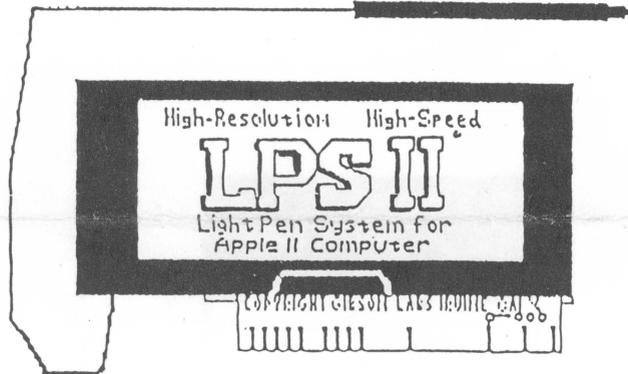


GIBSON LABORATORIES
PENTRAK GRAPHICS JOURNAL



Written by:

Dana Smith & Denise Strogatz

We are happy to present the first issue of the
Gibson Labs Pentrak Graphics Journal

It has taken a while, but here we are. Filled to the brim with all sorts of new stuff from Gibson Laboratories and other sources. Before anything else is said in this Journal, a quick experiment. If this Journal is your friends' and not yours, and you are wondering why your Journal has not arrived. You better check to see that Gibson Laboratories has your registration card. If we don't, send it in to us immediately or call.

Now that we know that everyone has their own copy, we're all set to start. There's been a great deal of interest in the development of a users group for Pen owners. As a forum for Pen discussion, this newsletter will disseminate any and all thoughts, discoveries and news from any users and Pen-compatible software developers. If you want to communicate to the over 4,000 owners of the LPS II system, write to us - we'll print it in the next issue of the "Journal".

The first issue discusses many routines that you can add to present software, append preliminary software, or use as a stand alone module for special purposes. Let's begin with some PENTRAK "one-liners". Most of these one-liners are useful. Some of them, if not useful, are at least amusing. All are fun!!

```
10 FOR Y = 0 TO 160 STEP 20: FOR X = 0 TO 240 STEP 20:  
    &ZOOM(X,Y): NEXT X : NEXT Y  
  
10 X=INT(240 * RND(X)):Y=INT(192 *RND(X)):& ZOOM(X,Y):GOTO 10  
  
10 &P(X,Y,KEY,ZV0):X1= 280-X: Y1=192-Y: &ZOOM(X1,Y1): GOTO 10
```

This next one is not exactly a one-liner! But we think you'll get a big kick out of it! (And it may open your eyes to the power of PENTRAK).

This routine may be used as a stand alone module, or can be inserted as a subroutine into any of your new PENTRAK application programs. Following the listing is a detailed description of each of the statements that make up the "FLOATER" subroutine.

```
-----  
1 REM ..... FLOATER ..... PROGRAM WILL FLOAT AN IMAGE OVER  
AN EXISTING IMAGE. IT MAY THEN BE SET DOWN OVER ANY THING  
10 HGR : &INIT, DHP, NEG, MIXED: HCOLOR= 0  
20     &TRACK(XX,YY,KEY,LINES,NC)  
30     IF KEY = 0 THEN 90  
40     &TRACK(XX,YY,KEY,SKETCH3,NC)  
50     GOTO 20  
90 X=XX: Y=YY: &TRACK(XX,YY,KEY,RUBRECT,NC)  
100 IF KEY <>21 THEN 20  
110 DX=ABS( XX-X)  
120 DY=ABS( YY-Y) + 1  
130 &FRAME(X TO XX AT Y FOR DY, 16384)  
140 &DHS,NEG,DHP: GOTO 180  
150 &PEN(X,Y,KEY,ZV0): IF KEY <> 255 THEN 20  
160 IF X=XL AND Y= YL THEN 150  
170 GOSUB 210  
180 XL=X: IF XL + DX > 279 THEN XL=279-DX  
190 YL=Y: IF YL - DY < 0 THEN YL=DY  
200 GOSUB 210: GOTO 150  
210 &FRAME(X,16384, XL TO XL+DX AT YL-DY FOR DY):RETURN  
-----
```

Program explanation:

Line 10 above sets up the screen and various initial information, selects the primary Hi-Res screen, clears and negates that screen and sets the screen for both mixed text and graphics. Line 20 starts the Pen tracking, there will be crosshairs and no cursor since we are on a white screen. The KEY escape mode is used and the TRACKING is terminated when a key or button is pressed. Line 30 checks the Applesoft variable KEY, If the key pressed was the left arrow '<', which is ASCII code 8, then control jumps to line 90 to begin the frame procedures. If the key is anything other than the left arrow, program control falls sequentially to line 40 where sketching in black takes place until another key is pressed. This next key hit toggles the sketching off, another key hit of anything but left arrow will again toggle the sketching back on.

Line 90 is the first instruction for setting up the frame. The initial X and Y coordinates are saved in the variables X and Y. After these values have been saved for future use, TRACKING resumes. This time a rubber-banding rectangle will show the user where the frame is to be set up. The outline of the rectangle is the frame boundary. In line 100 a check is made to see whether the key is a right arrow. If it is not, then the procedure will restart from the beginning. If the key is a right arrow (ASCII code 21) then the program will calculate the frame. Lines 110 and 120 calculate the difference between the first X and Y and the second X and Y. The second X and Y is that point where the Pen is now being pointed. Line 130 saves the current frame starting at the first location of the Hi-Res secondary page. In line 140 the second page is negated. This is done so the frame will be seen on the first page in opposite colors. It will be easier to see the relation between the frame and the page of graphics this way.

Starting at line 150 with the Pen returning X and Y every sixtieth of a second, if the escape is not caused by a key hit, you will see the frame floating over the current picture. Line 160 checks to see if the pen has moved since the last update, if it has, it will float the object. If not then the object stays right there. Line 170 is a GOSUB to the subroutine which actually places the image plane (frame) onto the Hi-Res screen. Lines 180 and 190 check to make sure that the frame is always staying inside the hires screen area. And finally, line 200 removes the frame with the GOSUB, then goes back for the next pen hit.

It is a good idea to understand this simple concept. You will find incredible flexibility and power in the use of this routine.

Now we'll look at some routines that complete gaps in the preliminary software, and others that compliment all of the software.

Since all the programs and routines thus far have been written with PENTRAK 1.1, we will continue to support both PENTRAK 1.1 and the preliminary software that you now have.

The following information is for the PRELIMINARY SOFTWARE ONLY.

In this part of the Journal we will address trouble users have had with the standard software, and solutions in the form of simple routines that will eliminate these problems. One of the biggest problems encountered by some novice users is the ability to save and load pictures from the disk. This is a problem that can be fixed very simply. First of all, in the preliminary software most of the pictures reside in page one of the Apple Hires screen. This is an area of memory that is between locations \$2000 and \$3FFF. An easy way to save any picture that resides in this location is to get out of the program you are in by hitting your <RESET> key. This will stop the execution of the program, yet it will not erase the program or the pictures in memory. After you hit the reset key, you will be looking at the text page. While looking at the text page you can type in:

```
BSAVE (filename),A$2000,L$1FF8
```

where (filename) is the name you give to the picture. This will save the area of memory starting at location \$2000 with a length (L) of \$1FF8 on a diskette. We use this length to save an extra sector on the disk. Once your picture has been saved you can just type run to restart the program. (Note that restarting the program will erase the picture in memory.)

If your picture happens to be on page 2, like the BOXES program, PENPAINTER, and GEOMED II, then you must change the \$2000 to \$4000. This will save the memory area starting at \$4000 with the same length of \$1FF8.

NOTE: In the GRID DRAW program, I recommend you type the following command prior to saving the picture. This will eliminate the grid dots.

&DOTS

If you wish to incorporate the above capabilities for loading and saving in the preliminary software, you may type the routines right into existing software.

The first thing we have to determine is what keyboard key will trigger loading and saving. We could do it by having one key for load and another key for save. In this example we will use one key to set up a screen that will prompt the user for either a load or a save operation. Also in this routine we will use the key board for data input. In our next example, we will use the light pen for data input. After we have determined whether we want to load or save, we must find the name of the file and then the disk drive that file resides on. A flow chart for this routine would look something like this:

Here is the routine:

```

100 REM THIS ROUTINE IS FOR PRELIMINARY SOFTWARE ONLY
110 REM -----
120 REM IT IS NOT NECESSARY IN ANY OF THE REVISION 2.0 SOFTWARE
130 REM SET UP VARIABLES
140 FL$ = "": REM SET FILENAME TO NULL STRING
150 PS = PEEK (43626):PD = PEEK (43624): REM GET INITIAL SLOT & DRIVE
160 LS$ = "    LOAD AND SAVE PICTURE ROUTINE    "
170 REM SET UP SCREEN FOR INPUT
180 & TP: HOME : PRINT LS$
190 PRINT "-----"
200 VTAB 8:MS$ = "LOAD OR SAVE": GOSUB 480: REM CENTER THE STRING
210 VTAB 20:MS$ = "PLEASE MAKE A SELECTION 'L' OR 'S'": GOSUB 480
220 VTAB 8: HTAB 38: GET A$: PRINT A$:
230 IF A$ = "S" THEN 420
240 IF A$ = "L" THEN 360
250 PRINT CHR$ (7): GOTO 220
260 REM GET NAME OF FILE IF NO NAME THEN RETURN TO MAIN PROGRAM
270 HOME : PRINT LS$
280 VTAB 8:MS$ = "PLEASE ENTER THE FILE NAME": GOSUB 480
290 VTAB 12:MS$ = ".....": GOSUB 480
300 VTAB 12: HTAB HTX: INPUT FL$
310 IF LEN (FL$) < 1 THEN 550: REM GO BACK
320 VTAB 14:MS$ = "DRIVE NUMBER -> ": GOSUB 480
330 VTAB 14: HTAB HTX + LEN (MS$) - 1: INPUT DR$:
    IF LEN (DR$) < 1 THEN DR$ = STR$ (PD): RETURN
340 IF DR$ = "1" OR DR$ = "2" THEN RETURN
350 PRINT CHR$ (7): GOTO 330
360 REM LOAD FILE FROM DISK
370 GOSUB 260: REM GET FILE NAME AND DRIVE
380 ONERR GOTO 510
390 PS$ = STR$ (PS)
400 PRINT :PRINT CHR$ (13): CHR$ (4): "LOAD": FL$: ",A$2000,S": PS$: ",D": DR$
410 RETURN
420 REM SAVE FILE FROM DISK
430 GOSUB 260: REM GET FILE NAME AND DRIVE
440 ONERR GOTO 510
450 PS$ = STR$ (PS)
460 PRINT :PRINT CHR$ (13): CHR$ (4): "SAVE ": FL$: ",A$2000,L$1FF8,S":
    PS$: ",D": DR$
470 RETURN
480 REM AREA FOR CENTERING STRINGS
490 HTX = INT (20 - LEN (MS$) / 2): HTAB HTX: PRINT MS$
500 RETURN
510 REM ERROR HANDLING
520 POKE 216,0: REM CLEAR ERROR FLAG
530 ER = PEEK (222): HOME : PRINT "ERROR NUMBER ";ER;" @ LINE ";
    PEEK (218) + PEEK (219) * 256
540 VTAB 22: INPUT "PRESS RETURN TO CONTINUE": X$
550 RETURN

```



Now that we have determined the flow of information, we can see how the routine will work. First we must insert a line into the main program. The line should read:

IF KEY = 1 THEN GOSUB 100:REM GO TO THE LOAD & SAVE ROUTINE
Using KEY = 1, the load and save routine will be called when the user presses the CTRL-A key.

This routine can be added to all your current Applesoft programs, except ANIMATOR, PENPAINTER, and PATTERN EDITOR. These programs already have the necessary disk file features. You will find a more complete example of this concept, called the FILER, in many of the new programs that were sent with this newsletter.

Now let's look at the problem of generating hard copy (paper) printouts. The following routines will allow the user to print the picture from the program. Here is a small routine for those of you who own GRAPPLER, MICROBUFFER, OR SPIES printer interface cards. There are also other cards that use this same protocol. In general, if your interface card contains built-in screen-dump software, and if it is upward compatible with one of the cards just mentioned, it will work here. Otherwise, look at the manual that came with your printer card for the correct protocol. If you want, you can insert this routine into every one of your preliminary programs. It is a small, one line routine that can be typed into all existing programs:

```
(linenum) IF KEY = 16 THEN PRINT CHR$(4);"PR#1":PRINT
CHR$(9);"G1":PRINT CHR$(4);"PR#0"
```

If the picture is on page two then a '2' should appear after the G1. ex:

```
(linenum) IF KEY = 16 THEN PRINT CHR$(4);"PR#1":PRINT
CHR$(9);"G12":PRINT CHR$(4);"PR#0"
```

The same concept can be used with the PKASO interface using this line:

```
(line ~) IF KEY = 16 THEN PRINT CHR$(4);"PR#1":PRINT
CHR$(9);"32H":PRINT CHR$(4);"PR#0"
```

and again if the picture is on page two then the number in the quotes must be changed. In this case it is incremented by one.

```
(linenum) IF KEY = 16 THEN PRINT CHR$(4);"PR#1":PRINT
CHR$(9);"33H":PRINT CHR$(4);"PR#0"
```

where linenum = the line number you want this statement to be executed at. The KEY value 16 in this case, is the ascii value of CTRL-P. You may use any key you want. I like CTRL-P because it is a fairly standard statement for those who are users of CP/M and APPLEWRITER // and ///. The "G1" and "32" part of the statement can also be changed to any valid GRAPPLER or PKASO command. Please be sure to specify what page you are dealing with in this phrase; G1 for page 1, and G12 for page 2 of Apple Hires, or 32H for page 1 or 33H for page 2 with the PKASO card.

The correct location to put this command is at the beginning of the KEY testing section. The KEY section usually follows an &TRACK command and is used for a TRACKING escape.

Here is a routine that can be very helpful in any graphics program.

```
1010 REM THIS PROGRAM CAN BE USED TO ERASE IN EITHER BLACK OR
      WHITE BY CHANGING THE 'B' OR 'W' IN THE FRAME COMMAND
      IN LINE 1170
1020 REM YOU CAN CHANGE THE SIZE OF THE ERASER BY CHANGING THE
1030 REM VALUES LOCATED IN THE FRAME COMMAND
1040 REM BUT REMEMBER TO CHANGE THE BOUNDARY VALUES
1100 R=5:REM RADIUS OF FRAME
1110 &T(X,Y,KEY,ZV0): REM INITIALIZE PEN TRACKING
1115 REM THE FOLLOWING LINES CHECK AND CORRECT FRAME BOUNDARIES
1120 IF X+R>279 THEN X=279-R :REM RIGHT BOUNDARY FIX
1130 IF X-R<1 THEN X=R :REM LEFT BOUNDARY FIX
1140 IF Y>187 THEN Y=192-R :REM BOTTOM BOUNDARY FIX
1150 IF Y-R < 1 THEN Y=R :REM TOP BOUNDARY FIX
1160 IF KEY = 27 THEN RETURN
1165 REM THIS WILL RETURN CONTROL TO THE CALLING PROGRAM
1170 &F(W,X-R TO X+R AT Y-R FOR R*2)
1180 REM FRAMES WHITE 'R' PIXELS TOP, BOTTOM, RIGHT, AND
      LEFT OF THE PEN POINT
1190 GOTO 1110
```

On January 19, 1983 Apple Computer Inc. blessed some of us with the Apple //e. (Thanks Guys!!) It took the place of the Apple II+. Since the 64k Extended 80 column card combined with some neat circuitry gave us the ability to display 80 columns, some people went a step farther and took advantage of the double hires that could be achieved. This led to some articles in one of our favorite magazines that showed us how to use that extra memory and all those colors. One of those articles laid it all on the line with a bunch of assembly language routines that we could use to set color and plot points and lines on the double hires page. The article name is APPLE SOFT BRUSHES FOR DOUBLE HI-RES ART. It was published in the September 83 issue of SOFTALK MAGAZINE. The routines in the article were linked together with PENTRAK 2.0 to provide partial PENTRAK support to Apple//e double hires graphics!!! The following program is a very simple example of the neat features to be found using the double hires capability. The program requires the following:

an Apple//e w/ extended 80 column card (the one w/64k extra memory), and a REV B board (one w/part number 820-0064-B on the motherboard identification plate).

a jumper installed on the eighty column card connecting the two molex connectors.

and the routine on page 96 of the September issue of SOFTALK magazine.

These routines must be typed in and then BSAVED on disk under the name 'COLOR DBL HIRES'.

The COLOR DBL HIRES routine also uses the ampersand (&) vector, but we are not going to tie this routine into that vector. Instead, using PENTRAK 2.0's command &ER, we will link the routine into PENTRAK, and let PENTRAK send the command to COLOR DBL HIRES when appropriate. (Only if PENTRAK cannot understand the command.)

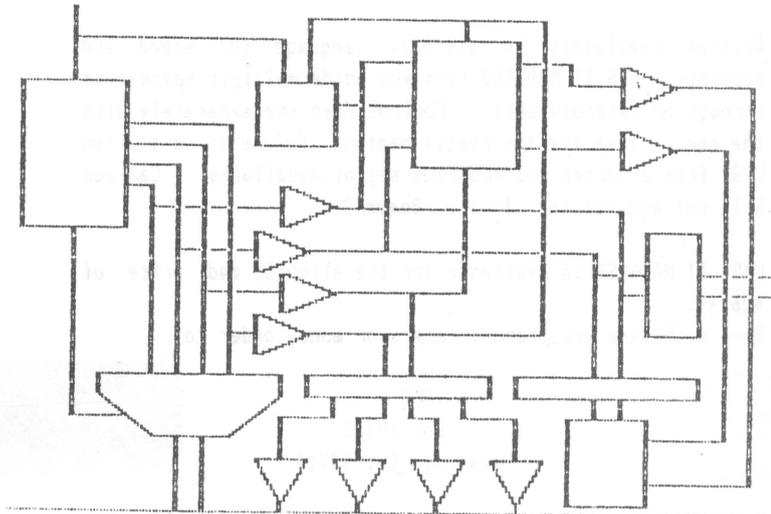
By the way, once the routine is typed in and on your disk you may use it without PENTRAK by simply BRUNING it. This will allow COLOR DBL HIRES to link into the AMPERSAND hook and thus parse its own commands. Note this will disconnect PENTRAK.

NOTE: This program will not work with PENTRAK 1.1 since it does not have an &ER command.

```

100 DIM CS$(16),XZ(16),YZ(16):YZ(0) = 3:YZ(1) = 3
110 FOR I = 0 TO 15 STEP 2: READ CS$(I):XZ(I)=4:READ CS$(I+1):
  XZ(I + 1) = 24: NEXT
120 FOR I = 2 TO 15 STEP 2:YZ(I) = YZ(I - 2) + 2:YZ(I + 1) =
  YZ(I): NEXT
130 PRINT CHR$(4);"BLOAD COLOR DBL HIRES,A$6000,D1,S4"
140 POKE 24696,255: REM ALLOW CLEAR TO WHITE
150 & AMPER24592: REM LINK COLOR DBL HIRES TO PENTRAK
160 & H
170 & P(X,Y,KEY,ZV0): IF KEY = 255 AND DR THEN & LX / 2,Y:
  GOTO 170
180 IF KEY = 49 OR KEY = 8 THEN DR = 1: & PX / 2,Y: GOTO 170
190 IF KEY = 50 OR KEY = 21 THEN DR = 0: GOTO 170
200 IF KEY = 67 THEN GOSUB 230: PRINT CHR$(18): & HP:
  GOTO 170
210 IF KEY = 69 THEN & H:DR = 0
220 GOTO 170
230 REM CHANGE COLOR
240 TEXT : PRINT CHR$(17): HOME
250 PRINT "PLEASE POINT TO THE COLOR OF YOUR CHOICE": PRINT
260 FOR I = 0 TO 15 STEP 2: GOSUB 350: PRINT CS$(I);:
  HTAB 20: GOSUB 350: PRINT CS$(I + 1): PRINT
270 NEXT :DR = 0
280 VTAB 22:MS$ = "USE PEN TO SELECT": GOSUB 360
290 & P(X,Y,ZV20):X1 = INT (X / 7):Y1 = 1 + INT (Y / 8)
300 FOR I = 0 TO 15 STEP 2: IF Y1 = YZ(I) OR Y1 - 1 = YZ(I)
  THEN CL = I: GOTO 320
310 NEXT I
320 IF X1 > 4 THEN CL = CL + 1: & CCL: RETURN
330 & CCL: RETURN
340 END
350 INVERSE : PRINT " "; NORMAL : PRINT " "; RETURN
360 HTAB (20 - LEN (MS$) / 2): PRINT MS$: RETURN
370 DATA BLACK,MAGENTA,DARK BLUE,VIOLET,DARK GREEN,GREY1,
  MEDIUM BLUE,LIGHT BLUE,BROWN,ORANGE,GREY2,PINK,
  GREEN,YELLOW,AQUA,WHITE

```



Since the Gibson Light Pen System was introduced over a year ago, there have been many inquiries about the availability of software that uses the GLPS. Here is a list of some of the packages that we have seen.

NOW AVAILABLE, THE FIRST HIRES GAME FOR THE GIBSON LIGHTPEN

LPS II MADNESS

It was inevitable that someone would start playing with the Gibson Light Pen System. Especially when it took over a year for the final software to become available! LPS II MADNESS is the first result of some of that playing. It makes complete use of the pen's unique ability to select random locations on the screen.

Written completely in assembly language for speed and accuracy, LPS II MADNESS lets you guide multiple spaceships through an asteroid belt. Control each one separately with the pen. Then the fun really begins. Defend space station LPS2 from a varied and numerous set of assailants. Can you hold out against the 'Inviso' Bomber?

LPS II MADNESS is available for the slightly mad price of \$26.44.

Send cash (are you crazy?), check or money order to:

BEZ

Box 19633

Irvine, Ca. 92714

LPS II MADNESS requires:
Apple II + w/48k
16 Sector Apple disk drive
Gibson Light Pen System

Like all other Gibson Light Pen System products, this game will not work on an Apple Monitor III, AMDEK 300, or a USI P13 monitor.

HIRES VERSATILE CALCULATOR

One of the most creative pieces of Apple software is the HIRES VERSATILE CALCULATOR. This program turns your Apple computer into an RPN scientific calculator that also operates in four bases. Key in numbers and functions from the Apple keyboard or the Gibson Light Pen System. Useful tool for programmers, engineers and educators. Outstanding display including visible 4-register stack with oversized numbers. Looks just like a calculator on your monitor or TV screen. Contains more than 50 functions. Operates in decimal, hex, octal, and binary. Features trig, logs, statistics, logical operations, ASCII tables, ten internal storage registers and more. Includes 114 page User's manual with 50 page Tutorial, internal Text Reference Guide, and 10 minute Demo Program.

The HIRES VERSATILE CALCULATOR is available from Tackaberry software for only \$59.95 plus 2.00 shipping (Florida residents please add 5% sales tax).

Tackaberry Software

Box 2857-G

Ormond Beach, FL 32075

(904) 677-2054

THE GRAPHIC SOLUTION

Animation, mixed text and graphics, three dimensional perspectives, and color-coding are just some of the features of THE GRAPHIC SOLUTION. Accent Software has linked together their GRAPHIC SOLUTION software with the Gibson Light Pen System, creating one of the finest graphics/animation packages available for the Apple. Users can develop educational material, business presentations, and much more by using a frame by frame graphic sequencer that allows you create and animate a video sequence. This is a highly professional system, designed with detail in mind.

More information regarding The Graphic Solution may be obtained from :

Accent Software

3750 Wright Place

Palo Alto, CA 94306

(415) 856-6505
