OUR NEXT MEETING will be on Friday,

AUGUST 19, 1983 at 7:30 pm

PLACE: KEY BANK BLDG.

SW corner of Rt. 20 and Rt. 155

THE SEPTEMBER MEETING will be on Friday.

SEPTEMBER 16, 1983 at 7:30 pm

PLACE: KEY BANK BLDG.

SW corner of Rt. 20 and Rt. 155

August's meeting will include Jon Daggett on dimensioned arrays

UPSTATE 99/4A USERS GROUP
P.O. BOX 13522
ALBANY, N.Y. 12212

VOL I, NO. 6 AUGUST, 1983

SOFTWARE REVIEW

Are you looking for a game where you don't have to shoot them before they shoot you? I have just the games for you. TI's A-MAZE-ING and Milton Bradley's CONNECT FOUR. Two very easy to learn games. Both are simple enough for young children to play, yet challenging enough for any adult. Both are either one or two player games. I feel that this is a real plus, as there are times when you want to play but have no one to play with. Let's take a look at each of them.

A-MAZE-ING is just that, a maze game. But not just a simple maze, there are over 5,000 possible variations, including dangerous cats, delicious cheese, and obstacles or mouseholes to aid or hinder your escape. In this game you are a mouse trying to find your way out. Watch out for those cats.

With the various options you can create a maze on almost any skill level. With the one player option you can play against yourself, the clock or the cats. Against yourself you can play a number of games trying to better your score each time. You can either work to find your way out or eat all ten pieces of cheese before you leave. You can have a simple or complex maze, visible or invisible, with mouseholes or obstacles, and a fast or slow mouse. Then there are those dangerous cats. There can be from zero to two cats. Their speed ranges from slow to lightening, they can be either dumb or smart (they almost seem to know right where you are) and either standard or pouncing, with varing pouncing frequencys. In the two player mode you have all the one player options an also the option to play either a competitive or cooperative game.

To me A-MAZE-ING is more than just a game. It can be a learning tool as well as a game. It requires reasoning, and memory skills to play the game. As a mother of small children this is an important feature in this game. I have only one complaint with this game, the mouse is too small. Sometimes when he is in a corner he is almost impossible to see.

CONNECT FOUR is a game of stategy for one or two players. You play it almost the same way the origional connect four is played. You try to place four checkers in a row horizontally, vertically, or diagonally before your opponent does. Try to develope a winning strategy with this mind teasing game.

With the one player option you choose the skill level of your opponent. Novice (being the least challenging) the degree of skill progressing to Master (the most challenging).

When you select two players you have more options to choose from. You can play a Basic game, Drop Out, or play with a Wild Spot. In the basic game you compete to be the first to get four checkers in a row on a regular playing board. In Drop Out you have the option to drop one of your checkers out the bottom instead of dropping one down the top, but beware if upon doing this you both connect four your opponent wins. With the wild spot you play a regular game except that somewhere on the board is a wild spot that can be used by either player to connect his four checkers.

The question is, where do you put your checker and where will your opponent put his? Who will CONNECT FOUR first? I only wish that you could play Drop Out with the one player option, maybe then I could beat the computer in more than just the Novice level.

Both of these games are excellent for both children and adults and are thought provoking games. You have to use you head not just have a fast wrist to win in these games.

Sally Lane

MINI MEMORY

This article is for those of you who didn't take my advice and bought the Mini-Memory anyway. If you are like most people, you haven't gotten anywhere with it yet either! Programming Assembly language with the Mini-Memory's Mini-Assembly is actually harder than using the larger Editor/Assembler system. You have fewer Op-Codes and also the naming of your program is much more complex. In this article, I will attempt to help you with some of these common problems.

Naming your program is perhaps the most difficult to comprehend. With the "big" assembler you just DEFine your start name in your source code, and that's all there is to it! With the Mini-Assembler, you must go through a long process to name your program. Here is a step by step method of naming your program.

1) Type in your program (do not put in the END command yet)

2) Make sure to write down the memory location where your program started

3) Type in AORG >701E (this is the pointer to the REF/DEF table)

4) Subtract 8 from the number in >701E (ex. >70E8 - 8 = >70E0)

5) Type in DATA >(the number you just came up with from above)
   You have just changed the start of the REF/DEF table

6) Type in AORG >(the same above number)
   You are now at the new beginning of the REF/DEF table

7) Enter you program name. This MUST be 6 characters in length. If shorter than 6 characters, you must pad with spaces. You use the TEXT Op-Code for this. (ex. TEXT 'START ')

8) Now type in DATA >(where your program starts)

9) Finally you can type in END, which tells you of any unresolved references, and then exits the assembler.

10) Run your program using the RUN option in the Mini-Memory

Sounds confusing enough, doesn't it! (It is)

But, don't give up. After doing it once, it will become an easy process. Basically what you are doing is entering the starting memory location of your program into the REF/DEF table. Just prior to this location is your start name (remember, it must be 6 characters long). The reason for subtracting 8 from the hex number at >701E is because you are adding a new program to the list. If you were just changing the name of a program, you could keep the REF/DEF table where it is and just change the name using the TEXT Op-Code.

Things to remember:

- Memory location >701E "points" to the start of the REF/DEF table.
  To look at this location you use AORG >701E. To change the contents of this pointer, you use DATA >(the new number).
  note: when subtracting 8 to add a new program name, remember that you are subtracting in HEX. (base 16)

- Write down the starting memory location of your program when you go to type it in. (It could be hard to find later)

- Use single quotes (') not double quotes (").

- If you have any problems, don't call me... Buy an Editor/Assembler.

Jon Daggett

SOFTWARE REVIEW

After what seemed to be many days of impatient waiting by the mail box, our DISK FIXER from Navarone Industries finally arrived. This assembly language program requires 32K memory expansion, at least 1 disk drive and the Editor/Assembler cartridge. It's cost was $29.95 and in my opinion is worth every penny.

I opened the package not really knowing what to expect to find. Inside was a disk and a small 10 page instruction book. After just a few minutes of reading the manual, and playing around with the program, I realized it's full potential.

This program allows you to read or write from a disk by individual sectors. This may not sound like much, but until now, the only way to access the disk drive was in one of the normal Open, Close, Save or Load ways. Also this allows you to change any byte on any sector of your disk.

Great! But what is it good for. Well, have you ever had a "frazzled" disk? This is a disk which you were using, and all of a sudden, it wouldn't let you at your programs! You can repair your disk with this program, and not lose the information on your disk!

Now a little bit about how it works. Anyone buying the Disk-Fixer should have some knowledge of the computer and disk handling. The manual provided tells you how to use the program and has some useful information, but it is up to you to really find out what it can do for you.

Sectors 0 and 1 are referred to as the CONTROL TRACK of the disk. This contains the diskette name, and available space left on the disk, an alphabetical pointer list of your files, and also whether or not the disk is protected.

Sectors 2 through 33 contain the directory entries for each file. This tells you the File Name, the File Type, the File Size, and the Record Length of each file. This also contains the sectors used for this file, and the length of each fraction of a file. The appendix of the manual describes this in more detail. One thing I would like to add though, that the manual doesn't mention. If you run into problems finding the starting sector of your file, you may have to look at the 2nd nibble of the 2nd byte of the 3 byte BLOCK LINE in the directory entry. This contains the overflow if the sector number is greater than >FF.

The program is easy to use and very user-friendly. But beware... You can cause permanent damage to a disk if you do not know what you are doing. You are actually altering pieces of a disk.

The program is a lot like the debugger. It uses one letter commands, followed by any information needed for the operation. Commands provided are as follows:

W)rite sector
A)lter data
D)isplay buffer
M)odify RAM
H)elp
Q)uit

The Help command gives you the above list and how to use them so you don't have to keep refering to the manual.

In less than an hour, we learned some interesting things that this program can do for us. For example, if you have a disk which you are unable to back-up try looking on sector 0 byte >0010. If you A)lter >0010 to >2028, you may find some interesting results.

Also for those of you who accidently protected an undebugged Extended Basic program and need to get a Bug out, you may try finding the first word of the program on the disk. This word is a two's compliment number. Try changing this number from a negative number to a positive number?!? (A knowledge of two's compliment numbers and hex is recommended to be able to do this)

As I mentioned earlier, this program is very powerful and useful, but be careful because you could really mess up a disk with this if you don't really understand it. The program comes on an uncopyable disk. (but that can be FIXED)

Those of you who are interested in seeing how this works, bring your disks in to the next meeting, and I will answer any questions about this program.

Hints From Henry

Welcome to the first installment of an advice column for TI 99 users. No, it's not an advice column of the "Dear Abby" type, but rather like the 'Hints from Heloise' type seen in some newspapers. The object is to provide hints, tricks, short-cuts, etc. for use with your TI 99. We'll start off with items I've picked up in a variety of places. After that we'd like to see reader input to this column. Send us your favorite hints, tricks, and/or short-cuts that you think would benefit 99-users, and we'll share them with other readers. The more reader input we get, the better this column should be. With the realization that our club membership changes continuously, occasionally we'll repeat the more popular and/or useful items. Join in and share!

Item 1: TI BASIC doesn't have a "SIZE" command like EXTENDED BASIC, but you can still get a good idea of how much memory is left in TI BASIC with a simple algorithm. Put the following two lines at the front of your program:

```
1 MEM=MEM+8
2 GOSUB 1
```

Now RUN the program. When you get a "MEMORY FULL" message, type the following imperative command:

```
PRINT MEM
```

The computer will print out a number that is the number of remaining free bytes. This number will in general be accurate to within a few hundred bytes. (NOTE: Be sure to delete lines 1 and 2 before using your program.)

Item 2: In TI's Basic interpreter its much more efficient to multiply than to divide. You can use this information to speed up programs that do lots of calculations in FOR-NEXT loops. For example: dividing a number by 2 takes 4.9 milli-seconds, but multiplying by 0.5 (an equivalent operation) takes only 2.6 milli-seconds. You can use this trick anywhere. When it comes to FOR-NEXT loops, you need one more piece of data. It takes 10.6 mili-seconds to invert a typical non-integer number (like 17.632) and to store the results as a variable. For such a number you save 4.3 milli-seconds if you multiply by the reciprocal instead of dividing by the number. Therefore if your FOR-NEXT loop will execute more than twice, you'll save time by inverting a constant outside the loop and multiplying by that rather than by dividing by the original constant. Loops of 100, 200 or 1000 could result in considerable time savings at a rate of 4.3 milli-seconds per loop per calculation for non-integer constants.

ITEM 3: Do you sometimes feel like you need more than four lines for a statement in console BASIC? Due to a flaw in the 99/4A BASIC interpreter, you can. Suppose you wanted to print this statement on your screen: There are times when I wish that I could put more than four lines into statements that I am using in TI BASIC programs. Type in:

```
100 PRINT "There are times w
hen i wish that i could put
more than four lines into st
atements that i am using in"
```

You notice that the computer will not accept anything past the last ("). However if you edit the line by typing 100 and hitting FCTN-X and then skip to the end of the line you can continue to type until the end of the fifth line. You can do the same thing again for a sixth line. (NOTE: The fourth line must be full for this to work; that is you must type to the end of the fourth, or fifth, line if you wish to expand it by editing.

Mike Henry

SWAPCLUB NEWS

Swap club business has been relatively slow during the summer. We've only had two new additions to the club since our July meeting. As a result, we won't publish a new catalog this month. We will have copies of the last catalog available at the August meeting for those who didn't get a copy in July. The two new additions (both contributed by James Fairweather) are:

SC-081 MASTER BLASTER (1P,RTG,G) (XB) (Ages 6-adult) You must destroy invading space ships before they blow-up your planet.

SC-082 CHOMPS (1P,RTG,G) (XB) (Ages 6-adult) You must track down a bug in a maze and CHOMP him before time runs out.

Because activity has been low and because I'll be on vacation in the Adirondacks, we won't offer a new tape of four programs at the August meeting. There are still extra copies of the June and July tapes and these will be available for purchase for $5 on August 19th. Those tapes contain:

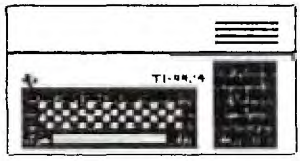
June: SC-055 Quintus (1P,STG,G)
SC-056 Boa Alley (1P,STG/RTG,G)
SC-065 Blackjack (1-5P,STG,G)
SC-067 Rocket Blast (1P,RTG,G) (XB,JS)

July: SC-057 Jumping Jack (1P,RTG,G)
SC-064 Lifeline to Titan (1P,RTG,G) (XB,JS(o))
SC-070 Lost Ruins (1P,STG,G)
SC-077 Camel (1P,STG,S)

We are considering opening the Swap Club for business for 15 minutes before the meeting start time of 7:30 P.M. This would help those of you who cannot always stay around until the end of the meeting. Let Rich Lane know how you feel about this. More on the subject in the next Newsletter.

See you in September.

Mike Henry



FROM THE CONSOLE OF THE PRESIDENT

This month's general meeting will have another first. One of the several dealers who attend our meetings will be offering a free piece of cassette software to some lucky person as a door prize.