Edited by J. Peter Hoddie

# The December Meeting

The December meeting of the Boston Computer Society's TI-99/4A User Group will take place on Wednesday December 21at 7:30 PM at the Massachusetts College of Art on Huntington Avenue in Boston. In the event of bad weather call me at 375-6003 or the office at 367-8080 for news of cancelation. Otherwise assume there will be a meeting. It is only a ten minute walk from my dorm room anyway.

The meeting topic is somewhat up in the air. I hope and expect we will be able to show Press, the new word processor by Charles Earl, but it is not yet out so nothing is for sure. We will see DISKASSEMBLER Version 2 for the 9640 in some detail, as well as some other 9640 programs. Suggestions for other topics is welcome. I expect there may also be a segment on communicating with other computers: or in otherwords data transfer

As usual, the December meeting falls during my exam period, so if you'd like to do a demo, please make this the month to come forward. This newsletter is moreru shed than usual due to my recent return from Chicago (less than 48 hours ago I was racing towards O'Hare in a rental car...).

Get your Christmas shopping done early this year, so you won't have an excuse to miss the December meeting. Not that this warning usually helps.

And yes, the software library will be at the next meeting as usual. And so wil the new catalogs we had printed for Chicago (we were also running out). Actually I think the new ones are in short supply already!

# poT-pourrl by Mike Wright

First, a correction to last month's column on page 1. The following should have appeared as the first six paragraphs of the main article:

September's supplement should have got you get you past the technical problem of plotting a point on the screen in bitmap mode. A natural next step would be to implement algorithms to draw lines and circles.

There is a fundamental problem with drawing straight lines on a bitmap screen — it cannot really be done unless the line is horizontal, vertical or at 45 degrees (assuming square pixels). You can confirm this next time you are taking a shower by trying to draw straight lines from one tile to another, by considering each tile as a pixel.

You will soon find that you will have to resort to the "jaggies" — jagged line segments which combine to represent the best approximation to a straight line. This is the technique that is used in such programs as TI Artist. It is also easily seen in the LINES demo program.

Many people have looked at the "jaggy" problem. The most popular algorithm for drawing such a line is called Bresenham's algorithm. It converts the equation of a line — generally given as y=mx+c — into an iterative process of comparing test values over the horizontal distance subtended by the line. The values are incremented and tested repeatedly.

Each time a new value of x or y increases to make a new pixel possible, the pixel is plotted. The test increments are derived, before the line is drawn, from the slope of the line and the differentials of its horizontal and vertical components.

Now this is pretty heady stuff unless you are geometrically inclined. Also Bresenham's algorithm in assembly code may be a bit too much to swallow at one sitting. Instead, as a guide to understanding the algorithm, we will state it in pseudocode, and then develop it in Xbasic.

The Boston Computer Society
TI-99/4A User Group
One Center Plaza
Boston, MA 02108

Pseudocode is a generalized way ...

## The CPUG exposition

Justin Dowling, co-director of the BCS TI-99/4A user group, and I attended the 1988 TI Exposition, held at the Carlisle fairgrounds, PA on October 16. The exposition was organized by the Central PA TI Users Group (CPUG).

We took an Amtrak train from Route 128 station, just south of Boston, to Philadelphia, PA. There we changed to another Amtrak train and headed west to Harrisburg, PA.

The journey took about 8 hours. When compared with taking a plane, and allowing for early airport arrival, change of planes, and baggage retrieval, we probably could have saved 3 hours. However, the comfort of the wide train seats and the abundant legroom far outweighs the cramped confines of a 727 or DC-9. The luxury of being able to get up and walk around at will, the convenience of the cafe car, and the knowledge that your baggage is tucked safely overhead and not on the way to Tel Aviv make this a viable alternative to the cattle processing mentality of the airlines.

Justin and I were met at the Harrisburg train station by Barry Long, secretary/treasurer of the CPUG. He drove us to our hotel in his nearly-new 88 T-bird and we exchanged TI stories, rumors and information along the way.

At the hotel we tracked down the BCS disk library that Justin had mailed to the hotel. We then met with Wayne Stith, author of Triad, which was discussed last month (p8).

The following morning found us at the fairgrounds at 6.30 am, since the exposition opened officially at 7:00 am. The show is a combined effort attended by the local TI groups, Atari and Commodore users, and radio hams. According to Barry Long, the early hours are due to the hams. As a service to exhibitors, the CPUG makes available free doughnuts and coffee while you are setting up.

The commercial enterprises attending the show included Barry Traver of Genial Computerware, Jim Horn of CompuServe, John McDevitt of Rave 99, Bud Mills of Bud Mills Services (Horizon ramdisks) and Chris Bobbitt of Asgard. There were also the usual user group tables and two TI dealers with large stocks of everything from cartridges to sidecar and other peripherals. A production Gramulator was available for inspection (and sale) next to the BCS stand.

I was unable to attend any of the presentations, but heard that Wayne Stith showed Triad and that Barry Traver showed Genial's First Base database program, written in c99 by Warren Agee. It was selling for \$49.95 and includes extensive documentation. It will be interesting to see which of TI Base or First Base win the database battle.

One of the most interesting parts of attending a TI show is to put faces to names. I had the pleasure of meeting: Mickey Schmitt,

author of "Getting the most out of your cassette system"; Chris Lang, author of the Wit series of games, and who was giving out a disk titled the Orphaned Instruments Title Screen Adventure Game, version 3.1; Harry Brashear, the outspoken president of the Western New York 99ers; and Chip Chapin, who writes the Chez Geneve column for the Nittany User Group. In addition, there were numerous people who came to the table to buy BCS disks and exchange pleasantries.

The show closed at 2:00 pm and we were taken back to the Harrisburg train station by Barry Long. Back in Philadelphia, we boarded a sleeper car attached to the Amtrak Night Owl and were soon being rocked to sleep by the motion of the train.

Both Justin and I would like to thank the CPUG in general, and Barry Long in particular, for their hospitality and for letting us participate in a well-organized, enjoyable TI fair. Here's looking forward to next year.

#### The book list

In the July Issue we published a 6-page list of all known TI-99/ 4A related books. I was recently fortunate to acquire some original TI material which should be added to the list.

When the 4A was being designed, TI produced a set of specifications for the computer, software and peripherals. These documents contain information fundamental to understanding the design and operation of the 4A. In the material that follows titles are shown exactly as they appear on the documents. Note that in some cases the 99/4 was referred to as a personal, not home, computer. The specifications include:

File Management Specification for the TI 99/4 Home Computer. 16 November 1979. Version 2.4. Sections include: 1.0 Introduction; 2.0 Applicable documents; 3.0 I/O handling, 3.1 File organization and use, 3.2 File management overview; 4.0 Implementation, 4.1 Peripheral Access Block definition, 4.2 I/O opcodes, 4.3 Directory handling, 4.4 Error codes; 5.0 DSR operations. 5.1 DSR Actions and Reactions. 5.2 Memory Requirements, 5.3 GPL interface to DSRs; 6.0 Linkage to Basic, 6.1 Basic PAB modifications, 6.2 Basic PAB linkage.

File Management Specification for the TI-99/4 Home Computer. February 25, 1983. Version 2.5. Similar to the above, but expanded to include the 4A.

TI-99/4 and 99/4A Personal Computer System Software Comprehensive Specification. February 25, 1983. Sections include: 1.0 Introduction, 1.1 Purpose, 1.2 Scope, 1.3 Terminology; 2.0 Applicable documents; 3.0 General description, 3.1 Hardware description, 3.2 Software description; 4.0 TI-99/4A keyboard scan routine. 4.1 Introduction, 4.2 State of the keyboard, 4.3 Keyboard levels, 4.4 Returned information and debounce, 4.5 Split keyboard and joystick scans, 4.6 Assembly language interface; 5.0 Console software, 5.1 System power-up sequence, 5.2 GPL application support, 5.3 Basic interpreter, 5.4 Peripheral support; 6.0 TI-99/4A Basic, 6.1 Functional changes, 6.2 Bug

fixes; 7.0 GPL interpreter modifications, 7.1 CRU in, 7.2 Case, 7.3 Fetch; 8.0 VDP interrupt handling; 9.0 Other modifications, 9.1 Break key routine, 9.2 ROM-only applications, 9.3 New character definitions; 10.0 File Management, 10.1 Introduction, 10.2 I/O handling, 10.3 Implementation, 10.4 DSR operations, 10.5 Linkage to Basic; Appendixes: A1 TI Basic, A2 TI Extended Basic, A3 Graphics Programming Language, A4 UCSD Pascal, A5 TMS9900 Assembly; B Compatibility.

Device Service Routine Specification for the TI-99/4(A) Personal Computer. March 28, 1983. Version 2.0. Sections include: 1.0 Introduction, 1.1 Interface in general; 2.0 I/O bus, 2.1 I/O bus pin assignments and descriptions; 3.0 Hardware structure of DSR, 3.1 DSR ROM, 3.2 CRU mapping; 4.0 Software structure of DSR, 4.1 Symbol definition block, 4.2 Header and linkage block, 4.3 Power-up routine, 4.4 Interrupt routine, 4.5 Main device service routine.

Functional Specification for the 99/4 Disk Peripheral. March 28, 1983. Version 3.0. Sections include: 1.0 Introduction; 2.0 Applicable documents: 3.0 Supported file management options; 4.0 Interface to Basic, 4.1 OPEN statement, 4.2 CLOSE statement, 4.3 PRINT statement, 4.4 INPUT statement, 4.5 RESTORE statement, 4.6 DELETE statement, 4.7 OLD command, 4.8 SAVE command, 4.9 EOF function; 5.0 Catalog file access from Basic; 6.0 File protection; 7.0 FILES subprogram; 8.0 I/O error codes.

TI-99/4 Home Computer EIA RS232C Peripheral General Software Interface and Operational Specification. March 28, 1983. Version 2.0. Sections include: 1.0 Introduction, 1.1 Functional capabilities; 2.0 Applicable documents; 3.0 Default parameters; 4.0 RS232 peripheral Basic language interface, 4.1 OPEN command, 4.2 CLOSE command, 4.3 INPUT command, 4.4 PRINT command, 4.5 LIST command, 4.6 OLD command, 4.7 SAVE command; 5.0 Sample programs and commands; 6.0 Procedure for program exchange with OLD/SAVE programs; 7.0 Interface restrictions; 8.0 Special comments; 9.0 Errors and error codes; 10.0 Decimal coded Ascii table.

Contained in the specs are references to other material. Note from the date references how much came together on March 28, 1983. This should probably be declared as TI-99/4A day:

Home Computer Basic Language Specification. 12 April 1979, Revision 4.1.

TI-99/4 Home Computer EIA RS232C Peripheral Detailed Software Functional Specification. 28 March 1983, Version 2.0.

TMS 9918A VDP Video Display Processor Data Manual (revised November 1982).

TMS 9919 Sound Generator Controller Specification (released 16 October 1979).

99/4 Home Computer Bus Specification (Electrical

Specification: document number 1037185).

Graphics Language Monitor specification.

Home Computer Disk Peripheral Hardware Specification.

Software Specification for the 99/4 Disk Peripheral. 28 March 1983, Version 2.0.

GPL Interface Specification for the 99/4 Disk Peripheral. 28 March 1983, Version 2.0.

In addition to the specs, other material includes:

A Guide to Using the Texas Instruments SN76489A Sound Generator. Ted Mahler, Applications Engineer, Texas Instruments. This is similar to the TMS 9919, SN94624N sound chip used on the 99/4A (U511). It also contains 3 programmable tone generators, each with its own programmable attenuator, and a noise source with its own attenuator. The report contains examples of the methods needed to generate control bytes for the chip, along with interfacting data and examples. Sounds include a bell or chime, basic bird, missile, and bomb drop and explosion. The Basic examples will not work in TI Basic without modification.

Technical Data. Texas Instruments, 1982. A boiled down version of the "TI-99/4A Console and Peripheral Expansion System Technical Data Manual". Everything in "Technical Data" is contained in the larger manual. ("Technical Data" is incorrectly listed as "TI-99/4A Technical Data Manual" in the book list.)

TMS 99000 Family Assembly Language Programmer's Guide. Texas Instruments. Covers the generic family of 9900 and 99000 CPUs — the 9900 (used in the 4A), 9940, 9980A, 9981, 9989, 9995, 99105 and 99110A. The 99000 family instruction set is composed of a base set of 69 instructions, plus extensions particular to each CPU, which provide for the input, output, manipulation, and comparison of words, bytes and Ascii character data. Only the base set is used in the TMS9900, TMS9980A and the TMS9981. Each instruction is covered in painstaking detail in this large manual.

Software Development Handbook. Texas Instruments, October 1981. (Only Chapter 8, Assembly Language, available.) This chapter contains 127 pages with many examples of assembly coding applicable to the 4A. For example, page 8-91 contains a routine to read data from the 9902 receive buffer register. The TMS9902 is the asynchronous communications controller used in the RS232 card. All in all a wealth of material requiring hours of study.

TM 99/U89 Microcomputer User's Guide. Texas Instruments, 1981. The TM 990/U89 was a self-contained, single-board microcomputer system. It was intended for use as a learning aid in the instruction of microprocessor fundamentals, machine and assembly language programming, and microcomputer interfacing. It also demostrated the TMS9900 family applications and advantages. This manual is a comprehensive

guide to all of the above. (A similar model, the TM 990/189 University Module, was reviewed in 99'er Magazine, May/June 1981, p78.)

## Home Publishing on the 99/4A

The Western New York 99ers user group has for sale a package called Home Publishing on the 99/4A by Harry Thomas Brashear. It consists of a disk and an 80-page manual on how to use the supplied programs to create banners, pictures and creative text. According to the group's flyer:

The manual "will show in detail what clips and instances are and how to use them. It also catalogues over 150 fonts and shows how they may be used.

"Home Publishing talks about the many graphic programs, just how they relate to each other, or how to put them together to work for you.

"But it doesn't stop there. The manual you get is 80 pages long... and it just wasn't enough, for the cost of the original manual you will also receive two 20-page supplement issues during the coming year. This is so your manual will always be up to date with information on the latest in graphics software.

"You will also receive a disk of programs to help you make your documents look their best. The programs include a TI-Artist instance printer that will not leave any skip lines in the printout, a program to organize your fonts and graphics, and the world's best columnizer.

The cost of the complete package is \$15.00, which includes postage and handling. User groups can buy a minimum of 10 packages for \$12.50 each. To order, or for more details, write to:

Western New York 99ers, c/o Harry T. Brashear, 2753 Main Street, Newfane, NY 14108.

#### Light pen offer

A light pen is a light sensitive device that can be attached to a computer and is typically used to "pick off" items on the display screen.

For example, a program could display a list of operations that it will perform. Instead of typing the usual selection digit and then <ENTER>, you can use a light pen to point to the line describing the operation on the display and have the computer respond. The application can be extended by having color boxes at the bottom of the display and having objects change to the color box pointed to, and so on.

TI did not manufacture a light pen for the 4A, but there have been some enterprising 99ers who have published details on how to make a light pen. The latest to do so are Don and Bob Emmel of the York Area 99ers Users Group. At the CPUG

exposition held recently in Harrisburg, PA, they were offering a sheet showing how to build a light pen that plugs into the joystick port using a Radio Shack photo-cell, an extension cord and a felt tip marker.

In addition, they are offering a disk containing light pen shareware programs, one of which is a contrast adjusting program. They request a prepaid mailer and disk to be sent to:

York Area 99ers Users Group, Box 7160, York, PA 17363.

## Disks to go

A company called "Disks To Go" is offering 5.25-in DSDD disks, black, in Tyvek sleeves with write protect tabs and color labels for 29c each, with no minimum purchase. The same disk is available in up to 16 colors for 41c each. The company also offers continuous form labels — for example, 2 3/4" x 7/16", 1 across, box of 10,000 at \$1.95/1000 — and disk mailers — for example, mlcro 4" x 5 3/8", brown at 31c (1-100).

You can order by calling 800-426-3303 (toll-free) or 609-456-6996, or you can write for a flyer to:

Disks To Go, 1040 Broadway. Westville, NJ 08093.

## c Column

## by Donald L.Mahler

Ron, our local Pascal expert, has been working with benchmark tests to compare the run times of programs written in different languages. He asked me to set up c99 files for comparison. All of the programs use the following equation:

x = x + ((y \* y) - y) / yIn each example, the equation is run 1000 times, using the new value of x each time, the results printed to screen, and then the whole process repeated 100 times. The file for integers is straight-forward and can be run on the 9640 or the 99/4A:

This took 137 minutes to do the 100,000 calculations and print the results, using the Geneve.

For floating point numbers, the situation is more complex. Tom Bentley's floating point library will NOT run on the 9640, so I had to set it up as a purely 99/4A c file:

```
/* benchmark for floating point numbers */
/* x = x + ((y * y) - y) / y */
extern printf();
#include "dsk2.floati"
/* on library dsk */
#define f 8
/* 8 bits for integer */
char op[2]; /* for calculations */
char s[f]; /* 8 bit string */
main()
{ float x[f],y[f]; /* 2 floating point
numbers */
  char *v, *w; /* 2 character pointers */
  int i.t;
  float a[f], b[f], c[f], d[f];
  /* floating pt nums for intermediate calcs
*/
  for(t=1;t<100;t++)
   (printf("Start loop %d /n",t);
  v="0.0"; w="9.9"; /* strings */
  stof(v,x); stof(w,y); /* string to fp */
 for(i=0;i<1000;i++)
  {fexp( y, "*", y, a);
                       /* a = y * y
                       /*b = y^2
  fexp(a,"-",y,b);
*/
  fexp (b, "/", y, c);
                      /* c = b / y
  fexp(c,"+",x,x);
                       /*x = x + c
  fpput(x,s);
  putchar('\n');})
printf("This is the end!"); }
```

I could not set up his third benchmark test; this involved trig functions, and I do not know of any implentation for them in c99.

## Intro to the UCSD P-System By Ron Williams

This month I will begin to cover some Pascal programming. Pascal is a language that like "C" is compiled, you must first type in the program with a text editor and then compile the text file to create an executable code file. The UCSD Pascal that the TI supports is the same UCSD Pascal that you could get for an Apple, IBM or other microcomputer the basic difference is memory, the TI just does not have as much but with some special programming you could get a lot of the Pascal programs written in UCSD Pascal on other machines to run on the TI. I have taken code that was created on other computers and with a few changes compiled and ran them on the TI. This code was brought in as text files and not code files, you still have to compile the files on the TI, as code files on other machines is compiled in that machine's assembly code.

The very first statement in every Pascal program is the program statement and it is at the beginning of every Pascal program.

It looks like this:

```
program Hello;
```

This statement must be included at the beginning of every Pascal program and it is changed for every program written, it should have a name of what the program is going to do. The declaration statements are next and there are three of them Constant, Type, and Variable and they are used in the order given but not all of them are used at any one time. The Constant declaration is used to declare program constants, these are identifiers that are never changed thru the execution of the program.

A example of a constant declaration is as follows:

```
Const pie = 3.14;
```

The next declaration statement is the type declaration and it is used to make a user defined type, that is you may create a record, or another type that is used by the program to mean something.

An example of the type declaration is as follows:

```
Type
Tens - (10,20,30,40,50,60,70,80,90);
```

This means that the identifer tens can have the value of the numbers that follow it and no others, it is a type defined by the programmer. The last declaration statement is the variable statement, it is used to declare program variables, these identifers can be changed by the program and all identifers must be declared before using them unlike Basic which you can make a new variable any time you need it. This means it is very easy to know what variables are being used by the program and you could also put comments next to them to tell anyone reading the program what they are.

An example of the variable statement is as follows:

```
Var
Number : integer:
```

When you declare an identifer you may use some data types that are already defined for you within the Pascal language and a few of them are:

Integer: This type is for numbers that are to be defined as integer

types(numbers without decimal points).

Real : This type is for real numbers (Numbers followed by a decimal point).

String: This type is used for characters of data.

Char : This type for one byte characters.

Boolean: This type accepts only a value of true or false.

Next month we will look at a few statements that the Pascal langauge uses to print out data and read data and also at the begin and end statements.

# Random Ramblings By J. Peter Hoddie

It is 3 AM before this months meeting. To say that I am slightly exhausted as a result of the Chicago Faire this past weekend would be an understatement. I would publish an unbiased show review here, but as of yet none have appeared on any of the networks. Instead I will present some random Information here. This information will be rather skewed towards BCS and Genial Computerware, I'm afraid, since this is about all I saw. Other information will have to come from others. I apologize for any blatant self promotion in these pages, but what can I say? If you want to see less of me in these pages, provide me with some articles by you.

## **Epnoymous**

i was listenling to the new R.E.M. album "Green" on the flight to Chicago. Somehow I got into a conversation with Mike Wright during the flight about the band. They have an album named "eponymous" - actually a recently released greatest hits collection. In any case, the album title became a running joke between Mike, myself, and Wayne Stith throughout the weekend. Unfortunately we didn't have any idea what it meant. Wayne provided me with the following information on the word. I'm not so excited by what the word actually meant, but I am kinda glad that it doesn't have any PG-13 or worse connotations!!

The word "eponymous" refers to the giving of a name, as in "Romulus was the eponymous founder of Rome." Eponymy refers to the state of having such a condition, and an eponym is the person from whom something receives a name. So there! I certainly hope this lays to rest all the wild speculation...

#### ChainLink

If you have been to a BCS meeting you have probably seen Walt Howe's popular Chainlink solitaire game. This program has been extremely popular in the TI community, a tribute to Walt's gaming and programming talents. Wayne Stith hand "compiled" the game into assembly language. The result is pretty cool. The program runs at break neck speed. Recently Wayne added animated card motion so that the cards slide into place when the move. He also added an animated deal. Some sound effects and other features are in the planning stages. The program will be

released by Genial Computerware on disk and cassette sometime towards the beginning of next year. At the demo in Chicago there were many oooo's and ahhhh's as the cards slide along. The bad part: I was doing the demo, and I'm a miserable Chainlink player. The good part: my lack of card playing talent provided a good laugh for the audience.

## The BCS Software Library

Our software library was at the show again this year. For the second year running Mike Wright was mostly responsible for watching over this esteemed collection of programs. The most recent reprint of our software catalog was also there. It was kind of exciting to have people come up to the table and comment "I've seem your advertising.." Just goes to show that our two MICROpendium ads really did pay off. Of could Justin's recent rush of software mail orders would also testify to the success of our advertising. At Chicago we did considerable sales, sending back a box more than half empty. I would provide sales figures here, but I haven't got them exactly. Needless to say, we did cover expenses with a little bit left over....

The software box was actually resting on two tables: the Genial table and CADD's table. Mark said he did pretty well with Grammulator sales. If you are looking for a GRAM simulator, I would seriously recommend checking into this one. It is the natural successor to MG's GRAM Kracker, and surpasses it in many ways. Thanks to Mark for letting the BCS take advantage of a corner of his table.

Many thanks to Mike for watching over the software library. Without his efforts I'm really not sure how the software box would have been watched over. The Genial area was a total zoo for most of the day. Mike's efforts on behalf of the BCS cannot be understated. He provides excellent articles for the newsletter and also manages our newsletter exchange with other user groups. Thank you. Thank you. Thank you.

## Browse

In Chicago I released this program I wrote called Browse. It was not pre-announced at all. It just appeared at the show. We sold out. I was happy. There is an interesting story about the program. I wanted to release something at Chicago this year. Unfortunately time was short. I had two half finished products somewhere. Browse was one, and a crossword program I was working on was the other. I decided to finish the crossword program since it was more fun. I sat down to do it and discovered that I only had the source code the Browse at school. The crossword source code was buried at home somewhere. So I finished Browse instead. When I get home for Thanksgiving I'll dig up the crossword source code. Can anyone guess what my next release could possibly be? Anyhow. What follows is a description of Browse ripped from our catalog. It is not a review.

However, I did give a copy of the program to Art Byers, newsletter editor for Central Westchester group and a Sysop on Delphi. His comment to me in a message last night was: "I did try it out last evening and found it to be a very nice USEFUL utility to have. I won't hesitate to recommend it to anyone!!" The list price is only \$10 (plus \$1 shipping). Here's the catalog text:

Browse is a utility to aid in the management of text files. Using Browse you can easily print, view, combine, and browse text files. You can easily select a group of files from a disk an have them printed. Browse has options which allow you to start each file on a new page, print the file name, use special modes of your printer, strip control characters, and more.

Browse has an extremely friendly user interface including pop up windows. File selection is handled through the familiar Disk Manager 1000 interface. Many functions are available through several different keypresses for maximum convenience of use on both the 99/4A and 9640.

Over 24000 bytes of text may be loaded into memory at one time. You can have several files loaded into memory at one time. These files may be viewed using a file viewer similar to the one found on Genial TRAVeIER. The viewer allows for 80 columns on the 9640. The viewer also includes keys to quickly move to the beginning and end of the file. Select groups of files may be printed from memory, so that you can view a set of files before deciding which to print.

Browse allows files to be cataloged on floppy and RAM disks as well as hard drives. It is fully configurable, with all user settings saved as part of the program for immediate availability whenever Browse is run.

Browse is an extremely useful and easy to use utility. Browse can read files from TI-Writer, Funnelweb, and MY-Word. It is a must for anyone who uses their TI for word processing. Because it is written in assembly language, Browse has an extremely quick response to your commands.

Browse requires Extended BASIC, Editor/Assembler, or TI-Writer to run. It runs on a TI-99/4a or 9640 computer.

## 9640 Software Lives

In Chicago this year for the first time, three commercial programs were available that actually ran from MDOS mode. The first of these was DISkASSEMBLER Version 2.0 from T + J Software. This is a new version of Miller Graphics classic program, updated and enhanced by Tom Freeman. I haven't had a chance to work with it yet, so a review will have to wait until next month. From reading the docs, and knowing Tom Freeman, I can recommend the program to anyone who has a 9640 and can use a program with the power

of DISKASSEMBLER. The program sells for \$20. For more information or to order write to T + J Software. 515 Alma Real Drive, Pacific Palisades, CA 90272 or call 213 454-1943.

T + J software also announced a new product called THE Bugger by Jim Lohmeyer. This is a debugger for the 9640. It looks incredible. Unfortunately it is in a limbo state which I can't really discuss here. Hopefully we'll have a full report in these pages next month.

The other two MDOS mode products available were from (surprise surprise) Genial Computerware. They were Mike Dodd's HyperCopy and Paul Charlton's Picture Transfer. Both sold out. Both were extremely well received during the demo. Both will be further discussed in these pages in the months to come.

## Other Ramblings

Press by Charles Earl being distributed by Asgard looks neat. I talked to Charles at great length Sunday night, but not much about Press. I hope to have a copy to demo as soon as it becomes available. The documentation by Ruth O'Neil should be a helpful addition - Ruth knows what she's doing. I hope Charles survives the next couple weeks of debugging - apparently that's about all that remains to do.

There were two time periods during which the Genial table was relatively calm. The first was during the Genial presentation when we shut it down. The second was during the demo of Press. Congratulations to Charles Earl. His arrival on the TI scene as a "major player" in the past few months has generated more excitement than anyone could of predicted. His high quality programs are setting new standards that we'll all benefit from.

FirstBase sales were very good, with Warren Agee autographing copies from our table. Amazing how many Ti-Base owners came by to buy a copy of FirstBase. I guess Ti-Base is not the answer to everyone's prayers.

The Genial presentation at 2:30 was packed. There were people standing in the back, sitting along the sides, and peering through the door. It was a great tribute to the talents of programmers like Paul Charton, Wayne Stith, Warren Agee (who were all present), and Mike Dodd that their programs could generate such excitement and interest. Many thanks to the Ti community for showing such support. We had a blast for that hour.

The Ottawa crew was a joy to be next to \_again\_ this year. We stole about 18 inches of their table. Encroaching on international borders. I wonder which government will prosecute....

I purchased Asgard's new Oliver's Twists adventure

by Mickey Schmidt and Lynn Gardner. The adventure runs from the adventure cartridge. According the Mickey the "twist" is that instead of having to find treasures and bring them to special room as in most adventures, in this one you start out with all the treasures and have to put the back. Sounds interesting. I'll have a report on my misadventures in a future issue.

If you want a free Genial Computerware catalog describing our entire product line drop a note to: Genial Computerware, P.O. Box 183, Grafton, MA 01519.

Asgard can be reached at: P.O. Box 10306, Rockville, MD 20850. Who says I don't give information about the competition?

Well its now 3:35 AM. I have to get some sleep or the meeting will be a bit stranger than usual. If you have something to contribute to these pages, don't hesitate to contact me. I can usually be reached at (617) 375-6003. Things are a bit calmer now that Chicago is over. Don't anyone say San Diego....

# Intro to the UCSD P-System By Ron Williams

Last month [last month I accidently reprinted an old P-System article, titus two this month. my fault. -jph] i begun to cover Pascal programming this month we will look at the write statement, the read statement, begin and end statements, and also at loops. The write statement allows you to print data to the screen, the printer, or a Pascal text file. It will print character data or numeric data.

An example is as follows:

```
WRITE('hello');
```

This statement will print out helio and will cause any other data printed out to be on the same line as hello.

```
The statement WRITELN('hello');
```

Will let any other data printed to be on the next line the writein statement will make a carriage return after it. Now to get the data to be printed on other columns you could write the statement like this: WRITELN ('hello':10);

This will make the data to be printed on column 10 right justified. Real numbers may also be printed, formatting them also to print out the number of decimal places to the right of the decimal point.

#### Like this:

```
NUMBER: =3.14;
WRITELN (NUMBER: 5:2);
```

The second number after the five is the number of decimal places to be printed.

The read statement will let you input data from the screen or a Pascal text file and is like this:

READLN (INDATA);

With INDATA being any variable previously defined. The read statement will cause any data printed out to be on the same line as the read input data so use readin if you want the next data read or printed to be on a different line. The read statement will read characters of data, a character, or real or integer numbers.

The begin and end statements are very important as they define groups of statements to be used by the program. To begin a program the main body must have a begin and end statement and the last end statement should have a period after it. A lot of statements in Pascal also need the begin and end statements to tell where they begin and where they end. A good example of this is the for loop, this loop is very much the same loop used in Basic.

```
An example is as follows:
```

```
FOR X:-1 TO 10 DO

BEGIN

WRITE ('NUMBER');

WRITELN(X);

END:
```

This example uses the write statement, and begin and end statements tell the beginning and the end of the loop. The for loop does not really need the begin and end statements all the time if there is only one statement after it the semi-colon after the statement will be the end of the loop.

#### Like this:

```
FOR K:=1 TO 10 DO WRITELN('NUMBER',X);
```

The next loop to show you is the repeat loop it is like this:

#### REPEAT

```
<PROGRAM STATEMENTS>
UNTIL CONDITION;
```

This loop uses a condition defined after the until to exit the loop, This is a boolean condition telling the loop to end.

The while loop is like the repeat loop but the condition is checked before entering the loop.

## Like this:

```
WHILE CONDITION DO
BEGIN
<PROGRAM STATEMENTS>
END:
```

The begin and end statements are required for this loop and it is good loop for checking data before entering a loop.

One more thing I wanted to mention is the use of semi-colons you may have noticed these after some of my program statements. These are required in the Pascal language after a lot of Pascal statements like the write and read statements and after the end statement in a group of statements.

So long until next month.