BOJANGLES II - A SYSTEM FOR THE DESCRIPTION AND ANALYSIS OF BODY MOVEMENTS

by SANFORD P. RESSLER
(HENRY RUTGERS SCHOLAR, 1977-1978)

A thesis submitted to
Rutgers College

in partial fulfillment of the requirements
for the degree of
Bachelor of Arts

Written under the direction of
Professor Philip Orenstein
of the Department of Art

New Brunswick, New Jersey
April 1978

Table of Contents

Abstract

The application of computer technology to problems in the arts is a relatively unexplored area. This project is concerned with the use of computers in the field of dance notation as well as the use of computers for the conceptualization of movements. A computer based dance notation system will be more useful than any of the existing notation systems because it is based on an animated figure, rather than a symbolic system. The method of controlling this figure is one of the major goals of this project. A system must be developed which is useful to the choreographer and compatible with present computer technology.

This same notation system can be used to aid in the design of movement for choreography and animation. The way one approaches choreography and various kinds of movements can be aided by the analytical power of a computer. New kinds of movements can be explored which would be hard to visualize without the aid of a computer.

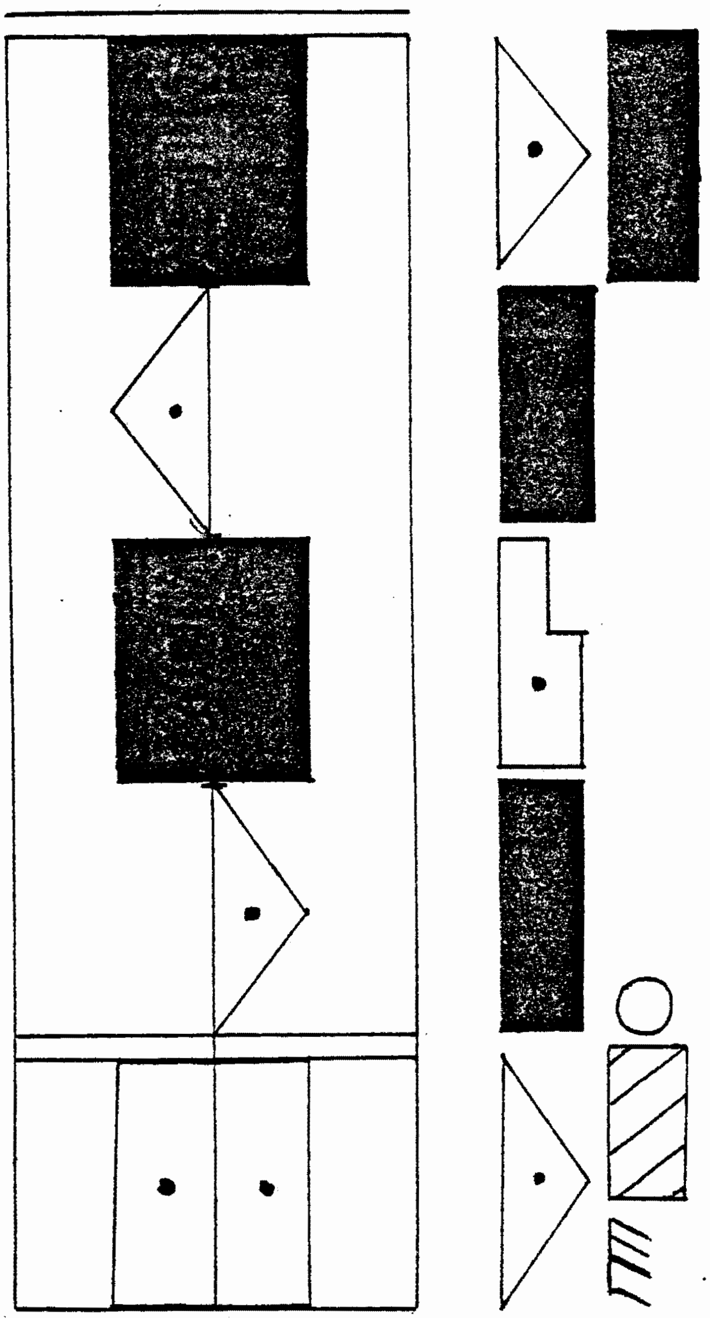
Notation

What is dance notation and why is there any need for it? A dance notation is a system or technique for recording either on paper, film, video or computer the motion of dancer(s). Musical notation is used to record and compose sounds visually in the same way a dance notation system is used to compose and record movement. A dance notation system enables the dancer or choreographer to read and reconstruct a dance or movement.

There are presently several notation systems in use, the main ones are the Labanotation, Sutton and Benesh systems. The practice of videotaping dance is also becoming very popular for recording dances. The primary one used today is Labanotation, invented by Rudolph Laban in 1930. It works by using a system of symbols and a chart which represents motion of the various body parts.(1) (see illus. 1) It takes a highly trained person to notate a dance with this system. It also takes a trained person to read the notations and translate them into motion.

There are hundreds of different symbols in Labanotation, each one representing a specific movement or body part. The complexity of a fully notated score is quite unreadable to an untrained person. Although the symbols are organized into a

ILLUSTRATION 1



logical framework, Labanotation is just too complicated for most dancers and choreographers. Dancers simply do not want to go through the effort of learning the notation system even though there is a strong push on the part of the Dance Notation Bureau[2] to achieve Labanotation literacy. Many dancers, choreographers and people involved in dance feel that a more direct and less symbolic notation system is necessary. Such a system would show the movement in a dynamic manner. Existing notation systems lack the capability of expressing the quality of movement. What is needed is a simple visual notation system which can express accurately the complexity and quality of movement.

Video taping is an example of a more direct system of recording movement. Due to its accessibility and ease of handling taping is becoming very popular, as a recording tool. At the present time, video taping is invaluable, because it shows movement and it is the most practical recording device around. It is used to teach dancers movements when they must learn a new dance. However, videotape as a form of notation has some significant inadequacies. One can not record a dance which does not yet exist. This makes video impossible to use as a tool to aid the creative process which should be a function of a notation system. Even video is to be used as a device solely for recording purposes it would require a great deal of production work in order to adequately record the movement. Several viewpoints would be very desirable as well as adequate lighting, equipment and technicians necessary to do a high quality recording. Once the recording is finished it is fixed and can not be altered, it is totally inflexible. Analyzing a movement in the detail sometimes desired can be very difficult with video, because the means of getting closeups and slow motion is very limited. Also using a dancer to analyze the movement can be difficult due to the complexity of that persons individual style.

The system I am developing makes use of computer technology because I feel that this technology can solve problems of dance notation as well as give the artist a creative tool. (See section on creativity) Computers have been used for some time to aid people in dealing with complex problems, usually scientific or business. More recently the artist has been getting involved in using the computer to help deal with complex problems in the arts. This project is one example of how computers can and will be used to aid artists in their creative works.

A major goal for this particular project is to show that a computer can be used as a conceptual tool for people involved in any kind of movement work and that a computer based notation system would be more flexible and practical than existing notation systems.

The idea of using computers for dance is not totally new. Animations were produced at Bell Labs in 1966 consisting of tiny stick figures moving on random paths. They looked like skaters in a rink.[3] It was not a very elaborate description of movement, but showed some of the possibilities. Recently work has been done by Dr. Steven Smolier for the Dance Notation Bureau. With Maxine Brown he has developed a computer graphics editor for Labanotation which allows a notator to manipulate the symbols on a graphics terminal.[4] He is also working on a way to control a stick figure via Labanotation and this aspect of his work seems very interesting, although not yet completed.[5]

Another person doing related work is Gideon Ariel in Amherst, Mass. He uses a computer to analyze the movements of athletes and to predict changes in performance based on changes of movement.[6] At Penn State Peter Cavanough is using computers to study human stride patterns and in Sweden Ingvar Fredrickson has used computers for ten years to study motion patterns of horses. All of this work shows that computers are being used in a great variety of ways to deal with problems of motion.

I am presently working with stick figures which can show the movement in a more analytical way than a symbolic notation system or video. An analysis of a stick figure can be a clearer way of looking at a movement than an analysis of a real person. With a stick figure one is not likely to get overwhelmed by the detailed intricacies of an individual. However, this figure can express a quality of movememt which other notation systems lack. The stick figure is manipulated and drawn on a Tektronix 4013 terminal by using a system of simple computer commands. These figures are then stored in the computers memory to be recalled when needed. (See Manual section)

Once the figures are in the computer the flexibility of the computer system can be used. These stick figures exist in a computer simulated three dimensional space made visible on the Tektronix terminal. The figures can be viewed from any angle and distance. The speed of any movement or section of movement can be accurately controlled. Lines can be drawn which show the path the body parts travelled in space. The viewpoints can change while the figure is moving. Sections of the figure can be separated so that only parts of the figure can be viewed as it dances. The proportions of the figure can be changed so that a tall and/or short person can be seen dancing the same piece. Every aspect of the figure, timing, and space of the system can be varied to give the viewer a control which is totally impractical with any other notation system. This analysis and recording can be done without ever having someone physically dance. The choreographer can use the computer as a tool to help visualize and thus clarify ideas.

Manual

The notation system described below was developed on an IBM 370/168 timesharing computer. The programs are all written in APL. All of the major programs are contained in a single workspace. After signing on the computer and loading the correct workspace the system is ready to go.

Now a choreographer can conceive or record a dance using the system. A person creates a dance by making various positions from the stick figure. One is presented with a stick figure (see illustration 2) and the means to manipulate it. The image may be manipulated by using a variety of commands, most within a control structure, which is the command language. This language is only operative when in command mode. To get the command mode just type GO. Now the following commands are available: (see table 1)

P: The primary positioning command, enables you to pick out a body part (see table 2) and chose a direction to move the body part towards. There are 26 directions and these correspond to the points of a 3 by 3 cube with each cube being one direction. (see illustration 3 and table 3) The center of the 3 by 3 cube is at the active joint.

R: Changes body to initial default position. (see illustration 2)

V: Places you into viewpoint mode and allows you to change the viewpoint in several ways.
  1) Select one of the preset viewpoints. Top, bottom, left, right, front or back.
  2) Allows you to move the eye relative to where you are by RIGHT, UP, or IN commands.

Illustration 2

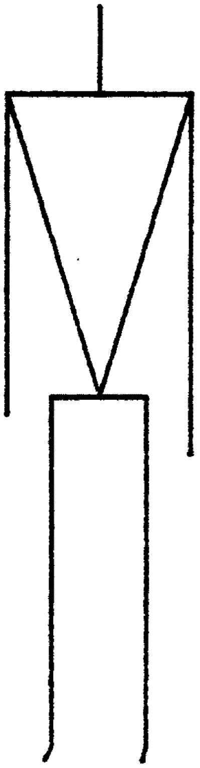
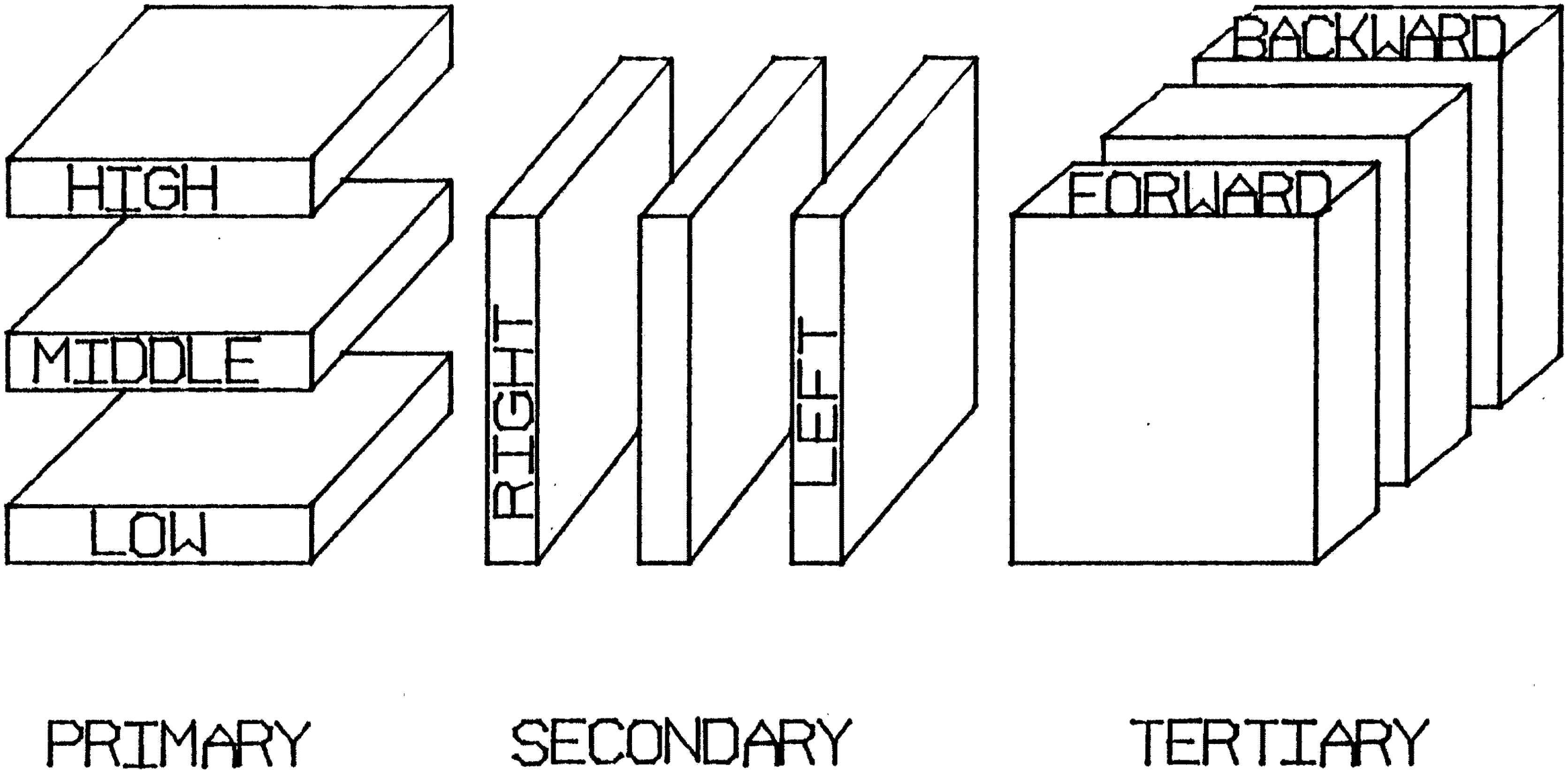


Illustration 3



D: Displays the active body.

E: Allows you to position the eye according to absolute x y z coordinates.

N: Let's you assign a name to the currently displayed body.

F: Puts you into refining mode. Prompts for a body part and one of the following directions:
HIGH LOW LEFT RIGHT FORWARDS BACKWARD
You must type in at least three letters for the above directions. (see table 3)

M: Produces a mirror image of the current body position.

B: Let's you pick another body (previously defined) and allows you to work with it as the active body.

C: Allows you to position the body with cursors.

I: Let's you spin the body around its own axis a specified degrees. (clockwise as viewed from top)

H: Gives you a list of body parts and/or commands.

S: Gets you out of command mode.

Following is a typical terminal session:

(Computer's responses in capitals)

```
go

     COMMAND:

position

     WHAT IS THE BODY PART YOU WISH TO POSITION?

la

     WHERE DO YOU WANT TO MOVE TO

mf
    (see illustration 4)

     COMMAND:

f    (for refine)

     WHAT IS THE BODY PART YOU WISH TO POSITION?

rl

     WHICH DIRECTION

rig
rig  (see illustration 5)
rig

     COMMAND:

view

     WHICH VIEWPOINT OR MODE T FOR TOP, B BOTTOM, L LEFT,
     R RIGHT, F FRONT, M TO MOVE, S TO STOP.

l
    (see illustration 6)

s   (only stops viewing mode)

     COMMAND:

position

     WHAT IS THE BODY PART YOU WANT TO POSITION?

ra

     WHERE DO YOU WANT TO MOVE?
```

Illustration 4

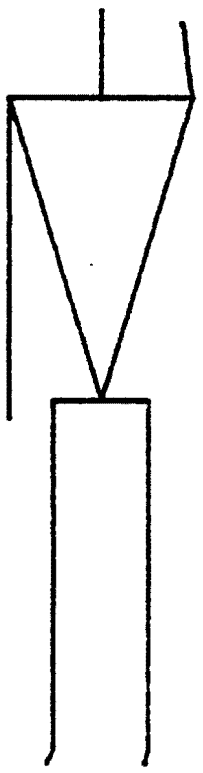


Illustration 5

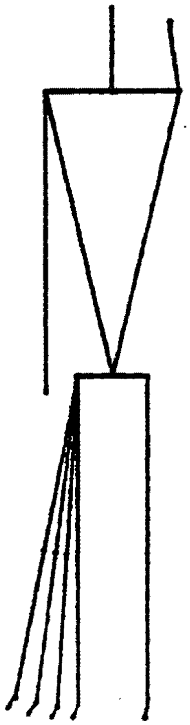
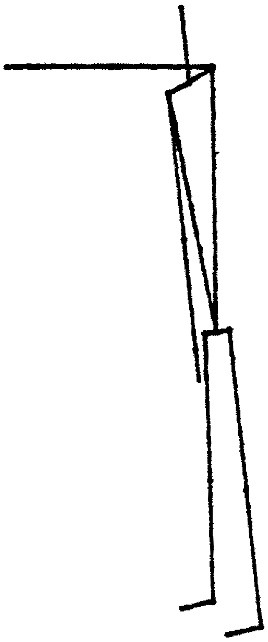


Illustration 6



```
hf
    (see illustration 7)

     COMMAND:

eye

     PLEASE GIVE THE COORDINATES FOR THE EYE

100 1000 -2000

     COMMAND:

display

    (see illustration 8)

     COMMAND:

name

     WHAT IS THE NAME FOR THIS PRESENT POSITION?

henryl

     COMMAND:

b    (for body pick)

     WHAT BODY DO YOU WANT TO DEAL WITH?

nub40

     COMMAND:

display

    (see illustration 9)

     COMMAND:

view

     WHICH VIEWPOINT OR MODE T FOR TOP, B BOTTOM, L LEFT,
     R RIGHT, F FRONT, M TO MOVE, S TO STOP.

f    (for refine)

    (see illustration 10)

     COMMAND:
```

Illustration 7

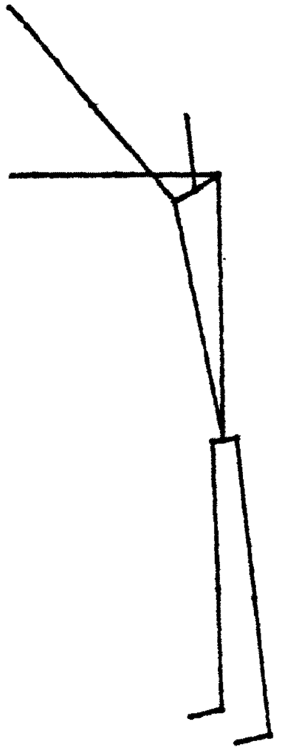


Illustration 8

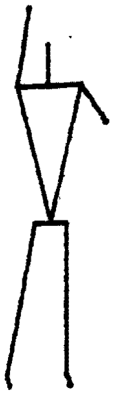


Illustration 9

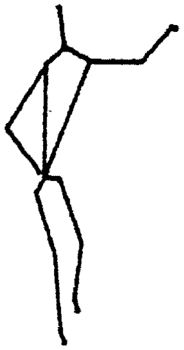
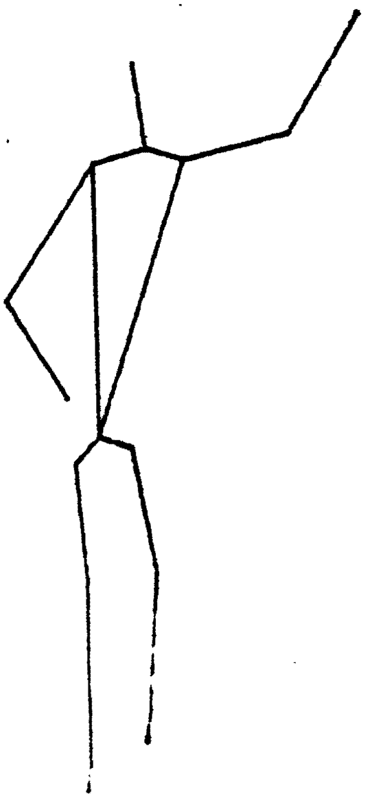


Illustration 18



```
position

     WHAT BODY PART DO YOU WANT TO POSITION

arms

     WHERE DO YOU WANT TO MOVE?

11

    (see illustration 11)

     COMMAND:

name

     WHAT IS THE NAME FOR THE PRESENT POSITION?

henry2

     COMMAND:

mirror

     THE MIRROR IMAGE IS NOW CALLED MBOD.
    (see illustration 12)

     COMMAND:

b    (for body pick)

     WHAT BODY DO YOU WANT TO DEAL WITH?

mbod

     COMMAND:

c    (for cursor position)

     PLACE THE BODY WITH CURSORS.
    (see illustration 13)

     COMMAND:

i    (for spin)

     HOW MANY DEGREES?

50
    (see illustration 14)

     COMMAND:
```

Illustration 11

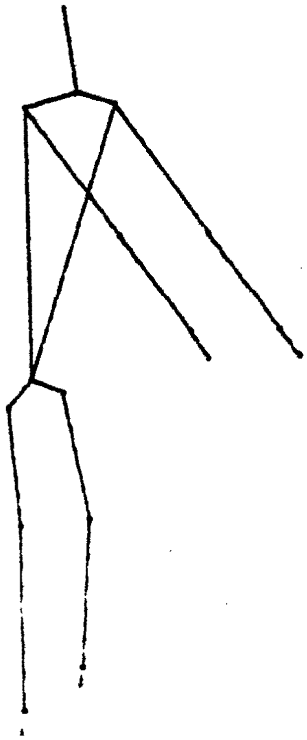


Illustration 12

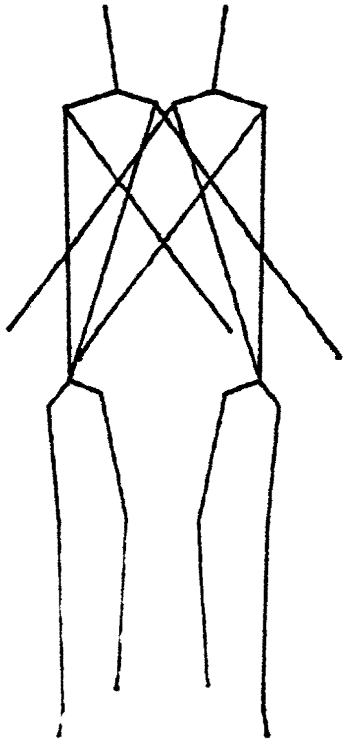


Illustration 13

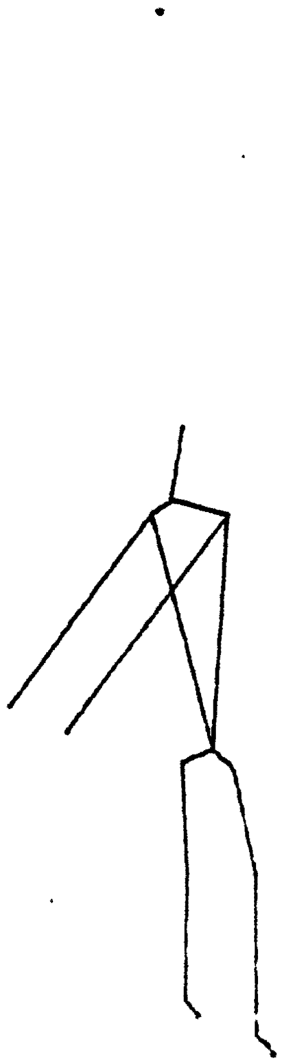


Illustration 14

```
name

     WHAT IS THE NAME FOR THE PRESENT POSITION?

henry3

     COMMAND:

stop
```

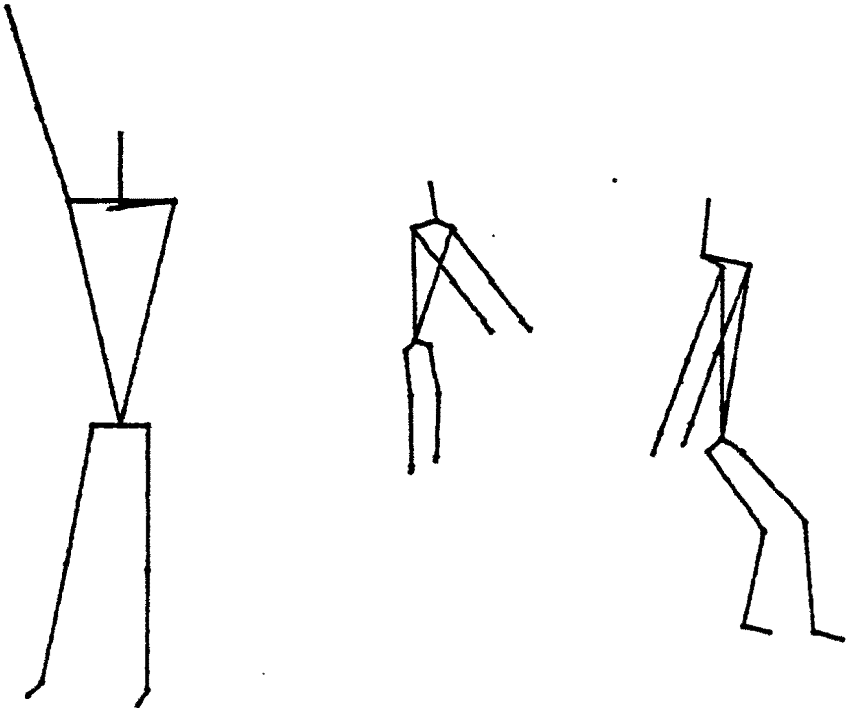
We now have three bodies to play with so lets position them on stage somewhere. Imagine the screen as a top view of the stage. You position the bodies by typing:

```
stage

     (Then enter a figure name.)

henry1

     (And you use the cursors to pick a point on the
      stage.)

   AGAIN?

y

henry2

     (pick the point)

   AGAIN?

y

henry3

     (pick the point)

   AGAIN?

n
     (see illustration 15)
```

The next step is o create the timing sequence for movements. All movements are transformations from one position to another within a certain time span. Now we may define a scene. First you type:

```
scene

     ENTER TIME SPAN.
```

Illustration 15



```
0 1
        ENTER TRANSFORMATION.
henry1/henry2
        AGAIN?
y
        ENTER TIME SPAN.
1.5  3
        ENTER TRANSFORMATION.
henry2/henry3
        GO TO DISPLAY?
```

(see illustration 16)

Now all that's left to do is film the position and then you can see the movement. In order to see any two figures transform you type:

```
x banima 'henry1/henry2'
```

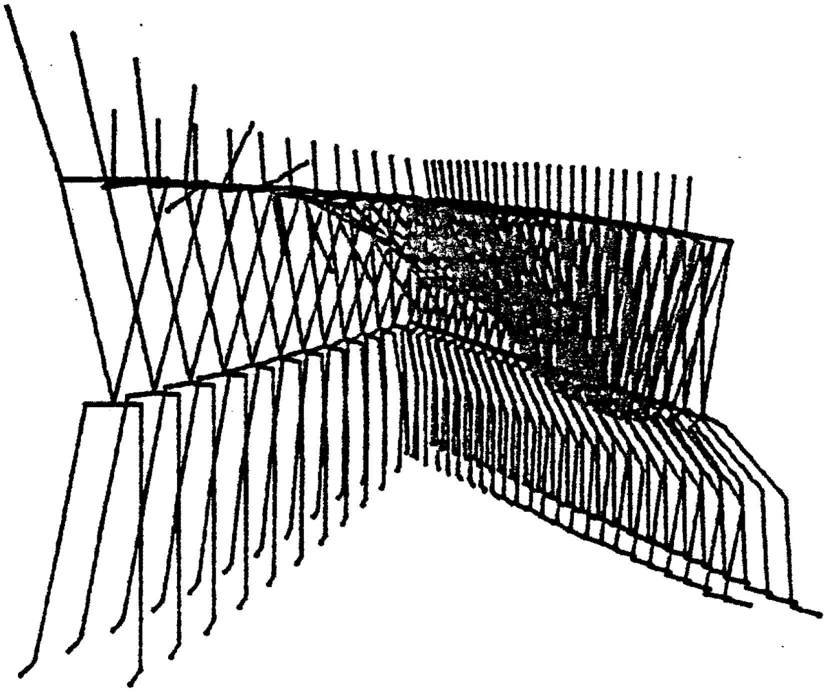
Where x is any number. It corresponds to frames per seconds as set in the variable fps.

One also has any of the standard graphic capabilities of the Interart Graphics system available to you. Several animated films have been made which show the figure dancing. One was of the beginning of "Nubian Woman" a dance choreographed by John Parks.

Following is a more technical and detailed discussion of some of the fine points involved in the system.

Positioning the body parts with the P command is accomplished by using a single sphere consisting of 27 points. The sphere is continuously moving around depending

Illustration 18



on what body part you are dealing with. The center of the sphere is located at the joint and the radius is equal to the length of the body part being positioned. Picking a place to move i.e. ml or ll picks the corresponding point on the sphere. Moving the body part is actually a trick in that there is really no movement involved. Instead the second point of the body part becomes the designated point of the sphere. This makes for a much faster cpu operation.

A body part such as the left arm is actually three body parts: the left upper arm (lua), left forearm (lfa), and left hand (lh). When you position the left arm first the upper arm is moved then the forearm then the hand. In this process the body parts become detached and then reattached.

The refine routine (command f) works by figuring out the lines of rotation and correct angle direction to rotate the body part given one of six possible direction (high low left right forwards backwards).

The function SSIZE is used to keep the size of the body constant or to change the body to other proportions. The variable sizevec1 and sizevec2 are variable which contain lengths for the body parts.

The view point routines are based on the following routines developed largely by Josh Hall.

Given an eye which is treated as a three element vector you have yaw pitch and roll controls. These allow you to rotate about any of the three vectors. One can move the eye by typing UP, where is distance in screen units. The function IN moves the eye towards whatever you are currently looking at. The function RIGHT moves the eye right. The function LOOK points the eye towards the point specified in the delta acenter variable. For the dance system delta acenter is center stage.

The GROUND function is used to give a sense of gravity. i.e. the figure is not allowed to be off the ground or below it. This can be turned on or off with the Gswitch, a three element character variable 'on' or 'off'. It is used primarily in Command mode.

(see illustration 17 for differences)

The BANIMA function is used to "fill in" the positions or movement between the key positions. The body parts do not shrink up as they do with a simple linear interpolation. This is accomplished by keeping the length of the parts constant. (see illustrations 18 and 19 for a comparison) See illustrations 20 and 21 for additional BANIMA examples.

The ANAL function lets you analyze the paths which body parts take during a transformation. (see illustrations 22 and 23)

Illustration 17

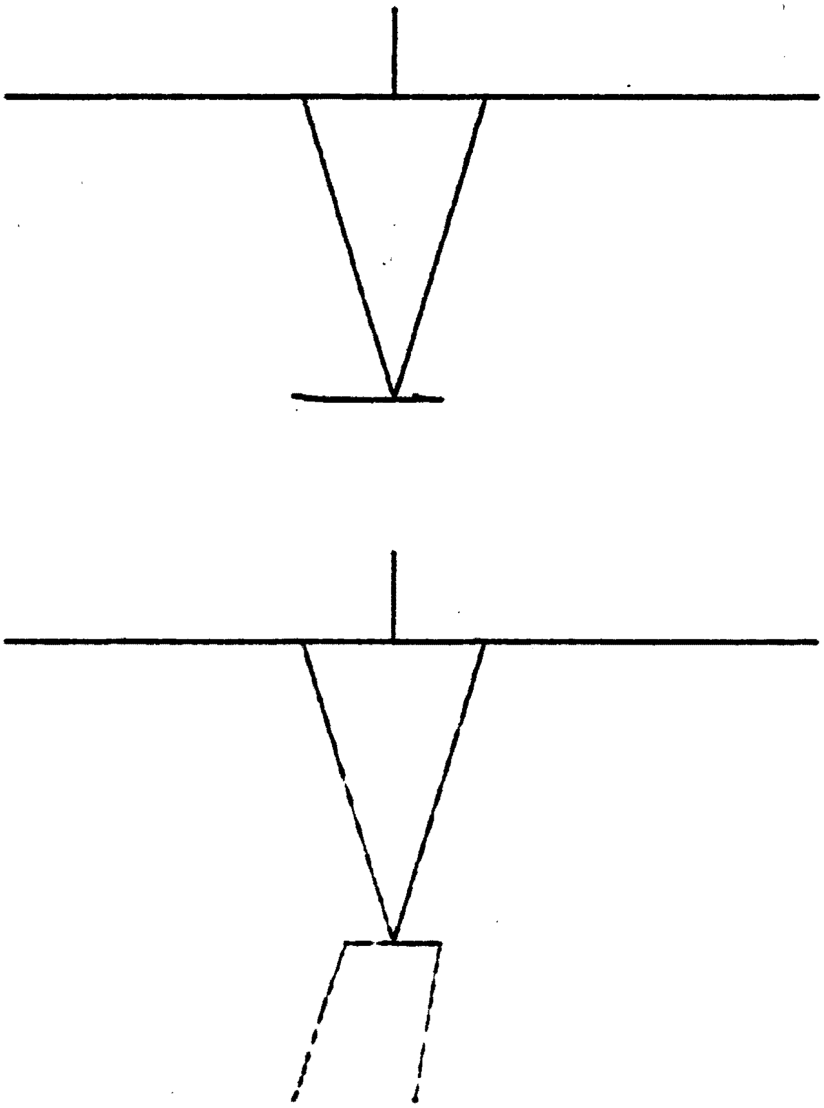


Illustration 18

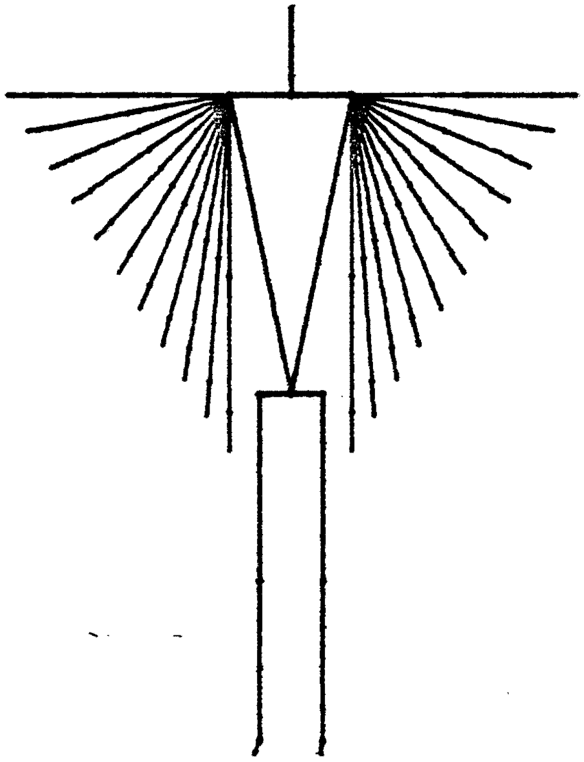


Illustration 19

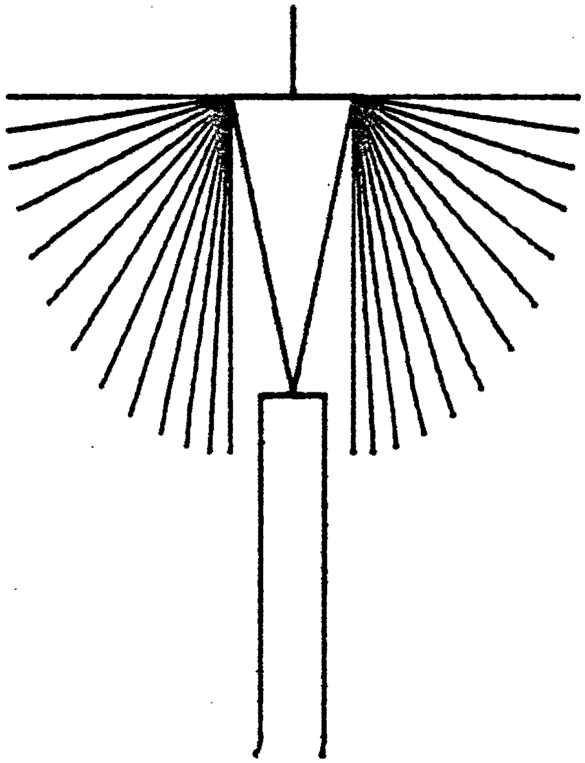


Illustration 20

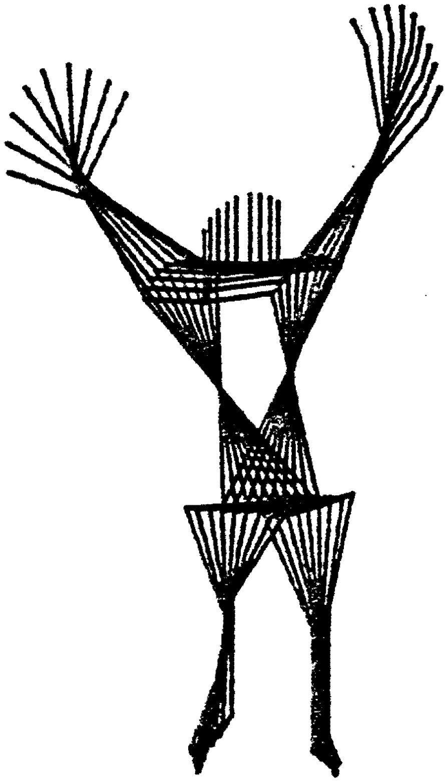


Illustration 21

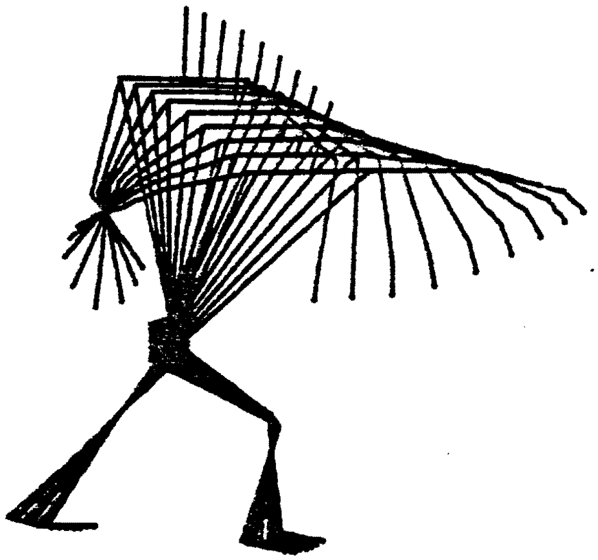


Illustration 22

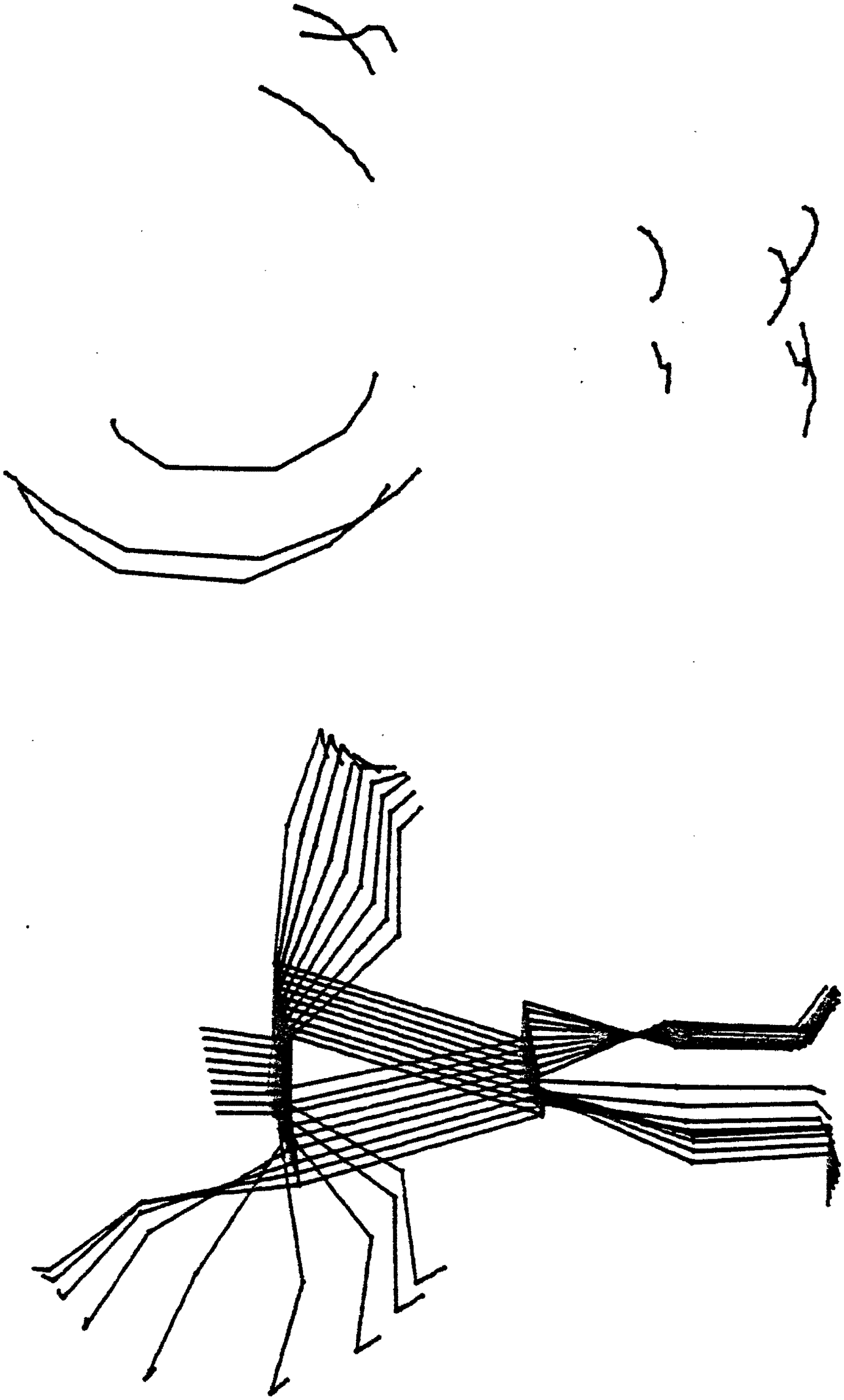
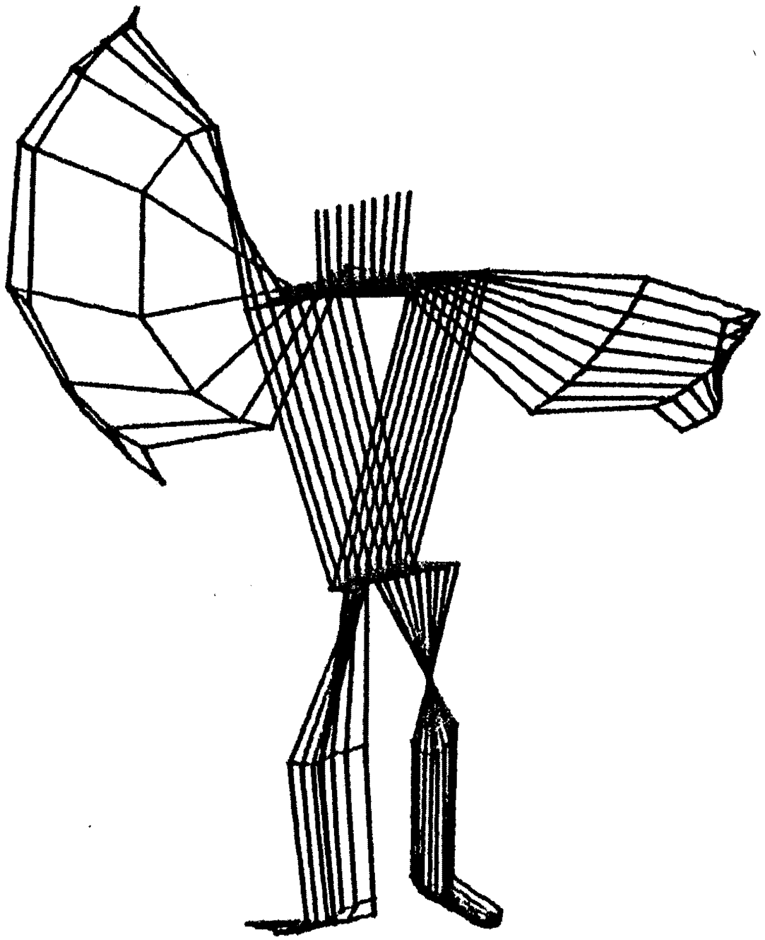


Illustration 23



Creativity

Rosalee Goldberg's article "The Art of Notation" shows how many different artist-dancers have turned to notation as a conceptual tool.(7) Many artists have invented their own limited forms of notation either for a single dance or for several. Artists such as Trisha Brown, Lucinda Childs, Yvonne Rainer, and Steve Paxton have developed their own notations systems to help them work out the problems of a particular piece. Foe example Lucinda Childs began to use notation as a conceptual tool because it became impossible for her to coordinate movements of several dancers simultaneously. "Notation, as a way of thinking out a piece and of presenting those abstract ideas to other performers", has now become indespensible to her working method.

In his book "Changes" Merce Cunningham predicted the future in 1968 by saying:

> "It seems clear that electronic technology has given us a new way to look. Dances can be made on computers, pictures can be punched out on them, why not a notation for dance that is immediately visual?... One would have the image (on a video screen) Next to it on the second screen images in stick-figures that work in depth." (8)

Notation as a conceptual tool is gaining interest among performance artists to script out movements. There is a great need for a notation system and Ms. Goldberg notes.

> "Notation as yet has no general system to express contemporary attitudes. Rather it is made up of a series of personal systems, which limits their readability to the performers themselves. Many performers feel that the development of a general system is essential. Without such a "thinking tool" a "descriptive tool" and even a "conversational tool" the difficulty in using words to describe music, dance and live performance remains." (9)

Choreographers will soon be able to have their own computers at home and will be able to put down ideas quickly in a visual format. Their 'late night ideas' could then be recorded and expressed much more easily to the performers. A computer at home is not a fantasy. At present there are thousands of people with small microcomputers in their homes and with the prices of computer equipment going down artists owning their own computers is not such a remote possibility.

The computer system is set up to deal with both traditional problems of choreography and notation. It also enables the artist to use the computer as an aid to the creative process. As the choreographer is creating the dance with the computer,notation is being made simultaneously. The notation is used as a practical conceptual tool. The relationships of dancers to each other and to the space they are in can be analyzed more carefully than ever before.

Given a relatively small process, a computer program, the information one can get is phenomenal. Finding a workable process or algorithm is the key to any problem on a computer, particularly in the arts. One must be very careful that this process is understandable to the computer, and usable for the artist. The fact that the system one designs is the interface between the artist and the machine must constantly be kept in mind. A system which is very efficient for the computer but difficult to use is useless. A fully implemented computerized dance notation system can give the choreographer a tremendous amount of information, which can be used to aid creativity.

Structured Interaction

When a choreographer teaches a movement to a dancer there is a dialogue between the two. The result is the realization of the choreographer's conception of the dance. In creating a dance on the computer, a dialogue also takes place. This dialogue is between the computer system and the choreographer. There is a significant conceptual difference between these two dialogues. In the choreographer-dancer interaction the idea of a dance is directly transferred from the choreographer's mind to the dancer. In the coreographer-computer dialogue there is an additional step. The idea is first worked out on a computer and is then transferred from the computer to the dancer. This added step enables the choreographer to put down preliminary ideas of a movement into a clear form which can easily be readable at a later time.

Another aspect to the use of a computer as a conceptual tool is that when one is defining these positions one actually has a dialogue with the notation. There is a much more dynamic manner of interaction with a computer notation system than any other type of notation. Having this visual dialogue results in a clarification of ones ideas. In a sense, this system becomes an extension of the choreographer's mind in that it allows one to visualize ideas previously existing only in the mind.

Once a choreographer gets used to the way the system operates, ideas are stimulated by the way it acts and from its structure. The computer-choreographer interaction has a structure which is much more defined than that between a choreographer and a dancer. There exists a structure in the way a choreographer works with a dancer, conscious or unconscious, and the result of a movement idea processed through this structure is a dance which has the style of the choreographer. When dealing with a system of computer programs this structure is very explicit.

Upon examination it becomes clear that this structure is built upon many substructures. Different combinations of substructures result in different systems to work with. For example motion is dealt with as the transition between an ordered series of positions. The computer can very easily randomize this order and create a 'random dance'. Another example of a reordered substructure is the creation of random figures. The body is put together from a series of lines which happen to look like a stick figure. It becomes very 'natural' to put a single body together using parts from many different bodies. This concept of combining bodies like parts of a tinker toy set would be very strange, for the conception of movement, if not for a computer based structure.

Randomness

Randomness has been a significant part of art since Arp's and Duchamp's wood cut outs and has been experimented with in a great variety of ways. What is randomness and how can it be used in choreography? Randomness is simply the idea that given a number of possible choices the probability of picking any one is equal to the probability of picking any other.

One way of creating a random dance is to choose a number of possible dance positions. For example, given a library of positions for a dance, (i.e. the positions of "NUBIAN WOMAN"), one could keep picking, at random, from those positions and the resulting order of positions is a random dance. A movement of this particular type would consist of the exact same key positions as the original 'real' dance but they are performed in a random order. Randomness is something which the computer is very well suited to deal with. Many different random aspects are possible. The manner and degree of randomness which you allow into the dance and figure, results in a wide variety of outcomes.

The above described random dance is one method of applying randomness. Another method is to alter the body parts of a key position random amounts and directions. Another possibility would be to let the computer generate the key positions. This too can be done in many ways. The computer can generate body positions without any anatomical limitations i.e. every body part has a full sphere of movement possibilities. The limitations of the body can be programmed to enable the computer to place the body parts anywhere within that limited range and limit specific body parts to specific ranges but allow random positioning within that range.

Yet another possibility comes from realizing that the computer treats the body merely a collection of lines organized and attached in a specific way. Random positions can also be created by picking subsitute body parts from a library of positions. For example, new position can be created by replacing the left arm of POS27 with the left arm of POS42. The body, used to substitute the parts is picked at random. When this is done for the whole body (all 18 body parts) a random position is generated. (see illustration 24) Random positions can be categorized by keeping track of the random number generated by the computer used to create these positions. (see the number in illustration 25) Surely there are many other possibilities and this can only show that randomness is a quality of dance particularly suited to the computer.

Illustration 24

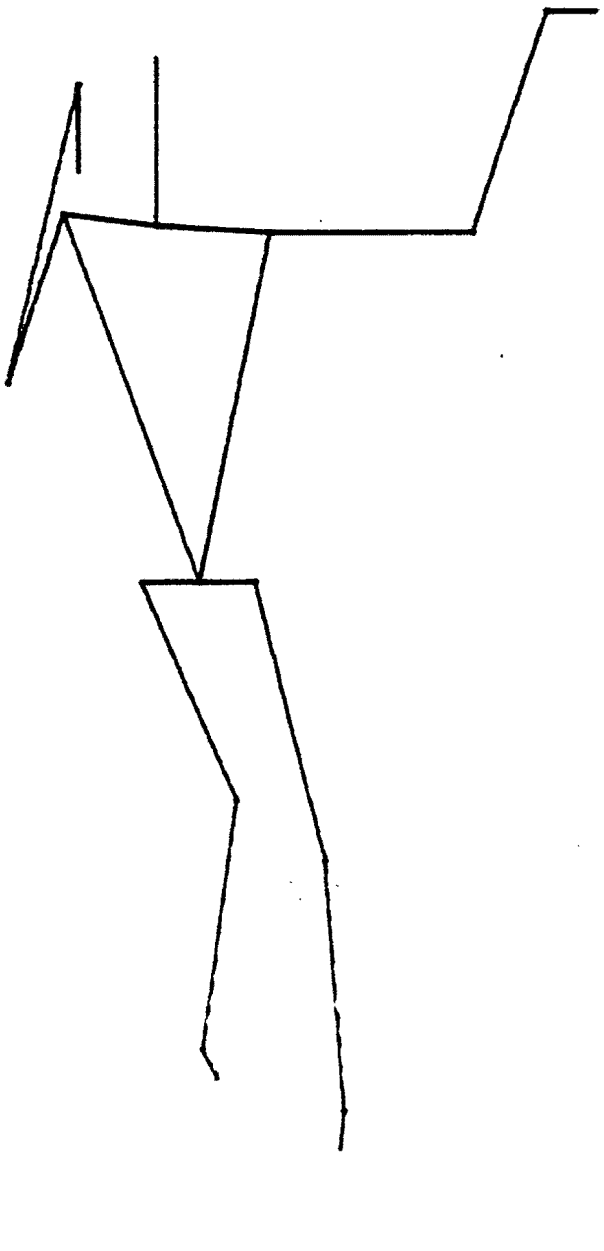
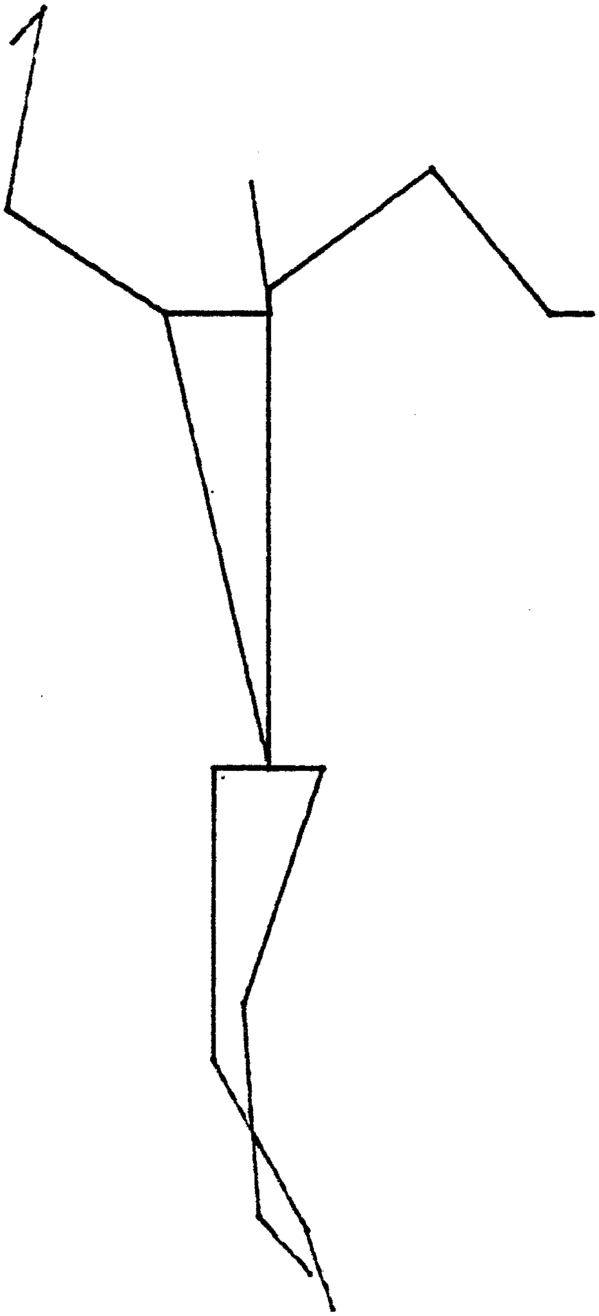


Illustration 25



A great variety of random qualities could be explored and categorized. Without a computer such an in depth exploration into randomness would probably be inconceivable and surely impractical.

Future

At the present stage of development this system can not be considered a practical notation tool. It should be viewed as a rather extensive feasability study into the future of computerized dance notation. As a creative tool it can be used now, particularly in the field of random dance or for studying the spatial relationships of people (proxemics). By far the primary barrier towards making this system practical and valuable for choreographers is the lack of a real time animation system, refresh graphics. This would allow the user to see the motion on the computer terminal without having to go through the very arduous task of animation. With such capability the computer's potential as a tool for dance would be realized. I believe that the system, as it stands now, proves that the computer can be used by choreographers as an extension of their mental processes into a visual format.

The future possibilities for such a system can have far reaching effects in the field of dance, dance notation, and movement studies. A choreographer could have a terminal or whole microcomputer in the studio and create the figures by twisting some knobs and pushing some buttons. Another way of creating the key positions would be to manipulate a mechanical model, a manequin which would send positional information to the computer. Better still the stick figures could be created in real time by the computer itself analyzing the images from two or three video cameras. The notation would be instantaneous, accurate and very readable.

Another extension of the system would be to define movements and use them as a basis for more complicated actions. For example, if walking is defined as a set of positions with certain distances between them one could simply pick two points and have the figure walk between them. The entire set of ballet movements can be defined for example pliet, jete, or arabesque. A ballet could be created by typing all the detailed words...the verbal notation of ballet.

Complete dances could be created with dozens of individually moving figures existing in space. The lighting, as well as the props, could be designed and controlled in full color. A whole theatrical experience could be conceived tested and executed on the computer without ever having to get the dancers, lights, and musicians. All of these variables could be instantaneously synthesized based on the movement of the dancers. This would mean that individualized music and lighting for each performance could be achieved. This computer execution step could become a brand new art form in itself, "Computer performance" or "Cybernetic performance" art.

Various movements can be integrated from several different dances, new dances can be created based on a library of movements. With the extension into 3D displays the images produced by the computer would be seen as existing in your "real" space.(10) The potentials of a computer based dance system are infinite.

Acknowledgements

I would like to thank Phil Orenstein for all his efforts, thoughts and guidance in this project and art in general. I would also like to thank Richard Bardlift whose idea it was to computerize dance notation. Without his help and efforts this project would simply not exist. And of course thanks to all of the hackers for their helpful comments.

Footnotes

1. Hutchinson, Ann, Labanotation, 2nd edition. New York: Theater Arts Books, 1954.

2. The Dance Notation Bureau is an international organization based in New York, devoted to increasing knowledge of Labanotation as well as being involved in movement research.

3. Spencer, Donald, Computers in Society. N.J.: Hayden Inc, 1974 p. 155.

4. Brown and Smolier, "A Graphics Editor for Labanotation", Siggraph Proceedings, Vol 10 No. 2, Summer 1976, p. 60-65.

5. Personal Communication.

6. Moore, K., "Gideon's Magic Machine", Sports Illustrated, Aug 1977, p. 52-60.

7. Goldberg, Rosalee, "The Art of Notation", Studio International, July/August 1976, p. 54-58.

8. Cunningham, Merce, Changes, New York: Something Else Press, 1968.

9. Goldberg, p. 54-58.

10. Davis, Douglas, Art and the Future, New York: Praeger, 1973, p. 104

Bibliography

1. Brown and Smolier, "A Graphics Editor for Labanotation", Siggraph Proceedings, Vol 10 No. 2, (Summer 1976) 60-65.

2. Cunningham, Merce. Changes. New York: Something Else Press, 1968.

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

4. Gilman and Rose. APL an Interactive Approach. Wiley Press, 1975.

5. Goldberg, Rosalee. "The Art of Notation", Studio International, July/August (1976), 54-58.

6. Goldberg, Rosalee. "Oskar Schlemmer's Performance Art", Artforum, Sept (1977), 32-37.

7. Hutchinson, Ann. Labanotation 2nd Edition. New York: Theater Arts Books, 1954.

8. Laban, Rudolph. The Language of Movement. Boston: Plays Inc., 1974

9. Mara, Thala. The Language of Ballet. Dance Horizons.

10. Moore, K., "Gideon's Magic Machine", Sports Illustrated, Aug (1977), 52-60.

11. Mueller, Robert. The Science of Art. New York: John Day Co., 1967.

12. Newman and Sproul. Principles of Computer Graphics. New York: McGraw Hill, 1973.

13. Parkman, Ralph. Cybernetic Society. New York: Pergamon Unified Engineering Series, 1972.

14. Rouse, William, "Design of Man-Computer Interfaces for On-Line Interactive Systems", Proceeding of IEEE, vol 63 (1975), 847-857.

15. Schlemmer, Oskar. The Letters and Diaries of Oskar Schlemmer. Edited by Tut Schlemmer, Translated by K. Winston. Conn: Wesleyan Univ. Press, 1972.

16. Spencer, Donald. Computers in Society. N.J.: Hayden Inc., 1974.

17. Tracey, Martin, A review of Computer Dance, TV, TV/Dance. Dance Research Jrnl, Fall/Winter, (1975-1976) , 40.

18. Turner, Margery. New Dance Approaches to Nonliteral Choreography. University of Pittsburgh Press, 1971.

19. Wein and Bortnyk, M., "Interactive Skeleton Techniques for Enhancing Motion Dynamics in Key Frame Animation", Communications of ACM, Oct, (1976), 564-569.

Table 1

Commands

P	position
V	view
S	stop
H	help
E	eye
N	name
F	refine
R	reset
M	mirror
B	body pick
C	cursor position
I	spin

Table 2

Individual Body Parts	Body Part Groups
HE----head	LA----left arm
LS----left shoulder	RA----right arm
RS----right shoulder	LL----left leg
SP----spine	RL----right leg
LUA---left upper arm	LLA---left lower arm
RUA---right upper arm	RLA---right lower arm
LFA---left fore arm	LLL---left lower leg
RFA---right fore arm	RLL---right lower leg
LH----left hand	ARMS--both la and ra
RH----right hand	LEGS--body ll and rl
LHIP--left hip	HNDS--both lh and rh
RHIP--right hip	FEETS-both lf and rf
LT----left thigh	
RT----right thigh	
LC----left calf	
RC----right calf	
LF----left foot	
RF----right foot	

Table 3

Major Positions	Refining Directions
HF----high forward	HIG----high
HRF---high right forward	LOW----low
HR----high right	LEF----left
HRB---high right backward	RIG----right
HB----high backward	FOR----forward
HLB---high left backward	BAC----backward
HL----high left	
HLF---high left forward	
PH----place high	
MF----middle forward	
MRF---middle right forward	
MR----middle right	
MRB---middle right backward	
MB----middle backward	
MLB---middle left backward	
ML----middle left	
MLF---middle left forward	
LF----left forward	
LRF---low right forward	
LR----low right	
LRB---low right backward	
LB----low backward	
LLB---low left backward	
LL----low left	
LLF---low left forward	
PL----place low	

Appendix

```
        ∇AEQ[⎕]∇
      ∇ R←X AEQ Y;X;Y
[1]     R←∧/∆CT≥|X−Y
      ∇



        ∇ANAL[⎕]∇
      ∇ ANAL;BPL;SI;IND;CNT
[1]     CNT←1
[2]     AIND←⍳0
[3]     BPL← 0 4 ⍴''
[4]     'WHAT BODY PARTS DO YOU WANT ANALYSED?'
[5]     'ENTER THEM ONE AT A TIME PLEASE.'
[6]   GETIT:→(0=⍴TEM←⍞)/DOIT
[7]     BPL←BPL,[1] 4↑TEM
[8]     →GETIT
[9]   DOIT:IND←(∨/BPLOC∧.=⍉BPL)/⍳1↑⍴BPLOC
[10]    IND←,IND×2
[11]    SI←(1↑⍴ANI)÷39
[12]  GETA:AIND←AIND,IND[CNT]+(0,¯1↓⍳SI)×39
[13]    →((1↑⍴BPL)≥CNT←CNT+1)/GETA
[14]    PIC←ANI[AIND;]
[15]    PIC[;1]←1
[16]    PIC[1+SI×¯1+⍳1↑⍴BPL;1]←0
      ∇



        ∇ANGFIND[⎕]∇
      ∇ ANG←L1 ANGFIND L2;BAVEC;BCVEC;L1;L2;DISBA;DISBC;DI
[1]     BAVEC←L1[2; 2 3 4]−L1[1; 2 3 4]
[2]     BCVEC←L2[2; 2 3 4]−L2[1; 2 3 4]
[3]     DIS L1
[4]     DISBA←DI
[5]     DIS L2
[6]     DISBC←DI
[7]     ANG←¯2○((BAVEC+.×BCVEC)÷(DISBA×DISBC))
[8]     ANG←360×ANG÷2×○1
      ∇


        ∇ANIMA[⎕]∇
      ∇ I ANIMA N;J;DL;TEMP
[1]     TEMP← 0 4 ↑0
[2]     ANI← 0 4 ↑0
[3]     J←0
[4]   LP:DRAW TEMP←J TRANS N
[5]     →(1≥J←J+I)/LP
      ∇
```

```
       ∇ABLINE[⎕]∇
     ∇ R←PARM ABLINE LINE;X;Y;Z;RAT
[1]    DIS LINE
[2]    RAT←PARM÷DI
[3]    X←LINE[1;2]+((LINE[2;2]-LINE[1;2])×RAT)
[4]    Y←LINE[1;3]+((LINE[2;3]-LINE[1;3])×RAT)
[5]    Z←LINE[1;4]+((LINE[2;4]-LINE[1;4])×RAT)
[6]    LINE[2; 2 3 4]←X,Y,Z
[7]    R←LINE
     ∇
```

```
       ∇ABSCALE[⎕]∇
     ∇ R←PARM ABSCALE PIC;SFAC;COUNT;COUNT2;X;Y;Z
[1]    COUNT2←COUNT←1
[2]   GUTS:OLDSIZE←(⌈/PIC[;COUNT+1]-⌊/PIC[;COUNT+1])
[3]    OLDSIZE←((OLDSIZE=0),(OLDSIZE≠0))/SM,OLDSIZE
[4]    SFAC←PARM[COUNT]÷OLDSIZE
[5]    →(GETX,GETY,GETZ)[COUNT]
[6]   GETX:X←SFAC
[7]    →(3≥COUNT←COUNT+1)/GUTS
[8]   GETY:Y←SFAC
[9]    →(3≥COUNT←COUNT+1)/GUTS
[10]  GETZ:Z←SFAC
[11]  XYZ:→((0=X),(0=Y),(0=Z))/REX,REY,REZ
[12]  DOIT:→(3≥COUNT2←COUNT2+1)/XYZ
[13]   R←(X,Y,Z) SCALE PIC
[14]   →0
[15]  REX:X←1
[16]   →DOIT
[17]  REY:Y←1
[18]   →DOIT
[19]  REZ:Z←1
[20]   →DOIT
     ∇
```

```
       ∇ACTIVATOR[⎕]∇
     ∇ ACTIVATOR;LIM
[1]    ACTLIM←1↑⍴STACK
[2]    →(ACTLIM<2)/0
[3]   GETBODF:BODF← 2 4 ↑STACK
[4]    STACK←((2-ACTLIM),4)↑STACK
[5]    →0
     ∇
```

```
        ∇ATTACH[⎕]∇
      ∇ ATTACH
[1]     LS←(SP[2; 2 3 4]-LS[1; 2 3 4]) MOVE LS
[2]     RS←(SP[2; 2 3 4]-RS[1; 2 3 4]) MOVE RS
[3]     HE←(SP[2; 2 3 4]-HE[1; 2 3 4]) MOVE HE
[4]     LUA←(LS[2; 2 3 4]-LUA[1; 2 3 4]) MOVE LUA
[5]     RUA←(RS[2; 2 3 4]-RUA[1; 2 3 4]) MOVE RUA
[6]     LFA←(LUA[2; 2 3 4]-LFA[1; 2 3 4]) MOVE LFA
[7]     RFA←(RUA[2; 2 3 4]-RFA[1; 2 3 4]) MOVE RFA
[8]     LH←(LFA[2; 2 3 4]-LH[1; 2 3 4]) MOVE LH
[9]     RH←(RFA[2; 2 3 4]-RH[1; 2 3 4]) MOVE RH
[10]    LHIP←(SP[1; 2 3 4]-LHIP[1; 2 3 4]) MOVE LHIP
[11]    RHIP←(SP[1; 2 3 4]-RHIP[1; 2 3 4]) MOVE RHIP
[12]    LT←(LHIP[2; 2 3 4]-LT[1; 2 3 4]) MOVE LT
[13]    RT←(RHIP[2; 2 3 4]-RT[1; 2 3 4]) MOVE RT
[14]    LC←(LT[2; 2 3 4]-LC[1; 2 3 4]) MOVE LC
[15]    RC←(RT[2; 2 3 4]-RC[1; 2 3 4]) MOVE RC
[16]    LF←(LC[2; 2 3 4]-LF[1; 2 3 4]) MOVE LF
[17]    RF←(RC[2; 2 3 4]-RF[1; 2 3 4]) MOVE RF
[18]    BODY←HE,[1] SP,[1] LS,[1] RS,[1] LUA,[1] RUA,[1] LFA,[1]
        RFA,[1] LH,[1] RH,[1] LHIP,[1] RHIP,[1] LT,[1] RT,[1] LC,
        [1] RC,[1] LF,[1] RF
[19]    BODY←⌊BODY
[20]    GROUND
      ∇


        ∇BANIMA[⎕]∇
      ∇ I BANIMA N;J;DL;TEMP
[1]     FNUM←0
[2]     SFLAG←0
[3]     FPS←18
[4]     I←(((3=+/TSWITCH='ON  ')),((3=+/TSWITCH='OFF')))/(1÷I×FPS)
        ,(I)
[5]     TEMP← 0 4 ↑0
[6]     ANI← 0 4 ↑0
[7]     J←I
[8]    LP:TEMP←J TRANS N
[9]     BODY← 36 4 ↑TEMP
[10]    NEXTRACT
[11]    →(⎕LC≥40)/INM
[12]   DISPLAY:DRAW PLOT SSIZE BODY
[13]    ANI←ANI,[1] SSIZE BODY
[14]    FNUM←FNUM+1
[15]    SFLAG←1
[16]    →(1≥J←J+I)/LP
[17]    →0
[18]   INM:IN 50
[19]    →DISPLAY
      ∇
```

```
        ∇BODROT[⎕]∇
      ∇ BODROT BOD;A;B;C
[1]     BODY←BOD
[2]     SP←BODY[3 4 ;]
[3]     'HOW MANY DEGREES?'
[4]     BODY←BODY ROT ⎕,(SP[2; 2 3 4],SP[2;2],(SP[2;3]-2),SP[
        2;4])
[5]     NEXTRACT
[6]     GROUP
[7]     ATTACH
[8]     DRAW PLOT SSIZE BODY
      ∇



        ∇BPICK[⎕]∇
      ∇ BPICK;T;T1
[1]     'WHAT BODY DO YOU WANT TO DEAL WITH?'
[2]   GET:T←⍞
[3]     BODY←⍎T
[4]     NEXTRACT
[5]     ATTACH
      ∇



        ∇BPINPUT[⎕]∇
      ∇ BPINPUT;COUNT
[1]     'WHAT IS THE BODY PART'
[2]   GETBPNAME:NAME←4↑⍞
[3]     COUNT←1
[4]   SCAN:→(∧/NAME=BPNAMES[COUNT;])/COUNTHOLD
[5]     →((1↑⍴BPNAMES)≥COUNT←COUNT+1)/SCAN
[6]     'PLEASE REENTER A BODY PART'
[7]     →GETBPNAME
[8]   COUNTHOLD:CHOLD←COUNT
      ∇



        ∇C[⎕]∇
DEFN ERROR
        ∇C
          ∧

        ∇CAT[⎕]∇
      ∇ Z̲←X̲ CAT Y̲
[1]     Z̲←X̲,[1] Y̲
      ∇
```

```
        ∇CDRAW[⎕]∇
      ∇ R̲←CDRAW Z̲
[1]     Z̲[;2]←Z̲[;2]+⌊(Z̲[;4]÷2↓∆VPOINT)×∆VPOINT[1]-Z̲[;2]
[2]     Z̲[;3]←Z̲[;3]+⌊(Z̲[;4]÷2↓∆VPOINT)×∆VPOINT[2]-Z̲[;3]
[3]     Z̲← 0 ¯1 ↓Z̲
[4]     R̲←∆DRAW
      ∇



        ∇COMHELP[⎕]∇
      ∇ COMHELP
[1]     ∆ZAP
[2]     ⎕TS←⎕DL 0.5
[3]     'FOLLOWING IS A LIST OF COMMANDS AND A BRIEF EXPLANATIO
[4]     'OF WHAT THEY DO. TO USE A COMMAND TYPE IN THE UNDERLIN
         LETTER.'
[5]     'P̲OSITION----------PLACES THE ASKED FOR BODY PART IN ONE
        F 26 '
[6]     '                  POSIBLE PLACES.  THE POINTS OF A 3 BY
         CUBE.'
[7]     '                  NAMES OF POSITIONS ARE:',,MAJOR,' '
[8]     '                  M--MIDDLE'
[9]     '                  H--HIGH'
[10]    '                  L--LOW'
[11]    '                  F--FORWARDS'
[12]    '                  B--BACKWARDS'
[13]    '                  P--PLACE (THE CENTER)'
[14]    '                  L--LEFT'
[15]    '                  R--RIGHT'
[16]    'THE FIRST L IN ANY POSITION IS LOW.'
      ∇


        ∇COMINPUT[⎕]∇
      ∇ R←COMINPUT
[1]    AG:'COMMAND'
[2]     INP←⍞
[3]     →(1↑INP='P')/0,R←PCOM
[4]     →(1↑INP='V')/0,R←VCOM
[5]     →(1↑INP='R')/0,R←RCOM
[6]     →(1↑INP='S')/0,R←SCOM
[7]     →(1↑INP='H')/0,R←HCOM
[8]     →(1↑INP='N')/0,R←NCOM
[9]     →(1↑INP='E')/0,R←ECOM
[10]    →(1↑INP='F')/0,R←REFINE
[11]    →(1↑INP='T')/0,R←TCOM
[12]    →(1↑INP='M')/0,R←MCOM
[13]    →(1↑INP='D')/0,R←DCOM
[14]    →(1↑INP='B')/0,R←BCOM
[15]    →(1↑INP='Z')/0,R←ZCOM
[16]    →(1↑INP='I')/0,R←ICOM
[17]    'PLEASE REENTER A COMMAND OR TYPE HELP'
[18]    →AG
      ∇
```

```
        ∇DIS[⎕]∇
      ∇ DIS MAT;I;J;TMAT;BMAT
[1]     I←1
[2]     DIV←⍳0
[3]     TMAT←BMAT←0
[4]    GUTS:TMAT←MAT[I;]
[5]     BMAT←MAT[I+1;]
[6]     DI←(((TMAT[2]-BMAT[2])*2)+((TMAT[3]-BMAT[3])*2)+((TMAT[
        4]-BMAT[4])*2))
[7]     DI←DI*0.5
[8]     DIV←DIV,DI
[9]     →((1↑⍴MAT)>I←I+1)/GUTS
      ∇
```

```
        ∇DISPLAYMAPS[⎕]∇
      ∇ DISPLAYMAPS ARG;COUNT
[1]     ΔZAP
[2]     ⎕TS←⎕DL 1
[3]     →(ARG='L')/ROTDIS
[4]     COUNT←1
[5]    DISPLAY:DRAW⍎MAPNAMES[COUNT;]
[6]     →(18≥COUNT←COUNT+1)/DISPLAY
[7]     →0
[8]    ROTDIS:COUNT←1
[9]    RDISPLAY:DRAW⍎MAPNAMES[COUNT;],' ROT 45,LINE'
[10]    →(18≥COUNT←COUNT+1)/RDISPLAY
      ∇
```

```
        ∇DRAW[⎕]∇
      ∇ DRAW OBJ;COUNT
[1]     ⎕IO←1
[2]     COUNT←1
[3]    GUTS:OB←OBJ[COUNT; 2 3 4]
[4]     OB← 512 390 ¯1000 +EYEO+.×(OB-ΔVPOINT)
[5]     OB←((OB-EYE)÷(OB-EYE)[3]÷-EYE[3])+EYE[1 2],0
[6]    ⍝PUT CLIPPING HERE
[7]    ⍝OBJ IS NOW XY
[8]     OBJ[COUNT; 2 3 4]←(2↑OB),0
[9]     →((1↑⍴OBJ)≥COUNT←COUNT+1)/GUTS
[10]    Z̲← 0 ¯1 ↓⌊OBJ
[11]    ⍞←ΔDRAW
[12]    ⎕←''
      ∇
```

```
       ∇ENDROT[⎕]∇
     ∇ ENDROT;AT;TO;DOWN;SIDE;EXAMPT;ID;ROTAT;ROTTO
[1]    AT←LOC
[2]    TO←PTS[2]
[3]    ID←1
[4]    EXAMPT←AT
[5]   START:DOWN←SIDE←1
[6]   FOUND:→(EXAMPT=CONTAB[DOWN;SIDE])/OUTPUT
[7]    →(8≥SIDE←SIDE+1)/FOUND
[8]    SIDE←1
[9]    →(5≥DOWN←DOWN+1)/FOUND
[10]   'SOMETHING DONE GONE WRONG IN ENDROT'
[11]  OUTPUT:→(ID=2)/R2
[12]   ROTAT←DOWN
[13]   EXAMPT←TO
[14]   ID←ID+1
[15]   →START
[16]  R2:ROTTO←DOWN
[17]   CHECK26
[18]   ROTANG←(ROTAT−ROTTO)×¯45
[19]   ROTLINE←6⍴SPHERE27[ROTCODE[RING2;]; 2 3 4]
     ∇


       ∇FROM[⎕]∇
     ∇ OBODY←FROM POS;SBODY;PART;RESP
[1]   TOP:'WHAT IS THE BP?'
[2]    BP←⍞
[3]    SBODY←BODY
[4]    BODY←⍎'POS'
[5]    NEXTRACT
[6]    PART←⍎BP
[7]    BODY←SBODY
[8]    NEXTRACT
[9]    ⍎BP,'←PART'
[10]   ATTACH
[11]   MAPPER
[12]   OBODY←BODY
[13]   'AGAIN?'
[14]   RESP←⍞
[15]   →('Y'=1↑RESP)/TOP
     ∇
```

```
        ∇FILL[⎕]∇
      ∇ MAT1 FILL MAT2
[1]     'HOW MANY INTERMEDIATE POSITIONS?'
[2]     INTNO←⎕
[3]     SMOVEFAC←MOVEFAC←(MAT2[21; 2 3 4]-MAT1[21; 2 3 4])÷INTNO
[4]     MOVEFAC
[5]     ∆ZAP
[6]     ⎕TS←⎕DL 0.5
[7]     DRAW MAT1
[8]     MAKEAVEC
[9]     ROTER
[10]    DRAW MAT2
      ∇

        ∇GETEYE[⎕]∇
      ∇ GETEYE;TEMP
[1]     'WHAT ARE THE X Y Z COORDINATES OF THE EYE?'
[2]     REYESET
[3]     TEMP←⎕
[4]     RIGHT(TEMP[1]-∆VPOINT[1])
[5]     UP(TEMP[2]-∆VPOINT[2])
[6]     IN(TEMP[3]-∆VPOINT[3])
[7]     ∆ZAP
[8]     LOOK
[9]     DRAW BODY
      ∇

        ∇GROUND[⎕]∇
      ∇ GROUND;LOWBP;YDIF;SW
[1]     SW←GSWITCH
[2]     →(1↑1⌽SW='F')/0
[3]     →((LOWBP←⌊/BODY[;3])≠GLEVEL)/MOVER
[4]     →0
[5]    MOVER:YDIF←GLEVEL-LOWBP
[6]     BODY←(0,YDIF,0) MOVE BODY
      ∇
```

```
       ∇GO[⎕]∇
     ∇ GO
[1]    ∆ZAP
[2]    ⎕TS←⎕DL 0.5
[3]   COM:→COMINPUT
[4]   VCOM:SEE
[5]    →COM
[6]   SCOM:→0
[7]   ECOM:GETEYE
[8]    →COM
[9]   REFINE:REFINER
[10]   →COM
[11]  MCOM:DRAW PLOT MIRROR BODY
[12]   →COM
[13]  ICOM:BODROT BODY
[14]   →COM
[15]  ZCOM:TRANSLATION
[16]   →COM
[17]  BCOM:BPICK
[18]   →COM
[19]  DCOM:DRAW PLOT SSIZE BODY
[20]   →COM
[21]  RCOM:RESET
[22]   →COM
[23]  HCOM:HELP
[24]   →COM
[25]  NCOM:'WHAT IS THE NAME FOR THIS PRESENT POSITION?'
[26]   POSNAME←⍞
[27]   ⍎POSNAME,'←BODY'
[28]   →COM
[29]  TCOM:TWIST
[30]   →COM
[31]  PCOM:BODYMEM←BODY
[32]   BPINPUT
[33]   POSINPUT
[34]   STACKER
[35]   LIM←+/(POINTERS[CHOLD;]≠0)
[36]   INC←1
[37]  ACT:ACTIVATOR
[38]   →(ACTLIM<2)/DISPLAY
[39]  ROTSHIT:MOVSPHERE
[40]   LOCATOR
[41]   BODP[2;]←SPHERE27[PTTO;]
[42]   BODP[2;1]←1
[43]   LOCATOR
[44]   ⍎BPNAMES[POINTERS[CHOLD;INC];],'←BODP'
[45]   INC←INC+1
[46]   ATTACH
[47]   →ACT
[48]  DISPLAY:GROUND
[49]   ∆ZAP
[50]   ⎕TS←⎕DL 1
[51]   MAPPER
[52]   DRAW SSIZE BODY
[53]   →COM
     ∇
```

```
        ∇GROUP[⎕]∇
      ∇ GROUP
[1]     LA←LUA,[1] LFA,[1] LH
[2]     RA←RUA,[1] RFA,[1] RH
[3]     SHOULDERS←LS,[1] RS
[4]     UBOD←HE,[1] LA,[1] RA,[1] SP
[5]     LL←LT,[1] LC,[1] LF
[6]     RL←RT,[1] RC,[1] RF
[7]     HIPS←LHIP,[1] RHIP
[8]     LBOD←LLEG,[1] RLEG,[1] HIPS
[9]     FEET←LF,[1] RF
[10]    HNDS←LH,[1] RH
[11]    RLA←RFA,[1] RH
[12]    LLA←LFA,[1] LH
[13]    ARMS←LA,[1] RA
      ∇
```

```
        ∇HOLD[⎕]∇
      ∇ HOLD
[1]     ∆ZAP
[2]     ⎕TS←⎕DL 1
[3]     ⍞←SIGN
[4]     →('S'=⍞)/0
[5]     1000⍴⎕AV[174]
[6]     →4
      ∇
```

```
        ∇IN[⎕]∇
      ∇ IN N
[1]     ∆VPOINT←∆VPOINT+N×EYEO[3;]
      ∇
```

```
        ∇INCEYE[⎕]∇
      ∇ INCEYE
[1]     →((TIME<SPA[1])∨(TIME>SPA[2]))/0
[2]     MLINE←MLINE ROT INC,RLINE
[3]     ∆VPOINT←MLINE[2; 2 3 4]
[4]     LOOK
[5]     STAND
      ∇
```

```
        ∇INTREFINE[⎕]∇
      ∇ INTREFINE
[1]     AD←(2 1 ⍴0), 2 3 ⍴MEMA,MEMD
[2]     DIS AD
[3]     DIST1←DI
[4]     TESTP←BODP ROT ANG,RLINE
[5]     TL2←TESTP[2; 2 3 4],[1] D
[6]     TL2←(2 1 ⍴0), 2 3 ⍴TL2
[7]     DIS TL2
[8]     DIST2←DI
[9]     →(DIST2<DIST1)/CONT
[10]    ANG←-ANG
[11]   CONT:INC←0
[12]   POP:ACTIVATOR
[13]    DRAW BODP←BODP ROT ANG,RLINE
[14]    INC←INC+1
[15]    ⍎BPNAMES[POINTERS[CHOLD;INC];],'←BODP'
[16]    ATTACH
[17]    →(0<1↑⍴STACK)/POP
[18]    →0
      ∇


        ∇LINEFIND[⎕]∇
      ∇ LINEFIND
[1]     →((RING2=1),(RING2=5),(RING2=2),(RING2=6),(RING2=
        3),(RING2=7),(RING2=4),(RING2=8))/L1,L1,L2,L2,L3,L3,L4,
[2]    L1:ROTLINE←,SPHERE27[8 4 ; 2 3 4]
[3]     →0
[4]    L2:ROTLINE←,SPHERE27[11 14 ; 2 3 4]
[5]     →0
[6]    L3:ROTLINE←,SPHERE27[17 20 ; 2 3 4]
[7]     →0
[8]    L4:ROTLINE←,SPHERE27[26 23 ; 2 3 4]
      ∇


        ∇LOCATOR[⎕]∇
      ∇ LOCATOR
[1]    ⍝FINDS PRESENT SPHERE27 LOCATION OF BP⍝
[2]     LCOUNT←2
[3]    SEARCH:→(BODP[2; 2 3 4] AEQ SPHERE27[LCOUNT; 2 3
        4])/FOUND
[4]     ⎕CT←1E¯13
[5]     →(27≥LCOUNT←LCOUNT+1)/SEARCH
[6]    FOUND:LOC←LCOUNT
[7]     ⎕CT←1E¯13
      ∇
```

```
        ∇LOOK[⎕]∇
      ∇ LOOK;X
[1]     X←ΔACENTER-ΔVPOINT
[2]     X←(EYEO+.×X)÷(+/X*2)*0.5
[3]     X←180×(¯1○X)÷○1
[4]     YAW X[1]
[5]     PITCH X[2]
      ∇


        ∇MAKEAVEC[⎕]∇
      ∇ MAKEAVEC;COUNT;INDEX;SINDEX;L1;L2;LL1;LL2;TLINE;DIST1;
        DIST2;MOVEFAC
[1]     FSAVE← 0 4 ⍴0
[2]     COUNT←1
[3]     ANGVEC←18⍴0
[4]     MOVEVEC← 18 3 ⍴0
[5]    MOVER:INDEX←¯1+COUNT×2
[6]     L1←MAT1[INDEX,(INDEX+1);]
[7]     L2←MAT2[INDEX,(INDEX+1);]
[8]     L1←(L2[1; 2 3 4]-L1[1; 2 3 4]) MOVE L1
[9]     ANGVEC[COUNT]←L1 ANGFIND L2
[10]    →(18≥COUNT←COUNT+1)/MOVER
[11]    MEMANGVEC←ANGVEC
[12]    ANGVEC←ANGVEC÷INTNO
[13]    COUNT←1
[14]   ⍝CODE FOR SIGN CHECKING⍝
[15]   LOOP:→(0≠ANGVEC[COUNT])/DOIT
[16]    →(18≥COUNT←COUNT+1)/LOOP
[17]    →0
[18]   DOIT:SINDEX←COUNT×2
[19]    LL1←MAT1[(¯1+SINDEX),SINDEX;]
[20]    LL2←MAT2[(¯1+SINDEX),SINDEX;]
[21]    MOVEFAC←(LL2[1; 2 3 4]-LL1[1; 2 3 4])
[22]    LL1←MOVEFAC MOVE LL1
[23]    MOVEVEC[COUNT;]←MOVEFAC
[24]    TLINE← 2 4 ⍴LL1[2;],LL2[2;]
[25]    DIS TLINE
[26]    DIST1←DI
[27]    LL1 ROTL LL2
[28]    LL1←LL1 ROT ANGVEC[COUNT],RLINE
[29]    TLINE← 2 4 ⍴LL1[2;],LL2[2;]
[30]    DIS TLINE
[31]    DIST2←DI
[32]    →(DIST2≤DIST1)/ENDTEST
[33]    ANGVEC[COUNT]←-ANGVEC[COUNT]
[34]   ENDTEST:→(18≥COUNT←COUNT+1)/LOOP
      ∇
```

```
        ∇NEXTRACT[⎕]∇
      ∇ NEXTRACT
[1]     HE←BODY[1 2 ;]
[2]     SP←BODY[3 4 ;]
[3]     LS←BODY[5 6 ;]
[4]     RS←BODY[7 8 ;]
[5]     LUA←BODY[9 10 ;]
[6]     RUA←BODY[11 12 ;]
[7]     LFA←BODY[13 14 ;]
[8]     RFA←BODY[15 16 ;]
[9]     LH←BODY[17 18 ;]
[10]    RH←BODY[19 20 ;]
[11]    LHIP←BODY[21 22 ;]
[12]    RHIP←BODY[23 24 ;]
[13]    LT←BODY[25 26 ;]
[14]    RT←BODY[27 28 ;]
[15]    LC←BODY[29 30 ;]
[16]    RC←BODY[31 32 ;]
[17]    LF←BODY[33 34 ;]
[18]    RF←BODY[35 36 ;]
      ∇
```

```
        ∇PITCH[⎕]∇
      ∇ PITCH X
[1]     EYEO←(3 3 ⍴ 1 0 0 0 ,(2 1 ○-X),0, 1 2 ○X←○X÷180)+.×EYEO
      ∇
```

```
        ∇PLOT[⎕]∇
      ∇ TBOD←PLOT BOD
[1]     TBOD←BOD[18 14 10 6 1 8 12 16 20 1 2 34 30 26 22
        21 24 28 32 36 ;]
[2]     TBOD←TBOD,[1] BOD[6 21 8 ;]
[3]     TBOD[22 15 16 5 ;1]←1
[4]     TBOD[21 12 9 1 ;1]←0
      ∇
```

```
        ∇PUT[⎕]∇
      ∇ R̲←L̲ PUT X̲
[1]     L̲←(3↑L̲)-⌊/[1] X̲[; 2 3 4]
[2]     R̲←⍉(⌽⍴X̲)⍴X̲[;1],(X̲[;2]+L̲[1]),(X̲[;3]+L̲[2]),X̲[;4]+L̲[
        3]
      ∇
```

```
        ∇MOVE[⎕]∇
      ∇ Z̲←L̲ MOVE X̲
[1]     L̲←3↑L̲
[2]     Z̲←⍉(⌽⍴X̲)⍴X̲[;1],(X̲[;2]+L̲[1]),(X̲[;3]+L̲[2]),X̲[;4]+L̲[
        3]
      ∇
```

```
        ∇MOVSPHERE[⎕]∇
      ∇ MOVSPHERE
[1]     DIS BODP
[2]     DI←DI×2
[3]     SPHERE27←(DI,DI,DI) ABSCALE SPHERE27
[4]     SPHERE27←(BODP[1; 2 3 4]-SPHERE27[1; 2 3 4]) MOVE
        SPHERE27
      ∇
```

```
        ∇NDIS[⎕]∇
      ∇ NDIS L1;L1
[1]     DIST←(((L1[1;2]-L1[2;2])*2)+((L1[1;3]-L1[2;3])*2)+((L1[1;
        4]-L1[2;4])*2))*0.5
      ∇
```

```
        ∇NEW[⎕]∇
      ∇ R←NEW OBJ;COUNT
[1]     ⎕IO←1
[2]     COUNT←1
[3]    GUTS:OB←OBJ[COUNT; 2 3 4]
[4]     OB←((OB-EYE)÷(OB-EYE)[3]÷-EYE[3])+EYE[1 2],0
[5]     OBJ[COUNT; 2 3 4]←(2↑OB),0
[6]     →((1↑⍴OBJ)≥COUNT←COUNT+1)/GUTS
[7]     R←⌊OBJ
      ∇
```

```
        ∇NREV[⎕]∇
      ∇ POS1←NREV POS2
[1]     BODY←POS2
[2]     NEXTRACT
[3]     BODY←HE CAT SP CAT RS CAT LS CAT RUA CAT LUA CAT RFA CAT
        LFA CAT RH CAT LH CAT RHIP CAT LHIP CAT RT CAT LT CAT RC
        CAT LC CAT RF CAT LF
[4]     NEXTRACT
[5]     ATTACH
[6]     POS1←BODY
      ∇
```

```
        ∇MAP[⎕]∇
      ∇ MAP MAT2;COUNT;COUNT2;INDEX
[1]     RESETMAPS
[2]     MAT1←S̲BODY
[3]     MAKEAVEC
[4]     ANGVEC[((ANGVEC≠0)/⍳⍴ANGVEC)]←INTNO×ANGVEC[((ANGVEC≠0)/
        ANGVEC)]
[5]     COUNT←1
[6]    CHECKER:→(ANGVEC[COUNT]≠0)/DOIT
[7]     →(18≥COUNT←COUNT+1)/CHECKER
[8]     →DISP
[9]    DOIT:INDEX←COUNT×2
[10]    LMEM←⍎MAPNAMES[COUNT;]
[11]    LMEM←LMEM[1 2 ;]
[12]    LTO←MAT2[(¯1+INDEX),INDEX;]
[13]    LMEM ROTL LTO
[14]    ⍎MAPNAMES[COUNT;],'←',MAPNAMES[COUNT;],' ROT ANGVEC[COUI
        ],RLINE'
[15]    NEXTRACT
[16]    MAPAT
[17]    →(18≥COUNT←COUNT+1)/CHECKER
[18]   DISP:COUNT2←1
[19]   DISPLAY:DRAW⍎MAPNAMES[COUNT2;]
[20]    →(18≥COUNT2←COUNT2+1)/DISPLAY
      ∇
```

```
        ∇MAPPER[⎕]∇
      ∇ MAPPER
[1]     BODY← 36 4 ↑BODY
[2]     NEXTRACT
[3]     MAPSP← 3 4 ⍴LS[2;] CAT LHIP[1;] CAT RS[2;]
[4]     MAPSP[;1]←1
[5]     MAPSP[1;1]←0
[6]     BODY←BODY CAT MAPSP
      ∇
```

```
        ∇MIRROR[⎕]∇
      ∇ SPOS←MIRROR POS
[1]     'MIRROR IMAGE IS CALLED MBOD'
[2]     SPOS←POS[;1],(|POS[;2]-1024),POS[; 3 4]
[3]     MBOD←REVERSE SPOS
[4]     BODY←POS
[5]     NEXTRACT
[6]     GROUP
[7]     ATTACH
      ∇
```

```
       ∇POSINPUT[⎕]∇
     ∇ POSINPUT;COUNT
[1]   GETPOS:'WHERE DO YOU WANT TO MOVE?'
[2]    MOVETO←3↑⎕
[3]    COUNT←1
[4]   SEARCH:→(3=(+/MOVETO=MAJOR[COUNT;]))/FOUND
[5]    →(27≥COUNT←COUNT+1)/SEARCH
[6]    'PLEASE REENTER A LOCATION'
[7]    →GETPOS
[8]   FOUND:PTTO←COUNT
     ∇


       ∇RANDPOS[⎕]∇
     ∇ RANDPOS
[1]    ΔZAP
[2]    SPHERE27←S̲PHERE27
[3]    BODY←S̲BODY
[4]    RBOD←BODY
[5]    LIM←?18
[6]    COUNT←1
[7]   GETBP:INDEX←?18
[8]    INDEX2←?18
[9]    POINTER←¯1+2×INDEX
[10]   BODP←BODY[(POINTER,POINTER+1);]
[11]   DIS BODP
[12]   DI←DI×2
[13]   SPHERE27←(DI,DI,DI) ABSCALE SPHERE27
[14]   SPHERE27←(BODP[1; 2 3 4]-SPHERE27[1; 2 3 4]) MOVE
       SPHERE27
[15]   RNO←1+?26
[16]   RBOD[(POINTER+1);]←SPHERE27[CONSTRAINTS[INDEX;RNO];]
[17]   →(LIM≥COUNT←COUNT+1)/GETBP
[18]   HE←RBOD[1 2 ;]
[19]   SP←RBOD[3 4 ;]
[20]   LS←RBOD[5 6 ;]
[21]   RS←RBOD[7 8 ;]
[22]   LUA←RBOD[9 10 ;]
[23]   RUA←RBOD[11 12 ;]
[24]   LFA←RBOD[13 14 ;]
[25]   RFA←RBOD[15 16 ;]
[26]   LH←RBOD[17 18 ;]
[27]   RH←RBOD[19 20 ;]
[28]   LHIP←RBOD[21 22 ;]
[29]   RHIP←RBOD[23 24 ;]
[30]   LT←RBOD[25 26 ;]
[31]   RT←RBOD[27 28 ;]
[32]   LC←RBOD[29 30 ;]
[33]   RC←RBOD[31 32 ;]
[34]   LF←RBOD[33 34 ;]
[35]   RF←RBOD[35 36 ;]
[36]   ATTACH
[37]   BODY[(2×⍳18);1]←1
[38]   PPLOT BODY
[39]   PEND
[40]   PLOTNO←PLOTNO+1
[41]   'PLOT-';PLOTNO
```

```
          ∇RESETMAPS[⎕]∇
        ∇ RESETMAPS;COUNT
[1]       HEMAP←S̲HEMAP
[2]       SPMAP←S̲SPMAP
[3]       LSMAP←S̲LSMAP
[4]       RSMAP←S̲RSMAP
[5]       LUAMAP←S̲LUAMAP
[6]       RUAMAP←S̲RUAMAP
[7]       LFAMAP←S̲LFAMAP
[8]       RFAMAP←S̲RFAMAP
[9]       LHMAP←S̲LHMAP
[10]      RHMAP←S̲RHMAP
[11]      LHIPMAP←S̲LHIPMAP
[12]      RHIPMAP←S̲RHIPMAP
[13]      RTMAP←S̲RTMAP
[14]      LTMAP←S̲LTMAP
[15]      RCMAP←S̲RCMAP
[16]      LCMAP←S̲LCMAP
[17]      RFMAP←S̲RFMAP
[18]      LFMAP←S̲LFMAP
        ∇
```

```
          ∇REVERSE[⎕]∇
        ∇ POS1←REVERSE POS2;NLUA;NRUA;NLFA;NRFA;NLH;NRH;NLHIP;NRHI
          ;NLT;NRT;NLC;NRC;NLF;NRF;NBODY
[1]       NBODY← 0 4 ↑0
[2]       BODY←POS2
[3]       NEXTRACT
[4]       NLHIP←RHIP
[5]       NRHIP←LHIP
[6]       NLS←RS
[7]       NRS←LS
[8]       NLUA←RUA
[9]       NRUA←LUA
[10]      NLFA←RFA
[11]      NRFA←LFA
[12]      NLH←RH
[13]      NRH←LH
[14]      NLT←RT
[15]      NLC←RC
[16]      NLF←RF
[17]      NRT←LT
[18]      NRC←LC
[19]      NRF←LF
[20]      NBODY←HE CAT SP CAT NLS CAT NRS CAT NLUA CAT NRUA CAT
          NLFA CAT NRFA CAT NLH CAT NRH CAT NLHIP CAT NRHIP CAT NLT
          CAT NRT CAT NLC CAT NRC CAT NLF CAT NRF
[21]      BODY←NBODY
[22]      NEXTRACT
[23]      ATTACH
[24]      POS1←BODY
        ∇
```

```
       ∇RNDPOS[⎕]∇
     ∇ RNDPOS;SBODY;PART;RESP;X;Y
[1]   VVTOP;POSMEM←POSMEM,⎕RL
[2]    X←70
[3]    Y←50
[4]   VTOP;COUNT←1
[5]   TOP;POS←⍎'NUB',⍕?765
[6]    BP←BPNAMES[BPPOI[COUNT];]
[7]    SBODY←BODY
[8]    BODY←POS
[9]    NEXTRACT
[10]   PART←⍎BP
[11]   BODY←SBODY
[12]   NEXTRACT
[13]   ⍎BP,'←PART'
[14]   ATTACH
[15]   MAPPER
[16]   →(17≥COUNT←COUNT+1)/TOP
[17]   BODY←L 300 50 0 PUT 1.5 1.5 1.5 SCALE SSIZE BODY
[18]   SETSP
[19] ⍝          ⍎(('RDPOS',(⍕⍎'RDNUM')),'←BODY')
[20] ⍝          RDNUM←RDNUM+1
[21]   BODY[4;1]←0
     ∇


       ∇RESET[⎕]∇
     ∇ RESET
[1]    'SURE?'
[2]    100ρ⎕AV[174]
[3]    →('N'=1↑⍞)/0
[4]    SPHERE27←S̲PHERE27
[5]    GROUP
[6]    HE←S̲HE
[7]    LS←S̲LS
[8]    RS←S̲RS
[9]    LUA←S̲LUA
[10]   RUA←S̲RUA
[11]   LFA←S̲LFA
[12]   RFA←S̲RFA
[13]   LH←S̲LH
[14]   RH←S̲RH
[15]   LHIP←S̲LHIP
[16]   RHIP←S̲RHIP
[17]   LT←S̲LT
[18]   RT←S̲RT
[19]   LC←S̲LC
[20]   RC←S̲RC
[21]   LF←S̲LF
[22]   RF←S̲RF
[23]   SP←S̲SP
[24]   ATTACH
[25]   MAPPER
[26]   ΔZAP
[27]   ⎕TS←⎕DL 1
[28]   DRAW SSIZE BODY
     ∇
```

```
        ∇ROTER[⎕]∇
      ∇ ROTER;LIMCOUNT;COUNT;INDEX;L1;L2;TEMP
[1]     MEML2←MEML← 0 8 ⍴0
[2]     RLMAT← 0 6 ⍴0
[3]     LIMCOUNT←1
[4]   TOP:COUNT←1
[5]   LOOP:→(0≠ANGVEC[COUNT])/DOIT
[6]     →(18≥COUNT←COUNT+1)/LOOP
[7]     MOVEFAC←MOVEFAC+SMOVEFAC
[8]     BODY←⌊BODY←MOVEFAC MOVE BODY
[9]     NEXTRACT
[10]    FSAVE←FSAVE CAT BODY
[11]    MAPPER
[12]    DRAW BODY←⌊BODY
[13]    TEMP←BODY
[14]    →(INTNO>LIMCOUNT←LIMCOUNT+1)/TOP
[15]    →0
[16]  DOIT:INDEX←2×COUNT
[17]    L1←MAT1[(¯1+INDEX),INDEX;]
[18]    L2←MAT2[(¯1+INDEX),INDEX;]
[19]    L2←(L1[1; 2 3 4]−L2[1; 2 3 4]) MOVE L2
[20]    L1 ROTL L2
[21]    DIS L1
[22]    L1←L1 ROT ANGVEC[COUNT],RLINE
[23]    L1←DI ABLINE L1
[24]    RLMAT←RLMAT CAT RLINE
[25]    MAT1[(¯1+INDEX),INDEX;]←L1
[26]    BODY←MAT1
[27]    NEXTRACT
[28]    ATTACH
[29]    MAT1←BODY
[30]    →(18≥COUNT←COUNT+1)/LOOP
[31]    MOVEFAC←MOVEFAC+SMOVEFAC
[32]    'MOVEFAC−';MOVEFAC;'LIM−';LIMCOUNT
[33]    BODY←MOVEFAC MOVE BODY
[34]    'MOVED BODY DOWN HERE'
[35]    NEXTRACT
[36]    MAPPER .
[37]    FSAVE←FSAVE CAT BODY
[38]  DISPLAY:DRAW BODY
[39]  ENDCHECK:→(INTNO>LIMCOUNT←LIMCOUNT+1)/TOP
      ∇

        ∇RTNSET[⎕]∇
      ∇ RTNSET S
[1]     13⌶ΔPROMPT[ΔASCΔ⍳,S]
      ∇
```

```
         ∇RINGID[⎕]∇
       ∇ RINGID PTS
[1]      PTFROM←PTTOPT[PTS[1]-1;RING2]
[2]      PTTO←PTTOPT[PTS[2]-1;RING2]
[3]      PTFROM
[4]      PTTO
       ∇

         ∇ROLL[⎕]∇
       ∇ ROLL X
[1]      EYEO←(3 3 ⍴(2○X),(1○X),0,(-1○X),(2○X←-X×○÷180), 0 0 0 1)
         .×EYEO
       ∇

         ∇ROT[⎕]∇
       ∇ Z̲←D̲ATA ROT A̲NGPT;A̲;D̲IR;V̲;R̲1;R̲2;R̲;P̲
[1]      →(1≠⍴,A̲NGPT)/D̲TR
[2]      A̲NGPT←A̲NGPT,C,C
[3]     D̲TR:A̲←○(1↑A̲NGPT)÷180
[4]      Z̲←(0 1 ↓D̲ATA),(1↑⍴D̲ATA)⍴1
[5]      D̲ATA←D̲ATA[;1]
[6]      Z̲[;1]←Z̲[;1]-A̲NGPT[2]
[7]      Z̲[;2]←Z̲[;2]-A̲NGPT[3]
[8]      Z̲[;3]←Z̲[;3]-A̲NGPT[4]
[9]      P̲←|A̲NGPT[5 6 7]-A̲NGPT[2 3 4]
[10]     D̲IR←P̲÷(+/P̲*2)*0.5
[11]     V̲←(+/(D̲IR*2)[2 3])*0.5
[12]     R̲1← 4 4 ⍴(5↑1),(D̲IR[3 2]÷V̲), 0 0 ,(¯1 1 ×D̲IR[2 3]÷V̲),
         ¯5↑1
[13]     R̲2← 4 4 ⍴V̲,0,D̲IR[1], 0 0 1 0 0 ,(-D̲IR[1]),0,V̲,¯5↑1
[14]     R̲← 4 4 ⍴(2○A̲),(-1○A̲), 0 0 ,(1 2 ○A̲),10⍴¯5↑1
[15]     A̲←1,0/V̲←1↑⍴Z̲
[16]    L̲OOP:Z̲[A̲;]←⌊((((Z̲[A̲;]+.×R̲1)+.×R̲2)+.×R̲)+.×⌹R̲2)+.×⌹R̲1
[17]     →(V̲≥A̲←A̲+1)/L̲OOP
[18]     Z̲[;1]←Z̲[;1]+A̲NGPT[2]
[19]     Z̲[;2]←Z̲[;2]+A̲NGPT[3]
[20]     Z̲[;3]←Z̲[;3]+A̲NGPT[4]
[21]     Z̲←D̲ATA, 0 ¯1 ↓Z̲
       ∇

         ∇ROTL[⎕]∇
       ∇ VER ROTL ENDPT;RL
[1]      MEMA←A←ENDPT[2; 2 3 4]
[2]      MEMB←B←VER[1; 2 3 4]
[3]      MEMD←D←VER[2; 2 3 4]
[4]      VECBA←A-B
[5]      VECBC←D-B
[6]      X1←VECBA[1]
[7]      Y1←VECBA[2]
[8]      Z1←VECBA[3]
[9]      X2←VECBC[1]
[10]     Y2←VECBC[2]
[11]     Z2←VECBC[3]
[12]     RL←(((Y1×Z2)-(Y2×Z1)),((Z1×X2)-(Z2×X1)),((X1×Y2)-(X2×Y1)
         )
[13]     RLINE←B,(B+RL)
       ∇
```

```
        ∇REYESET[⎕]∇
      ∇ REYESET
[1]     ∆VPOINT← 512 390 ¯1000
[2]     EYEO← 3 3 ⍴ 1 0 0 0
      ∇
```

```
        ∇RFINDER[⎕]∇
      ∇ VER RFINDER ENDPT;RL;B
[1]     SIXDIR←SPHERE27[2 6 4 8 17 20 ; 2 3 4]
[2]     MEMA←A←ENDPT[2; 2 3 4]
[3]     MEMB←B←VER[1; 2 3 4]
[4]     MEMD←D←,SIXDIR[MEMINDEX;]
[5]     VECBA←A−B
[6]     VECBC←D−B
[7]     X1←VECBA[1]
[8]     Y1←VECBA[2]
[9]     Z1←VECBA[3]
[10]    X2←VECBC[1]
[11]    Y2←VECBC[2]
[12]    Z2←VECBC[3]
[13]    RL←(((Y1×Z2)−(Y2×Z1)),((Z1×X2)−(Z2×X1)),((X1×Y2)−(X2×Y1)
        )
[14]    RLINE←B,(B+RL)
      ∇
```

```
        ∇RINGFIND[⎕]∇
      ∇ RINGFIND POINTS
[1]     PTS←POINTS
[2]     ID←1
[3]    START:DOWN←SIDE←1
[4]    FOUND:→(POINTS[ID]=PTRING[DOWN;SIDE])/OUTPUT
[5]     →(5≥SIDE←SIDE+1)/FOUND
[6]     SIDE←1
[7]     →(8≥DOWN←DOWN+1)/FOUND
[8]     'SOMETHING DONE GONE WRONG IN RINGFIND'
[9]    OUTPUT:→(ID=2)/R2
[10]    RING1←DOWN
[11]    ID←ID+1
[12]    →START
[13]   R2:RING2←DOWN
      ∇
```

```
        ∇RIGHT[⎕]∇
      ∇ RIGHT N
[1]     ∆VPOINT←∆VPOINT+EYEO[1;]×N
      ∇
      ∇SEE[⎕]∇
    ∇ SEE;FARM
[1]   MAPPER
[2]   'PICK A VIEPOINT OR MODE NOW'
[3]   'TOP,BOTTOM,RIGHT,LEFT,FRONT OR ARBITRARY OR MOVE'
[4]  RE:FARM←⍞
[5]   ∆ZAP
[6]   FARM←1↑FARM
[7]   →((FARM='T'),(FARM='B'),(FARM='R'),(FARM='L'),(FARM='F'),
      (FARM='A'),(FARM='M'),(FARM='S'))/TOP,BOT,RIGH,LEFT,FRON,
      ARB,MOV,STOP
[8]   →0
[9]  FRON:REYESET
[10]  DRAW BODY
[11]  →RE
[12] BOT:REYESET
[13]  UP ¯1000
[14]  IN 500
[15]  LOOK
[16]  DRAW BODY
[17]  →RE
[18] RIGH:REYESET
[19]  RIGHT ¯1000
[20]  IN 500
[21]  LOOK
[22]  DRAW BODY
[23]  →RE
[24] LEFT:REYESET
[25]  RIGHT 1000
[26]  IN 500
[27]  LOOK
[28]  DRAW BODY
[29]  →RE
[30] TOP:REYESET
[31]  UP 1000
[32]  IN 500
[33]  LOOK
[34]  DRAW BODY
[35]  →RE
[36] ARB:'YOU MAY NOW MOVE THE EYE AT WILL ENTER ONE AT A TIME
      S TO STOP'
[37]  REYESET
[38] GET:→('S'=R←⍞)/DIS
[39]  ⍎R
[40]  →GET
[41] DIS:∆ZAP
[42]  LOOK
[43]  DRAW BODY
[44]  →RE
[45] MOV:'ENTER COMMANDS ONE AT A TIME S TO STOP'
[46] GETM:→('S'=R←⍞)/DISM
[47]  ⍎R
[48]  →GETM
[49] DISM:∆ZAP
[50]  LOOK
[51]  DRAW BODY
[52]  →RE
[53] STOP:→0
```

```
      ∇SETPATH[⎕]∇
    ∇ SETPATH
[1]   'WHAT IS THE STARTING AND ENDING TIME FOR THE PATH?'
[2]   SPA←⎕
[3]   TIME←SPA[2]-SPA[1]
[4]   REYESET
[5]   SPHERE27← 3000 3000 3000 ABSCALE SPHERE27
[6]   SPHERE27←(,ΔACENTER-,SPHERE27[1; 2 3 4]) MOVE SPHERE27
[7]   POSINPUT
[8]   POS←PTTO
[9]   POSINPUT
[10]  ANGLE←SPHERE27[(POS,1,PTTO);]
[11]  LIM←ANGLE[1 2 ;] ANGFIND ANGLE[2 3 ;]
[12]  LIM←((LIM=0),(LIM≠0))/180,LIM
[13]  SPHERE27[(1,POS);] ROTL SPHERE27[(1,PTTO);]
[14]  INC←TIME÷FPS
[15]  MLINE←SPHERE27[(1,POS);]
[16]  ANG←0
[17]  DIS 2 4 ρ(MLINE[2;],SPHERE27[PTTO;])
[18]  MEMDIS←DI
[19]  TL←MLINE ROT INC,RLINE
[20]  DIS 2 4 ρ(TL[2;],SPHERE27[PTTO;])
[21]  INC←((MEMDIS>DI),(MEMDIS<DI))/INC,-INC
    ∇
```

```
      ∇SETSP[⎕]∇
    ∇ SETSP
[1]   NEXTRACT
[2]   BODP←SP
[3]   MOVSPHERE
[4]   SP[2;]←SPHERE27[SPVEC[?ρSPVEC];]
[5]   ATTACH
[6]   MAPPER
[7]   BODY[3;1]←0
[8]   DRAW PLOT BODY
    ∇
```

```
      ∇TOP[⎕]∇
    ∇ TOP PIC
[1]   REYESET
[2]   UP 1000
[3]   LOOK
[4]   DRAW PIC
    ∇
```

```
      ∇UP[⎕]∇
    ∇ UP N
[1]   ΔVPOINT←ΔVPOINT+EYEO[2;]×N
    ∇
```

```
       ∇STAGE[⎕]∇
     ∇ STAGE;CNT;ST;LIM
[1]    POS← 0 10 ρ''
[2]    ST← 0 3 ρ0
[3]    'TYPE IN POSITION THEN MARK POSITION  WITH CURSOR'
[4]   AG:POS←POS,[1] 10↑⍞
[5]    ST←ST,[1] C
[6]    'AGAIN?'
[7]    →('Y'=1↑⍞)/AG
[8]    ST← 1 1 2 SCALE(0,ST) ROT-90, 0 0 0 100 0 0
[9]    LIM←1↑ρPOS
[10]   CNT←1
[11]  DOIT:⍎POS[CNT;],'←ST[CNT;2 3 4] PUT ',POS[CNT;]
[12]   →(LIM≥CNT←CNT+1)/DOIT
[13]   CNT←1
[14]  LP:DRAW⍎POS[CNT;]
[15]   →(LIM≥CNT←CNT+1)/LP
     ∇
```

```
       ∇STACKER[⎕]∇
     ∇ STACKER;COUNT;LIM
[1]    TSTACK←STACK← 0 4 ρ⍳0
[2]    COUNT←1
[3]    LIM←+/POINTERS[CHOLD;]≠0
[4]   GETSTACK:TSTACK←⍎,BPNAMES[POINTERS[CHOLD;(¯1↑(COUNT↑⍳LIM)
       ];]
[5]    PLHOLD←(¯1↑(COUNT↑⍳LIM))
[6]    JOISTACK←⍎,JOINAMES[POINTERS[CHOLD;(¯1↑(COUNT↑⍳LIM))];]
[7]    STACK←STACK CAT TSTACK
[8]    JOISTACK←JOISTACK,[0.5] JOISTACK
[9]    →(LIM≥COUNT←COUNT+1)/GETSTACK
[10]   BODP← 2 4 ↑STACK
     ∇
```

```
       ∇TRANS[⎕]∇
     ∇ X̲←N̲ TRANS M̲;A̲;B̲;I̲
[1]    B̲←¯1↓(M̲⍳'/')↑M̲
[2]    A̲←(M̲⍳'/')↓M̲
[3]    M̲←1↑ρX̲←⍎A̲
[4]    I̲←1
[5]   L̲OOP:X̲[I̲;]←⌊(N̲×X̲[I̲;])+(1-N̲)×⍎B̲,'[',(⍕I̲),';]'
[6]    →(M̲≥I̲←I̲+1)/L̲OOP
     ∇
```

```
      ∇TRANSLATION[⎕]∇
    ∇ TRANSLATION;RESP
[1]   'PLACE THE BODY WITH THE CURSORS'
[2]   BODY←C PUT BODY
[3]   NEXTRACT
[4]   GROUP
[5]   ATTACH
    ∇


      ∇VIEW[⎕]∇
    ∇ VIEW Z̲
[1]   ⍞←ΔDRAW
[2]   ⍞←''
    ∇


      ∇YAW[⎕]∇
    ∇ YAW X
[1]   EYEO←(3 3 ⍴(2○-X),0,(1○-X), 0 1 0 ,(1○X),0,2○X←○X÷
      180)+.×EYEO
    ∇


      ∇ZOOM[⎕]∇
    ∇ ZOOM PARMS;COUNT
[1]   COUNT←1
[2]  GUTS:⍎S̲BPNAMES[COUNT;],'←PARMS SCALE ⍎S̲BPNAMES[COUNT;]'
[3]   →(18≥COUNT←COUNT+1)/GUTS
    ∇


      ∇CHORUS[⎕]∇
    ∇ CHORUS;TEMP;LIM;FMAT;CNT
[1]   FMAT← 0 20 ⍴''
[2]   'TYPE IN POSITIONS (ONE AT A TIME)'
[3]  TOP:TEMP←⍞
[4]   →(0=⍴TEMP)/DISPLAY
[5]   TEMP←20↑TEMP
[6]   FMAT←FMAT,[1] TEMP
[7]   →TOP
[8]  DISPLAY:REYESET
[9]   IN ¯2000
[10]  RIGHT 1200
[11]  CNT←1
[12]  LIM←1↑⍴FMAT
[13] TD:BOD←⍎,FMAT[CNT;]
[14]  DRAW PLOT SSIZE((CNT×250),0,(CNT×1)) PUT BOD
[15]  →(LIM≥CNT←CNT+1)/TD
    ∇
```

```
        ∇SSIZE[⎕]∇
      ∇ POS1←SSIZE POS;VEC
[1]     VEC←SIZEVEC1
[2]     BODY←POS
[3]     NEXTRACT
[4]     HE←VEC[1] ABLINE HE
[5]     LS←VEC[2] ABLINE LS
[6]     RS←VEC[3] ABLINE RS
[7]     LUA←VEC[4] ABLINE LUA
[8]     RUA←VEC[5] ABLINE RUA
[9]     LFA←VEC[6] ABLINE LFA
[10]    RFA←VEC[7] ABLINE RFA
[11]    LH←VEC[8] ABLINE LH
[12]    RH←VEC[9] ABLINE RH
[13]    RHIP←VEC[10] ABLINE RHIP
[14]    LHIP←VEC[11] ABLINE LHIP
[15]    LT←VEC[12] ABLINE LT
[16]    RT←VEC[13] ABLINE RT
[17]    LC←VEC[14] ABLINE LC
[18]    RC←VEC[15] ABLINE RC
[19]    LF←VEC[16] ABLINE LF
[20]    RF←VEC[17] ABLINE RF
[21]    SF←VEC[18] ABLINE SF
[22]    ATTACH
[23]    NEXTRACT
[24]    MAPPER
[25]    POS1←BODY
      ∇
```

```
        ∇STAND[⎕]∇
      ∇ STAND
[1]     ROLL 180×(¯1○EYEO[1;2])÷○1
      ∇
```

```
        ∇SWITCH[⎕]∇
      ∇ ONE SWITCH TWO;TEMP
[1]     TEMP←ONE
[2]     ONE←TWO
[3]     TWO←TEMP
      ∇
      ∇ΔXRD[⎕]∇
    ∇ Z̲←ΔXRD X̲;ΔXBF
[1]   ΔXBF←⍳0
[2]   →(6≥⍴X̲)/L̲T7
[3]   ΔXBF←14⌶C̲APL[1+ΔXBF,¯6↓X̲]
[4]   →C̲OM,⍴X̲←¯6↑X̲
[5]  L̲T7:ΔXBF←14⌶C̲APL[1+ΔXBF]
[6]  C̲OM:13⌶,ΔPROMPT[1+X̲]
[7]   Z̲←¯1+C̲APL⍳⍞
[8]   13⌶ΔPROMPT[8 8]
    ∇
```

```
      ∇ΔZAP[⎕]∇
    ∇ R̲←ΔZAP
[1]   C̲APL[28 13]
[2]   R̲←0/⎕DL 1
    ∇
```

MANY OF THE PURELY GRAPHIC ROUTINES
WERE WRITTEN BY DAVID TOURETZKY AND
ARE PART OF THE INTERART GRAPHICS SYSTEM.

```
        ∇ΔCMX[⎕]∇
      ∇ Z̲←ΔCMX X̲
[1]     Z̲←⌽Z̲[3 1]+(Z̲← 32 1 32 1 ×32|X̲)[4 2]
      ∇


        ∇ΔDRAW[⎕]∇
      ∇ R̲←ΔDRAW;⎕IO;C
[1]     ⎕IO←0
[2]     C←Z̲[;0]
[3]     Z̲← 0 1 ↓Z̲
[4]     Z̲←((⍴Z̲)⍴ 32 96 32 64)+Z̲←((⌊Z̲÷32),(32|Z̲))[; 1 3 0
        2]
[5]     C← 29 ¯1[0 1 ⍳C]
[6]     R̲←C̲APL[29,((Z̲≠¯1)/Z̲←,C,Z̲),31]
      ∇



        ∇ΔFCON[⎕]∇
      ∇ R̲←ΔFCON A̲;R̲R;I̲;D̲;F̲
[1]     R̲←3⍴I̲←⍴0/R̲R←999999
[2]   F̲L:→((1↑⍎'⍴',Q̲)<I̲←I̲+1)/R̲ET
[3]     F̲←⍎Q̲,'[',(⍕I̲),';]'
[4]     →(~F̲[1]∊ 0 1 2)/F̲L
[5]     D̲←+/(2↑A̲-F̲←1↓F̲)*2
[6]     →(D̲>R̲R)/F̲L
[7]     →F̲L,R̲R←D̲,0/R̲←F̲
[8]   R̲ET:→~⍳R̲R<ΔMAXDIS*2
[9]     20⍴C̲APL[8]
[10]    R̲←⍳0
      ∇



        ∇ΔFSC[⎕]∇
      ∇ Z̲←ΔFSC
[1]     Z̲←4↑Z̲[1],ΔCMX 1↓Z̲←5↑ΔXRD(ΔTMO←31), 27 26 17
      ∇


        ∇ΔFXMC[⎕]∇
      ∇ Z̲←ΔFXMC X̲
[1]     Z̲←⌊ 32 96 32 64 +((X̲÷32),32|X̲←,X̲)[2 4 1 3]
      ∇


        ∇ΔLINES[⎕]∇
      ∇ ΔLINES;D̲;N̲EW1;N̲EW2;N̲EWDATA;I̲NC1;I̲NC2
[1]     D̲← ¯3 ¯3 ↑Z̲,[1] D̲ATA
[2]     I̲NC1←(D̲[2;]-D̲[1;])÷ΔNUMLINES
[3]     I̲NC2←(D̲[3;]-D̲[2;])÷ΔNUMLINES
[4]     N̲EW1←(⍳ΔNUMLINES)∘.×I̲NC1
[5]     N̲EW2←(⍳ΔNUMLINES)∘.×I̲NC2
[6]     N̲EW1←N̲EW1+(⍴N̲EW1)⍴D̲[1;]
[7]     N̲EW2←N̲EW2+(⍴N̲EW2)⍴D̲[2;]
[8]     N̲EWDATA←⌊0.5+((2×ΔNUMLINES),4)⍴(0,N̲EW1),1,N̲EW2
[9]     VIEW N̲EWDATA
[10]    D̲ATA←D̲ATA,[1] N̲EWDATA
```

```
          CONTAB
   2   2   2   2   2   2   2   2
   9   3  15  10  21  16  27  22
   8   4  14  11  20  17  26  23
   7   5  13  12  19  18  25  24
   6   6   6   6   6   6   6   6


          MAJOR
FC
FH
HL
ML
LL
FL
LR
MR
HR
HLF
MLF
LLF
LRB
MRB
HRB
HF
MF
LF
LB
MB
HB
HRF
MRF
LRF
LLB
MLB
HLB
```

```
POINTERS
  1  0  0  0  0  0  0  0  0  0  0
  2  0  0  0  0  0  0  0  0  0  0
  3  0  0  0  0  0  0  0  0  0  0
  4  0  0  0  0  0  0  0  0  0  0
  5  0  0  0  0  0  0  0  0  0  0
  6  0  0  0  0  0  0  0  0  0  0
  7  0  0  0  0  0  0  0  0  0  0
  8  0  0  0  0  0  0  0  0  0  0
  9  0  0  0  0  0  0  0  0  0  0
 10  0  0  0  0  0  0  0  0  0  0
 11  0  0  0  0  0  0  0  0  0  0
 12  0  0  0  0  0  0  0  0  0  0
 13  0  0  0  0  0  0  0  0  0  0
 14  0  0  0  0  0  0  0  0  0  0
 15  0  0  0  0  0  0  0  0  0  0
 16  0  0  0  0  0  0  0  0  0  0
 17  0  0  0  0  0  0  0  0  0  0
 18  0  0  0  0  0  0  0  0  0  0
  5  7  9  0  0  0  0  0  0  0  0
  6  8 10  0  0  0  0  0  0  0  0
 13 15 17  0  0  0  0  0  0  0  0
 14 16 18  0  0  0  0  0  0  0  0
 15 17  0  0  0  0  0  0  0  0  0
 16 18  0  0  0  0  0  0  0  0  0
  7  9  0  0  0  0  0  0  0  0  0
  8 10  0  0  0  0  0  0  0  0  0
  2  1  3  4  5  7  9  6  8 10  0
 11 12 13 15 17 14 16 18  0  0  0
 13 15 17 14 16 18  0  0  0  0  0
  5  7  9  6  8 10  0  0  0  0  0
 11 12  0  0  0  0  0  0  0  0  0
  9 10  0  0  0  0  0  0  0  0  0
 17 18  0  0  0  0  0  0  0  0  0
```

```
      CONVERT26
 15 19
 21 25
 10  6
 16 12
 22 18
 28 24
  2  6
  9 13
```

```
      PTRING
 16 17 18  2  6
 22 23 24  0  0
  7  8  9  0  0
 13 14 15  0  0
 19 20 21  0  0
 25 26 27  0  0
  3  4  5  0  0
 10 11 12  0  0
```

```
      CONSTRAINTS
   3   9   8   4  16  17  21  20  18  10  11  14  15  22  23  27  26   2   2   2   2   2   2   2
           2   2   2
   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
           2   2   2
   3   3   3   3   3   3   3   3   3   4   4   4   4   4   4   4   4   4   5   5   5   5   5   5
           5   5   5
   7   7   7   7   7   7   7   7   7   8   8   8   8   8   8   8   8   8   9   9   9   9   9   9
           9   9   9
   2   3   4   5   6   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24
          25  26  27
   2   3   4   5   6   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24
          25  26  27
   2   3   4   5   6   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24
          25  26  27
   2   3   4   5   6   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24
          25  26  27
   2   3   4   5   6   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24
          25  26  27
   2   3   4   5   6   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24
          25  26  27
   4   4   4   4   4   4   4   4   4   4   4   4   4   4   4   4   4   4   4   4   4   4   4   4
           4   4   4
   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8
           8   8   8
   2   3   4   5   6   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24
          25  26  27
   2   3   4   5   6   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24
          25  26  27
   2   3   4   5   6   6   7   8   9  10  11  12  13  14  15   6   6   6  19  20  21  22  23  24
          25  26  27
   2   3   4   5   6   6   7   8   9  10  11  12  13  14  15   6   6   6  19  20  21  22  23  24
          25  26  27
  17   3   4   5   6   6   7   8   9  10  11  12  13  14  15  16  17  18  17  17  17  22  23  24
          25  26  27
  17   3   4   5   6   6   7   8   9  10  11  12  13  14  15  16  17  18  17  17  17  22  23  24
          25  26  27
          BPLOC
HE
SP
LS
RS
LUA
RUA
LFA
RFA
LH
RH
LHIP
RHIP
LT
RT
LC
RC
LF
RF
          BPNAMES
HE
SP
LS
RS
LUA
RUA
LFA
RFA
LH
RH
LHIP
RHIP
LT
RT
LC
RC
LF
RF
LA
RA
LL
RL
LLL
RLL
LLA
RLA
UBOD
LBOD
LEGS
ARMS
HIPS
HNDS
FEET
```