To the Graduate Council:

I am submitting herewith a thesis written by Chi-Lin Tom entitled "MAKER3D: Three-Dimensional Interactive Figure Creator." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

Mike Moshell, Major Professor

We have read this thesis
and recommend its acceptance:

Robert M. Aiken

Terry Ashley

Accepted for the Council:

Vice Chancellor
Graduate Studies and Research

MAKER3D: THREE-DIMENSIONAL INTERACTIVE FIGURE CREATOR

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Chi-Lin Tom

June 1981

ACKNOWLEDGEMENTS

I would like to thank Dr. Mike Moshell for offering me the chance to work with him on expanding the interactve graphics into three-dimension on APPLE II machine. And I appreciate the support that Dr. Terry Ashley has provided me. Also I would like to thank Dr. Robert Aiken for helping me on my thesis and studies.

ABSTRACT

A three-dimensional interactive graphics package, named MAKER3D, is proposed in this thesis. It is based on MAKER which is a two-dimensional figure creator for the Rascal animation system. Many computer graphics applications frequently require realism in the display of objects, for which two-dimensional graphics cannot meet. MAKER3D was created for a high degree of realism on APPLE II Microcomputers. It adds the third dimension for depth and clips the figure properly. Hidden surfaces and shadow control will be added on MAKER3D soon. The user can change the figure interactively, without prior experience or any knowledge of data structures.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE PAGE

CHAPTER I

INTRODUCTION

1.1 Interactive Graphics

Interactive computer graphics plays a very important role in computer graphics. Its display systems not only allow the users to have information reentered in graphical form but also to interact directly with the system to create and manipulate graphic objects. An interactive graphics system consists of the user, the data, and the machine [Lee, 1976]. A system has users who form an integral part of its operation. The data of the system is a link between the user and the machine. The flow of data must organized rationally to maximize the convenience of the user. The machine includes hardware and software. The software consists of programs which implement mathematical techniques and decision processes for the control of visual images. Things one needs to think about when producing a realistic image of a three-dimensional object on a two-dimensional display, are: How is depth, the third dimension, to be displayed on the screen? How are hidden surfaces to be removed from the image? How are the parts of the object which exceed the prescribed limits of the viewport to be removed? How is the shadow to be controlled?

Let us compare MAKER3D to some other graphics packages and editors. APPLE PILOT is a programming language designed

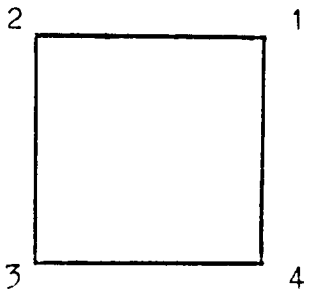
for writing teaching programs on APPLE II Microcomputers. Its editor allows the user to produce graphic images on the screen. It is a two-dimensional non-interactive graphics editor. The users can draw points, lines, boxes, and circles on the screen in positions determined by a coordinate of 560 (0-559) columns and 511 (0-510) rows which APPLE PILOT automaticlly converts into the correct APPLE II high-resolution graphics screen (280 by 192). Text characters can appear anywhere on the screen.

MAKER3D is a three-dimensional interactive graphics package. The user can draw lines on a three-dimensional coordinate. MAKER3D use a central projection to get a two-dimensional image on the same resolution screen. Text characters can appear on the whole screen in textmode or on the last four lines of the mixed screen, in graphics mode.

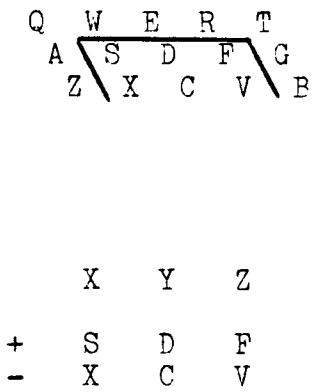
MAKER [Amann, 1980], a part of the Rascal animation system, is a two-dimensional interactive graphics editor which allows its user to create new figures and modify existing ones. MAKER3D takes MAKER as a model and expands it to three dimensions. The depth, the third dimension, of an object is displayed on the screen. MAKER3D controls all rotations, translations and scalings by matrix multiplication. In MAKER, when working on a single line, a temporary work line already exists; then the user modifies it as needed. But in MAKER3D, when working on a single line, the cursor flashes at the end of the last line and the

user has to use a KEYPAD around the letter pairs D,C to get a work line, then work on it. This avoids confusion in three-dimensional space. This difference is shown in Figure 1. The numbers in figure 1 are the data points of the figure in order. Additional procedures are added to MAKER3D such as: moving the location of the center of a figure, removing several lines of a figure, standardizing and normalizing a figure.

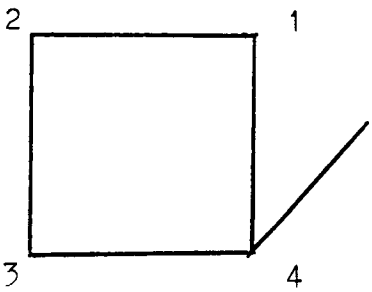
Another interactive graphics package named Three-Dimensional Graphics, was created on an IBM 1130 by Dr. Kaiman Lee of Environmental Design & Research Center, N.J. [Lee, 1976]. This three-dimensional graphics system is a tool used at any stage of design by the architect/planner to get a feeling of the spatial and massing effect of the design. It provides a means for simulation of a real view of a design as seen from any point of interest by the user. It can receive information in a number of ways--three-dimensional coordinate points, defining lines, planes, or cubes. Input may be done on-line interactively or off-line by using a digitizer. Hidden surfaces are removed from the screen. Shadows can be cast on a reference plane (i.e. ground) of an object (i.e. building) when the location of the light source is given. The users can obtain a printed copy of the perspective generated by the Display command.



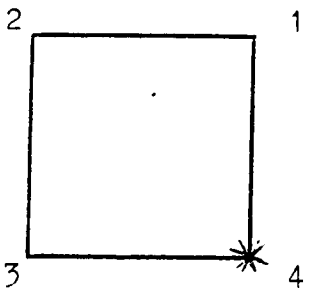
(a) Original figure



(b) The KEYPAD in keyboard and for three axes



(c) Temporary work line for MAKER



(d) Using KEYPAD to get work line for MAKER3D

Figure 1. Working on a Single Line

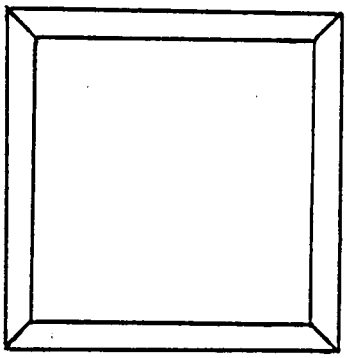
MAKER3D provides a particular location and direction to view the object on the screen. There is a procedure VIEW which allows the user to change the viewer's location and direction to any point of view. Expansion of the program to provide hidden surfaces and shadow control is planned for MAKER3D. A two-dimensional Tablet and a hardware matrix multiplication have been ordered and will be used by MAKER3D to increase its speed and input power.

1.2 Introduction to MAKER3D

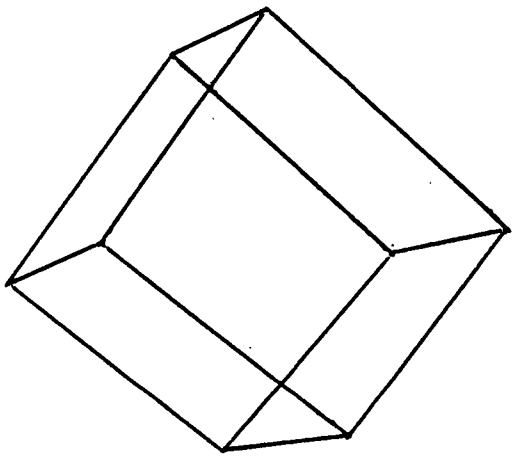
MAKER3D is an interactive three-dimensional graphics package which allows its user to create new figures and manipulate existing ones. A figure is shown on the screen by projecting the three-dimensional data into a two-dimensional image. The user can change the size, orientation and position of an entire figure in real time on the graphics screen. The user also can combine two figures to make a new figure. The shape of a figure can be changed by manipulating one line at a time or several lines of a figure can be deleted at the same time. A figure is a line drawing.

When working on single lines, the user starts working on the last line of the figure. The line being worked on can be stretched to anywhere on the screen. Its color can be changed, or the line can be removed. The user can add the line to the figure and work on a new line following the line just added.

The user can fill the background with a color or ask for help at anytime. Several figures can be loaded and displayed. The user can turn off figures as desired or turn off all the figures except the work figure. Asking for the directory gives the name of all the figures in memory, showing each figure for two seconds as its name appears. The first color of a figure can be changed. A figure can be given a new name. A figure can be changed back to the original figure as it was before any translations, rotations and scalings to avoid cumulative rotation errors and distortion. Also, the user can "normalize" a figure; that is, declare the current figure as the new "original figure". Figure 2 shows two cubes where drawn before and after a lot of rotations. The location of the center of a figure can be changed. In general, a user can do almost anything to make and manipulate a figure to her needs, doing it without knowledge of data structures. MAKER3D maintains relatively elaborate data structures in order to handle figures, but when the user wants to save a figure onto disk, MAKER3D only puts the data necessary. For instance, there are two sets of data points for any figure on MAKER3D. This will be discussed in Chapter IV. Those two sets can be derived from each other, so MAKER3D just puts one set of the data points on the disk.



(a) Original figure



(b) After a number of rotations

Figure 2. Normalizing and Standardizing a Figure

MAKER3D uses three basic structures which are shown in figure 3 in order to control figures and individual lines in figures. The data structures are linked lists in dynamic storage.

The FIGURE head structure is composed of a singly linked list. It is actually composed of two linked lists, but one is a subset of the other. The FIGURE head of each figure contains an ordinal value to identify it.

The second data structure is a head node for MAKER3D. It contains the name, the ordinal, and a pointer to the FIGURE head. This structure also contains a boolean variable indicating if the figure has been saved or not.

The line segment structure is a doubly linked list, and is the body of a figure. FIGURE traverses this list in SHOW, but all manipulation of the list itself is done by MAKER3D. This structure provides MAKER3D' ability to shape figures. This is the only list that can go down in size by a significant amount, so a freelist is maintained for line segments.

All the lines are drawn by a procedure MOVETO which is provided by APPLE TURTLEGRAPHICS. MOVETO allows the user to move the cursor in the x and y directions and draw a line to it.

There are basically two levels in MAKER3D, one being the figure level where the user can have several figures on the screen and all changes are upon the entire figure.

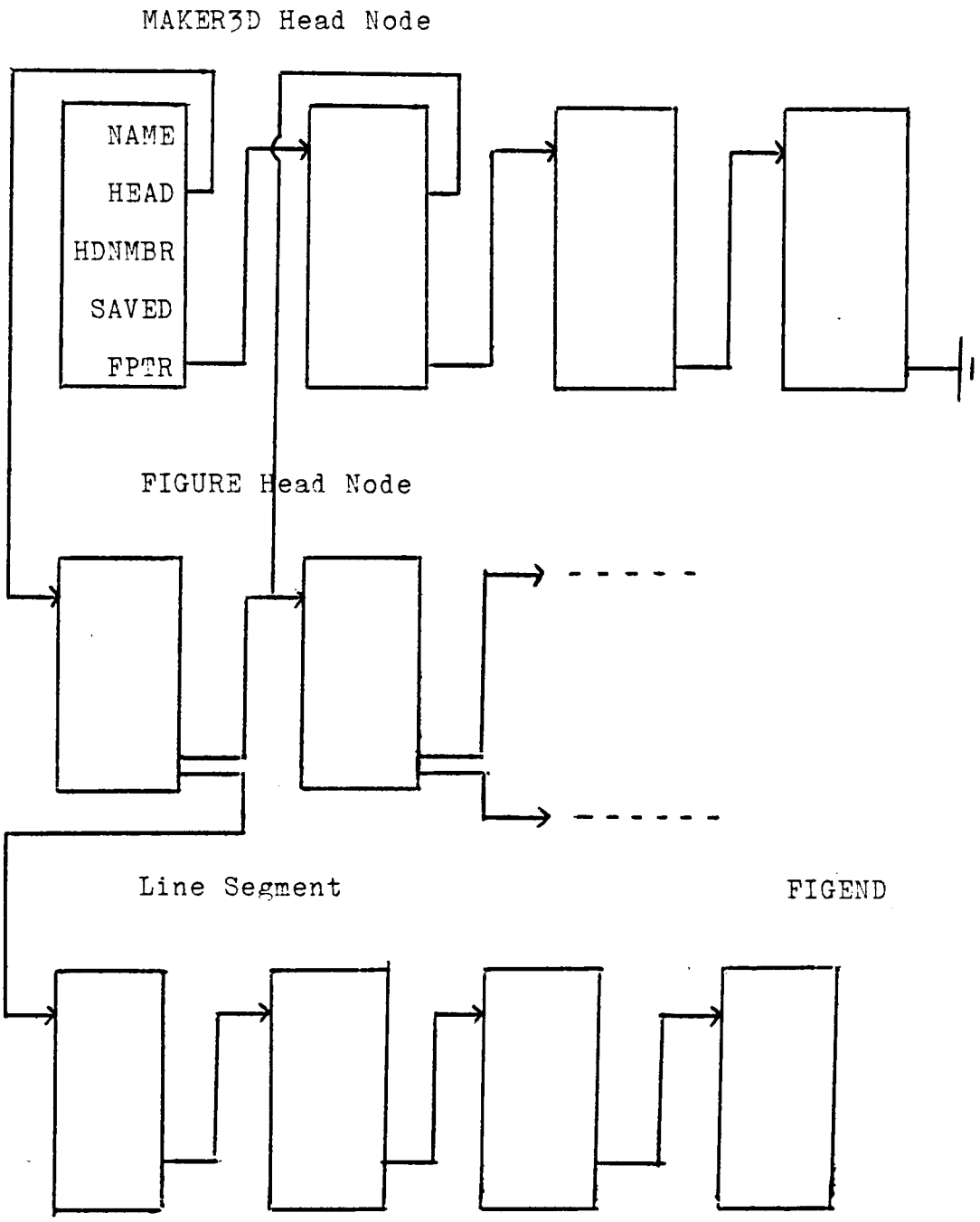


Figure 3. Data Structures of MAKER3D

Paddles are used to set a scale factor and angle for scaling and rotation. A keypad around the letters D,C is used to position the figure and the cursor. The above things will be discussed in more detail in Chapter III.

The other level is the line level where there is only one figure on the screen, and changes are made to individual lines in the figure. The user moves the cursor around the screen with a line "rubberbanding" from it back to the previous line. Rubberbanding is a technique where a line changes its length and angle as necessary to connect a fixed point (the end of the previous line) to a moving point (the cursor). She positions the line as desired, then adds it to the figure. Work then begins on the next line.

MAKER3D attempts to simplify its use while giving the user as much power as possible. It is aimed at users who have very little computer knowledge and experience.

CHAPTER II

THREE-DIMENSION TRANSFORMATIONS AND PERSPECTIVE

Geometric transformations play an essential role in generating images of three-dimensional scenes. Those transformations are used to achieve the effect of different viewing positions and directions. Since the display devices are two-dimensional, there is a perspective transformation required in many three-dimensional visualization techniques, to project the three-dimensional scene onto a two-dimensional screen [Newman, 1979].

Three-dimensional transformations can be represented by a 4 x 4 matrix. The transformation of a point (x,y,z) to a new point (x',y',z') by means of any sequence of translations, rotations, and scalings is then represented as (see Figure 4)

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1] \begin{pmatrix} a & b & c & 0 \\ d & e & f & 0 \\ g & h & i & 0 \\ j & k & l & 1 \end{pmatrix}$$

Figure 4. Matrix Representation

where the 4 x 4 matrix specifies the transformation.

Conversely, to get the point (x,y,z) back from (x',y',z') by means of the matrix inversion. It is

necessary to use the following formula (see Figure 5):

$$[x \quad y \quad z \quad 1] = [x' \quad y' \quad z' \quad 1] \begin{pmatrix} a & b & c & 0 \\ d & e & f & 0 \\ g & h & i & 0 \\ j & k & l & 1 \end{pmatrix}^{-1}$$

Figure 5. Matrix Inversion

2.1 Translation

To translate a point (x,y,z) to a new point (x',y',z') requires the formula (see Figure 6):

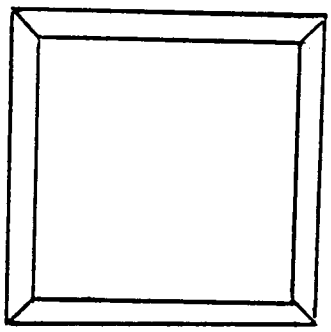
$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1] \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ Tx & Ty & Tz & 1 \end{pmatrix}$$

Figure 6. Translation Transformation

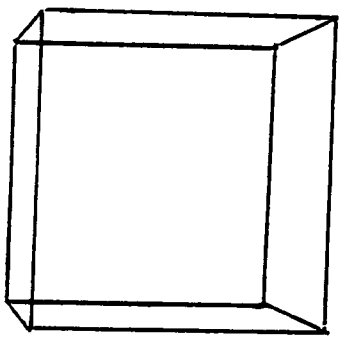
where Tx, Ty and Tz are the components of the translation in the x, y and z directions, respectively. Figure 7 is a series of three translations of a cube in x, y, z individual directions.

2.2 Rotation

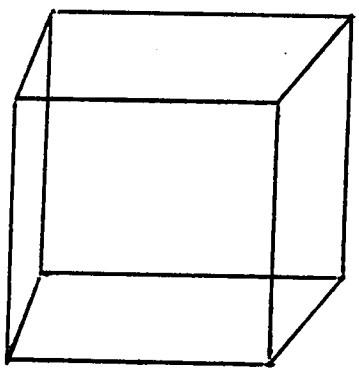
To rotate about a arbitrary point we must concatenate three transformations:



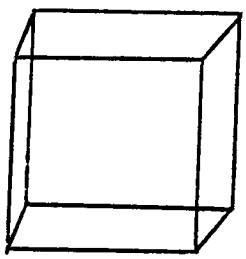
(a) Original figure



(b) After translation of (a) about x-axis



(c) After translation of (b) about y-axis



(d) After translation of (c) about z-axis

Figure 7. Translating a Cube

1. Translate the point to the origin.

2. Perform the rotation.

3. Translate the origin back.

To rotate a point (x,y,z) through an angle θ about the x-axis, it is necessary to use the equation (see Figure 8):

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 8. Rotation Transformation about X-axis

where θ is measured clockwise about the origin when looking along the x-axis toward the origin. Figure 9 shows this and the other two axes.

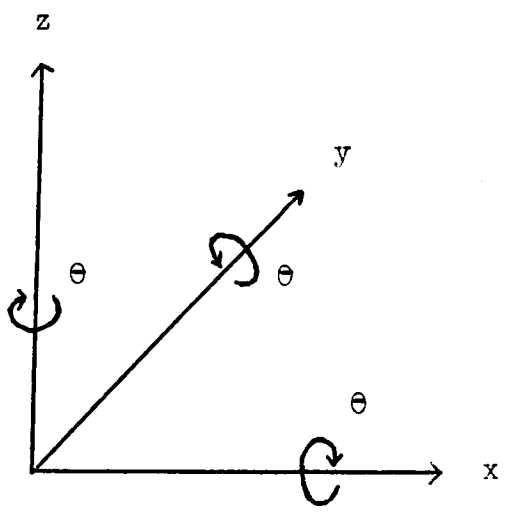


Figure 9. Three-Dimensional Rotations

To rotate a point (x,y,z) through an angle θ about the y-axis (see Figure 10):

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1] \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

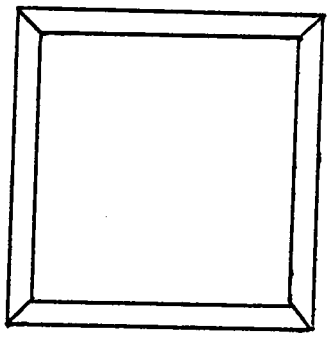
Figure 10. Rotation Transformation about Y-axis

To rotate a point (x,y,z) through an angle θ about z-axis (see Figure 11):

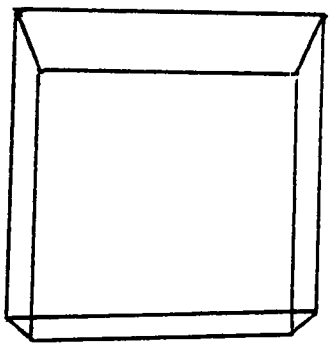
$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1] \begin{pmatrix} \cos 0 & -\sin 0 & 0 & 0 \\ \sin 0 & \cos 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 11. Rotation Transformation about Z-axis

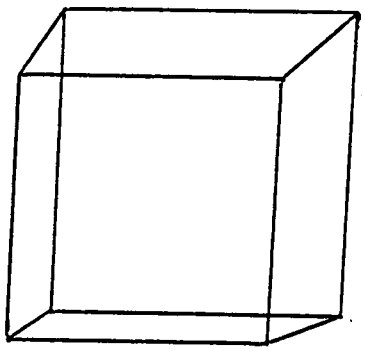
Figure 12 is a series of three rotations of a cube about three different axes.



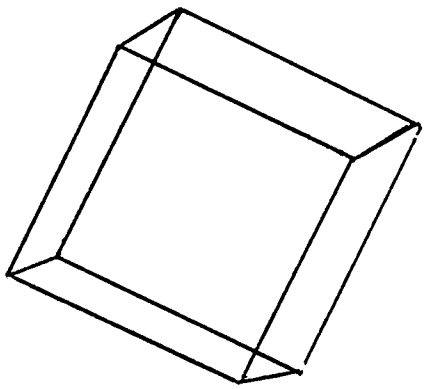
(a) Original figure



(b) After rotation of (a) about x-axis



(c) After rotation of (b) about y-axis



(d) After rotation of (c) about z-axis

Figure 12. Rotating a Cube Through 30 Degree

2.3 Scaling

To scale a point (x,y,z) to a new point (x',y',z') uses the formula (see Figure 13):

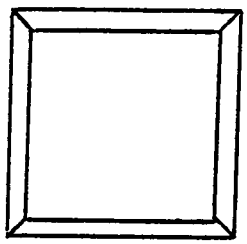
$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1] \begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 13. Scaling Transformation

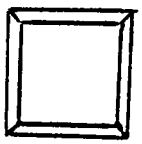
where Sx, Sy and Sz are the scale factor of the scaling in the x, y and z directions separately. Figure 14 shows two scalings of a cube. The scale factors of x, y, z axes are set to the same value.

2.4 Perspective

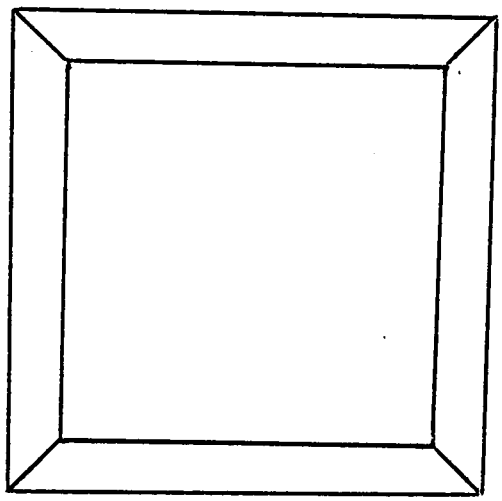
A display screen is a two-dimensional space, so we cannot directly display three-dimensional objects; we must project each point of an object onto the plane of the display screen. This operation is called a perspective projections. Before the object is projected on the display screen, we must first adjust the viewer's position for the point of view. This can be done by several translations and rotations that are determined by the viewing parameters. Then a central projection is used to get the image on the display screen. The projection produces an image point of the corresponding object point. In a central projection, the center of projection, also called the viewpoint, is



(a) Original figure



(b) After scaling through scale factor 0.5



(c) After scaling through scale factor 2.0

Figure 14. Scaling a Cube

located on one of the axes of the three-dimensional coordinate system. Figure 15 shows the projection and viewpoint located on the z-axis [Newman, 1979, Giloi, 1978].

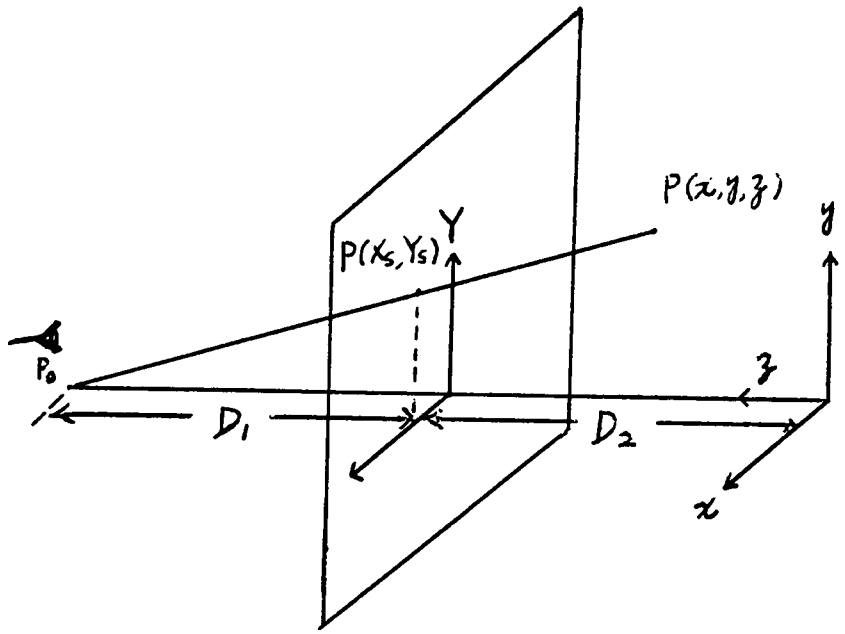


Figure 15. Perspective Projection

We can get relations Xs = D1 * x / (D1 + D2 - z) and Ys = D1 * y / (D1 + D2 - z) by triangle similarities. The numbers Xs and Ys can be converted into dimensionless fractions by dividing by the screen size. Alternatively, we can convert Xs and Ys to screen coordinates by including a specification of the location of the viewport in which the image is displayed (see Figure 16)

$$Xs = \left(\frac{D1 \cdot x}{(D1 + D2 - z) \cdot Sx} \right) \cdot Vsx + Vcx$$

$$Ys = \left(\frac{D1 \cdot y}{(D1 + D2 - z) \cdot Sy} \right) \cdot Vsy + Vcy$$

Figure 16. Two-Dimension (Xs,Ys) from (x,y,z)

where Sx and Sy are screen size, Vsx, Vcx, Vsy and Vcy are screen coordinate controls and can be calculated from four viewport parameters. Figure 17 shows the APPLE II high-resolution graphics screen and four viewport parameters. For instance, Vsx = (Vxr - Vxl) / 2 and Vcx = (Vxr + Vxl) / 2.

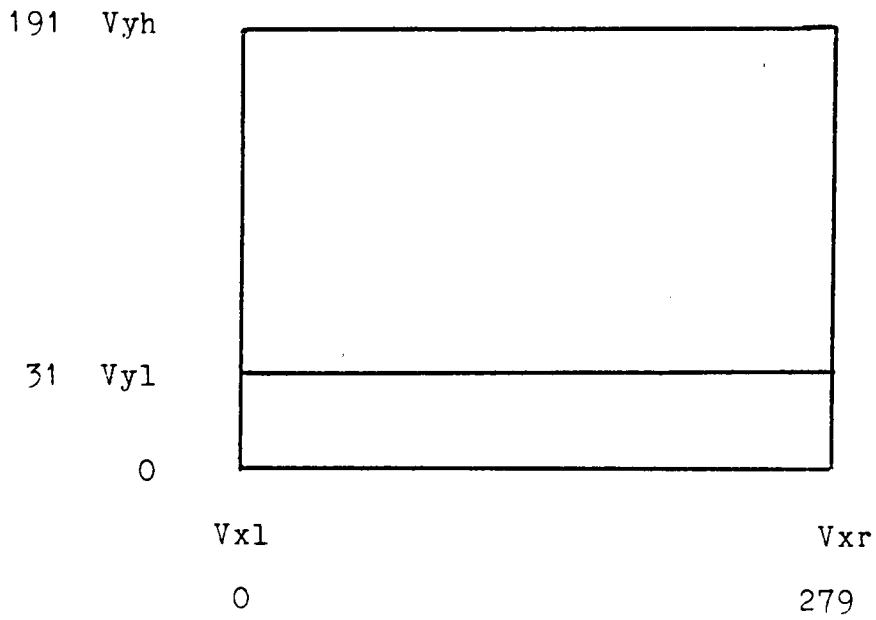


Figure 17. APPLE II High-Resolution Screen

CHAPTER III

USER'S MANUAL

MAKER3D is a graphics package that is used to build figures. These figures are composed of lines. In MAKER3D, the cursor moves by using keystrokes in three different directions and draws the line temporarily by using a rubberbanding move method. When you have the line the way you want it, you add it to the figure. The entire figure can be rotated, translated, and scaled. All this is done by using the two paddles provided with the APPLE II, and by keystrokes.

There is a 'menu' displayed at the bottom of the graphics screen. It consists of words which indicate the 'procedures' you can use to build figures. These procedures are processed by typing the first letter of the word being displayed. For instance, there is a procedure called GET which is invoked by typing G. It gets a figure from memory or from disk to display on the screen.

There are two basic menus in MAKER3D. The first one is used when working with entire figures. The second menu is processed through a procedure in the first menu. It is used for working on lines in a particular figure, called the work figure. For instance, one procedure called EXTEND is invoked by typing E. It goes to line mode(level) which allows you to work on the work figure. A second menu is

displayed at the bottom of the screen. It is easy to build three-dimensional figures with MAKER3D. The following sections explain how to use this graphics package.

3.1 Figure Control

Work is done on one figure (the work figure) at a time, even though you can have several figures in memory and on the screen and can switch back and forth among them to adjust each of them. The size and direction of the work figure are set by the game paddles. The position on the screen is set by a keypad. An explanation of these controls follows.

PADDLE 0

After selecting the coordinate axis for rotation, you can use PADDLE(0) to set a rotate angle to do the job. When the figure or the line is first set to be rotated, you have already got a value from PADDLE(0). So every time the image is redrawn, the rotate angle is compared to the last value of PADDLE(0). Type X, Y or Z to change the axis you want to rotate about. Typing any other key ends the rotation mode.

PADDLE 1

When you are ready to scale the figure, you can use PADDLE(1) to set a scale factor. When the figure is first set to be scaled, you already have a value from PADDLE(1). So every time the figure is redrawn, the

scale factor is in comparison to the last value of PADDLE(1). The scale factor is used for all the three axes. scale can change the size of the figure from as small as a half to twice its original size.

KEYPAD

The position of the figure on the screen can be controlled by a KEYPAD consisting of three pairs of keys: S,X; D,C; and F,V. Typing the letter S results in the figure moving one 'unit' along the +x axis. Typing the letter X causes the figure to move one unit along the -x axis. The other letters in the KEYPAD (D C, F V) cause similar movement in the y and z directions, respectively. Also you can use the KEYPAD to move the cursor along different axes and different directions individually, when in the EXTEND mode.

ADDITION FACTOR

Moving the figure or the cursor around the screen one unit at a time can be very slow, so you can set an addition factor to help you move faster. Typing a number between 1 and 4 changes this factor. The figure or the cursor is moved 1+2*FACTOR units. For instance, if you type a 3, the figure or the cursor moves seven units every time you type a key in the KEYPAD. After you finish moving, the FACTOR is reset

to zero. When you start MAKER3D, the FACTOR is set to zero.

3.2 Procedures

You build figures by invoking procedures with single keystrokes. The procedures are listed in a menu at the bottom of the screen. Figure 18 is the first menu in MAKER3D.

```
G(ET,W(RITE,E(XTEND,A(NEW,D(IR,T(URNOFF
5(FILL,Z(AP,O(CNTR,C(OMBINE,H(ELP,Q(UIT
R(OTATE,L(SCALE,M(OVE,Y(RECOLOR,K(ILL
S(TANDARD,B(NORMALIZE,N(AME:
```

Figure 18. Main Menu for MAKER3D

GET

GET is invoked when you want to get a figure from disk, or to get a figure that is already in memory, as the working figure. After you type G, it asks you for the name of the figure to be loaded. You just type the name in and hit the RETURN key. It then checks the figures in memory to see if there is one with that name. If there is not, it tries to find a figure by that name on disk. If it finds the figure, MAKER3D loads it and makes it the current work figure. The previous work figure is left as it is, and cannot be changed until you get it again. If the figure is not found, an error message is displayed on the screen and no current work figure exists.

WRITE

When WRITE is invoked, MAKER3D asks you for the name of the figure to be saved. It then finds that figure in memory and writes it out onto disk. If there was a previous figure on disk with the same name, the new one replaces it. This does not change anything that you are working on.

ANEW

ANEW starts a new figure by asking for the name of the new figure, then invoking EXTEND on the new figure. If the name already exists, you will get a message and then you can destroy the figure and recreate a new figure by typing Y, or can leave ANEW without changing anything by typing N.

TURNOFF

When you have several figures in memory, it can be beneficial to turn off some of them while working on others. This procedure TURNOFF prevents the showing of the figure you name. A figure can be turned back on by GETing it.

DIR

Typing D will give you a directory of the figures currently in memory. It lists each figure's name and shows that figure for two seconds. This includes showing any figures that are turned off.

NAME

When NAME is invoked, the figure is displayed and you are prompted for the new name. All future references to the figure are through the new name. If you are not prompted for a name, it does not change anything. The name of the work figure is displayed after NAME:.

5FILL

To get a new background color, you use procedure 5FILL to fill the background. After typing 5, you are given a menu showing you the different colors and asking you to choose one. You can change the background color to orange, black, white, blue, green and violet.. Trying to change color to NONE will retain the old background color.

ZAP

When you have several figures on the screen, they can be confused with the work figure. It is beneficial to turn off all the figures except the work figure. When you type Z, only the work figure is displayed on the screen.

OCNTR

In three-dimensional rotation, you have to be careful with the center of the figure, or sometimes the figure will be displayed off the screen. When starting a new figure, the center is set at the center of the screen.

When OCNTR is invoked, you can move the center of the figure to the real center or any other point you want by using the KEYPAD. When you find the point you want to be the new center of the figure, type G to get it and end OCNTR.

COMBINE

This allows you to combine two figures by adding one to the end of another. It asks you for the name of the first figure, then for the second figure's. MAKER3D adds a line of color NONE from the end of the first figure to the start of the second figure. The combined figure retains the name of the first figure. The individual figures are no longer accessible. If you have a figure that you want to use several times, you should save it by transferring it onto disk, then get it off the disk after each COMBINE.

ROTATE

When ROTATE is invoked, MAKER3D asks you about which axis you want to rotate the figure. After you are prompted for the axis, you can use PADDLE(0) to set the angle to rotate. Typing another axis will let you change the axis of rotation while typing any other key ends the rotation.

LSCALE

When LSCALE is invoked, you can use PADDLE(1) to scale

the figure from as small as half to twice its size. Any keystroke ends the scaling.

MOVE

After you type M, you are able to translate (move) the figure. You can use KEYPAD to control the translation along the different axis and directions respectively. Any other keystroke ends the translation.

YRECOLOR

This command is invoked when you want to change the first color of the work figure. After typing Y, you are given a menu which shows you the different colors and asks you to choose one. YRECOLOR will go over color NONE and keep going to a different color. It does not affect any color after the first color. For instance, if you make a figure with the first five lines blue, then one NONE color line, then three lines blue, then use green lines for the rest of the figure, YRECOLOR will change the color of the first nine lines except the NONE color line.

STANDARD

After a lot of rotations, translations and scalings, it is easy to have an accumulative error which results in considerable figure distortion. You can type S to standardize the figure; this results in a return to the original figure before any rotations, translations, and scalings.

BNORMALIZE

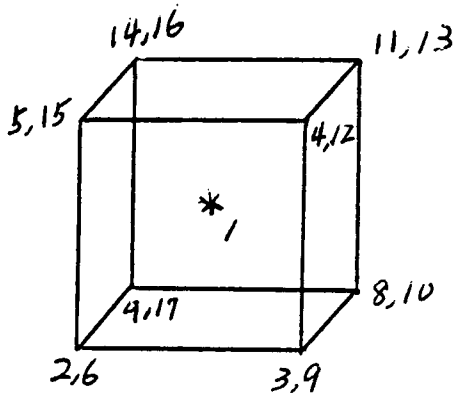
When BNORMALIZE is invoked, MAKER3D asks you to make sure that you want to take the figure shown on the screen as your original figure. You must type Y to confirm the action. Afterwards, when STANDARD is invoked, this figure becomes your original figure and shows on the screen.

QUIT

QUIT allows you to leave MAKER3D. It gives you a listing of the figures in memory and prints whether they have been saved since they were the work figure. Then you have two options: type R to return to MAKER3D or type E to exit it.

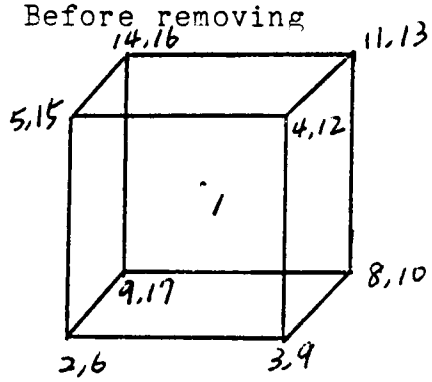
KILL

This command is invoked when you want to delete some lines on a figure. When you type k, the cursor flashes at the starting point and a small menu shown in Figure 19 (a) appears at the bottom of the screen. The numbers in the figure 19 are the data points of the figure in order. It asks you to delete from here by typing H, go to the next point by typing the SPACE key, or cancel the KILL mode by typing C. When you decide to delete from the point where the cursor flashes, you type H to delete from here, and the cursor goes to the next point and flashes. It asks



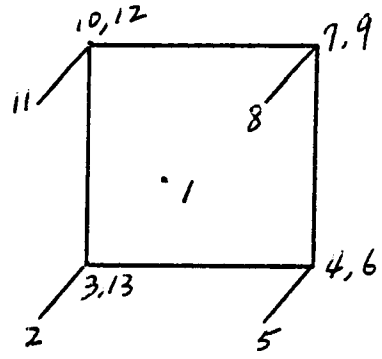
THE LINES OF THE WORK FIGURE
TO BE KILLED FROM:
<H>HERE <SPACE>GO AHEAD <C>CANCEL

(a) Before removing



THE LINES OF THE WORK FIGURE
TO BE KILLED TO:
<E>END <H>HERE <SPACE>GO AHEAD <C>CANCEL

(b) After choosing point 1 as starting point being removed



(c) After removing

Figure 19. Removing Lines from Data Point 1 to Point 6

you to delete to here or to the end, or go to the next point, or to cancel the request which is shown on Figure 19 (b). When you decide to delete to the point where the cursor flashes, you type H. KILL allows you to delete any line or any lines on a figure. Figure 19 (c) shows the result of removing lines from data point 1 to point 6.

HELP

HELP changes the screen to text mode and gives an explanation of each of the procedures available in the menu. There are two pages in this list, you can type <CTRL> A key to change the pages back and forth. When you have learned what you need, hitting the RETURN key ends the HELP procedure.

EXTEND

This procedure performs all work on the individual lines of a figure. It can be invoked on the current work figure by typing E. It has its own menu and submenu, which are shown in Figure 20 and Figure 21. Once entering EXTEND, all figures are removed from the screen except the work figure. A cursor flashes at the end of the last line of the figure. When you type a KEYPAD key, the cursor will move one unit in the direction you want, a line will be drawn from the end of the last line to the new cursor position. This

line is called the work line, and all changes you make occur on it until that line is added to the figure. When you get a work line, the submenu is shown at the bottom of the screen. The length of the work line is controlled by the position of the cursor. Before you add the line, you can rotate the line about three different axes or rotate the whole figure and the work line. When you decide to add the line, you may also choose a color for it. When starting a new figure, the first line starts at the center of the screen.

```
MOVE   X    Y    Z    <5>  FILLSCREEN
    +  <S>  <D>  <F>  <R>  REMOVE LINE
    -  <X>  <C>  <V>  <H>  HELP  <Q>  QUIT
```

Figure 20. Menu for EXTEND

```
MOVE   X    Y    Z    <W>  TO ROTATE LINE
    +  <S>  <D>  <F>  <R>  TO ROTATE FIG.
    -  <X>  <C>  <V>  <A>  TO ADD LINE
```

Figure 21. Sub-Menu for EXTEND

```
KEYPAD CONTROL  <S> <D> <F>
                <X> <C> <V>
```

When in the EXTEND procedure, the KEYPAD is used to set the new position of the cursor and get the corresponding work line.

<5>

Typing 5 is the same as the 5FILL in the first menu.

<R>

Typing R removes the last line of the figure and the cursor flashes at the new last line of the figure. If you only have one line in your figure, you cannot remove it.

<Q>

Typing Q ends EXTEND and returns you to the first menu.

<W>

When W is typed, MAKER3D asks you about which axis you want to rotate the work line. After you type one of the axes, you can use PADDLE(0) to rotate the figure and type another axis key to change the axis of rotation. Typing any other key ends the rotation.

<R>

Typing R in the submenu is almost the same as the ROTATE in the first menu. The only difference is the rotation here rotates the work figure and the work line.

<A>

When you decide you have the line the way you want, you type A to add the line to the figure with any color you want.

<H>

Typing H is similar to the HELP for the first menu, except that it contains information for the menu and submenu.

3.3 Quick Reference

A. Main Menu

PADDLE(0)	sets angle of work figure to rotate
PADDLE(1)	sets scale of work figure to scale
KEYPAD	moves work figure 1+2*FACTOR units
GET	gets an existing figure for the work figure
WRITE	writes a figure onto disk
ANEW	starts a new figure
TURNOFF	prevents the figure from being shown
DIR	lists the names and shows all figures in memory
NAME	allows you to rename the work figure
5FILL	fills the background with the chosen color
ZAP	turns off all figures except work figure
OCNTR	moves the center of work figure
COMBINE	adds a second figure to the end of the first one
ROTATE	rotates the work figure
LSCALE	scales the work figure
MOVE	translates (moves) the work figure
KILL	deletes selected lines on the work figure
YRECOLOR	recolors the first color of the

	work figure
STANDARD	standardizes the work figure
BNORMALIZE	normalizes the work figure
QUIT	ends MAKER3D
HELP	prints two pages of information (menu)
EXTEND	lets you extend (work on) a figure

B. EXTEND Menu

KEYPAD	moves cursor 1+2*FACTOR units
<5>	fills the background with a color
<R>	removes the last line of work figure
<Q>	ends EXTEND
<H>	prints a page of information (menu)

C. EXTEND Sub-Menu

PADDLE(0)	sets angle of work figure or work line to rotate
KEYPAD	moves cursor 1+2*FACTOR units
<W>	rotates the work line
<R>	rotates the work figure with the work line
<A>	adds the work line with a color

CHAPTER IV

IMPLEMENTATION

4.1 Basic Data Structures

This chapter assumes that the reader is familiar with the Pascal language. MAKER3D uses three basic data structures to implement its graphics package. The data structure in Figure 22 is a head node for a figure. A figure head node is the record which holds information that describles the entire figure. It is linked into lists called ONLIST and INLIST which control the status of figures in memory.

```
FIGHEAD = PACKED RECORD
   TXSTART:INTEGER;  (* STARTING POINT IN
   TYSTART:INTEGER;  (TXSTART,TYSTART,TZSTART)
   TZSTART:INTEGER;     WITH COLOR
   STARTMODE:SCREENCOLOR;   STARTMODE  *)
   TX      :INTEGER;  (* STARTING SHOW POINT IN
   TY      :INTEGER;  (TX,TY,TZ)        *)
   TZ      :INTEGER;
   XCENTER:INTEGER;   (* CENTER OF THE FIGURE IN
   YCENTER:INTEGER;   (XCENTER,YCENTER,ZCENTER)
   ZCENTER:INTEGER;                     *)
   MRX     :AR4;      (* 4 X 4 MATRIX   *)
   ORD     :BYTE;     (* POSITION IN ONLIST *)
   INORDER:BYTE;      (* POSITION IN INLIST *)
   FIG     :RGNSEGPTR; (* PTR TO START OF BODY *)
   FIGEND :RGNSEGPTR; (* PTR TO END OF BODY *)
   INPTR  :FIGPTR;    (* PTR TO NEXT IN INLIST *)
   FPTR   :FIGPTR;    (* PTR TO NEXT IN ONLIST *)
END;
```

Figure 22. FIGURE Head Node

The values TXSTART, TYSTART and TZSTART form a three-dimensional starting point. The first line of the figure is set by drawing to this point with color STARTMODE which is set to be NONE. There is another point formed by TX, TY and TZ, called the starting show point. So for each figure, there are two sets of three-dimensional data points. The first set is called original data while the second set is called show data. Only the show data will be transformed by converting these data to a two-dimensional data set and showing it on the screen. Chapter II tells you how to get show data from original data and how to get two-dimensional data from show data.

The variables XCENTER, YCENTER, ZCENTER form a center of the figure. When starting a new figure, the center is set to the starting point of the figure. It can be changed later on. MRX is a 4 X 4 matrix which is used to get a new set of show data from the original data when a rotation, translation, or scaling has been processed. In the beginning, the MRX is set to the identity matrix.

The figure's position in the onlist is provided by the integer ORD field. The onlist is linked through the FPTR with lower ORDs being shown first. Only those figures that are turned on are linked into the onlist. The inlist keeps track of all figures that are in memory and arranges them by their INORDER value. All references to figures are made by their INORDER value. The inlist is linked through the

INPTR. The onlist is actually a subset of the inlist, but it does not have to follow the same order as the inlist. The FIG points to the start of the body of the figure while the FIGEND points to the end of the figure. This will be discussed in more detail later. MAKER3D has another head node, shown in Figure 23.

```
INLST = RECORD
   NAME   : STR12;  (* DISK NAME OF FIGURE *)
   HEAD   : FIGPTR; (* PTR TO FIGHEAD *)
   HDNMBR: INTEGER; (* SAME AS INORDER *)
   SAVED  : BOOLEAN; (* STATUS OF FIGURE *)
   FPTR   : LISTPTR; (* PTR FOR THIS LIST *)
END;
```

Figure 23. MAKER3D Head Node

Each figure has a disk file name. The HEAD is used to know which name is attached to which figure. Once a new list is set up, it is easy to add more fields to ease figure manipulation. The HNDMBR, the same as INORDER value, is set to save tracing through lists to find the value. The SAVED field is set to true when a figure is put on disk, and set to false when the figure is loaded by procedure GET or just created. The list is linked through the FPTR.

The MAKER3D node list is traversed when MAKER3D is looking for a figure, causing minimum use of the figure head node. When the MAKER3D node is found for the work figure, the NAME field becomes global. All changes are straight forward and occur when the user handles the figure except procedure COMBINE that combines two figures.

Basically, combining two figures involves cutting the body of one figure and appending it to the body of another figures, as shown in Figure 24. In the following figures, cut links are shown by ---x---, and added links are shown by ------. Small changes must be made to the figure body in order to accomplish this properly. A line segment of color NONE, with the data which is from the FIGHEAD of the second figure, is added between the two figures to join them in their relative positions. Then the head of the second figure is cut. One more thing that must be done is changing the original data of the second figure by multiplying the show data of the second figure by the inverse matrix of the first figure.

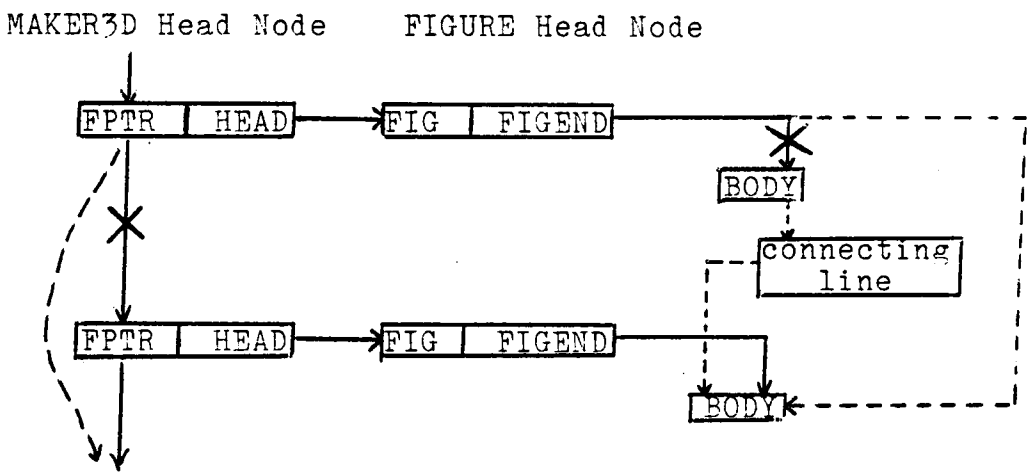


Figure 24. Combining Two Figures

Another data structure is the body of a figure which is composed of line segments shown in Figure 25. The body is a

doubly linked list which gives greater convenience in adding or removing line segments.

The RGNX, RGNY and RGNZ represent a three-dimensional point of the original data set. The X, Y and Z form a three-dimensional point of the show data set. The X, Y and Z hold the values of RGNX, RGNY and RGNZ before any rotations, translations or scalings. These two data points can be derived from each other by a matrix multiplication.

```
RGNLINESEG = RECORD
   RGNX    : INTEGER; (* ORIGINAL DATA POINT IN
   RGNY    : INTEGER; (RGNX,RGNY,RGNZ)
   RGNZ    : INTEGER;                  *)
   RGNMODE: SCREENCOLOR; (* COLOR FOR LINE *)
   X       : INTEGER; (* SHOW DATA POINT IN
   Y       : INTEGER; (X,Y,Z)
   Z       : INTEGER;                  *)
   BPTR    : RGNSEGPTR; (* ADDR OF PREVIOUS LINE *)
   FPTR    : RGNSEGPTR; (* ADDR OF NEXT LINE *)
END;
```

Figure 25. Line Segment Node

To get X, Y and Z, we multiply the RGNX, RGNY and RGNZ by the matrix of the FIGHEAD of the figure while we can get RGNX, RGNY and RGNZ by multiplying the X, Y and Z by the inverse matrix of the figure. Since X, Y and Z form a show data point, the steps for adding a line are as follows:

1. Get the point (X,Y,Z) you want.
2. Choose the color RGNMODE for the line.
3. Get the point (RGNX,RGNY,RGNZ) as original data from (X,Y,Z).

And the steps for showing a figure are as follows:

1. Get all show data points from original data points.
2. Convert the show data points to two-dimensional data points.
3. Draw one line by MOVETO to the next point with color RGNMODE and continue drawing the figure line by line.
4. When the line color has value RADAR this means end of the figure.

   (RADAR is an unused color type of TURTLEGRAPHICS)

Line manipulation is done by working on one line at a time, and gives the user the ability to work on any line in the figure, adding and removing lines as desired. The user controls the position of the cursor which is a three-dimensional point shown on the screen. The work line is stretched from the previous line to the cursor. The work line is referred to by a line segment called RGNNOWSEG. When a line is added, the current RGNNOWSEG is put into the figure and a new RGNNOWSEG is put in front of the previous RGNNOWSEG. The BPTR of RGNNOWSEG points to RGNPREVSEG, the FPTR of RGNNOWSEG points to the end of the figure. Figure 26 shows the linkage.

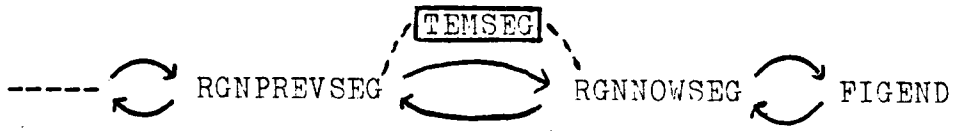


Figure 26. Linkage of RGNNOWSEG

There is a set of data TEMSEG, including the temporary show data point and original data point. It is used to generate the working line and change the size and the direction of the line until the user makes a decision about it. If the user decides to add this line, all the information in TEMSEG will be put into RGNNOWSEG and the FPTR of RGNNOWSEG is changed to point to a new RGNNOWSEG, and initialize the new RGNNOWSEG.

Removing a line involves unlinking RGNNOWSEG and relinking it in the new location. Line removal is done by moving the links to RGNNOWSEG back one line segment. This is shown in Figure 27. The only exception to this occurs when the first line is removed. In that case the links are moved forward one line segment.

Figure 27. Removing a Line Segment

Removing several lines at one time involves unlinking the previous line of the first line being removed and

relinking it to the line after the last line being removed. This is shown in Figure 28. For intance, if you want to remove the line from SEG 2 to SEG n-1, MAKER3D doubly links the SEG 1 and SEG n and frees SEG 2 to SEG n-1.

On exiting from this level of work, RGNNOWSEG is unlinked from the figure. The global RGNPREVSEG will hold the location of the line for RGNNOWSEG to link in front of when this level is entered again.

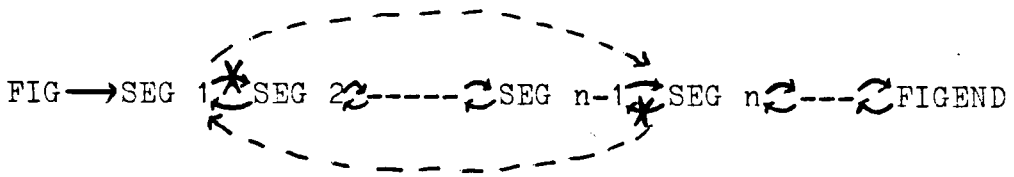


Figure 28. Removing Several Lines

4.2 Input / Output

Once a figure is built up, the user can write (save) this figure onto disk. If the user then wants to work on this figure, it can be read (loaded) from disk. Both of these things are done via a IOBUFFER which is shown in Figure 29. The size of the IOBUFFER is a block (512 bytes) in order to read or write one disk block a time. It is read and written using UCSD commands BLOCKREAD and BLOCKWRITE. IOBUFFER has two different data structures; the second one is used for a figure size larger than one block.

When the user wants to write a figure onto disk, all the information in FIGHEAD needs to be put into IOBUFFER except the show data which can be gotten from the original data. We put the starting point into XS, YS and ZS, the starting mode into MODES, the center point into XC, YC and ZC, the matrix into MATRX. We fill with blanks for FILLER to make IOBUFFER a full block size. Then we count the lines FIGRCOUNT and put the data point into XYZ, color into COLR. If FIGRCOUNT is less than or equal to 53, then one block is enough to save all the information for the figure and the field MORE is set to zero. Otherwise, we must use a second data structure to save the rest of the lines.

```
IOBUFFER = RECORD
    CASE INTEGER OF
    1 : (MORE   : INTEGER; (* NAME OF BLOCKS LEFT *)
         FIGRCOUNT: INTEGER; (* NUMBER OF LINES *)
         XS     : INTEGER; (* STARTING POINT
         YS     : INTEGER; (XS,YS,XS)
         ZS     : INTEGER;                    *)
         MODES  : SCREENCOLOR; (* STARTING COLOR *)
         XC     : INTEGER; (* CENTER OF A FIGURE *)
         YC     : INTEGER; (XC,YC,ZC)
         ZC     : INTEGER;                      *)
         MATRX  :AR4;  (* MATRIX OF A FIGURE   *)
         FILLER: STR6;  (* FILL BLANKS *)
         FIGR   : ARRAY[1..53] OF RECORD
                    XYZ : AR3; (* ARRAY OF ONE DATA *)
                    COLR: SCREENCOLOR; (* COLOR OF LINE *)
                  END;
    2 : (COUNT  : INTEGER; (* NMBR OF LINES *)
         FILLERA: STR6;    (* FILL BLANKS   *)
         FIGRD  : ARRAY[1..63] OF RECORD
                    MXYZ : AR3; (* ARRAY OF ONE DATA *)
                    MCOLR: SCREENCOLOR; (* COLOR *)
                  END;
```

Figure 29. Input / Output Buffer

Getting a figure from disk is easily done by putting the information into IOBUFFER from disk, then getting the figure from IOBUFFER. Also, the user needs to check MORE to see if more than one block was used for a figure on disk. If more, the user has to use another BLOCKREAD to accomplish getting a figure.

CHAPTER V

SPEED OF MAKER3D

The drawback of MAKER3D is speed. For all the rotations, translations, and scalings, MAKER3D requires matrix multiplication. Currently, all multiplication and division is done by software. That makes MAKER3D a little slow on responding to the user's request. A translation or a scaling of a cube which consists of sixteen data points requiring one multiplication of a 4 x 4 matrix by a 4 x 4 matrix and seventeen multiplications of a 1 x 4 matrix by a 4 x 4 matrix. After typing a key for translation of the cube, the user needs to wait for almost three seconds to see the image after the translation on the screen. Of the almost three seconds, more than two seconds are used for the matrix multiplications. The rest are spent on filling the background of the screen, projecting the cube and showing it on the screen. For a rotation of the cube, this delay is even slower, since it needs to calculate three multiplications of a 4 x 4 matrix by a 4 x 4 matrix and seventeen multiplications of a 1 x 4 matrix by a 4 x 4 matrix. It takes almost five seconds to get an image after the rotation and four seconds is used for matrix multiplications. This results in MAKER3D being less interactive than desirable. One multiplication of a 4 x 4 matrix by a 4 x 4 matrix needs to calculate 64 floating

point multiplications and 48 floating point additions. One floating point multiplication, using software only, takes 6 milli-seconds while one floating point addition takes 4 milli-seconds. Thus we can compute that a software matrix multiplication takes about 0.6 second.

A hardware floating point multiplication or addition only takes about 0.1 milli-second, using a commonly available hardware multiplication chip. So a hardware multiplication can make MAKER3D much faster by a factor of approximately 60. Fortunately, it has been ordered courtesy of Dr. Ashley and will solve the speed problem of MAKER3D.

However, there is another way to solve the speed problem using software by decreasing the calculation for multiplications and additions. For a 4 x 4 matrix, not all the elements of the matrix are needed to be multiplied or added, since some values of the elements are zero. We can check the value of the element before a multiplication or addition. If the value of the element is zero, we ignore it to save one multiplication and one addition. This will make the speed of MAKER3D somewhat faster.

CHAPTER VI

CONCLUSION AND FUTURE STUDIES

There exists a two-dimensional interactive graphics package, MAKER, which is used as a teaching tool for high school students for the creation of video games and cartoons. However, the two-dimensional graphics does not meet all requirements in applications. For instance, when biologists try to use it to observe chromosomes in different views on the screen, it can not provide the necessary realism. An extention of MAKER, MAKER3D, was created for a high degree of realism in the display of objects and scenes.

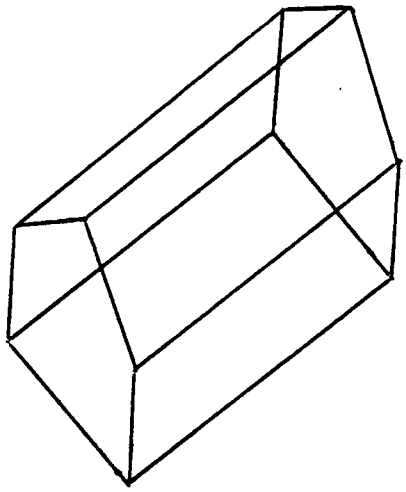
MAKER3D is an interactive three-dimensional graphics package for the APPLE II machine. It can display a figure being worked on, showing changes as they are made. All the data points of a figure are in three-dimensions. They are projected on two-dimensional screen coordinate data points and shown on the screen. The size, orientation and position can be changed. Figures can be combined to make a new figure. The location of the center of a figure can be changed. Figures can be normalized and standardized. The shape of a figure can be changed by adding, removing, or otherwise adjusting individual lines. All control of figures is simplified by the two paddles and by single keystrokes. A two-dimensional Tablet has been ordered and we plan to extend MAKER3D to use the Tablet as a three-dimensional input device.

The effectiveness of interactive computer graphics is measured by the speed with which the user of the computer can assimilate the displayed information. A serious flaw in MAKER3D is its slowness. However, A commonly available hardware multiplication chip has been ordered which can make MAKER3D much faster.

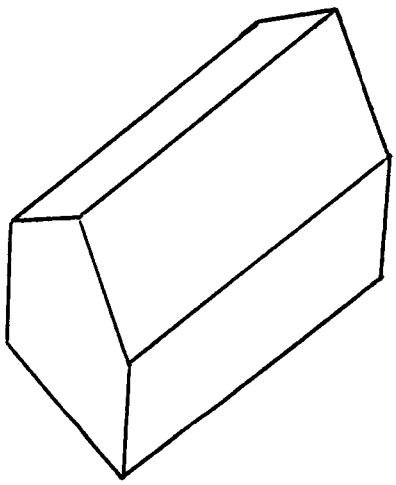
Several features that have not been implemented but that are being considered are:

1. Hidden surfaces

   Hidden line elimination is a very important feature of three-dimensional objects and scenes. Lines which are blocked by other surfaces in space are clipped or eliminated. Figure 30 shows two polyhedron figures where drawn before and after hidden line elimination. The problem with hidden line elimination is computation time. A number of hidden line algorithms exist [Sutherland, 1974]. These algorithms either compare every line against every surface or use nondeterministic methods to look for conflicts and remove them [Artwick, 1979]. Both methods are very time consuming. Another problem on MAKER3D for hidden line elimination is memory. We are currently exploring means of eliminating the hidden surfaces.



(a) A perspective view on a polyhedron



(b) After hidden line elimination

Figure 30. Hidden Line Elimination on a Polyhedron

2. Defining built-in figures

We can apply MAKER3D to a biological problem by observing chromosomes in different views on the screen. Since each chromosome occupys only a small portion of the nucleus, we have to build some circles to represent the surface of a sphere and use the sphere to represent a nucleus. All the chromosomes are inside the sphere and can be observed in different views.

3. Shadow control

Shadowing is another advanced graphics technique. Surfaces at different angles have different color shades when projected due to light angle and viewing angle. Shading adds a very realistic effect to three-dimensional pictures when used with hidden surface elimination. Shadowing is not an easy task and is not appropriate for all applications. Thus shadowing is still being considered.

Expansion of this package is planned to make it more powerful and easier to use.

BIBLIOGRAPHY

BIBLIOGRAPHY

Amann, G. W. "MAKER: Figure Creator for Rascal Animation System," M.S. Thesis, University of Tennessee, Knoxville, 1980.

Artwick, B. A. "Three-Dimensional Microcomputer Graphics, 6502/APPLE II Assembly Language," Sublogic Company, Illinois, 1979.

Giloi, W. K. Interactive Computer Graphics, Prentice-Hall, Inc., 1978.

Lee, Kaiman. Interactive Computer Graphics in Architecture, Environmental Design & Research Center, Boston, 1976.

Mckenzie, J.; L. Elton.; R. Lewis. Interactive Computer Graphics in Science Teaching, Ellis Horwood Limited, N.Y., 1978.

Newman, W. M and R. F. Sproull. Principles of Interactive Computer Graphics, Mcgraw-Hill Book Company, N.Y., 1979.

Sutherland, I. E.; R. F. Sproull; R. A. Schumacker. "A Characterization of Ten Hidden-Surface Algorithms" Computing Surveys, Volume 6, Number 1, March, 1974.

VITA

Chi-Lin Tom was born in Taiwan, Republic of China on February 8, 1953. He lived in Taichung, Taiwan from his birth till graduating from Taichung First High School in June, 1971. He then entered the Tamkang University, Taiwan in September 1971, and graduated from there in June, 1976, receiving a Bachelor's degree in Computer Science.

After graduation, he was employed by Tang-Lin Computer Company as a programmer for a short period. Then he returned to his alma mater as a teaching assistant in Computer Science until June, 1979.

After that, he entered The University of Tennessee, Knoxville in September, 1979 to begin work on the Master of Science in Computer Science. He will receive that degree upon the completion of this thesis, and will be take a job to get some working experience. Then he will go back to Taiwan to contribute his learning and knowledge to his country.