YESTERDAY'S NEWS

VOLUME 4 NUMBER 4 Established 2016 APRIL 2019

30 Years Ago...

Historical Information taken from Bill Gaskills TIMELINE

APRIL 1989:

Andi Wise of the Eugene, Oregon TI User Group releases a 500 record database on names and addresses of past and current 99/4A User Groups. The file is written for MarK Beck's Creative Filing System. A TI-Base adaptation of the Andi Wise UG database is released by Bill GasKill.

MICROpendium announces that it will begin offering programs on disK that appear within the pages of its magazine.

The 4th Annual TI Fayuh taKes place on April 1st in Woburn, MA.

The Ottawa TI Fair taKes place on April 28th. Ottawa User Group member Gary Bowser, dba as Oasis Pensive Abucators (OPA) releases DISKODEX during the fair.

Charles Earl, author of the popular TELCO terminal emulator tells attendees at he Ottawa TI Fair that he will be releasing HOT BUG, a program he developed for debugging the PRESS word processor. The program runs out of a GRAM KracKer or SuperCart, leaving a full 32K RAM for your program.

Archiver III v3.03 is released by Barry Boone.

My-Art Picture Viewer is released by Barry Boone.

Jesse Slicer, Olathe, Kansas, releases first utility to parK the read/write head on a Myarc HFDC managed hard disK.

Randy Moore, Edwards AFB, California, releases Sector One, the first sector editor that's designed to worK on hard drive sectors.

Page Pro 99 by Ed Johnson is announced by Asgard Software. It debuts at the Ottawa TI Fair.

Music Pro, by David Caron and Lucie Dorais of the Ottawa TI-99/4A Users, is released by Asgard Software.

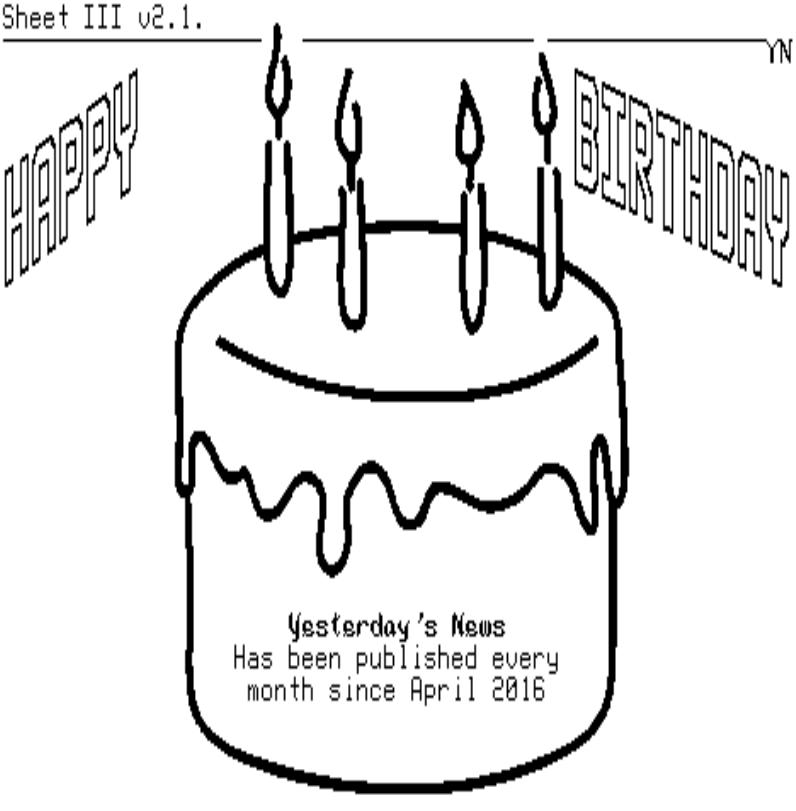
MICROpendium announces that it will be selling disKs with the programs from each issue.

TexComp Users Supply announces that it will be sponsoring a conference in July in Hawaii, on business applications for the TI-99/4A computer.

Using the Navarone Data Base Manager booK is released by Bill GasKill.

Glenn BernaseK, dba GEE BEE BASICS, releases TI Short Sheet III v2.1.

INSIDE INFORMATION



ELEMENTS OF BASIC

By Dave Howell

Courtesy of the Erie 99'ers

Part 18

STRING FUNCTIONS - continued

LEN Function

LEN(X$) is a string function which gives the length of, or number of characters in, the string X$. LEN is used to do special formatting or to make sure someone entered a string answer rather than a number. Try this:

```
10 A$="THE TI IS A NEAT MACH
INE."
20 PRINT LEN(A$)
30 PRINT LEN("THE TI IS A NE
AT MACHINE")
```

The TI will print 25 for both print statements. This number includes all spaces and punctuations as well as characters between the quotes. If A$ in the example above is a null string (""), that is, it has no characters or spaces between the quotes, then the computer will print 0. This is especially useful in determing if someone hit the ENTER Key without actually entering a string answer.

LEN can be used to assign a value to a variable, like this:

```
10 A = LEN(X$)
20 B = B + LEN(X$)
```

Or, LEN can be used as part of an expression:

```
10 FOR I = 1 TO LEN(X$)
50 IF LEN(X$) = 0 THEN 40
```

This last example (line 50) checks to see if the answer (X$) was a null string (0). If so, then the computer is directed to go back to line 40 and repeat this question.

POS Function

The POS(A$,B$,n) is the position function. A$ and B$ are string expressions. The numeric expression n is evaluated and rounded to a whole number, if necessary. POS finds the first occurance of the second string B$ in the first string A$ starting at the nth character in the first string A$. If B$ is found in A$, the location of the first character of B$ in A$ is given. If B$ is not found in A$, then the value of zero is given.

```
10 A$ = "ABCDEFGHIJKL"
20 B$ = "HI"
30 P = POS(A$,B$,3)
40 PRINT P
```

In the above example, the computer will begin looking for "HI" after the 3rd character "C" in "ABCDEFGHIJKL." ^The computer will find the "HI" beginning with the 8th character in "ABCDEFGHIJKL." ^Therefore, the position P will equal 8.

Here are more examples:

```
10 A$="BOXES"
20 B$="X"
30 PRINT POS(A$,B$,1)
RUN
3
```

```
10 A$="BOBBY DOLE"
20 B$="B"
30 PRINT POS(A$,B$,5)
RUN
0
```

```
10 A$="BOBBY"
20 B$="BOB"
30 PRINT POS(A$,B$,1)
RUN
1
```

```
10 A$="BOBBY"
20 B$="X"
30 PRINT POS(A$,B$,1)
RUN
0
```

SEG$ Function

The SEG$ function will find a substring (part of a string) that is part of a longer string. For example,

```
10 A$="MISSISSIPI"
20 PRINT SEG$(A$,7,3)
```

The SEG$ function will print "SIP" which begins with the 7th character in MISSISSIPI and is 3 characters in length. What is actually printed depends on what the 3 characters are beginning with the 7th character in the given string. Consider these examples:

```
10 A$="PENNSYLVANIA"
20 PRINT SEG$(A$,10,5) RUN
NIA
```

```
10 A$="OHIO"
20 PRINT SEG$(A$,6,2)
RUN
```

When a position is selected that is beyond the end of the string as in the second example, a null string will result. That is, nothing will be printed. The null string is a character string whose length is zero. It contains no characters.

If the value for the length carries the substring beyond the end of the original string, as in the first example above, the computer will print only the characters it finds up to the end of the string. The following are more examples of the SEG$ function.

```
10 A$="HERE IS A MESSAGE"
20 PRINT SEG$(A$,1,4);
30 PRINT SEG$(A$,3,5);
40 PRINT SEG$(A$,12,3);
50 PRINT SEG$(A$,12,12);
60 PRINT SEG$(A$,20,3);
70 PRINT SEG$(A$,LEN(A$)-4,5);
80 PRINT SEG$(A$,-1,0)
RUN
HERERE ISESSESSAGESSAGE
* BAD VALUE IN 80
```

In line 70, the SEG$ function establishes the starting point of the substring by figuring the length (LEN) of A$ and subtracting 4 from it. Line 80 illustrates that it is an error to use a negative value for the length or to use a negative value or a zero for the starting location.

SEG$ can also be used as part of a string expression like this:

```
10 S1$="HELLO HI  "
20 S2$="THERE I AM"
30 P$="..?!"
40 PRINT SEG$(S1$,7,3)&&SEG$(
S2$,1,5)&&SEG$(P$,4,1)
```

Try it!

YN

GRAVITY MASTER

Review by Chris Bobbitt

MICROPENDIUM December 84

REPORT CARD	
PERFORMANCE	A
EASE OF USE	B+
DOCUMENTATION	B-
VALUE	A
FINAL GRADE	B+

It is sort of a truism that every computer game comes set up to play right out of the box. For practically every game, once It is loaded, the player simply begins playing. It doesn't take genius or great artistry to set up or play most games. There is, however, a new genre of games being produced which have educators and parents on their ears. These are the interactive games, where the player has the opportunity to create his own game environment, or screen, using a set of predefined figures.

This type of game allows the budding artist in everyone to get a workout, as it is in our own interest to create interesting games for ourselves. It also gets a number of children interested in doing something else with a computer than just play games, namely, create them. About the most well-Known of this type, and, alas, not available for the TI, is Pinball Construction Set from Electronic Arts. Since Electronic Arts didn t think it was reasonable to write their games for the two million TI-99/4As out there, we will have to do without their excellent game.

However, we don't necessarily have to settle for second best. There is another game of this type exclusively for the T199/4A, which is everything the quality program Pinball Construction Set is. CSI, formerly Challenger Software, may not have the big name and reputation of Electronic Arts, but in my humble (and often heard) opinion, their program Gravity Master is the equal of anything Electronic Arts has produced.

Gravity Master actually is sort of ,a hybrid, since it can be played directly after loading. It includes two sample games of different levels of difficulty, each with a number of screens. These two games are merely examples, albeit alone worth the price of the game. The real star of the program is the game screen editor, which allows the player to create games, each with up to 20 screens. Each of the screens can be designed in an infinite number of ways. If the player gets tired of the game, he merely has to design more screens.

The game has a very familiar plot and playing screen. The screen is covered with girders of various types at different levels, ladders connecting some of the girders, and robots moving back and forth menacingly on some of the levels. The player's character, vaguely shaped like a robot, is placed on another area of the screen, safe for the time being.

The game play resembles the play of 2049'er, from Tigervision. The player must move his character over the girders, back and forth, until they are a particular color. Once all the girders have been painted that color, the player moves on to the next screen. The player may jump from one girder level to another, eluding robots, and changing girders to different colors. Some girders are beneficial, allowing the player to destroy robots once these particular girders have been touched. Some disappear once they have been stepped on, destroying a possible means of escape from those terrible enemy robots.

The gravity part of Gravity Master has to do with the jumping one can do. The higher the gravity level of the screen, the more likely it is that the character will be Killed if it jumps off a ledge. This is a game in which the player must balance a complicated set of factors. The object: to survive to see the next level.

Performance: Gravity Master requires a fair amount of hardware to run. The user must have an Extended BASIC cartridge, disK drive system and a memory expansion. Though this may seem to be a lot, it is quite reasonable for a game liKe this. Most, if not all, of the advanced applications and better games today require the ability to access Assembly language routines. BASIC simply doesn't have the power of Assembly. If the disK containing the program is in disK drive one, the program loads and automatically runs when Extended BASIC is selected. Once loaded, the user is faced with a menu containing a number of options.

The first option available to the user, and the focal point of the program, is the Builder, or game screen editor. The editor is extremely user friendly, a term used in conjunction with many programs nowadays.

To create the game screen, the user uses one of two different-sized arrows to picK up objects arrayed in a menu and place them where he wants them to be. The smaller arrow is used to picKup the smaller pieces, such as the various types of girders, while the larger arrow is used to move around the player's character, or the opposing robots. To picK up a piece, the user simply moves the appropriate arrow over the desired object and presses the space bar. To place a copy of the object being carried by the arrow, the user only has to put it over the desired area on the screen and press the space bar again. Once the user has created the desired screen, he can save it to disK.

If the user wishes to test the screen, the next option is just what the doctor ordered: the Tester. This option lets the user try out each screen, to maKe sure all the girders are accessible and in the proper places. The robots are not active, so the user doesn't have to worry about dying. The screens can be edited again with the Builder after being tested.

After the screens have been tested, the LinKer option is used to put them all together in one game. The LinKer can be used to play a screen already in the program's memory, or may be used to create a game out of several screens already saved to disK. Individual screens may be linKed together in any order the user desires.

The program also contains an option which allows the user to catalog disKs, by program files or game screen files, and delete files. I found this to be pretty useful when it comes time to linK together game screens into one game, since I often forgot the filenames I'd given the screens.

The program seems to perform perfectly. The only difficulty I had with it, which I traced backK to my console, was in saving a game screen. I didn't find out it was the console's fault until after I asKed the manufacturer for a new review copy. I am happy to report that CSI has very fast service; I received a guaranteed copy within a weeK after I mailed my letter.

Ease of Use: For the most part, this program gets top marKs for ease of use. The screen editor is extremely easy to use, and the many options are menu-oriented and easy to access. The program is arranged logically, a plus, and worKs very quicKly.

Documentation: This is my only real complaint with this pacKage. Several facets of the program, in the LinKer and Editor, go unexplained. In the review copy a number of crucial things went undocumented; for one, the fact that the user is to use the arrow Keys in the editor.

The manual is not arranged in a logical fashion. You would expect the simplicity of the program would be reflected in the manual, but there is no table of contents, explanations for the parts of the LinKer are not in the order that the program requires them and explanations for use of the Tester and disK utility options are inadequate. The manual is also difficult to read, assuming the user has a lot of Knowledge he may not have, and it assigns acronyms haphazardly. The various sections seem to be arranged rather randomly. To top it off, the manual I received had two page eights and no page seven. (This has been corrected in subsequent manuals, according to CSI.)

The manual isn t without its strengths, however. The explanation of how to play the game is quite good, and the cover graphics are excellent.

Value: An arcade game is an arcade game, you either liKe it or you don't. This game, I believe, has more value than most arcade-type games. The option allowing the user to maKe an infinite number of screens insures infinite variations. The player isn't as apt to get tired of it as quicKly as most games, and the game will always be as challenging as the player wants it to be. Never mind that the actual plot of the game is rather old, the game situations can always be new.

The beginner may find this game to be difficult to use, but once the constructs of the game are familiar, the user is assured of hours of excitement and scores of interested and challenging situations.

YN

Review by Bob Carmany

MICROPENDIUM May 91

REPORT CARD	
PERFORMANCE	C
EASE OF USE	A
DOCUMENTATION	A
VALUE	C
FINAL GRADE	B

High Gravity, written by Tom Wible, is one of the many games that are an exercise in ballistics. By altering the speed and trajectory of an object, you attempt to hit a target with it. Of course, you have gravity to contend

with, hence the name of this program. As you attempt to choose the correct path from your spaceship to the space station, there are anywhere from one to nine planets to contend with on the way. The gravitational pulls vary, so you can forget trying to wind your way through the lot of them on the first try. It becomes an interesting exercise in physics to chart a successful course to the space station.

Performance: The program loaded easily from all the load environments supplied with no surprises along those lines at all. All the commands are simple single Key-presses and easy to remember once you read the documentation (the first step in any program).

The program does exactly what the documentation says it will and I encountered no problems except for the two mentioned in the documentation. The first causes a reverse in motion because of simulations calculations and the second causes a crash in apparently empty space. Since the documentation forewarns you, they can be accepted.

Why the relatively low grade? Despite the promise of a program written in c99, High Gravity has some weaKnesses. The graphics are second-rate at best. The planets appear as colored disKs with flattened ends and the space station (your target) is merely an X. I found it quite a let-down from the promise a c99 program held. In fact, I have seen many Extended BASIC programs with similar ballistic simulations with much superior graphics.

Conversely, options to change velocity and even reconfigure your solar system as you wish largely balance out the primitive graphics. In short, an average grade of C.

Ease of use: The program is easy to use. It doesn't require much preparation and the documentation can be easily read and understood. All the commands are simple, single Keypresses and don't need a lot of study to master. For example, "I" increases the velocity of the probe capsule. When you go to the reconfigure section, the other Keypresses are just as easy to remember (e.g., P for Planets, C for Capsules). In fact, you don't even have to Know what ballistics entails to taKe advantage of this program. You can save the space station without being a physicist. This is one of the major strong points of the program. They don't maKe programs much easier to use than this one.

Documentation: The eight-page booldet that comes with High Gravity is thorough and easy to read. Each command is explained in detail and one section even contains a digression on physics and gravitation for those of you interested in the mechanics of the program. Although it won't pass for a textbooK, this digression will give the user a rudimentary idea of simple ballistic trajectories and how they are affected by gravitational variables. The documentation is well written and concise, containing only a couple of minor typographical errors. In short, it is another of the program's strong points.

Final grade: The fact that the documentation is quite well done and the program easy to use maKes it just a bit above the average commercial program. It isn't as good as some but a good deal better than most efforts I have seen recently. Remember, C is average, and High Gravity is above that. If the graphics didn't detract so much from the performance, it would have been a truly excellent effort. Maybe there is a version 3.0 in the future.

Value: This is a tough category! The real question concerns whether or not the performance of the program justifies the purchase price. Besides the shortcomings in the graphics, I didn't find the program nearly as addictive as the hype in the documentation. The application as a ballistics tutor is limited in interest. The program does have some redeeming value as a pure game if you are willing to put up with the frustration of seeing probe capsule after probe capsule crash unceremoniously into one of the planets. Over all, it doesn't qualify as an exceptional value. It is just about average.

The program is a welcome respite from the "shoot-em-up" space game programs usually seen. High Gravity isn't a steal at $14.95, but you probably won't regret spending your money to buy it!

YN

FOUNDATION 128K MEMORY EXPANSION

REPORT CARD	
PERFORMANCE	A
EASE OF USE	A
DOCUMENTATION	B+
VALUE	B
FINAL GRADE	A-

MICROPENDIUM January 1985 By John Koloen

Foundation Computing's 128K memory expansion card is a product that has remained unique in the TI marKetplace. The card has been out for more than a year, and still there are no imitators.

Since introducing the card, Foundation has made a significant improvement to it by marKeting DisK File Emulator firmware that allows users to access the extra memory in a straight-forward manner. Prior to the availability of the DisK File Emulator several months ago, the card was of greatest use to those who used it in programming and could write programs to access the cards 128K of Random Access Memory. The DisK File Emulator chip option changed all that, maKing three 32K memory banKs easily accessible to even casual users.

Purchasers may order the 128K card without the emulator firmware, but I do not Know why anyone would want to. For an extra $35 or so the company will ship the card with firmware implanted, and I recommend the firmware option to

any would-be purchasers. It is the brains of the outfit, so to speaK.

Performance: With the exception of the activity light in the front the card (which is green), once it is installed in the PEB a user is not liKely to notice any difference between the Foundation card and the standard TI issue. The card itself is as sturdy as a TI card and fits the PEB liKe a glove.

The card actually consists of four memory banKs. The lower one functions in the same manner as the standard 32K of RAM found in the TI card, and the other three banKs have 32, 32 and 24 Kilobytes of RAM, respectively. 8K of the last banK is used for the disK emulator software.

The upper three memory banKs provide the user with potential not available in any 32K card. Although it is possible to address these banKs through user-written Assembly language routines, most users will probably prefer to have the banK switching and accessing done via the DisK File Emulator.

The emulator is the heart of the card, as far as I am concerned. Without it, one would not be able to access the three upper memory banKs through such cartridges as TI-Writer and Microsoft Multiplan.

Basically, the emulator allows the user to access the upper three memory banKs in much the same way as one would access a disK drive. Only the names are changed. Instead of entering DSKI to represent disK drive 1, the user enters DSKX to access a memory banK. (Prior to using the expansion memory, it is necessary to enter DELETE MEMINIT in BASIC. This initializes the expansion memory as a psuedo device. There is another command, DELETE MMM (Memory Management Module) that functions as a catalog/manager for the three upper memory banKs. When this command is entered, a display reports the name of the file or program in each banK, the type of file and its size in bytes. (Each banK also has a name: MEM96A, MEM96B and MEM96C.)

Through MMM, the user may delete or rename files or programs or clear all three banKs simultaneously. The command options are displayed at the bottom of the screen when DELETE MMM is used. MMM is accessed through BASIC.

The three memory banKs are very useful with programs such as TI-Writer. One can write files to a memory banK and read them in one-tenth the time or less that it taKes to do the same operation with a disK drive. With Extended BASIC, programs can be chained through use of a RUN DSKX.(filename) command. With Terminal Emulator II, entire screens can be sent to a memory banK through use of the output option. When finished receiving data one simply closes the file through the TEll close command.

The 128K card will also facilitate the use of Multiplan files with TI-Writer.

Programmers may incorporate the memory banKs in programs via the same input/output routines used with BASIC and Extended BASIC.

Without the DisK File Emulator, the memory beyond the initial 32K is considered to be a single large file for storage of relative records only. The firmware allows the memory to be accessed as three pseudo-devices that may be opened as relative or sequential files.

Concerning file sizes, I found that the memory banKs quite adequately handle very lengthy files generated by TI-Writer and other programs. However, I was unable to retrieve programs longer than about 12K from the memory banKs. This apparently has to do with how the TI system interprets lengthy programs. Programs longer than about 12K are not registered as programs by are stored as 256 byte internal/variable files. It seems that the Foundation memory banKs do not treat long programs as programs per se. Although few of my BASIC programs are longer than 12K, I have some financial-oriented programs that incorporate many functions and taKe up about 60 or so sectors on a disK. Although I was able to load these into a memory banK I could not get them to run after trying to retrieve them from the memory banK.

As a point of clarification, the use of psuedo-devices is Foundation s way of circumventing the limitations built into the TI99/4A. DSKX is a psuedo-device that fools the computer into treating the memory banKs as I/O devices. MEM96A, B and C are psuedo-devices that are accessed in conjunction with DSKX. MMM, another psuedo-device, can be thought of as a very specialized version of a disK manager cartridge. MEMINIT is a reset switch that deletes all files from DSKX and initializes internal variables.

The DELETE command is used to access MMM and MEMINIT from BASIC. It does not actually delete anything. Foundation used the DELETE command because it is the only I/O command that does not include error checKing. It is used simply to transfer control to MMM.

It is not possible to load the MMM through a program such as TI-Writer to review the contents of the three banKs of expansion memory.

Ease of Use: The 128K card with accompanying firmware is easy to use as a RAM disK. Writing and reading files requires virtually no adjustment for even casual users. About the only caution that I would recommend to users is to not turn the console off until the data stored in the card's memory has been written to disK. Turning the console off causes the files to degrade.

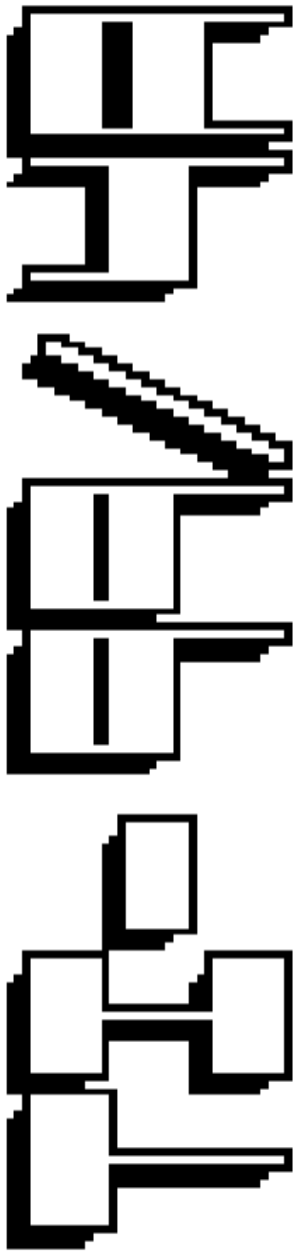
Documentation: The 128K card comes with a 12-page manual about the DisK File Emulator and a six-page manual about the 128K card itself. Both manuals go into some detail that will give users a good idea of the potential of the card for a variety of applications. However, most users will probably do a lot of experimenting to learn exactly what they can and can't do with the card.

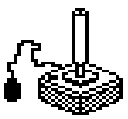
Value: The 128K card sells for about $230, plus about $35 for the DisK File Emulator firmware. The card is about double the price of third-party 32K cards, and about the same price as TI's 32K memory expansion card when it was still being marKeted.

The only program that I Know of that actually maximizes the use of the 128K card is the TIBBS program developed by Ralph Fowler. However, I Know of one programmer who has developed a program using Forth that uses the 128K card to copy disKettes in a single pass. This may become available in the near future. Other uses for the card depend more on the ingenuity of the user than third-party marKeters. I used it with several terminal emulator programs and found the card a real time saver in terms of dumping data. There are several applications for the card in data transmission. Those who use modems extensively will find the ability to store up to three files in memory, a time-saver while sending data. It is also quite convenient for receiving data, which can later be written to disK.

Getting down to the bottom line, this remains the only 128K card available for the TI. Using the memory banKs as a psuedo disK drive, programmers can speed up software development. It provides users with a speedy way of temporarily storing data and programs.

```
1 !VICIOUS CIRCLE
10 @=1 :: _=2 :: CALL CLEAR :: CALL SCREEN(_):: RANDOMIZE :: CALL MAGNIFY(3):: FOR A=@ TO 14 :: CALL COLOR(A,16,@):: NEXT A :: CALL COLOR(11,11,@):: CALL CHAR(48,"007C44444444447C")
20 CALL CHAR(96,"070B1321418 1E3FFFFE3814121130B07E0D0C88 48281C7FFFFC7818284C8D0E")
30 CALL CHAR(100,"030C302040 4080808080404020300C03C0300C 0402020101010102020040C30C",108,"007E7E7E7E7E7E001C2A497F492A1C")
40 CALL CHAR(112,"8080808080 8080800101010101010101FF0000 00000000000000000000000000FF") :: DISPLAY AT(@,8):"VICIOUS CIRCLE"
50 DISPLAY AT(4,@):"AVOID TH E CIRCLES WHILE":"CLEARING T HE GRID.": :"USE THE JOYSTIC K OR ARROW KEYS TO MOVE."
60 DISPLAY AT(10,@):"YOU REC EIVE 10 PTS FOR EACH SQUARE, OR 1000 PTS FOR AN ENTIRE GRID.": :"ONCE YOU HAVE BEEN HIT 10 TIMES, THE GAME WI LL END."
70 DISPLAY AT(17,@):"FOR EVE RY 5000 PTS, YOU":"GET AN EX TRA LIFE." :: DISPLAY AT(23,7):"JOYSTICKS?(Y/N)"
80 CALL KEY([,B,C):: IF C=[ THEN 80
90 D=[ :: IF B=89 OR B=121 T HEN D=@ :: IF B=121 THEN 120
100 IF D=[ THEN 120
110 FOR A=@ TO 4 :: DISPLAY AT(23,4):"RELEASE ALPHA-LOCK KEY" :: FOR E=@ TO 40 :: NEXT E :: DISPLAY AT(23,4):: NEXT A
120 CALL CLEAR
130 F=85 :: G=117 :: H,I,J=[ :: K=10 :: L=5000 :: CALL HCHAR(_,9,115,17):: CALL HCHAR(20,9,114,17):: CALL VCHAR(3,8,113,17):: CALL VCHAR(3,26,112,17)
140 DISPLAY AT(@,_):"00000"
150 FOR A=4 TO 18 STEP _ :: DISPLAY AT(A,8)SIZE(16):RPT$("1 ",8):: NEXT A :: FOR A=_ TO 9 :: IF A/_=INT(A/_)THEN M=-@ ELSE M=@
160 M=M*INT(RND*10+12+J):: CALL SPRITE(#A,100,INT(RND*14+3),200,A*16+37,M,[):: NEXT A :: GOSUB 330
170 DISPLAY AT(22,9):"PRESS ANY KEY" :: DISPLAY AT(22,9) :: CALL KEY([,N,O):: CALL KEY(@,P,Q):: IF O=[ AND Q=[ THEN 170 ELSE CALL SOUND(500,262,3,330,3,392,3)
180 CALL SPRITE(#@,96,15,F,G ):: GOTO 210
190 IF D=[ THEN 300 ELSE CAL L JOYST(@,B,C):: IF ABS(B)=A BS(C)THEN 230
200 G=MIN(181,MAX(69,G+B*4)) :: F=MIN(133,MAX(21,F-C*4)): : CALL LOCATE(#@,F,G)
210 CALL GCHAR(INT(F/8)+_,IN T(G/8+_),R):: IF R<>108 THEN 230 ELSE CALL SOUND(140,-6,3,900,4,1100,5,1300,6)
220 CALL HCHAR(INT(F/8+_),IN T(G/8+@),32,_):: I=I+@ :: IF I=64 THEN 240
230 CALL COINC(ALL,R):: IF R =[ THEN 190 ELSE CALL SOUND( 200,-6,_):: K=K-@ :: GOSUB 3 30 :: IF K=[ THEN 260 ELSE 1 90
240 CALL SOUND(1600,131,_,39 2,_,1047,_):: J=J+_ :: CALL DELSPRITE(ALL):: I=[ :: F=85 :: G=117 :: H=H+1000 :: DIS PLAY AT(@,7-LEN(STR$(H)))SIZ E(6):STR$(H)
250 IF H=L THEN K=K+@ :: GOS UB 330 :: L=L+500 :: GOTO 15 0 ELSE 150
260 FOR A=@ TO I :: H=H+10 : : CALL SOUND(30,523,_):: DIS PLAY AT(@,7-LEN(STR$(H))):ST R$(H):: CALL SOUND(20,200,30 ):: NEXT A :: DISPLAY AT(22, 11):"GAME OVER" :: FOR A=@ TO 340 :: NEXT A
270 DISPLAY AT(22,8):"PLAY A GAIN?(Y/N)"
280 CALL KEY([,B,C):: IF C=[ THEN 280
290 IF B=89 OR B=121 THEN CA LL DELSPRITE(ALL):: CALL CLE AR :: GOTO 130 ELSE END
300 CALL KEY([,N,O):: B,C=[ :: IF N=83 OR N=115 THEN B=-4 ELSE IF N=68 OR N=100 THEN B=4
310 IF N=69 OR N=101 THEN C=4 ELSE IF N=88 OR N=120 THEN C=-4
320 GOTO 200
330 DISPLAY AT(@,16):RPT$(" ",13-K)&RPT$("m",K):: RETURN
```





INTERNATIONAL FUN & GAMES

GAME TITLE	SCORE	JOYSTICK JOCKEY	TI CLUB	DATE
BACKSTEINE	155900	STEVEN JAKABFY	OSHTI UG	09/95
BIGFOOT	290500	DAVID HANDLE	OZARK 99	01/95
BLASTO	44880	MIKE CENDROWSKI	W/PENN 99	11/94
BREAKTHROUGH	1850	RAY FRANTZ	VAST	11/93
BURGER BUILDR	1000000	ELEANOR ZIC	W/PENN 99	03/94
BURGERTIME	82600	MICKEY CENDROWSKI	W/PENN 99	09/85
CAR WARS	6050	JIM WAYNE	VAST	11/93
CENTIPEDE	301930	MICKEY CENDROWSKI	W/PENN 99	01/87
COLORS	1000000	HARRY HOFFMAN	CLEVELAND	03/95
COMBAT	750	AIRSHACK	VAST	02/19
DIG DUG	262460	FRANK ZIC	W/PENN 99	03/94
ENTRAPMENT	3668	FRANK ZIC	W/PENN 99	11/93
HOPPER	4031826	TOM BEERSMAN	OZARK 99	06/94
HUSTLE	WON 52	ELEANOR ZIC	W/PENN 99	03/94
JAWBREAKER	15025	JIM WAYNE	VAST	11/93
JUMPY	131900	ELEANOR ZIC	W/PENN 99	03/94
MICRO PINBALL	1776500	NORM ROKKE	W/PENN 99	05/87
MIONITE MASON	27100	FRANK ZIC	W/PENN 99	11/93
MOON PATROL	73150	MIKE SEALY	W/PENN 99	03/94
MUNCHMAN	202170	PAUL BROCK SR.	W/PENN 99	09/87
PACMAN	153000	GARY TAYLOR	W/PENN 99	09/87
PARSEC	47300	MICKEY CENDROWSKI	W/PENN 99	09/87
PKR SOLITAIRE	3790	JACKIE REMENSKI	VAST	11/93
POLE POSITION	57700	MICKEY CENDROWSKI	W/PENN 99	12/94
SUPER YAHTZEE	615	JACKIE REES	VAST	11/93
THE ATTACK	31800	JIM WAYNE	VAST	11/93
TI INVADERS	15930	PAUL BROCK SR.	W/PENN 99	09/87
TI TRIS	2208	FRANK ZIC	W/PENN 99	11/93
TOMBSTNE CITY	154400	DANNY MCGUIRE	OZARK 99	11/94
TRN SOLITAIRE	351	CAROL HOFFMAN	CLEVELAND	03/95
TREASURE ISLE	37800	MIKE CENDROWSKI	W/PENN 99	10/94
TRIS (ASGARD)	8393	MICKEY CENDROWSKI	W/PENN 99	12/94
YOUR GAME	0000000	YOUR NAME	GROUP?	00/00
YOUR GAME	0000000	YOUR HANDLE	STATE?	00/00
YOUR GAME	0000000	YOUR NAME	COUNTRY?	00/00
YOUR GAME	0000000	YOUR HANDLE	GROUP?	00/00
YOUR GAME	0000000	YOUR NAME	STATE?	00/00
YOUR GAME	0000000	YOUR HANDLE	COUNTRY?	00/00
YOUR GAME	0000000	YOUR NAME	GROUP?	00/00
YOUR GAME	0000000	YOUR HANDLE	STATE?	00/00
YOUR GAME	0000000	YOUR NAME	COUNTRY?	00/00
YOUR GAME	0000000	YOUR HANDLE	GROUP?	00/00
YOUR GAME	0000000	YOUR NAME	STATE?	00/00
YOUR GAME	0000000	YOUR HANDLE	COUNTRY?	00/00
YOUR GAME	0000000	YOUR NAME	GROUP?	00/00
YOUR GAME	0000000	YOUR HANDLE	STATE?	00/00

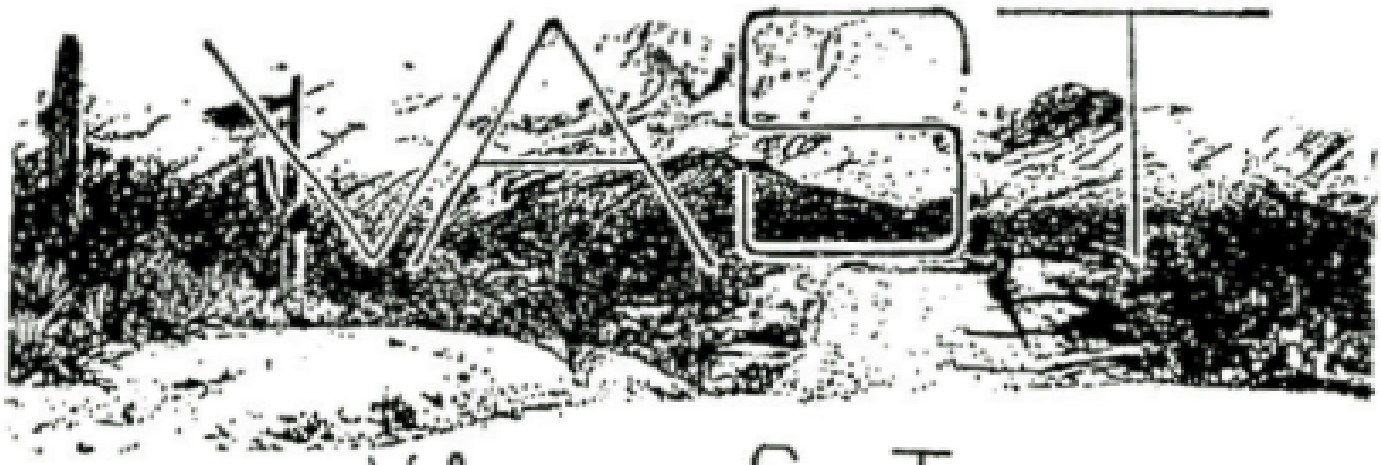
BOLD LINES INDICATE NEW HIGH SCORE OR GAME SUBMITTED

Please submit all scores to SPARKDRUMMER via private message on the ATARIAGE TI-99/4A forum.

THE

DIFFERENCE

IS



VAlley of the Sun TI-99/4A

USER GROUP

INVITES

YOU

and all Personal Computer owners and users to Join them the 2nd Saturday Morning of each month between 9:00am and 12:00pm (noon) at the PHLE ADULT RECREATION CENTER in Tempe.

VAST OFFERS:

. 24 hour a day BBS

. Monthly Newsletter

. Download from BBS for members

. Computer "Hands On" demonstrations

. Loan of a MODEM System for members

. A library of programs for disc and cassette

. A friendly forum for discussion of your PC problems

For more Information on VAST, contact either:

Mike Groppe (Pres)...272-4315 or Mike Marfisi (Sec/Treas)...491-1552

Yesterday's News Information

Yesterday's News is a labor of love offered as a source of pleasure & information for users of the TI-99/4A and Myarc 9640 computers.

TI-99/4A HARDWARE

TI99/4A COMPUTER
MODIFIED PEB
WHT SCSI AND SCSI2SD
MYARC DSQD FDC
MYARC 512K MEMORY
HORIZON 1.5 MEG HRD
TI RS232
CORCOMP TRIPLE TECH
1 360K 5.25 DRIVE
1 360K 3.50 DRIVE
1 720K 5.25 DRIVE
1 720K 3.50 DRIVE

TI-99/4A SOFTWARE

PAGEPRO 99
PAGEPRO COMPOSER
PAGEPRO FX
PAGEPRO HEADLINER
PAGEPRO GOFER
PAGEPRO FLIPPER
PAGEPRO ROTATION
PIXPRO
PICASSO PUBLISHER
BIG TYPE
TI ARTIST PLUS
GIF MANIA

PC HARDWARE

COMPAQ ARMADA 7800
COMPAQ ARMADASTATION
SAMSUNG SYNCMASTER

PC SOFTWARE

DEAD WINDOWS 98SE
FILECAP
PRN2PBNS
IRFANVIEW
ADOBE DISTILLER
ADOBE ACROBAT

Yesterday's News is composed entirely using a TI-99/4A computer system. It consists of 11 PagePro pages which are "printed" via RS232 to PC to be published as a PDF file.