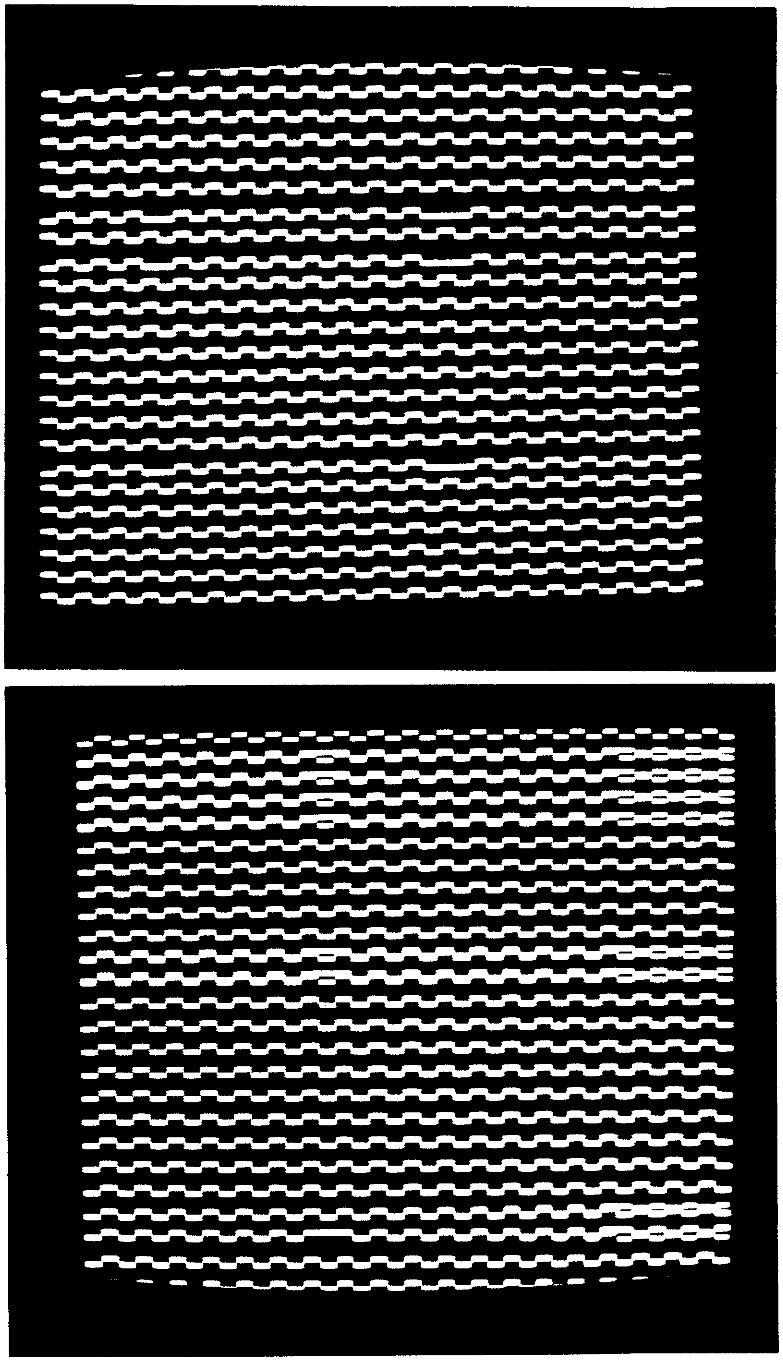


SUBROUTINE
11ØØØ

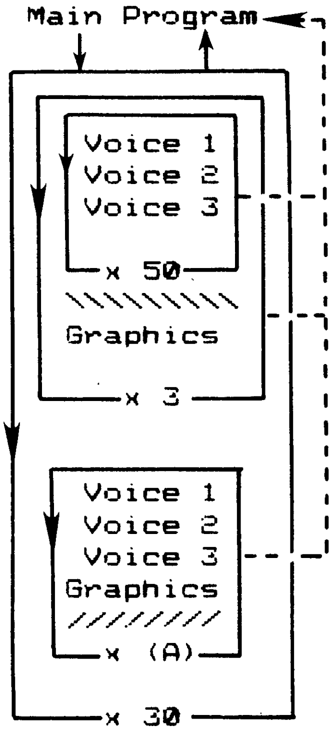


RN = a random number between 1 and 38
R = a random number between 1 and K
K = a random number between 1 and 2Ø

SUBROUTINE 11ØØØ

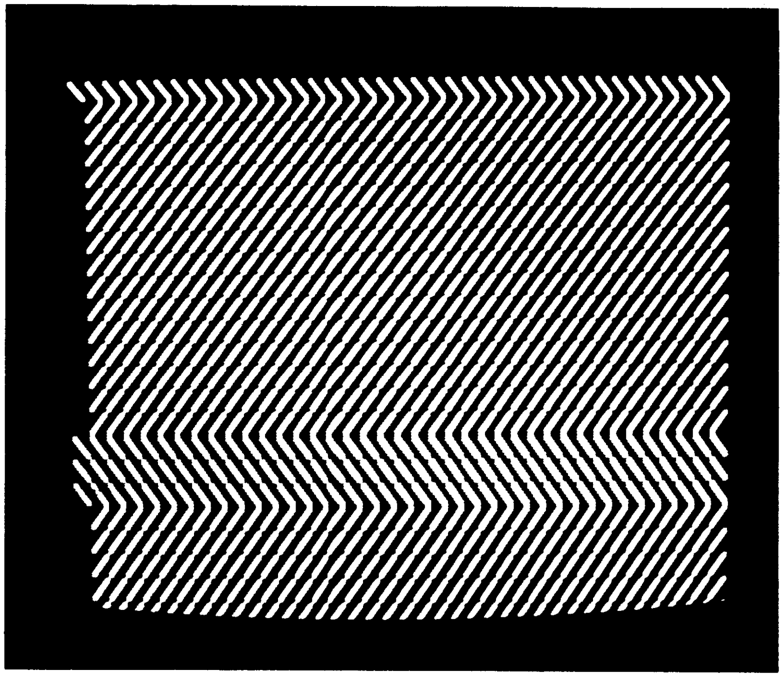


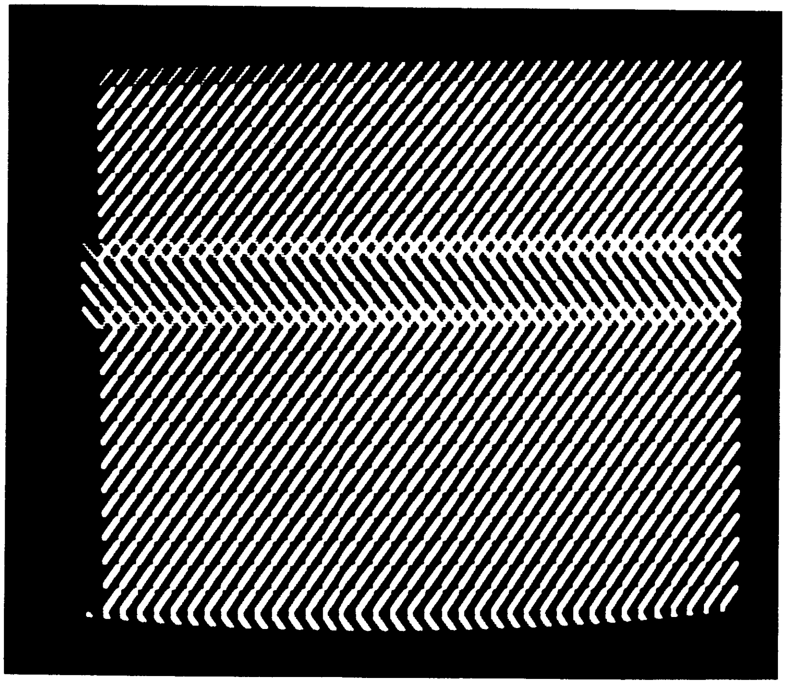
SUBROUTINE
12000



A = a random number between 1 and 50

SUBROUTINE 12000





HISTORICAL PRECEDENTS

> What stranger enterprise
> could be imagined in the
> whole field of art than to
> make sound visible, to make
> available to the eyes those
> many pleasures Music affords
> the ears? (2)

Synergism is "the cooperative action of discrete agencies such that the total effect is greater than the sum of the two effects taken independently."(3)

"Songs in the Language of Information" was an installation which merged graphics, music, light and interactivity in an environmental installation synergistically. Each of the components of this installation has its own history, some of which overlap. In this section I will discuss examples of works by artists and a scientist working in these areas.

Color organs

An early mechanical realization of visual music was the color organ built by Jesuit priest and mathematician, Father Louis-Bertrand Castel in 1734. In this instrument transparent color bands were illuminated by candles regulated by the operating keys and hammers of an organ. Another well-known variation of the color organ was built by an American, Bainbridge Bishop around 1880. In this instrument combinations of colors were projected onto a small screen above it when played. A different color was assigned to each key on the keyboard and as the loudness increased so did the intensity of the colors.

The "color organs" now available at electronic hobby stores work in a different manner. Instead of assigning a different color to each pitch, a filter network divides the audio spectrum into 3 to 4 ranges rather than the direct, one to one correlation between pitch and color as in the old color organs.

Projected Light

The score for Alexander Scriabin's orchestral work,"Prometheus, Poem of Fire", written in 1908, included a stave of notated music marked, "tastier per luce", for light keyboard. Scriabin dreamed of filling the concert hall with colors which would correspond to tonalities and chordal complexes. Unfortunately, the performance of this work did not live up to his hopes due to the scale of the light instrument built for this piece. It was placed behind the orchestra and projected color only onto a small screen which left little impression upon the audience.

Thomas Wilfred began building light instruments at the turn of this century when he put incandescent lights into cigar boxes to produce light effects. By the nineteen-twenties his instruments were equipped with keyboards enabling him to produce moving, colored light effects. Wilfred called his work "Lumia", the art of light. He never developed sound for his instruments for he saw a discrepancy in the perception of sounds and colors. He prefered to make music solely for the eyes.

In the early nineteen-twenties Bauhaus artist Ludwig Hirshfeld-Mack formed a group which experimented with reflected light compositions by superimposing templates of various colors. By moving the templates back and forth in front of a spotlight the audience would see colored and kinetic abstract forms projected on the back of a transparent screen. Hirshfeld-Mack added music to these planned and improvised sequences. It is said that he attempted to find a relationship between acoustical and visual rhythms.

In the late 1950's Otto Piene first publically presented his "Light Ballet". Using hand-held lamps to project light through stencils, he produced an environment of moving light patterns accompanied by sound. In 1969 Piene produced a videotape as part of the WGBH television program, "The Medium is the Medium" entitled, "Electronic Light Ballet". In it he aimed a strong light source through multi-perforated stencils while moving both the camera and stencil. This caused "a sperm-shaped burn-in of intense colors"(4). Some of the images are very similar to those seen in "Songs in the Language of Information", but Piene's simple technique produced variations in color and movement which are infinitely more complex and lyrical than anything yet possible with a computer.

Video Music/Music Video

The "music videos" shown on commercial television networks generally are simple narrative fantasies full of sex and violence which reflect the words or ideas inspired by the music. One of the recurrent themes is gang wars. I feel that these videos promote anti-social behavior and limit the imagination by translating the music into cliches.

Nam June Paik, the noted composer/sculptor/media artist has used the electronic manipulation of television sets to produce numerous sculptures, installations, performances and videotapes which incorporate audio/visual elements. Even though he has studied musical composition, Paik usually uses outside sources for the soundtracks on his videotapes. It is Paik's video scuplture and installations that most inspire me. His 1984 installation at the Boston Institute for Contemporary Art entitled, "BSO and Beyond" incorporated 180 color television monitors playing a tape produced by adding analog and digital effects to a standard video recording of the Boston Symphony orchestra playing the third movement of Beethoven's "Fourth Piano Concerto". In addition to music and images, this mass of television screens gave off an imposing glow of artificial light and electric buzz.

Ron Hays has used video to synthesize electronic archetypes and symbols in order to visualize pre-existing sources of music. One such work is his 1974 production, "Visualization of an Experience Within Music: Prelude and Liebestod from Tristan and Isolde". For this piece of Richard Wagner's music, Hayes developed a brilliantly colored, electronically generated flow of abstract images which he says were "symbols revealed to me through feelings and intuitive knowledge more than from thought and reason." (5)

Vin Grabill rhythmically edits his video source material to create audio/visual patterning which maintains a high degree of sight/sound synchronism. The rapid rate of his edits produce sounds which reflect the rhythms of motors and factories. Grabill says, "The visual element remains a textural reflection of the dominant audio rhythms."

Sight and Sound

Many people have constructed light and sound installations in many different ways. Otto Piene and Group Zero experimented with projected light and sound accompaniment. In Gyorgy Kepes' 1971 "Flame Orchard", flames were transformed by different musical tones. Wen-Ying Tsai has created light and sound installations using vibrating strobe lights. Takis uses electromagnetic forces in the creation of sound sculptures. Liz Phillips employs interactive electronic systems to sense, characterize and synthesize sound structures.

Robert Rauschenberg was one of the founders of E.A.T. (Experiments in Art and Technology). Through this affilliation he collaborated with engineers to add a sound and participatory element to several of his works. One such piece, "Soundings" reacts to voices by illuminating a thirty-six foot long plexiglas wall covered with silkscreened images. Another is a five-part construction entitled "Oracle". Each of the parts of this construction is programmed to emit sounds or music in sequence.

Nicholas Schoffer has experimented with various programmed towers which project light and sounds. An example of one of his large scale light and sound works is the "Formes et Lumieres", constructed in the early 1960s, at the Palais de Congres, Liege. Here he designed a giant screen made up of blinds which are lowered across the facade of the Palais. Three hundred and sixty projectors and numerous spotlights are linked to a complex system of stops, much like an organ. Sounds are emitted from loudspeakers mounted on columns at the front of the projection screen. The effects are produced as a spectacle which lasts about twenty minutes. Schoffer has also built cybernetic sculptures and towers using light and sound.

Paul Earls has composed music which has been perfomed in many different environments. His interest in the heightened total experience provided by the stimulation of multiple senses has led him to compose works for visual as well as musical instruments. One of the systems he has developed for this purpose is a computer driven, analog laser system. Otto Piene supplies drawings which Earls then digitizes and projects. The laser projections are transformed in size and shape by the music. Although the image changes do not always correspond directly to the music, there is an organic relationship between the two due to the analog modulation of the image and the music. Earls' projection surfaces range from steam, water and operatic stage sets to Otto Piene's huge inflatable sculptures.

Graphics

Designs like those projected in "Songs in the Language of Information" have been used decoratively in architecture, textiles, pottery, and book design for centuries. The components of these designs often carry symbolic meanings. Zig-zag lines for example, are the Eqyptian hieroglyphic character sign for water. In the West the simple "V" shape of the chevron has symbolized ownership and rank since the Middle Ages. Islamic mathematicians, as far back as the thirteenth century A.D., have interpreted the cosmic order of the universe through Magic Squares. These symbols are now also emerging as part of computer graphics.

The term "computer graphics" was coined in 1960 by Boeing Company to describe a flight simulation program developed to determine the pilot's position in the cockpit (6). Since that time many artists and scientists have experimented with this medium. A.Michael Noll, Charles Csuri and John Whitney Sr. were early computer graphic pioneers. During the mid-sixties Stan VanDerBeek worked with Ken Knowlton at Bell Labs to create computer graphic movies with mosaic-like imagery. The development of inexpensive personal computers has now made it possible for artists to produce computer graphics at home.

For a number of years Ed Emshwiller has combined computer graphics with electronically generated sound and produced densely textured videotapes. He has recently completed a new work entitled "Skin Matrix". This piece has a very dense soundtrack and imagery yet it was produced using much simpler technology. He wrote a number of graphics programs on a $50. computer, then used the blockiness of the low resolution computer graphics to evoke a sense of primitive art forms. By keying video of different masks, faces and textures onto the computer graphics he achieved a layered effect with rich sound accompaniment.

ANALYSIS OF WORK

The intent of the exhibition was to investigate the use of multiple computers to create expanded visual and aural experiences. Two Commodore 64 computers were employed for this purpose. A series of subroutines were programmed to produce scrolling patterns of light. Programmed sounds emphasized the movement of the patterns. The sounds were composed to be joined in a number of different ways and and each computer projected scrolling patterns, making two continually changing tapestries of video light.

The installation presented viewers with an environment undergoing cyclical changes. Patterns of sound and projected light were repeated at intervals set by the computer ranging from several seconds in length to loops which would end only with human interaction. At the end of an interval, the computer would randomly select the next cycle. Viewers could trigger the selection of another cycle prematurely by walking through the light beams but they could not select what the next cycle would be.

Randomness was used to model the unexpected and the unpredictable. All life situations have some forces which we cannot predict or control. I wanted this work to reflect both the cyclical and the unpredictable nature of life. Changes in the cycles initiated by participants and the computers themselves were controlled by random elements which influenced the musical and visual composition of this work.

I used video projectors to address the entire physical space on an environmental scale. They also minimized the associations with television and video games. I placed the projectors overhead on two existing platforms. This left the floor area free of everything but the pedestals and light beams, thus emphasizing the sounds, graphics and interaction rather than the mechanics of the installation.

By projecting the video patterns onto walls, rather than onto scrims or screens, the moving light patterns became part of the wall surface. The projections occupied two walls at right angles to each other. These walls were unbroken by stairways or other passageways and windows.

I used Commodore computers as tools because I wanted to employ the video game technology of the Commodore to create an environment of subtlety and beauty rather than violence and the voyeuristic destruction that is typical of video games. The Commodore is equipped with a sound synthesizer and an alternate keyboard providing sixty-six graphic symbols.

The Commodore graphic symbols can be positioned in various ways to compose pictures or patterns. Combining these symbols, I created patterns similar to decorative ornamentation in architecture, textiles, pottery and book design. The components of these patterns carry individual symbolic meanings. I used computer technology to layer them into designs and added sound and movement in an attempt to translate symbols from the past into symbols of the present.

The relationship between the sound and visual elements in the exhibition differed slightly with each subroutine. In some subroutines the sounds changed according to the duration of the subroutine loop. In other subroutines the sounds and images remained constant. All of the sounds and patterns were created by repeating a series of commands a set number of times. The patterns scrolled from the top to the bottom of the screen as a symbol of the passing of time.

The Commodore sound synthesizer made it possible to create a wide variety of possible sounds. Selecting the sounds was an arbitrary intuitive process. As I described in the chapter entitled, "The Program", the Commodore made sounds by reading numerical values from specific locations in the computers' memory. Sometimes I used the random variable which determined the duration of the loop to set the attack, decay, sustain and release (envelope parameters), and the high or low frequency (pitch), and the duration of each note. Pitch, duration, and envelope parameters, therefore, were directly related to the length of each cycle. Other sounds were always fixed.

The Commodore 64 computer offers a choice of sixteen colors ranging from black to white. The colors for the background, border and characters must be defined by poking a specific value from zero to sixteen into the appropriate address in the computers' memory. Border, screen and character colors can be changed at any point in the program. I defined the background and border colors to be black and the character color to be white throughout this program. This maximized the contrast in black and white or color video. It also made the visual transitions between subroutines more fluid. A low light level was maintained for the duration of the exhibition to further increase the contrast in the black and white projected images.

The floor and pedestals were painted white in order to reflect light and to add a floating quality to the room, making it a space outside of time.

CONCLUSION

"Songs in the Language of Information" for me is both a beginning and an end. It was the end of a long struggle to learn to think simultaneously like a machine and an artist. It is the beginning of my efforts to use the power of information technology to produce works of beauty and inspiration on an environmental scale.

The work discussed in this thesis concerns patterns in cycles of time and the unpredictable events which cause fluctuations within the cycles. To me it was like a series of thoughts trapped in a space of time, bouncing between each other, examining muliple ways of being.

"Songs in the Language of Information " was a continually changing light and sound environment. Although some aspects of the fluctuations within the cycles were random, the number of cycles never changed, nor did their individual character. The system continually changed, but it never evolved.

Using artificial intelligence techniques, artist Harold Cohen and others have developed computer programs which do evolve. Cohen's programs, for example, tell pen-plotters and turtles (motorized drawing machines which are able to crawl around large areas) to make drawings. The machines create one drawing after another, each one unique.

This progression is made possible through recursion. Instead of repeating tasks in loops like in "Songs in the Language of Information", recursive programs progress by defining elements in terms of simpler versions of themselves. Examples of recursion are boxes inside of boxes, stories inside of stories, or faces inside of faces.

In the future I will combine recursive elements with the principles described in this thesis to create light and sound installations which will slowly progress and evolve through stimulation by the outside environment.

The power that computer technology gives us to manipulate information is seductive. The computer gives artists the ability to economically combine interdisciplinary elements in complex ways to produce works of art. I want my work to both reflect and transcend the process.

I will end with a quote from Luigi Russolo's 1913 Futurist Manifesto, "The Art of Noises".

> The variety of noises is infinite. Today, when we have perhaps a thousand different machines, we can distinguish a thousand different noises, tomorrow, as new machines multiply, we will be able to distinguish ten, twenty, or THIRTY THOUSAND DIFFERENT NOISES, NOT MERELY IN A SIMPLY IMITATIVE WAY, BUT TO COMBINE THEM ACCORDING TO OUR IMAGINATION. (7)

Appendix

The Program

```
1 REM MAIN PROGRAM/ ASSIGNS VARIABLES/ RANDOM NUMBER GENERATOR/ GOTO LOOPS
2 REM LINE 2 ASSIGNS INPUT/OUTPUT VARIABLES
3 ICR=56833:OCR=56832:OD=56834
5 POKE 53280,0:REM SETS BORDER COLOR TO BLACK
6 POKE 53281,0:REM SETS BACKGROUND COLOR TO BLACK
7 REM LINES 9-50 ASSIGN SOUND VARIABLES TO MEMORY LOCATIONS
9 VOL=54296:REM POKE SETTING FOR VOLUME
10 W1=54276:W2=54283:W3=54290:REM POKE SETTINGS FOR WAVEFORM
15 AD1=54277:AD2=54284:AD3=54291:REM POKE SETTINGS FOR ATTACK/DECAY
20 SR1=54278:SR2=54285:SR3=54292:REM POKE SETTINGS FOR SUSTAIN/RELEASE
40 H1=54273:H2=54280:H3=54287:REM HIGH VOICE SETTINGS
50 L1=54272:L2=54279:L3=54286:REM LOW VOICE SETTINGS
55 REM LINE 57 GENERATED RANDOM INTEGER FROM 1-12 AND ASSIGNS IT TO "NUM"
57 NUM=INT(RND(1)*(13-1)+1)
90 REM LINES 100-123 SEND COMPUTER TO APPROPRIATE SUBROUTINE
91 REM IF NUM EQUAL TO NUM CHOSEN IN LINE 57 COMPUTERS GOES TO THAT SUBROUTINE
92 REM IF NUM NOT EQUAL TO NUMBER THEN PROGRAM CONTINUES
100 IF NUM<>1 THEN GOTO 102
101 GOSUB 1000
102 IF NUM<>2 THEN GOTO 104
103 GOSUB 2000
104 IF NUM<>3 THEN GOTO 106
105 GOSUB 3000
106 IF NUM<>4 THEN GOTO 108
107 GOSUB 4000
108 IF NUM<>5 THEN GOTO 110
109 GOSUB 5000
110 IF NUM<>6 THEN GOTO 112
111 GOSUB 6000
112 IF NUM<>7 THEN GOTO 114
113 GOSUB 7000
114 IF NUM<>8 THEN GOTO 116
115 GOSUB 8000
116 IF NUM<>9 THEN GOTO 118
117 GOSUB 9000
118 IF NUM<>10 THEN GOTO 120
119 GOSUB 10000
120 IF NUM<>11 THEN GOTO 122
121 GOSUB 11000
122 IF NUM<>12 THEN GOTO 126
123 GOSUB 12000
125 REM LINE 126 SENDS COMPUTER BACK TO RANDOM NUMBER GENERATOR
126 INPUT X,X
127 IF X=1 THEN GOTO 200
130 GOTO 57
200 END
```

```
1000 REM SUBROUTINE 100 CREATES REPEATING SOUNDS AND GRAPHICS
1150 REM LINES 1156-1157 INITIALIZE THE INPUT DEVICE
1156 POKE ICR,0:POKE ID,0:POKE ICR.4
1157 POKE OCR,0:POKE OD,255:POKE OCR.4
1158 REM LINES 1159-1950 ARE "X" LOOP WHICH REPEATS SOUNDS AND GRAPHICS
1159 FOR X= 1 TO 30
1160 POKE VOL,15
1162 REM LINES 1160-1195 POKE MEMORY ADDRESSES TO CREATE SOUNDS
1165 POKE AD1,10:POKE AD2,90:POKE AD3.200
1170 POKE SR1,200:POKE SR2,10:POKE SR3.10
1175 POKE H1,45:POKE H2,90:POKE H3,33
1180 POKE L1,20:POKE L2,69:POKE L3.90
1185 POKE W1,16:FOR A= 1 TO 400:NEXT:POKE W1,17
1190 POKE W2,16:FOR A=1 TO 20:NEXT:POKE W2,17
1193 REM LINE 1195 POKES VOICE 3 TRIANGLE WAVEFORM AND BEGINS "A" LOOP
1194 REM "A" LOOP CREATES REPEATING SOUNDS AND GRAPHICS
1195 POKE W3,16:FOR A=1 TO 29:POKE W3,17
1197 REM 1198 CHECKS INPUT DEVICE
1198 REM 1200-1201 PRINT LINES OF GRAPHIC CHARACTERS
1200 PRINT" ┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌┌"
1201 PRINT"┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘┘"
1210 REM IF PHOTOCELL GETS NOT LIGHT 1220 SENDS COMPUTER TO MAIN PROGRAM
1220 IF B<>0 THEN RETURN
1229 REM 1230 IS THE END OF "A" LOOP
1230 NEXT A
1300 REM LINES 1350-1930 ARE "N" LOOP WHICH REPEATS SOUNDS AND GRAPHICS
1350 FOR N=1 TO 3
1390 REM 1400 PRINTS BLANK LINE FOR SPACING
1400 PRINT"                                      "
1420 REM 1425 PRINTS STRING OF GRAPHIC CHARACTERS
1425 PRINT"○●○●○●○●○●○●○●○●○●○●○●○●○●○●○●○●○●○●"
1440 REM LINES 1450-1900 POKE MEMORY ADDRESSES TO CREATE SOUNDS
1450 POKE AD1,200:POKE AD2,250:POKE AD3,230
1550 POKE SR1,100:POKE SR2,100:POKE SR3,80
1600 POKE H1,17:POKE H2,61:POKE H3.54
1700 POKE L1,85:POKE L2,126:POKE L3.169
1800 POKE W1,33:POKE W2,33:POKE W3,16
1900 POKE W1,32:POKE W2,32:POKE W3,17
1918 REM IF NO LIGHT IS ON PHOTOCELL, 1920 SENDS COMPUTER TO MAIN PROGRAM
1920 IF B<>0 THEN GOTO 1975
1925 REM 1930 END "N" LOOP & 1950 ENDS "X" LOOP
1930 NEXT N
1950 NEXT X
1970 REM LINE 1975 RETURNS THE COMPUTER TO THE MAIN PROGRAM
1975 RETURN
```

```
2000 REM SUB 2000
2002 REM 2005 & 2040 INITIALIZE INPUT DEVICE
2005 POKE ICR,0:POKE ID,0:POKE ICR,4
2040 POKE OCR,0:POKE OD.255:POKE OCR,4
2043 REM LOOP "L", FROM 2045-2430, POKES EACH ADDRESS ON SOUND CHIP
2045 FOR L=54272 TO 54296
2047 REM "M" LOOP IS FROM 2050-2850
2050 FOR M= 1 TO 5
2100 REM 2250 POKES VOLUME TO HIGHEST LEVEL
2250 POKE VOL,15
2275 B=PEEK(ID):B=B AND (1)
2290 REM 2300 POKES ATTACK/DECAY
2300 POKE AD1,120:POKE AD2,200:POKE AD3,130
2349 REM 2350 POKES SUSTAIN/RELEASE FOR VOICE 1,2,3 WITH TIMING LOOP FOR VOICE
2350 POKE SR1,64:POKE SR2,200:FOR A=1 TO 100:NEXT :POKE SR3.90
2359 REM SAME AS 2260
2360 IF B()0 THEN RETURN
2399 REM 2400 SETS HIGH FREQUENCY FOR ALL THREE VOICES
2400 POKE H1,57:POKE H2,72:POKE H3,34
2429 REM 2430 POKES 200 TO EACH POINT ON THE SOUND CHIP
2430 POKEL,200:NEXT
2435 REM 2450 SAME AS 2260
2449 REM 2450 POKES LOW FREQUENCIES FOR ALL 3 VOICES
2450 POKE L1,33:POKE L2,100:POKEL3,120
2499 REM 2500 POKES TRIANGLE WAVEFORM TO VOICE 1
2500 POKE W1,17
2600 POKE W2,17:FOR A=1 TO 400:NEXT
2625 IF B()0 THEN RETURN
2650 POKE W3,17
2655 PRINT "\-/-\-/-\-/-\-/-\-/-\-/-\-/-\-/-\-/-\-/"
2660 IF B()0 THEN RETURN
2700 POKE W1,16:FOR A=1 TO 100:NEXT
2750 POKE W2,16:FOR A=1 TO 300:NEXT
2800 POKE W3,16:FOR A=1 TO 500:NEXT
2804 REM LOOP "A" FROM 2805-2819
2805 FOR A= 1 TO 7
2806 REM 2807 TURNS ON INPUT INTERFACE DEVICE
2807 B=PEEK(ID):B=B AND (1)
2810 PRINT" . .      \       ┘    .      ‾     .   "
2814 REM 2815 & 2825 SEND COMPUTER TO MAIN PROGRAM IF NO LIGHT IS ON PHOTOCELL
2815 IF B()0 THEN RETURN
2817 REM 2819 & 2850 END "A" AND "M" LOOPS
2819 NEXT A
2825 IF B()0 THEN RETURN
2850 NEXT M
2899 REM WHEN "M" HAS REPEATED 5 TIMES, 2900 RETURNS COMPUTER TO MAIN PROGRAM
2900 RETURN
```

```
3000 REM SUB 3000 CREATES REPETITIVE SOUNDS AND OPTICAL GRAPHICS
3001 REM 3002-3002 INITIALIZE INPUT INTERFACE DEVICE
3002 POKE ICR,0:POKE ID,0:POKE ICR,4
3003 POKE OCR,0:POKE OD,255:POKE OCR,4
3004 REM "A" LOOP IS FROM 3005-3800. IT REPEATS SOUNDS AND OPTICAL GRAPHICS
3005 FOR A= 1 TO 100
3099 REM 3100 SETS VOLUME TO HIGHEST LEVEL
3100 POKE VOL,15
3199 REM 3200 SETS ATTACK/DECAY FOR ALL 3 VOICES
3200 POKE AD1,30:POKE AD2,55:POKE AD3,100
3249 REM 3250 SETS SUSTAIN/RELEASE FOR ALL 3 VOICES
3250 POKE SR1,200:POKE SR2,76:POKE SR3,240
3299 REM 3300 SETS HIGH FREQUENCIES FOR ALL 3 VOICES
3300 POKE H1,100:POKE H2,85:POKE H3,87
3324 REM 3325 TURNS ON INPUT DEVICE
3325 B=PEEK(ID):B=B AND (1)
3349 REM 3350 PRINTS A LINE OF GRAPHICS
3350 PRINT"/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/"
3399 REM 3400 SETS LOW FREQUENCIES FOR ALL 3 VOICES
3400 POKE L1,33:POKE L2,79:POKE L3,100
3499 REM 3500 SETS VOICE 1/PULSE,VOICE 2/SAWTOOTH,VOICE 3/TRIANGLE WAVEFORMS
3500 POKE W1,65:POKE W2,33:POKE W3,17
3599 REM 3600 TURNS OFF WAVEFORM SETTINGS
3600 POKE W1,64:POKEW2,32:POKEW3,16
3649 REM 3650 PRINTS A BLANK LINE FOR OPTICAL EFFECT
3650 PRINT"                                      "
3652 REM IF NO LIGHT HITS PHOTOCELL LINE 3653 RETURNS TO MAIN PROGRAM
3653 IF B<>0 THEN RETURN
3799 REM 3800 ENDS LOOP
3800 NEXT A
3899 REM IF "A" COMPLETES LOOP 100 TIMES 3900 RETURNS TO MAIN PROGRAM
3900 RETURN
```

```
4000 REM SUB 4000
4001 REM LINES 4002 & 4004 INITIALIZE INTERFACE DEVICE
4002 POKE ICR,0:POKE ID,0:POKE ICR,4
4004 POKE OCR,0:POKE OD,255:POKE OCR,4
4009 REM LINE 4010 POKES VOLUME TO HIGHEST LEVEL
4010 POKE VOL,15
4014 REM LINE 4015 BEGINS "N" LOOP
4015 FOR N= 1 TO 80
4018 REM LINE 4020 ASSIGNS ATTACK/DECAY & LINE 4030 ASSIGNS DECAY/RELEASE
4020 POKE AD1,180:POKE AD2,90:POKE AD3,200
4030 POKE SR1,69:POKE SR2,200:POKE SR3,20
4038 REM LINE 4040 PICKS RANDOM INTEGER BETWEEN 1 AND 200
4040 R=INT(RND(1)*(200-1)+1)
4048 REM LINES 4050 AND 4060 POKE MEMORY ADDRESSES FOR HIGH AND LOW FREQUENCIES
4050 POKE H1,R:POKE H2,R:POKE H3,R
4052 POKE L1,R:POKE L2,R:POKE L3,R
4054 REM LINE 4062 BEGINS "C" LOOP WHICH IS A RANDOM LENGTH LOOP
4055 REM LINE 4063 CHECKS INPUT DEVICE
4056 LINES 4065 AND 4066 PRINT LINES OF CHARACTERS
4057 REM LINE 4068 SENDS COMPUTER TO MAIN PROGRAM IF NO LIGHT DETECTED
4058 REM LINE 4069 ENDS "C" LOOP
4060 POKE L1,R:POKE L2,R:POKE L3,R
4061 REM LINE 4062 BEGINS "C" LOOP WHICH IS A RANDOM LENGTH LOOP
4062 FOR C=1 TO (R*.1)
4063 B=PEEK(ID):B=B AND(1)
4064 POKE W1,65:POKE W2,65:POKE W3,65:POKE W1,64:POKE W2,64:POKE W3,64
4065 PRINT"────────────────────────────────────────"
4066 PRINT"────────────────────────────────────────"
4068 IF B<>0 THEN RETURN
4069 NEXT C
4070 FOR M=1 TO 100
4071 REM LINE 4072 STARTS "J" LOOP WHICH POKES ATTACK/DECAY ADDRESSES
4072 FOR J=1 TO 3:POKE AD1,R:POKE AD2,R:POKE AD3,R:NEXT
4073 B=PEEK(ID):B=B AND (1)
4074 REM LINE 4073 PEEKS INPUT DEVICE / LINE 4075 PRINTS LINE OF CHARAACTERS
4075 PRINT"───────────────────────────────────────"
4076 REM LINE 4078 SENDS COMPUTER TO MAIN PROGRAM IF NO LIGHT ON PHOTOCELL
4077 REM LINE 4080 ENDS "M" LOOP / LINE 4081 STARTS "V" LOOP
4078 IF B<>0 THEN RETURN
4079 REM LINES 4082,4083,4086 POKE MEMORY ADDRESSES FOR SOUND
4080 NEXT M
4081 FOR V=1 TO R*.2
4082 POKE H1,(R*.5):POKE H2,90:POKE H3,79:POKE L1,20:POKE L2,10:POKE L3,5
4083 POKE AD1,R:POKE AD2,R*.5:POKE AD3,200
4084 PRINT" ||||||||||||||||||||||||||||||||||||||||"
4085 PRINT"███████████████████████████████████████████"
4086 POKE W1,17:POKE W2,17:POKE W3,17:POKE W1,16:POKE W2,16:POKE W3,16
4087 REM LINES 4084 AND 4085 PRINT LINES OF CHARACTERS
4088 NEXT V
4089 REM LINE 4088 ENDS "V" LOOP / LINE 4090 ENDS "N" LOOP
4090 NEXT N
4199 REM LINE 4200 RETURNS COMPUTER TO MAIN PROGRAM
4200 RETURN
```

```
5000 REM SUBROUTINE 5000 MAKES GRAPHICS AND MORSE CODE-LIKE SOUNDS
5001 REM LINES 5002 & 5004 INITIALIZE INPUT DEVICE
5002 POKE ICR,0:POKEID,0:POKE ICR,4
5003 REM LINE 5005 CLEARS SOUND CHIP
5004 POKE OCR,0:POKE OD,255:POKE OCR,4
5005 FOR L=54272 TO 54296:NEXT
5009 LINE 5010 POKES VOLUME TO THE HIGHEST LEVEL
5010 POKE VOL,15
5079 REM BEGIN "R" LOOP IN 5080
5080 FOR R=1 TO 100
5089 REM 5090 PEEKS THE INPUT DEVICE
5090 B=PEEK(ID):B=B AND(1)
5099 REM LINE 5100 PRINTS A LINE AND A HALF OF CHARACTERS
5100 PRINT"─────────────────────────────────────────────────────────────────"
5104 REM IF NO LIGHT HITS PHOTOCELL LINE 5105 RETURNS COMPUTER TO MAIN PROGRAM
5105 IF B<>0 THEN RETURN
5109 REM 5110 BEGINS "X" LOOP
5110 FOR X=1 TO 20
5114 REM LINE 5115 PEEKS INPUT DEVICE
5115 B=PEEK(ID):B=B AND (1)
5119 REM LINE 5120 PRINTS A LINE OF CHARACTERS
5120 PRINT"▼▼▼▼▼▼      ╱╲╱╲╱╲╱╲╱╲╱╲╱╲╱╲╱╲╱╲╱╲╱╲╱╲╱╲╱╲╱╲╱╲╱╲╱╲╱╲╱╲╱╲╱╲╱╲╱╲╱╲╱"
5125 IF B<>0 THEN RETURN
5129 REM LINES 5130, 5140, 5150, 5151. 5152 POKE MEMORY ADDRESSES FOR SOUND
5130 POKE AD1,130:POKE AD2,145:POKE AD3,20
5139 REM SAME AS 5104
5140 POKE SR1,200:POKE SR2,20:POKE SR3,87
5150 POKE W1,65:FOR A=1 TO 20:POKE W1,64
5151 POKE W2,65:FOR S=1 TO 35:POKE W2,64
5152 POKE W3,17:FOR M=1 TO 10:POKE W3,16
5159 REM 5160 ENDS "X" LOOP
5160 NEXT X
5199 REM 5200 BEGINS "Y" LOOP
5200 FOR Y= 1 TO 15
5204 REM LINE 5205 PEEKS INPUT DEVICE
5205 B=PEEK(ID):B=B AND (1)
5209 REM LINE 5210 PRINTS ONE CHARACTER
5210 PRINT "█"
5214 REM 5215 IS SAME AS 5105
5215 IF B<>0 THEN RETURN
5219 REM LINE 5220 POKES MEMORY ADDRESSES FOR SOUND
5220 POKE AD2,40:POKESR2,10:POKE W2,17:FOR A=1TO 3:NEXT :POKE W2,64
5239 REM LINE 5240 ENDS "Y" LOOP / LINE 5260 ENDS "R" LOOP
5240 NEXT Y
5260 NEXTR
5269 LINE 5270 RETURNS COMPUTER TO MAIN PROGRAM
5270 RETURN
```

```
6000 REM SUB 6000
6004 REM LINES 6005 & 6007 INITIALIZE INPUT DEVICE
6005 POKE ICR,0:POKE ID,0:POKE ICR,4
6007 POKE OCR,0:POKE OD, 255:POKE OCR,4
6010 REM LINE 6020 CLEARS SOUND CHIP
6020 FOR L=54272 TO 54296:POKE L,0:NEXT
6029 REM LINE 6030 STARTS LOOP "T" / LINE 6040 POKES VOLUME TO HIGHEST LEVEL
6030 FORT=1 TO 9
6040 POKE VOL,15
6045 REM LINES 6050, 6060, 6065, 6070, 6080 POKE MEMORY ADDRESSES FOR SOUND
6050 POKE AD1,10:POKE AD2,20:POKE AD3,220
6060 POKE SR1,200:POKE SR2,5:POKE SR3,79
6065 POKE H1,40:POKE H2,60:POKE H3,30
6070 POKE L1,10:POKE L2,12:POKE L3,4
6072 X=1
6074 REM LINE 6075 BEGINS "D" LOOP
6075 FOR D=1 TO X
6077 REM LINE 6078 PEEKS INPUT DEVICE
6078 B=PEEK(ID):B=B AND(1)
6079 REM LINES 6080, 6082, 6083, 6084 POKE MEMORY ADDRESSES FOR SOUND
6080 POKE W1,17:FOR A=1 TO 40:NEXT A:POKE W1,16
6081 FOR E=1 TO Y
6082 POKE W2,17:POKE W2,16
6083 POKE W2,17:POKE W2,16
6084 POKE W3,17:POKE W3,16
6085 B=PEEK(ID):B=B AND(1)
6086 PRINT"▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁"
6087 IF B()0 THEN RETURN
6088 NEXT E
6089 B=PEEK(ID):B=B AND (1)
6090 POKE 1024+X+40*Y,67
6091 REM LINE 6090 POKES A BLANK SPACE IN A RANDOM PLACE ON THE SCREEN
6092 REM LINE 6087 RETURNS COMPUTER TO MAIN PROGRAM IF NO LIGHT DETECTED
6093 REM LINES 6085, 6089 PEEK INPUT DEVICE
6095 REM LINE 6095 SAME AS 6087
6098 REM 6100 PICKS RANDOM NUMBER BETWEEN 1 & 39
6099 REM 6110 PICKS RANDOM NUMBER BETWEEN 1 & 24
6100 X=RND(1)*(39-1)+1
6110 Y=RND(1)*(24-1)+1
6119 REM 6120 ENDS "D" LOOP / 6150 ENDS "T" LOOP
6120 NEXT D
6129 REM 6139 IS SAME AS 6092
6130 IF B()0 THEN RETURN
6150 NEXT T
6210 RETURN
```

```
7000 REM SUB 7000 MAKES RANDOM LENGTH LOOPS OF REPETITIVE SOUNDS AND GRAPHICS
7010 INITIALIZE INPUT DEVICE
7020 POKE ICR,0:POKE ID,0:POKEICR,4
7025 POKE OCR,0:POKE OD,255:POKE OCR,4
7050 REM 7070 PICKS RANDOM INTEGER BETWEEN 1 AND 29
7070 Q=INT(RND(1)*(29-1)+1)
7099 REM LINE 7100 BEGINS "X" RANDOM LENGTH LOOP
7100 FOR X=1 TO Q
7101 REM 7102 CHECKS INPUT DEVICE
7102 B=PEEK(ID):B=B AND(1)
7104 RN=INT(RND(1)*(100-30)+30)
7105 PRINT"\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\"
7106 PRINT"/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\"
7107 IF B<>0 THEN RETURN
7108 REM 7107 SENDS COMPUTER TO LINE 57 WHEN PHOTOCELL DETECTS NO LIGHT
7109 POKE VOL,15
7110 POKE AD1,130:POKE AD2,80:POKE AD3,2
7115 REM 7109, 7110, 7120, 7130, 7135, 7140, 7160, 7180 POKE SOUNDS ADDRESSES
7120 POKE SR1,136:POKE SR2,136:POKE SR3,136
7130 POKE H1,31:POKE L1,75
7135 POKE H2,40:POKE L2,200:REM C A,D D#
7140 POKE H3,61:POKE H3,126
7160 POKE W3,33:FOR A=1 TO 140:NEXT
7180 POKE W1,32:POKE W2,16:POKE W3,32
7188 REM 7190 ENDS "X" LOOP
7190 NEXT X
7220 REM 7240 STARTS "Y" RANDOM LENGTH LOOP
7230 FOR Y= 1 TO RN
7240 PRINT"⁑▟▟▟▟▟▟▟▟▟▟▟▟▟▟▟▟▟▟▬"
7244 REM IF PHOTOCELL DETECTS NO LIGHT, 7245 SEND COMPUTER TO LINE 57
7245 IF B<>0 THEN RETURN
7247 NEXT Y
7248 REM 7247 ENDS "Y" LOOP
7330 REM 7340 PRINTS A LINE OF CHARACTERS
7340 PRINT" ⁑  ⁑ ⁑ ⁑ ⁑ ⁑ ⁑ ⁑ ⁑ ⁑ ⁑ ⁑ ⁑ ⁑ ⁑ ⁑ ⁑ ⁑ ⁑ "
7341 REM IF NO ONE INTERUPTS LIGHT BEAM, LINE 7342  BEGINS SUBROUTINE AGAIN
7342 GOTO 7000
```

```
8000 REM SUB 8000 MAKES RANDOM LENGTH LOOPS OF RANDOM SOUNDS & OPTICAL GRAPHICS
8010 REM 8020 PICKS RANDOM INTEGER BETWEEN 1 AND 100
8020 RN = INT(RND(1)*100-1)
8040 REM 8050 AND 8055 INITIALIZE INPUT DEVICE
8050 POKE ICR,0:POKE ID,0:POKE ICR,4
8055 POKE OCR,0:POKE OD,255:POKE OCR,4
8060 REM 8070 BEGINS "X" LOOP
8070 FOR X=1 TO 7
8073 REM 8075, 8110, 8112, 8114, 8116, 8118, 8120 POKE SOUND MEMORY ADDRESSES
8075 POKE VOL,15
8110 POKE AD1,250:POKE AD2,80:POKE AD3,10
8112 POKE SR1,20:POKESR2,81:POKE SR3,35
8114 POKE W1,65:FOR A=1 TO 100:NEXT
8116 POKE W2,65:FOR A=1 TO 10:NEXT
8117 REM 8119 BEGINS "N" LOOP
8118 POKE W3,17:POKE W3,16
8119 FOR N=1 TO 50
8120 POKE W1,64:POKE W2,64
8124 REM 8125 CHECKS INPUT DEVICE
8125 B=PEEK(ID):B=B AND(1)
8128 REM 8129 PRINTS LINE OF GRAPHICS
8129 PRINT"\\\\\\\\\\\\\\\\\\\\////////////////////"
8130 IF B()0 THEN RETURN
8131 NEXT N
8132 REM 8130 SENDS COMPUTER TO LINE 57 IF NO LIGHT IS DETECTED BY PHOTOCELL
8133 REM 8131 ENDS "N" LOOP
8135 REM 8140 PRINTS LINE OF GRAPHICS
8140 PRINT"||||||||||||||||||||||||||||||||||||||| "
8144 REM 8145 ENDS "X" LOOP
8145 NEXT X
8146 M=INT(RND(1)*(12-1)+1)
8147 REM 8146 PICKS RANDOM INTEGER BETWEEN 1 AND 12 / 8148 BEGINS "Y" LOOP
8148 FOR Y = 1 TO M
8149 REM 8150 PICKS RANDOM INTEGER BETWEEN 1 AND 120
8150 R= INT(RND(1)*(120-1)+1)
8180 REM 8190 CHECKS INPUT DEVICE
8190 B=PEEK(ID):B=B AND(1)
8195 REM 8200 PRINTS LINE OF GRAPHICS
8200 PRINT"////////////////////\\\\\\\\\\\\\\\\\\\\"
8210 POKE VOL,15
8212 POKE SR1,R:POKE SR2,R:POKE SR3,R
8214 POKE AD1,R:POKE AD2,R:POKE AD3,R
8216 POKE H1,(R*.5):POKE H2,(R*.7):POKE H3,(R*.6)
8217 POKE L1,(R*.25):POKE L2,(R*.35):POKE L3,(R*.3)
8218 FOR D=1 TO (R*.5)
8219 B= PEEK(ID):B=B AND(1)
8220 PRINT"||||||||||||||||||||||||||||||||||||||||"
8221 REM 8218 STARTS "D" LOOP / 8219 CHECKS INPUT DEVICE / 8220 PRINTS GRAPHICS
8224 POKE W1,17:POKE W2,33:POKE W3,65
8230 POKE W1,16:POKE W2,32:POKE W3,64
8235 IF B()0 THEN RETURN
8236 REM IF PHOTOCELL DETECTS NO LIGHT 8235 AND 8245 SEND COMPUTER TO LINE 57
8239 REM 8240 ENDS "D" LOOP
8240 NEXT D
8245 IF B()0THEN RETURN
8249 REM END "Y" LOOP
8250 NEXT Y
8290 REM 8300 SENDS COMPUTER BACK TO LINE 57 IN MAIN PROGRAM
8300 RETURN
```

```
8999 REM SUB 9000 MAKES OPTICAL GRAPHICS AND HIGHER NOTES WITH EACH LOOP
9000 FOR T= 1 TO 100:REM 9000 STARTS "T" LOOP / 9001 STARTS "A" LOOP
9001 R=INT(RND(1)*(260-1)+1):FOR A=1 TO R
9002 POKE VOL,15:POKE AD1,R*.2:POKE AD2,R*.6:POKE AD3,R*.6
9003 POKE SR1,R*.5:POKE SR2,200:POKE SR3,R*.3
9004 POKE H1,R*.5:POKE H2,R*.3:POKE H3,R*.1
9005 POKE L1,R*.3:POKE L2,R*.2:POKE L3,2
9006 FOR D=1 TO .5 * R: REM START "D" LOOP/ 9009 ENDS "D" LOOP
9007 POKE ICR,0:POKE ID,0:POKE ICR,4
9008 POKE OCR,0: POKE OD,255:POKE OD 4:REM 9007 AND 9008 INITIALIZE INPUT
9009 POKE W2,16:POKE W3,32:POKE W1,64:NEXT D
9010 FOR D=1 TO .5 *R: REM START NEW "D" LOOP / 9014 CHECKS INPUT DEVICE
9011 REM LINES   9002 - 9005, 9009, 9012, 9016 POKE MEMORY ADDRESSES FOR SOUND
9012 POKE W1,65:POKE W2,17:POKE W3,33
9013 REM LINE 9015 PRINTS A LINE AND A HALF OF GRAPHICS CHARACTERS
9014 B= PEEK(ID):B=B AND(1)
9015 PRINT"    ▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚"
9016 POKE W2,16:POKE W3,32:POKE W1,64
9017 REM IF PHOTOCELL DETECTS NO LIGHT, 9018 SENDS COMPUTER TO MAIN PROGRAM
9018 IF B<>0 THEN RETURN
9019 NEXT D: REM END "D" LOOP
9020 FOR L=54272 TO 54296:POKEL,0:NEXT L
9021 REM 9020 CLEARS SOUND CHIP
9029 REM 9032, 9031, 9034, 9036, 9038, 9044, 9046 POKE MEMORY SOUND ADDRESSES
9031 POKE VOL,15
9032 POKE AD1,100:POKE AD2,160:POKE AD3,70
9034 POKE SR1,120:POKE SR2,20:POKE SR3,90
9036 POKE H1,80:POKE H2,59:POKE H3,.5*R
9038 POKE L1,10:POKE L2,9:POKE L3,40
9039 REM 9040 BEGINS "M" LOOP
9040 FOR M=1 TO R
9041 REM 9042 CHECKS INPUT DEVICE
9042 B=PEEK(ID):B=B AND(1)
9044 POKE W1,17:POKE W2,17:POKE W3,17
9046 POKE W1,16:POKE W2,16:POKE W3,16
9047 REM 9048 PRINTS 1/2 LINE OF GRAPHICS/ 9049 PRINTS OTHER 1/2 IF M > 35
9048 PRINT"              ████████████████████████"
9049 IF M>35 THEN PRINT"▒▟ ▟ ▟ ▟ ▟ ▟ ▟ ▟ ▟ ▟ ▟ ▟ "
9050 IFB<>0 THEN RETURN
9051 REM LINES 9050, 9120, 9145 SEND TO MAIN PROGRAM IF NO LIGHT ON PHOTOCELL
9100 NEXT M
9110 REM 9100 ENDS "M" LOOP/ 9140 ENDS "A" LOOP / 9150 ENDS "T" LOOP
9120 IF B<>0 THEN RETURN
9140 NEXT A
9145 IF B<>0 THEN RETURN
9150 NEXT T
9950 REM 9999 RETURNS COMPUTER TO MAIN PROGRAM
9999 RETURN
```

```
10000 REM SUB 10000 DRAWS LINES AND MAKES RANDOM MUSICAL SOUNDS
10040 REM 10050 AND 10060 INTITIALIZE INPUT DEVICE
10050 POKE ICR,0:POKE ID,0:POKE ICR,4
10060 POKE OCR,0:POKE OD,255:POKE OCR,4
10065 REM 10070 BEGINS "X" LOOP
10070 FOR X=1 TO R
10072 REM 10075 CHECKS INPUT DEVICE
10075 B=PEEK(ID):B=B AND(1)
10077 REM 10080 PICKS RANDOM NUMBER BETWEEN 1 AND 220
10080 R=INT (RND(1)*(220-1)+1)
10085 REM 10090 BEGINS "T" LOOP
10090 FOR T= 1 TO (R*.1)
10093 REM 10095 CHECKS INPUT DEVICE
10095 B=PEEK(ID):B=B AND (1)
10097 REM 10100, 10110, 10120, 10130 POKE MEMORY ADDRESSES FOR SOUND
10100 POKE VOL,15
10102 REM 10198 ENDS "K" LOOP / 10200 ENDS "X" LOOP
10103 REM 10105 BEGIN "S" LOOP
10105 FOR S= 1 TO R*.1
10106 REM 10107 CHECKS INPUT DEVICE
10107 B=PEEK(ID):B=B AND(1)
10110 POKE AD1,90:POKE AD2,120:POKE AD3,20
10120 POKE SR1,R:POKE SR2,R:POKE SR3,R
10130 POKE H1,80:POKE H2,50:POKE H3,30
10134 REM 10135, 10150, 10193 ALL PRINT LINES  WHICH SCROLL DOWN THE SCREEN
10135 PRINT"IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII"
10136 IF B<>0 THEN RETURN
10137 NEXT S
10138 REM 10136 RETURNS COMPUTER TO MAIN PROGRAM IF PHOTOCELL DETECTS NO LIGHT
10139 REM 10137 ENDS "S" LOOP / 10140 BEGINS "D" LOOP
10140 FOR D= 1 TO (R*.1)
10143 REM 10145 CHECKS INPUT DEVICE
10145 B=PEEK(ID):B=B AND(1)
10150 PRINT"IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII"
10153 REM 10155 SAME AS 10136
10155 IF B<>0 THEN RETURN
10157 REM 10160, 10170, 10182, 10184, 10186, 10188, 10189 POKE SOUND ADDRESSES
10160 POKE W1,35:POKE W2,35:POKE W3,17
10170 POKE W1,34:POKE W2,34:POKE W3,16
10175 REM END "D" LOOP
10180 NEXT D
10182 POKE AD1,R:POKE AD2,R:POKE AD3,R
10184 POKE SR1,R:POKE SR2,R:POKE SR3,R
10186 POKE H1,R*.5:POKE H2,R*.5:POKE H3,R*.2
10188 POKE L1,R*.8:POKE L2,R*.7:POKE L3,R*.9
10189 FOR K=1 TO (R*.1):POKE W1,65:POKE W2,65:POKE W3,33
10190 B=PEEK(ID):B=B AND(1)
10191 REM 10190 CHECKS INPUT DEVICE / 10194 ,10197, 10199 SAME AS 10155
10193 PRINT" IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII"
10194 IF B<>0 THEN RETURN
10195 POKE W1,64:POKE W2,64:POKE W3,32
10196 REM 10195 POKES MEMORY ADDRESSES FOR WAVEFORMS
10197 IF B<>0 THEN RETURN
10198 NEXT K
10199 IF B<>0 THEN RETURN
10200 NEXT X
10240 REM 10250 SENDS COMPUTER TO BEGINNING OF SUBROUTINE
10245 REM THE LIGHT BEAM MUST BE INTERUPTED TO STOP THIS SUBROUTINE
10250 GOTO 10070
10255 REM 10260 RETURNS COMPUTER TO MAIN PROGRAM
10260 RETURN
```

```
11000 REM SUB 11000 MAKES OPTICAL GRAPHICS & HIGHER SOUNDS EACH TIME IT REPEATS
11002 REM 12003,12004 INITIALIZE INPUT DEVICE/ 12005 BEGINS "F" LOOP
11050 REM 11100 AND 11105 INITIALIZE INPUT DEVICE
11100 POKE ICR,0:POKE ID,0:POKE ICR,4
11105 POKE OCR,0:POKE OD,255:POKE OCR,4
11150 REM 11159 PICKS RANDOM NUMBER BETWEEN 1 AND 20
11159 K= INT (RND(1)*(20-1 )+1)
11160 FOR R= 1 TO K
11161 REM 11160 BEGINS "R" LOOP /11165 CHECKS INPUT DEVICE
11165 B=PEEK(ID):B=B AND (1)
11167 REM 11170 CLEARS SOUND CHIP
11170 FORL=54272 TO 54296: POKEL,0:NEXT
11180 REM 11200, 11205, 11208, 11209, 11190 POKE SOUND MEMORY ADDRESSES
11190 POKE VOL,15
11200 POKE AD1,R*3:POKE AD2,R*3:POKE AD3,R*3
11205 POKE SR1,R*2:POKE SR2,R*3:POKE SR3,R
11208 POKE H1,R*2:POKE H2,R*3:POKE H3,R*2.5
11209 POKE L1,R:POKE L2,R*.5:POKE L3,R
11210 RN = INT (RND(1)*38-1)+1)
11211 REM 11210 PICKS RANDOM NUMBER BETWEEN 1 AND 38
11212 REM 11213 BEGINS "M" LOOP
11213 FOR M=RN TO R
11214 REM 11215 CHECKS INPUT DEVICE / 11216 PRINTS A LINE OF CHARACTERS
11215 B=PEEK(ID):B=B AND (1)
11216 PRINT"_-_-_-__-_-_-_-_-_-_-_-__-_-_-_-_-_-_-"
11217 REM 11218 SENDS COMPUTER TO MAIN PROGRAM IF PHOTOCELL DETECTS NO LIGHT
11218 IF B()0 THEN RETURN
11219 REM 11220, 11222, 11224, 11230 POKE MEMORY ADDRESSES FOR SOUND
11220 POKE W1,33:POKE W2,17:POKE W3,33
11222 POKE AD1,100:POKE AD2,80:POKE SR1,90:POKE SR2,70
11224 POKE H1,30:POKE H2,60:POKE L1,20:POKE L2,10
11225 FOR D= 1 TO (RN*.3)
11226 POKE W1,33:POKE W2,17:POKE W3,33
11228 B=PEEK(ID):B=B AND(1)
11229 REM 11228 CHECKS INPUT DEVICE / 11231 PRINTS LINE OF CHARACTERS
11230 POKE W1,32:POKE W2,16:POKE W3,32
11231 PRINT"-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-"
11232 IF B()0 THEN RETURN
11233 NEXT D
11234 IF B()0 THEN RETURN
11235 NEXT M
11236 IF B()0 THEN RETURN
11237 IF M)33 THEN GOTO 11210
11238 REM 11232, 11234, 11236 SAME AS 11217
11310 NEXT R
11315 REM 11233 ENDS "D" LOOP/ 11234 ENDS "M" LOOP/ 11310 ENDS "R" LOOP
11400 REM 11450 SENDS COMPUTER TO MAIN PROGRAM
11450 RETURN
```

```
12000 REM SUB 12000 PRINTS DIAGONAL LINES AND MAKES TRAIN-LIKE SOUNDS
12002 REM 12003 AND 12004 INITIALIZE INPUT DEVICE
12003 POKE ICR,0:POKE ID,0, POKE ICR,4
12004 POKE OCR,0:POKE OD,255:POKE OCR,4
12005 FOR F=1 TO 30 : REM BEGIN "F" LOOP
12006 REM 12007, 12012, 12097 CHECK INPUT DEVICE
12007 B=PEEK(ID):B=B AND (1)
12009 REM 12010 BEGINS "A" LOOP
12010 FOR A=1 TO 3
12012 B=PEEK(ID):B=B AND(1)
12060 REM 12070,12080, 12090, 12092, 12094, 12100, 12130 POKE SOUND ADDRESSES
12070 POKE VOL,15
12080 POKE AD1,100:POKE AD2,200:POKE AD3,186
12090 POKE SR1,100:POKE SR2,8:POKE SR3,90
12092 POKE H1,31:POKE H2,21:POKE H3,67
12094 POKE L1,12:POKE L2,19:POKE L3,31
12095 FOR D=1 TO 50
12097 B=PEEK(ID):B=B AND (1)
12100 POKE W1,17:POKE W2,17:POKE W3,17
12103 IF B() 0 THEN RETURN
12104 REM 12103 SENDS COMPUTER TO MAIN PROGRAM IF PHOTOCELL DETECTS NO LIGHT
12105 NEXT D
12130 POKE W1,16:POKE W2,16:POKE W3,16
12133 REM 12135 PRINTS A LINE OF CHARACTERS
12135 PRINT"\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\"
12136 REM 12137 SAME AS 12103
12137 IF B()0 THEN RETURN
12139 REM 12140 ENDS "A" LOOP / 12150 BEGINS "N" LOOP
12140 NEXT A
12153 REM 12155 CHECKS INPUT DEVICE
12155 B=PEEK(ID): B=B AND (1)
12157 REM 12160, 12170, 12180, 12190, 12200, 12220, 12222, 12230 POKE SOUND
12158 REM ADDRESSES IN MEMORY
12160 POKE VOL,15
12170 POKE AD1,50:POKE AD2,67:POKE AD3,120
12180 POKE SR1,30:POKE SR2,120:POKE SR2,2
12190 POKE H1,41:POKE H2,51:POKE H3,21
12200 POKE L1,20:POKE L2,46:POKE L3,3
12209 REM 12210 PICKS RANDOM NUMBER BETWEEN 1 AND 50 / 12212 BEGINS "N" LOOP
12210 A= INT(RND(1)*(50-1)+1)
12212 FOR N=1 TO A
12214 REM 12215 CHECKS INPUT DEVICE
12215 B=PEEK(ID):B=B AND (1)
12220 POKE W1,65:POKE W2,17:POKE W3,129
12222 POKE AD1,100:POKE SR1,100
12230 POKE W1,64:POKE W2,16:POKE W3,128
12235 REM 12240 PRINTS DIAGONAL LINES
12240 PRINT"////////////////////////////////////////"
12243 REM 12245 AND 12255 SEND COMPUTER TO MAIN PROGRAM IF PHOTOCELL DETECTS N
12244 REM LIGHT
12245 IF B()0 THEN RETURN
12250 NEXT N
12253 REM 12250 ENDS "N" LOOP / 12260 ENDS "F" LOOP/ 12300 SENDS COMPUTER TO
12254 REM MAIN PROGRAM
12255 IF B()0 THEN RETURN
12260 NEXT F
12300 RETURN
```

FOOTNOTES

[1] John Cage, "Preface to: Lecture on the Weather", in Empty Words. (Middletown, Connecticut: Wesleyan University Press, 1979), p.5.

[2] Louis-Bertrand Castel, cited by Frank Popper, Origins and Development of Kinetic Art (Greenwich, Connecticut: New York Graphics Company, 1968), p. 156.

[3] Webster's Seventh New Collegiate Dictionary (Springfield, Massachusetts: G.&C. Merriam Co., 1967), p. 894.

[4] Gene Youngblood, Expanded Cinema, (New York: E.P. Dutton & Co., Inc., 1970), p. 301.

[5] Ron Hays, cited in Arttransition, (Cambridge, Massachusetts: Center for Advaanced Visual Studies / Massachusetts Institute of Technology, 1975), p. 57.

[6] Jasia Reichardt, The Computer in Art, (New York: Van Nostrand Reinhold, 1971), p. 15.

[7] Luigi Russolo, "The Art of Noises (extracts) 1913", in Futurism, ed. Angelo Bozzolla and Caroline Tisdall. (New York: Oxford University Press, 1978), p.75.

BIBLIOGRAPHY

Apollonio, Umbro. Futurist Manifestos. New York: Viking Press, 1973.

Battcock, Gregory, ed. Breaking the Sound Barrier. New York: E.P. Dutton, 1981.

Brecht, George. Chance-Imagery. New York: A Great Bear Pamphlet, 1966.

Burnham, Jack. Beyond Modern Sculpture. New York: George Braziller, 1968.

Cage, John. Empty Words. Middletown Connecticut: Wesleyan University Press, 1979.

Cage, John. Silence. Cambridge, Massachusetts: M.I.T. Press, 1961.

Commodore 64 Programmer's Reference Guide. West Chester, Pennsylvania: Commodore Business Machines, Inc., 1982.

Compton, Michael. Optical and Kinetic Art. London: Tate Gallery Publications Department, 1967.

Davis, Douglas. Art and the Future. New York: Praeger Publishers, 1973.

Goldberg, Roselee. Performance: Live Art 1909-Present. New York: Henry N. Abrams, Inc. Publishers, 1977.

Hofstadter, Douglas R. Gödel, Escher, Bach: An Eternal Golden Braid. New York: Vintage Books, 1979.

Hornung, Clarence P. Handbook of Designs and Devices. New York: Dover Publications, Inc., 1932.

Jung, Carl G., ed. Man and His Symbols. Garden City, New York: Doubleday & Co., Inc., 1964.

Knapp, Jeff. Graphics for the Commodore 64 Computer. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1984.

Kostelanetz, Richard, ed. Moholy-Nagy. New York: Praeger Publishers, 1970.

Malina, Frank J. Kinetic Art. New York: Dover Publications. Inc., 1974.

Mansbach, Steven A. Visions of Totality. Ann Arbor, Michigan: UMI Research Press, 1980.

Piene, Otto and Heinz Mack. Zero. Cambridge, Massachusetts: M.I.T. Press, 1973.

Popper, Frank. Origin and Development of Kinetic Art. Greenwich, Connecticut: New York Graphic Society, 1968.

Reichardt, Jasia. The Computer in Art. New York: Van Nostrand Reinhold, 1971.

Smith, Reginald Brindle. *The New Music*. London: Oxford University Press, 1975.

Swan, Alfred J. *Scriabin*. New York: Da Capo Press, 1969.

Tisdall, Caroline, and Bozzolla, Angelo. *Futurism*. New York: Oxford University Press, 1978.

Wingert, Paul S. *Primitive Art*. New York: New American Library, 1962.

Youngblood, Gene. "Ed Emshwiller's Skin Matrix: An Interveiw." *Send*. Spring 1985. p.13.

Youngblood, Gene. *Expanded Cinema*. New York: E. P. Dutton & Co., Inc., 1970.

Exhibition Catalogs

1968 *The Machine*. Museum of Modern Art, New York.

1970 *Kinetics*. Hayward Gallery, London.

1975 *Arttransition*. Center for Advanced Visual Studies / Massachusetts Institute of Technology, Cambridge, Massachusetts.

1979 *Sound*. Los Angeles Institute of Contemporary Art, Los Angeles.

1980 *L. Moholy-Nagy*. Institute Contemporary

Arts, London.

1982 Nam June Paik. Whitney Museum of American Art, New York.

1984 New Music America '84. New Music America Festival, Hartford, Connecticut.