A COMPUTER-CONTROLLED SOUND-LIGHT ENVIRONMENT

James Seawright

The recent construction of a new terminal building for the Seattle-Tacoma International Airport provided me with the opportunity to design and construct a large-scale sound-light environment as part of a permanent installation of works of art within the building. The Richardson Associates, Seattle, Washington, architects for the Port of Seattle, had indicated a willingness to consider commissioning a highly experimental computer-controlled system to Peter Phillips, a New York composer, who originally conceived the project, and who played an important role in arranging the commission. A location designated as a waiting area, about 30' by 70' in dimension, was made available, with the only stipulation being that the space had to remain useful as a waiting area, with a certain amount of useable seating. A design study was conducted during the summer of 1971. The design was accepted in July of 1972 and work was completed 1 August 1973, at which time the work was placed in operation at the inauguration of the airport art project.

The environment, entitled NETWORK IV, is a development of previous pieces in which I sought to take advantage of feedback from viewers in order to control and vary programs of visual and sound phenomena within an environment-scale structure. The immediately previous piece, NETWORK III (1970), had used a PDP-8L computer to detect and analyse the movements of viewers on a pressure-sensitive floor, and to generate and display visual patterns related to their actions in a grid of lamps forming a ceiling. Another piece, ELECTRONIC PERISTYLE (1968), one of the pieces in the Kansas City Performing Arts Foundation's show Magic Theater, used viewer movements to govern the selection of parameters of electronically synthesized sounds. NETWORK IV then, in addition to offering a chance to develop further some of the ideas in these pieces, allowed by virtue of the permanent nature of the installation and the generous extent of the available budget a much higher degree of control over more factors in the perceptual environment.

The room itself is arranged as shown in Figure 1. The visual display, at A, consists of 1024 3-watt neon glow lamps in 64 vertical columns of 16 lamps, on 6" by 6" centers. The lamps are mounted on a screen of aluminum channels and appear to float in a black void behind the plate glass window at B. The lamps are individually addressable by the computer, and each lamp is associated with a one-bit memory in the driver circuitry so that moderate output data rates are sufficient to produce elaborate, rapidly-changing patterns. In order to permit viewing of the display from all parts of the room, floor-to-ceiling mirrors at locations marked M reflect the image of the display, yet because of their stepped, "Fresnel" arrangement do not take up much of the floor space of the room. The computer, a Data General Corporation Nova 1210, with 8K of core memory, with associated lamp-driver circuitry and sound synthesizer, is located in the enclosed area at D. Four loudspeakers are at L1, L2, L3, and L4. (Objects designated C are structural columns of the building) In the center of the area in front of the display, at E, is located a lectern-like panel of 64 pushbutton switches arranged 8 by 8. No instructions or explanations are provided and the viewer is left to his own intuition as to what may result if the buttons are pressed.

The basic state of operation, between periods in which there is viewer input consists of a large blob-shaped pattern which moves back and forth across the display, seeming to bounce off the top or bottom, and travelling at varying rates. A loosely-structured sound accompaniment of sustained bell- and gong-like sounds is heard. Whenever a viewer presses buttons, rapidly-changing patterns appear in the display and are accompanied by sound patterns which are synchronized in articulation and correlated in dynamics with the antics of the visual patterns. Initially, a geometrical correspondence between the pushbutton array and portions of the display offers clues to the viewer as to inherent possibilities; actually the program is timed to accept, most efficiently, grouped input, and other tests distinguish between spatially grouped pressings of buttons and more widely spaced, random pressings, so that a perceptive viewer can establish a greater and greater degree of control over the activities of the visual display and the sound accompaniment by learning how to time and pace his efforts and how to repeat certain structures by remembering the spatial arrangement and sequence of his button-pressings. In this way the effort is made to build into the program criteria by which idle, desultory input and systematic, explorative, "interested" input may be distinguished, and the latter rewarded. When input no longer continues, after an interval control returns to the program routines which restore the meandering blob and reestablish the sustained sound activity.

In developing the program the one overriding consideration was to attempt to achieve an effect of environmental sensitivity. In other words the viewer would ideally feel that in his encounter with the piece the responses were unique to him, reflections as it were of his own intelligence. Every effort had then to be made to derive from viewer input information which could modify visual and sound time-structure on every level while retaining consistency of detail and dynamic qualities sufficient to permit recognition of what could genuinely be perceived as response.

Figure 2 shows a simplified flowchart of the program. The background program, once started and initialized, consists simply of a loop of subroutine calls. The subroutines, each of which is concerned with some aspect of either ~~the~~ visual display generation, input evaluation and processing, or the formulation of sound parameters, are to a very high degree self-controlling and intercommunicative. Virtually every identifiable, measurable characteristic of the gross or detailed activity possible within the system is associated with variable quantities and correlation coefficients accessible to all of the subroutines in directly-addressible memory. Control flags, timers, and data tables are also accessible to the action subroutines and allow independent as well as interdependent activity throughout the flux of the program. The foreground, or interrupt program, running at a 100 Hz rate by a real-time clock, outputs the visual display fast enough to be flicker-free, and provides the overall time control of all events.

Once the program is started and the blob pattern has been initiated, (a prototype stored pattern is fetched, mapped to a particular place in the display coordinate system~~s~~, then periodically offset in certain directions by certain amounts of offset) sounds are synthesized by referring to tables of pitch, loudness, timbre (as a filter setting), rapidness of attack and decay, and duration. As each sound event is computed the appropriate digital output words are converted into analog voltages which vary the output of the various oscillators, amplifiers, filters, etcetera, of which the sound synthesizer consists. Data is initially taken from the tables without modification, but part of the process of formulating each sound event results in the computation of a series of displacements which are subsequently used to modify the raw data in systematic ways resulting in progressive changes and sequential development. In this way precomposed pitch sequences, rhythmic patterns, loudness contours, "phrasing", and the like may retain recognisable structure, yet through modification a relatively small amount of data, suitably chosen, is able to provide extremely diverse and continually interesting output.

When someone presses buttons on the lectern, the program begins to shift control to the routines which evaluate and process input data. A list of coordinate-pairs is prepared, mapped to a particular position in the display field, and is treated as an initial pattern in the Conway "life" game.* The rules of this game are applied

---

* The Conway life game is a mathematical diversion invented by John Conway, a Cambridge University mathematician. It is played in a cellular space in which each cell is said to be alive or dead. (On a rectilinear grid of lamps each lamp may be on or off) Each cell, or lamp, has eight neighbors, four orthogonal and four diagonal. In the usual form of rules a cell which is alive at a given time, or generation, will remain alive in the next generation if it has two or three live neighbors. A cell which is not alive will be "born" if it has exactly three neighbors. An initial pattern is examined to determined the future of each cell, then the pattern is replaced by the new generation. The game results in an amazing variety of patterns which grow and develop; some patterns eventually become stable, others oscillate, some grow explosively, seemingly without limit, while others die away altogether. The game may be played with pencil and graph paper, but is ideally suited to computers with graphic or display output, when the fantastic dynamics of rapidly changing generations becomes apparent. See Scientific American, Oct. 70, pp120-123.

to the initial pattern in order to determine the configuration of the next generation and the patterns are displayed, changing at a rate determined by the time taken to run the main program loop. An average rate is between one and ten ~~six~~ generations per second with patterns of a dozen or so live cells. One important modification has been made to the basic life rules and that is that the new pattern list is copied over the old list in memory instead of adding newborn cells to the list of surviving ones. The result of this is that all stable, many oscillating, and some translating patterns are viable, but patterns which grow efficiently and develop for hundreds of generations are considerably inhibited and the display does not easily become loaded down with dense patterns which slow all program activity due to the time it takes to compute them.

Use of the life game as a basic display generator results in having available a very efficient yet enormously versatile means of producing dynamically interesting, developing patterns. The actual algorithm used in the life subroutine manipulates lists of locations in a coordinate system, and consequently allows easy monitoring of levels of activity of the patterns, since one always knows how many cells are alive, whether the pattern is growing or decaying, how many generations has the pattern persisted, and more generalized characteristics such as direction of movement, identity of pattern, etcetera, are readily determinable. This is the type of information referred to earlier as performance variables

which

Synchronization with visual patterns is accomplished by the fortunate fact that due to the cyclical background program, sound output can only be changed in between times life generations are changed. When a change in sound does occur it is certain to be coincident, within the viewer's ability to determine, with a change in the visual display. Since the visual dynamics and the rhythmic characteristics of the sounds are always fairly well correlated by the program, the viewer's normal perceptual disposition to accept approximately similarly structured simultaneous events as actually intended counterpoints completes the illusion of purposeful and active responsiveness.

Figures 3 and 4 show the finished environment as it appears in operation.

I would like at this time to acknowledge the assistance of James Hussey and Gerald Williams of The Richardson Associates, M. V. Mathews and F. R. Moore of Bell Telephone Laboratories, William Hemsath, Emmanuel Ghent, Robert Miller Charles Keagle, Frederick Toevs, Michael Delay, William Campbell, and my wife, Mimi Garrard, without whose help I would not have been able to complete the project.

James Seawright

Captions:

Figure 1 - Floor plan of Sound-Light Environment, Network IV

Figure 2 - Simplified Flowchart of Program

Figure 3 - Sound-Light Environment, Network IV, in operation; general view, looking approximately south.

Figure 4 - View from behind pushbutton panel, facing display.

b

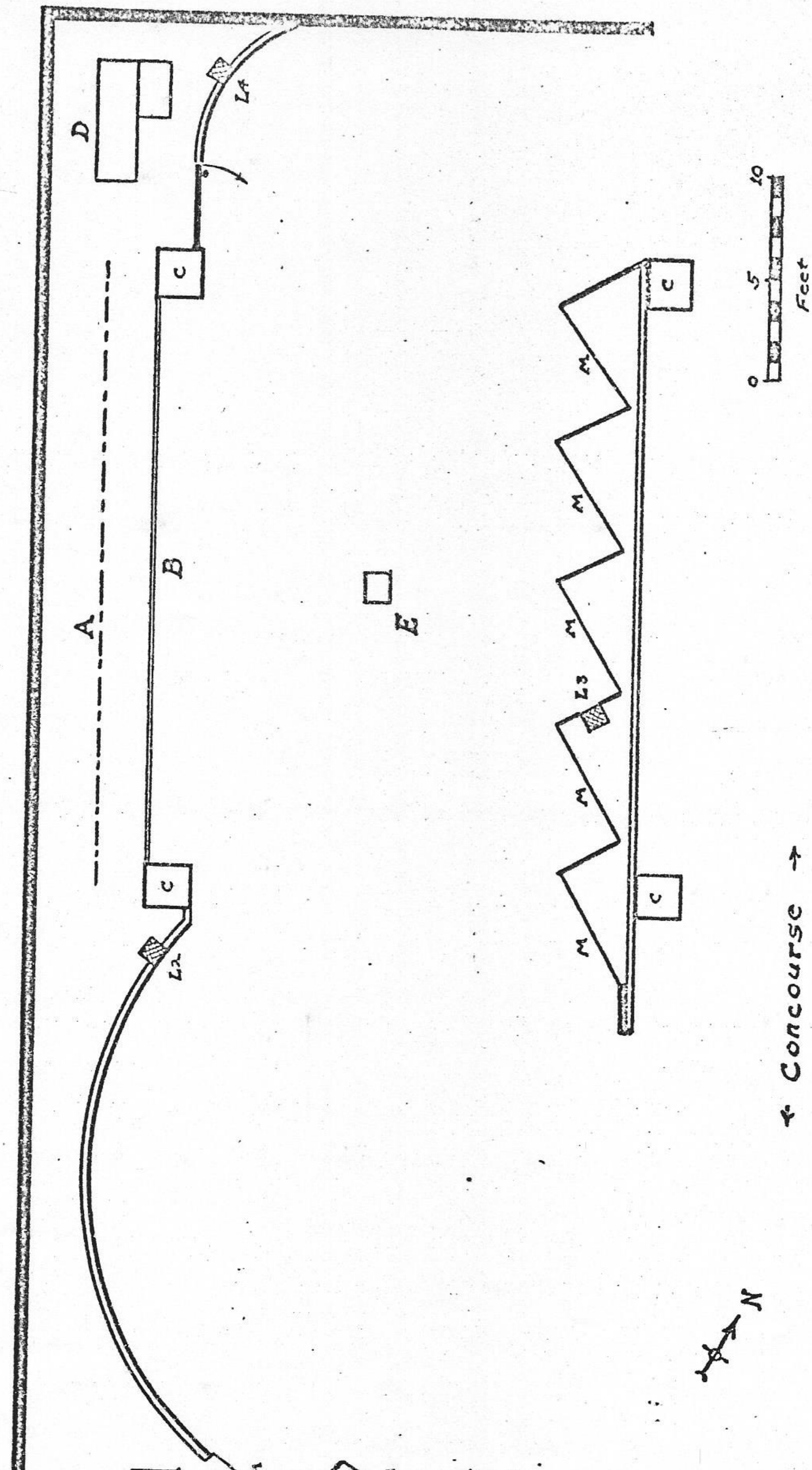
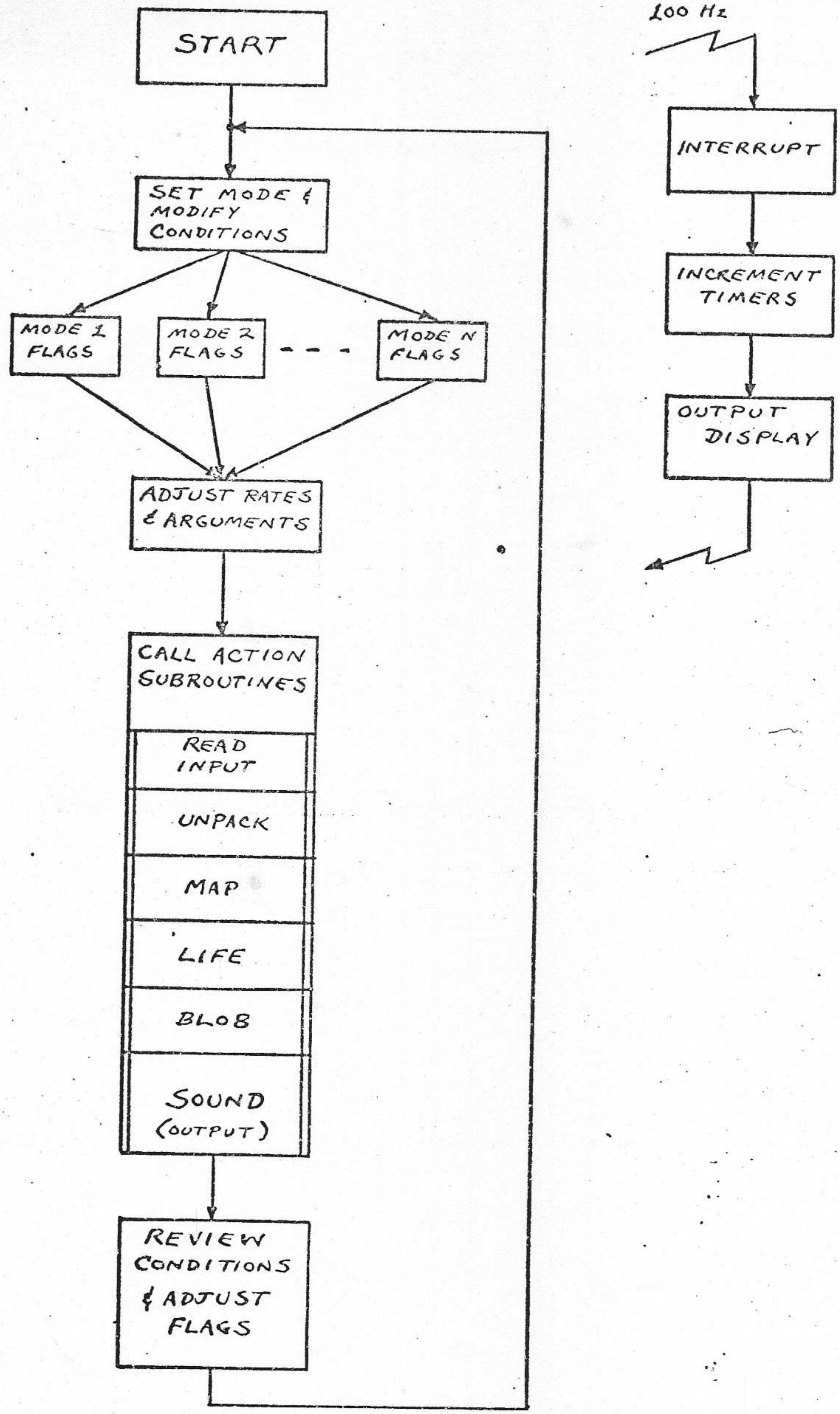


Fig 1.



START
SET MODE & MODIFY CONDITIONS
MODE 1 FLAGS
MODE 2 FLAGS
MODE N FLAGS
ADJUST RATES & ARGUMENTS
CALL ACTION SUBROUTINES
READ INPUT
UNPACK
MAP
LIFE
BLOB
SOUND (OUTPUT)
REVIEW CONDITIONS & ADJUST FLAGS
100 Hz
INTERRUPT
INCREMENT TIMERS
OUTPUT DISPLAY

Fig 2.