

Microcomputer 3D Graphics

Load & Go Version

Apple II

(Applesoft II ROM-16K)

THREE DIMENSIONAL MICROCOMPUTER GRAPHICS

APPLE II LOAD AND GO VERSION

(Applesoft II ROM - 16K)

Adapted by Jim Harter

Personal Computers Inc.
1610 S. Neil Street
Champaign, IL 61820

BASIC Program by Bruce Artwick

Sublogic

Load and Go Manual by Stu Moment

Sublogic

March 1979

Sublogic Company

Box V

Savoy, Illinois 61874

First Edition
First Printing

© Sublogic Co. 1979
All Rights Reserved

Table of Contents

Section	Title	Page
	Introduction	1
A	Running the Program	2
B	3D Familiarization	3
C	Creating and Entering Data Bases	8
D	Moving to High Resolution Page 2	13
E	View Finder Program	15
F	Written Listing	17
G	Notes	22

The 60 page BASIC 3D Graphics manual is referenced within these instructions by the notation "(ref p)" where p refers to the page number.

Package Contents

Load & Go 3D Manual

60 page BASIC 3D Graphics Manual

Applesoft II ROM cassette

side 1	Load & Go 3D Graphics
side 2	Demonstration

This manual is designed for your quick familiarization with 3D graphics.

3D graphic subroutines will be of major use in your educational, scientific, or pure graphic applications.

The Apple listing in Section F is similar to the Basic 3D manual listing except for commands that control the Apple's graphics. The core commands appear one statement per line and use simple variable notations. There are no trig functions used.

This BASIC version will give you a powerful understanding of 3D graphic theory and uses. This package is meant for your development of data bases and control programs over many years time. You may use all of the core statements in most BASICs used on computers ranging from a Level II TRS-80 to an IBM 370. You will also be able to use your data bases in Sublogics fast assembly language packages.

Section A Running the Program

Two copies of the program are provided on each side of the cassette. The first copy is located just after the leader. The second copy is located approximately 2 minutes (at normal playing speed) from the beginning of the tape.

OPTIONAL Type HIMEM:8192 if you would like to protect the graphic page. If you do not protect the graphic page then programs with large data bases will overwrite the page. If you set high memory then you will receive an out of memory error instead. At Sublogic, we prefer not to set HIMEM.

Load the program as usual. Most loading problems are caused by using a volume level which is too loud. Another cause of loading problems is tape quality just after the leader. If you suspect this problem then use the second copy.

Type RUN and Return. While the title page appears, the data base of line endpoints is transferred from data statements to the Q,R, and S arrays (ref 27). The data base provided is a cube and will be used for your familiarization.

Section B 3D Familiarization

The object that you will be viewing is a 50 X 50 X 50 unit cube located on the upper right side of the X,Y origin, 100 units back on the Z axis. Figure 1 shows a diagram of this cube according to the convention used in the BASIC manual (ref 6).

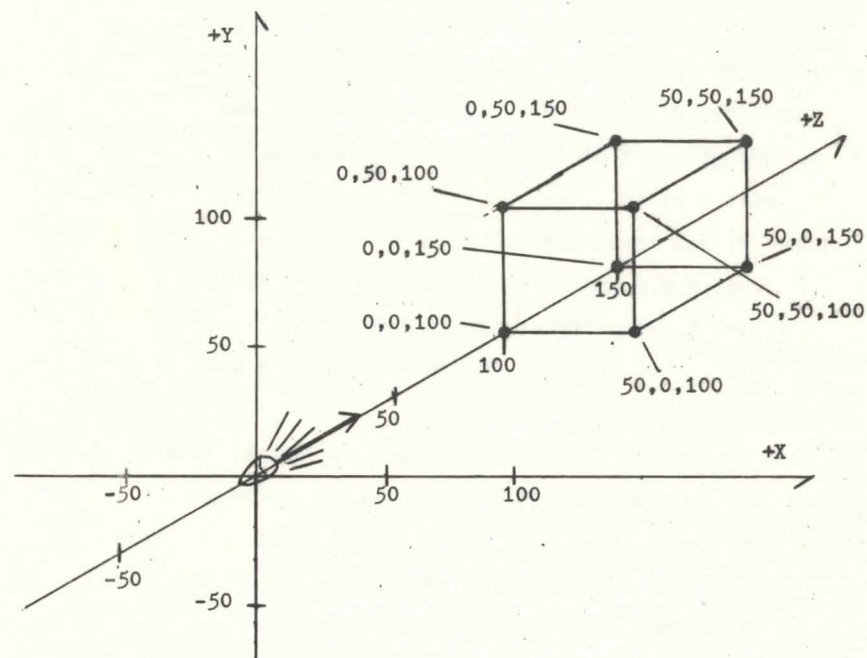


Figure 1
Sample Data Base - Cube

The first input requested is Field of View.

This parameter allows you to select wide angle or telephoto views (ref 9 and 52).

To begin with, type 1 and Return.

The program now asks you for object's location (X,Y,Z). The BASIC program manual's references to "viewer's location" should read "object's location". The BASIC program assumes that you will always be located at X,Y,Z=0,0,0. You will be moving the whole object with the objects location prompt. You will move it to a new relative position determined by you input (ref 12).

Type 0,0,0 and Return. The object is now in the location shown in Figure 1.

Do not use the View Finder yet. Type N and Return.

Your next prompt is for (P,B,H) or pitch, bank, and heading (ref 8). Your input will be standard degrees where 360 degrees completes a circle.

When you use a positive number for pitch, the projected images move up in the screen as if you looked down.

When you use a positive number for bank, you are rotating the object (relative to itself) to the right.

When you use a positive number for heading, you are looking to your right. A 0 degree heading is the same as looking straight down the Z axis in a positive direction.

Referring to Figure 1, the lower left corners of the cube are located at a P,B,H of 0,0,0.

Type 0,0,0 and Return.

To center the view of the closest face of the cube, move it down and to the left while holding pitch, bank, and heading constant. Press Return for new parameters. Enter 1 to change the view and 1 to erase the last view.

X,Y,Z -25,-25,0

P,B,H 0,0,0

To move the object closer to you, move it in a negative direction along the Z axis.

X,Y,Z -25,-25,-50

P,B,H 0,0,0

Now take time out to adjust the screen proportions. Measure the height and width of the closest face of the cube. If the height and width are not the same, then break into edit mode. LIST 8017 and change the value of H9. H9 will usually be within the range of 0.5 to 1. Run the program and enter the same parameters as last used.

Repeat modifying H9 until the closest face of the cube is square. SAVE your custom program.

Rotate the object relative to itself by changing bank.

X,Y,Z -25,-25,-50

P,B,H 0,30,0

Move the object image up by pitching down.

X,Y,Z -25,-25,-50

P,B,H 20,0,0

Note the automatic clipping of lines that run off the screen.

Move the object to the left side of the screen by looking more to the right (increase heading).

X,Y,Z -25,-25,-50

P,B,H 0,0,20

Move the actual object over to the right by increasing its X location and increase heading so that you will be able to see the object.

X,Y,Z 25,-25,-50

P,B,H 0,0,33

Move the actual object up by increasing its Y location and increase pitch so that you will be able to see the object.

X,Y,Z 25,35,-50

P,B,H -34,0,33

Experience with different entries will give you the ability to judge the P,B,H necessary to view the object. An object locating algorithm called the View Finder is discussed in Section E of this manual. Take it easy and use the View Finder as you place the cube in new locations during your familiarization.

DEMONSTRATION Program After you become familiar with the program construction, you may wish to customize the program to your own needs. The **DEMONSTRATION** is provided on the second side of the cassette. The location and viewing parameters are internal in the 100000 BASIC block and are requested in line 8040.

Section C Creating and Entering Data Bases

There are many ways to code images. If you do not have a method presently, then we suggest starting with a rough sketch of the object on an X,Y,Z oriented drawing as shown in Figure 2. Do not label line endpoints yet.

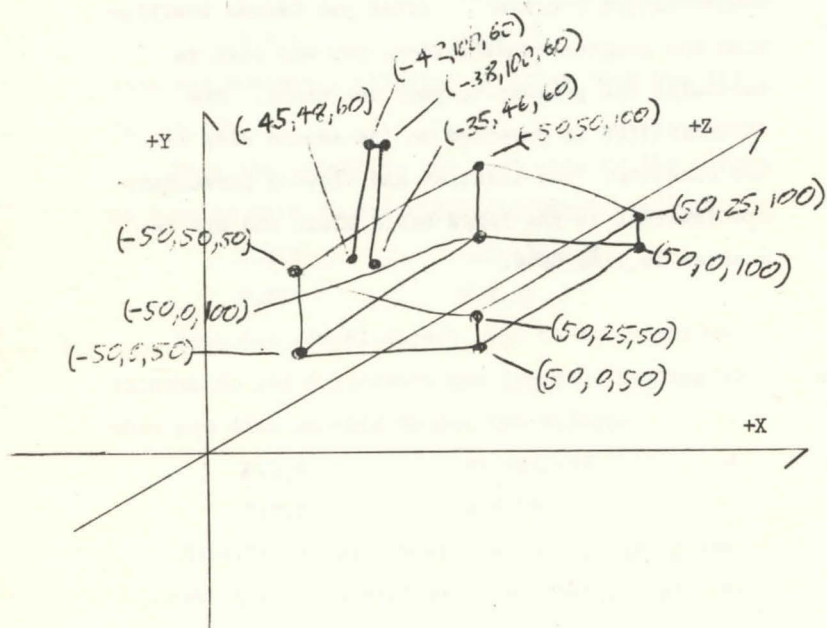


Figure 2
Factory

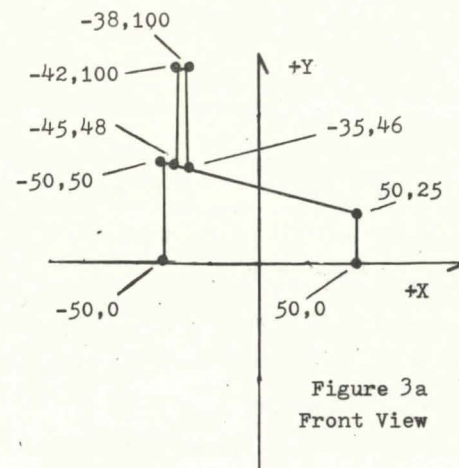


Figure 3a
Front View

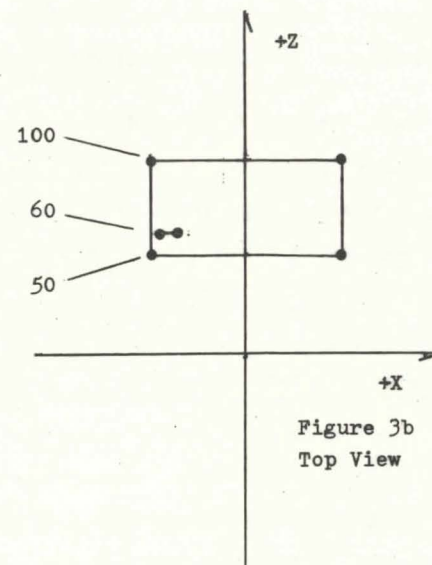


Figure 3b
Top View

Figure 3
Two Views of a Factory

Now draw 2D drawings of your object from a couple of different orientations as shown in Figure 3. Use graph paper if desired.

The goal of the 2D drawings is to find all X,Y,Z coordinates of endpoints in order to label your rough drawing endpoints.

For simple scenes, 2 views will be enough. The coordinates for each endpoint in Figure 2 were obtained by adding Z values from the top view to the X,Y coordinates found in the front view.

The data base must now be organized. A column of line start and endpoints will be easily transferred to the program. Group line endpoints in pairs until all the desired lines are coded.

2000 DATA 00		factory base
2010 DATA -50,0,50	start point	line 1
2011 DATA 50,0,50	end point	
2012 DATA 50,0,50		2
2013 DATA 50,0,100		
ect. 50,0,100		3
-50,0,100		
-50,0,100		4
-50,0,50		
-50,50,50		roof
50,25,50		5
50,25,50		
50,25,100		6

50,25,100	7
-50,50,100	
-50,50,100	8
-50,50,50	
	roof supports
50,0,50	9
50,25,50	
50,0,100	10
50,25,100	
-50,0,100	11
-50,50,100	
-50,0,50	12
-50,50,50	
	smoke stack
-45,48,60	13
-42,100,60	
-42,100,60	14
-38,100,60	
-38,100,60	15
-35,46,60	

Note that the smoke stack is really a 2D object. You may enter 2D objects for display. They will be shown as a line when viewed from the side. 2D objects can dress up images quickly.

You may enter points by giving the same start and end point. This pseudo pure point method gives nice presentations of relative positions of planets or molecular structures.

The first data statement at line 2000 must give the total number of start and endpoints. Remember to update this when adding to existing data bases.

There are quicker ways to input data. Through yearly catalog mailings we will keep you informed on products such as digitizers that meet industry standards.

The data base for a pyramid is given in the BASIC manual (ref 27). To code it, continue entering data to the following theme:

2000 DATA 12

2010 DATA 100,0,1100

2011 DATA 500,0,1100

2012 DATA 500,0,1100

ect.

Your data bases may use any number in the range of floating point BASIC.

BASIC blocks 2000-6000 are reserved for data bases.

Section D Moving to High Resolution Page 2

A move to page 2 will give you extra room for data bases and control programs. If you set high memory before running programs, use HIMEM: 16384.

The following statements must be changed if you move to page 2.

change line#

55 from: HGR:TEXT
to: HGR2:TEXT

8046 from: POKE -16297,0
to: POKE -16299,0

8047 from: POKE -16304,0:POKE -16302,0
to: POKE -16304,0

-Line 55 clears page 2.

-Line 8046 sets page 2 for display.

-Line 8047 displays the set page.

Out of interest, you may want to display an object on page 1, then change lines 55, 8046, and 8047 to the page 2 commands. Display an object on page 2, Return, and break (Control C). Manually POKE -16304,0 to view page 1.

To view page 2, enter TEXT (while blind). Manually POKE -16299,0:POKE -16304,0. Page 2

should now be displayed again. Using the given POKEs you should be able to modify the program to display the desired page at the desired time.

There are some restrictions to using page 2.

- Memory requirements are 24K minimum.
- No program such as an Applesoft language or disk operating system may occupy page 2.

If you use a Disk II to store the program, you may have to LOAD the program, disable the DOS with a Reset, and recover the program with a ⌘G in monitor mode. Page 2 references will now be acceptable.

Section E View Finder

The view finder assumes that you are located at the origin. Just pitch and heading are calculated. Bank is independent of pitch and heading.

The listing provided in this section is general purpose. It uses a trig function included in BASICS used in many microcomputers.

In your program, line 7014 sets the values of XD,YD and ZD at the center of the cube (25,25,125). This is the point that the view finder aims for. If you would like to set the center manually during the program, then use line 7014 from this listing. Be sure to enter a new center for a new creation.

If you would like to know the exact pitch and heading to the center, then use line 7300 given in this section. The values returned to your program are turned into integers in line 7300 for clean presentation.

View Finder by Stu Moment

View Finder © Copyright 1979 Sublogic

```
7000 PRINT "VIEW FINDER (Y/N)"
7005 INPUT F$
7008 IF F$="N" THEN RETURN
7011 IF E$="B" THEN GOTO 7040
```

```

7014 INPUT "CENTER OF OBJECT? (X,Y,Z)";XD,YD,ZD
7019 E$="B"
7040 XC=XD+X(3)
7041 YC=YD+Y(3)
7042 ZC=ZD+Z(3)
7045 P3=0:P4=0:P5=0
7048 IF XC<0 THEN GOSUB 7060
7050 IF YC<0 THEN GOSUB 7070
7052 IF ZC<0 THEN 7080
7054 GOTO 7100
7060 P3=1:XC=-XC:RETURN
7070 P4=1:YC=-YC:RETURN
7080 P5=1:ZC=-ZC
7100 IF ZC=0 THEN ZC=.001
7105 HE=ATN(XC/ZC)*57.29578
7110 IF P3=0 AND P5=0 GOTO 7200
7120 IF P3=1 AND P5=0 THEN HE=360-HE
7130 IF P3=1 AND P5=1 THEN HE=HE+180
7140 IF P3=0 AND P5=1 THEN HE=180-HE
7200 WC=((XCA2)+(ZCA2))^(1/2)
7202 IF WC=0 THEN WC=.001
7205 PI=ATN(YC/WC)*57.29578
7210 IF P4=1 THEN 7300
7220 PI=-PI
7300 PRINT "P= ";PI,"H= ";HE
7310 RETURN

```

To patch this routine to the program, add:

```

8013 E$="A"
8040 GOSUB 7000

```

Section F Listing

```

LIST
10 REM SUB-LOGIC 3D GRAPHICS
20 REM LOAD AND GO APPLE II VER
   SION
30 REM JIM HARTER, PERSONAL COMP
   UTERS
40 REM INC., CHAMPAIGN, ILL.
50 REM COPYRIGHT 1978 SUB-LOGIC
   INC.
55 HGR : TEXT
60 HOME : REM CLEAR SCREEN
70 VTAB (4)
75 FOR Z = 1 TO 40: PRINT "*"; : NEXT
   Z
80 VTAB (6): PRINT TAB( 10); "SU
   BLOGIC 3D GRAPHICS"
90 VTAB (8): FOR Z = 1 TO 40: PRINT
   "*"; : NEXT Z
95 VTAB (10): PRINT TAB( 8); "CO
   PYRIGHT 1979 SUBLOGIC"
100 VTAB (15): PRINT "APPLE II V
   ERSION"
110 VTAB (18): PRINT TAB( 10); "
   BY JIM HARTER"
115 VTAB (20): PRINT TAB( 10); "
   PERSONAL COMPUTERS INC. "
117 PRINT TAB( 10); "1610 S. NEIL
   "
118 PRINT TAB( 10); "CHAMPAIGN, I
   LL"
120 FOR Z = 0 TO 2500: NEXT Z
1000 REM
1010 READ L9
1015 DIM Q(L9),R(L9),S(L9)
1020 FOR I = 1 TO L9
1030 READ Q(I),R(I),S(I)
1040 NEXT I
1060 GOTO 8000
2000 DATA 24
2010 DATA 0,0,100
2011 DATA 50,0,100
2012 DATA 50,0,100
2013 DATA 50,50,100
2014 DATA 50,50,100
2015 DATA 0,50,100
2016 DATA 0,50,100

```

```

2017 DATA 0.0,100
2018 DATA 0.0,150
2019 DATA 50.0,150
2020 DATA 50.0,150
2021 DATA 50.50,150
2022 DATA 50.50,150
2023 DATA 0.50,150
2024 DATA 0.50,150
2025 DATA 0.0,150
2026 DATA 0.0,100
2027 DATA 0.0,150
2028 DATA 0.50,100
2029 DATA 0.50,150
2030 DATA 50.50,100
2031 DATA 50.50,150
2032 DATA 50.0,100
2033 DATA 50.0,150
8000 DIM X(5),Y(5),Z(5),C(8)
8005 HOME:PRINT "AFTER PLOT IS
      COMPLETE.HIT RETURN":PRINT
      "TO CHANGE VIEW DATA OR EXIT
      PROGRAM"
8010 PRINT:PRINT "THE APPLE WI
      LL BEEP WHEN PLOT IS OVER"
8012 PRINT
8015 W1 = 139
8016 W2 = 91
8017 H9 = .795
8020 CALL 62450
8025 INPUT "FIELD OF VIEW";V
8030 HCOLOR= 3
8035 INPUT "OBJECTS LOCATION(X,Y
      Z)";X(3),Y(3),Z(3)
8045 INPUT "DIRECTION OF VIEW-P,
      B,H";P,B,H
8046 POKE - 16297,0
8047 POKE - 16304,0:POKE - 16
      302,0
8050 GOSUB 8200
8055 FOR A1 = 1 TO L9 STEP 2
8060 X(1) = O(A1)
8063 Y(1) = R(A1)
8066 Z(1) = S(A1)
8069 X(5) = O(A1 + 1)
8072 Y(5) = R(A1 + 1)
8075 Z(5) = S(A1 + 1)
8078 GOSUB 8500
8080 GOSUB 9000
8098 NEXT A1
8099 PRINT CHR$(7):INPUT Z$:TEXT

```

```

8100 INPUT "ENTER 1 TO CHANGE VI
      EW,2 TO END PROGRAM":CHOICE:
      IF CHOICE < > 1 AND CHOICE
      < > 2 THEN 8100
8110 IF CHOICE = 2 THEN END
8120 PRINT "ENTER 1 TO ERASE LAS
      T VIEW,2 ELSE":INPUT CHOICE
      IF CHOICE < > 1 AND CHOICE
      < > 2 THEN 8120
8130 IF CHOICE = 1 THEN CALL 62
      450
8140 GOTO 8035
8200 F = P
8203 GOSUB 8300
8204 R1 = N
8206 F = P
8209 GOSUB 8310
8212 R2 = N
8215 F = B
8218 GOSUB 8300
8221 R3 = N
8222 F = B
8224 GOSUB 8310
8227 R4 = N
8230 F = H
8233 GOSUB 8300
8236 R5 = N
8237 F = H
8239 GOSUB 8310
8245 T1 = (N * R4 + R5 * R1 * R3)
      * H9
8248 I2 = (1 - N) * R3 + R5 * R1 *
      R4
8251 T3 = R5 * R2 * V
8253 T4 = R2 * R3 * H9
8256 T5 = R2 * R4
8259 T6 = (1 - R1) * V
8262 T7 = ((1 - R5) * R4 + N * R1 *
      R3) * H9
8265 T8 = R5 * R3 + N * R1 * R4
8269 T9 = R2 * N * V
8272 RETURN
8300 F = F - 90
8305 IF F < = (1 - 180) THEN F =
      F + 360
8310 IF F < 0 THEN F = - F
8315 S = F
8320 IF S > = 90 THEN F = 180 -
      F
8330 N = F * .01745329
8340 A = N * N
8350 M = A * A
8360 N = 1 - A / 2 + M / 24 - A *
      M / 720 + M * M / 40320
8370 IF S > = 90 THEN N = (1 - N)
      )
8380 RETURN

```

```

8500 FOR A = 1 TO 5 STEP 4
8510 G = X(A) + X(3)
8520 S = Y(A) + Y(3)
8530 K = Z(A) + Z(3)
8540 X(A) = G * T1 + S * T4 + K *
      T7
8550 Y(A) = G * T2 + S * T5 + K *
      T8
8560 Z(A) = G * T3 + S * T6 + K *
      T9
8570 NEXT A
8600 FOR A = 1 TO 5 STEP 4
8610 C(A) = 0
8612 C(A + 1) = 0
8614 C(A + 2) = 0
8616 C(A + 3) = 0
8618 IF X(A) < ( - Z(A)) THEN C(
      A) = 1
8620 IF X(A) > Z(A) THEN C(A + 1
      ) = 1
8622 IF Y(A) < ( - Z(A)) THEN C(
      A + 2) = 1
8624 IF Y(A) > Z(A) THEN C(A + 3
      ) = 1
8626 NEXT A
8632 FOR A = 1 TO 4 STEP 1
8634 IF C(A) = 0 THEN 8638
8636 IF C(A) = C(A + 4) THEN 866
      8
8638 NEXT A
8644 FOR A = 1 TO 4
8646 IF C(A) = 1 THEN 8676
8648 NEXT A
8654 FOR A = 5 TO 8
8656 IF C(A) = 1 THEN 8686
8658 NEXT A
8662 P2 = 1
8664 GOTO 8800
8668 P2 = 0
8670 RETURN
8676 A = 1
8678 S = 5
8680 GOTO 8694
8686 A = 5
8688 S = 1
8694 IF C(A) = 1 THEN 8728
8696 IF C(A + 1) = 1 THEN 8714
8698 IF C(A + 2) = 1 THEN 8742
8700 IF C(A + 3) = 1 THEN 8756
8706 GOTO 8662
8714 K = (Z(A) - X(A)) / (X(S) -
      X(A) - Z(S) + Z(A))
8716 X(A) = K * (Z(S) - Z(A)) + Z
      (A)

```

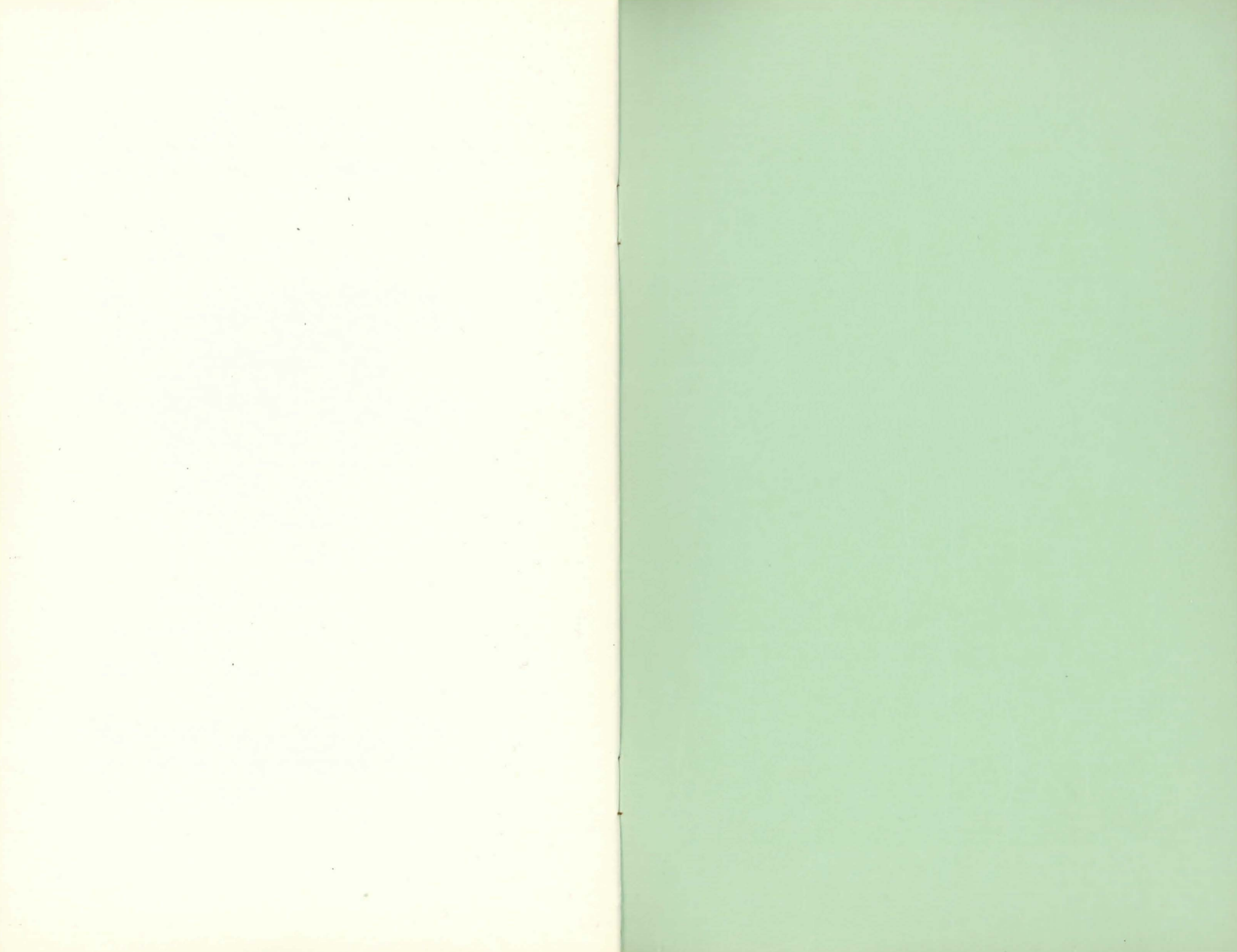
```

8718 Y(A) = K * (Y(S) - Y(A)) + Y
      (A)
8720 Z(A) = X(A)
8722 GOTO 8600
8728 K = (Z(A) + X(A)) / (X(A) -
      X(S) - Z(S) + Z(A))
8730 X(A) = K * (Z(A) - Z(S)) - Z
      (A)
8732 Y(A) = K * (Y(S) - Y(A)) + Y
      (A)
8734 Z(A) = ( - X(A))
8736 GOTO 8600
8742 K = (Z(A) + Y(A)) / (Y(A) -
      Y(S) - Z(S) + Z(A))
8744 X(A) = K * (X(S) - X(A)) + X
      (A)
8746 Y(A) = K * (Z(A) - Z(S)) - Z
      (A)
8748 Z(A) = ( - Y(A))
8750 GOTO 8600
8756 K = (Z(A) - Y(A)) / (Y(S) -
      Y(A) - Z(S) + Z(A))
8758 X(A) = K * (X(S) - X(A)) + X
      (A)
8760 Y(A) = K * (Z(S) - Z(A)) + Z
      (A)
8762 Z(A) = Y(A)
8764 GOTO 8600
8800 IF Z(1) = 0 THEN Z(1) = .00
      1
8845 IF Z(5) = 0 THEN Z(5) = .00
      1
8855 X(2) = X(1) / Z(1) * W1
8860 Y(2) = Y(1) / Z(1) * W2
8865 X(4) = X(5) / Z(5) * W1
8870 Y(4) = Y(5) / Z(5) * W2
8875 RETURN
9000 IF P2 = 0 THEN RETURN
9010 X(2) = X(2) + 140
9020 Y(2) = 91 - Y(2)
9030 X(4) = X(4) + 140
9040 Y(4) = 91 - Y(4)
9045 REM SAVE PREVIOUS VALUES H
      ERE FOR SCREEN ERASE
9050 H$PLOT X(2),Y(2) TO X(4),Y(4
      )
9060 RETURN

```

Section G Notes

1. Before storing the Demonstration on disk,
delete BASIC line #10.



sub**LOGIC**
Box V, Savoy, IL 61874

(217) 367-0299