



NEWS DIGEST

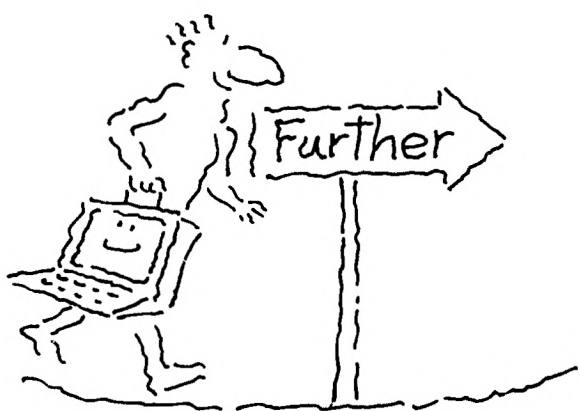
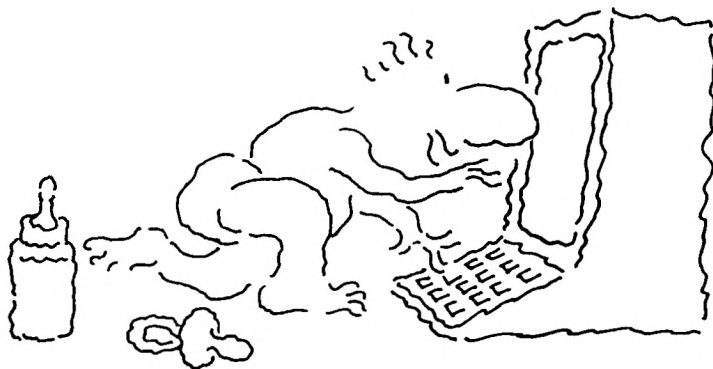
Focusing on the TI99/4A Home Computer

Volume 9, Number 7

August, 1990

Registered by Australia Post - Publication No. NBH5933

The one to
start with



is the one to
stay with



TI99/4A Home Computer

P.O. Box 214, Redfern, New South Wales, Australia, 2016

\$3

TIsHUG News Digest

Index

August 1990

All correspondence to:

P.O. Box 214
Redfern, NSW 2016
Australia

The Board**Co-ordinator**

Dick Warburton (02) 918 8132

Secretary

Terry Phillips (02) 797 6313

Treasurer

Geoff Trott (042) 29 6629

Directors

Rolf Schreiber (042) 84 2980

Russell Welham (043) 92 4000

Sub-committees**News Digest Editor**

Bob Relyea (046) 57 1253

BBS Sysop

Ross Mudie (02) 456 2122

BBS telephone number (02) 456 4606

Merchandising

Percy Harrison (02) 808 3181

Publications Librarian

Warren Welham (043) 92 4000

Software library

Rolf Schreiber (042) 84 2980

Technical co-ordinator

Lou Amadio (042) 28 4906

Regional Group Contacts**Carlingford**

Chris Buttner (02) 871 7753

Central Coast

Russell Welham (043) 92 4000

Coffs Harbour

Kevin Cox (066) 53 2649

Glebe

Mike Slattery (02) 692 0559

Illawarra

Lou Amadio (042) 28 4906

Liverpool

Larry Saunders (02) 644 7377

Northern Suburbs

Dennis Norman (02) 452 3920

Sutherland

Peter Young (02) 528 8775

Membership and Subscriptions

Annual Family Dues \$30.00

Overseas Airmail Dues A\$60.00

Overseas Surface Mail Dues A\$45.00

TIsHUG Sydney Meeting

The next meeting will start at 2.00 pm on 4th of August at Ryde Infant School, Tucker Street, Ryde.

Printed by

The University of Wollongong
Printery

Title	Description	Author	Page No.
4-D graphics	Software hints	Peterson, Jim	22
Assembly squeezing, part 1	Software hints	McGovern, Tony	22
Assembly squish	Software hints	McGovern, Tony	21
Boot, bug fix	Software hints	Schreiber, Rolf	6
Bulletin board users	General interest	Shaw, Stephen	9
Cataloguing with multiplan	Software hints	Relyea, Bob	19
Co-ordinators report	General news	Warburton, Dick	2
Communicators	BBS information	Mudie, Ross	9
Cracked	Software hints	Jack, Cracker	20
Cutthroats part 1	Adventure hints	Scorpi	10
Disk reviews	Software reviews	Shaw, Stephen	14
Editor's comment	General interest	Relyea, Bob	1
Games modules	Software reviews	Shaw, Stephen	13
Hard disk setting up	Software hints	Wilkinson, Derek	8
Letter to editor	Future of TIsHUG	Amadio, Lou	3
Living with spiders, part 1	Software hints	McGovern, Tony	17
Program information	BBS information	Mudie, Ross	9
Rambles	Software hints	Shaw, Stephen	11
Regional group reports	General interest		23
Secretary's notebook	Club news	Phillips, Terry	4
Techo time	Hardware hints	Amadio, Lou	7
TI-Writer once again	Software hints	McGovern, Tony	15
TIsHUG shop report	Club news	Harrison, Percy	6
TIsHUG software column	Club software	Schreiber, Rolf	5
Treasurer's report	Club news	Trott, Geoff	8
Younger set	Program	Maker, Vincent	14

TIsHUG Fairware Author of the Month

The Fairware Author for this month is Jim Peterson for all his Tips from the Tigercub. The shop has disks of all his software and disks of his public domain programs for sale. All Donations collected at the meeting and sent in will be mailed to him.

Kits now available

The shop has kits of parts for the following:
AT multifunction card disk controller - \$102.50
RGB colour monitor interface - \$21
MIDI/E Forti ... card, including PCB - \$65

Editor's Comment

by Bob Relyea

This has been a very quiet month (June) as far as editing is concerned. If it were not for Vincent Maker with the Younger Set and Rolf Schreiber, who gave me a disk of gear from Tony McGovern, I would not have had much to do apart from the regular contributions, such as the Co-ordinator's Report and the Meeting Summary. Come on gang! Where are those articles? Even if you just type up something interesting that you have seen in an overseas magazine or tell us about what you are doing with your computer at the moment, we want to hear about it. Anybody done anything interesting with their computer lately? I have heard members talk about closing down the BBS because of lack of support at the moment. I would not want to see that happen. Get those articles rolling in and not leave it up to a select few. Everybody can make a contribution even if you are just using a cassette. Hope to hear from you soon.

Co-ordinator's Report

by Dick Warburton

Last month I wrote about the uniqueness of Tishug and the advances which have been made over the past 10 years. I emphasised the qualities which differentiate TI users from other club members, particularly our Independence and Self Reliance. This month I wish to discuss our club's future.

There are, without doubt, some quite exciting hardware projects ahead. I expect more ramdisks to be made, and eproms on the ramdisk will soon be commonplace. There may be a cheaper hard disk controller available, and consequently more hard disk drives in use. Certainly more TI users will be the proud owners of multifunction double density drive controllers. With the use of such good monitors, 80 column cards become a viable alternative. I have no doubt that there is plenty more yet to come. To achieve our goals with the TI, we need to maintain sufficient membership to finance our magazine, and to help us stay solvent through the club shop. Recently a group of interested TIers came together to think about our club's future. Two main topics were discussed at length:

1. How we can maintain our present level of club activity, and improve our club functioning.
2. Opening club membership to other computer users.

It was generally agreed that we must act this year to determine our future, and that any actions taken should not threaten the basic stability of Tishug. Many suggestions were made which aimed at improving club functioning and keeping our existing membership while attracting new members. The major points made were:

* The most obvious deficiency in present club functioning, is in the provision of software. It is true that little or no new software is being written locally, and only limited access to overseas writers is available. It is acknowledged that the directors have tried to improve this situation, but changes are happening slowly. Only a small amount of overseas commercial software is available. A further criticism is that insufficient software is being demonstrated at the meetings. Members are unaware of what is available, and whether it is what they want. Software should be promoted more professionally through the club, particularly at club meetings. These points were clearly endorsed by the members who attended the July meeting. Members must know what is available and want to see it demonstrated. Hopefully this will improve at future meetings. To do this we will need more expanded systems at the meetings. Bring your system with you. The more computers available, the greater the variety of things we can do and demonstrate.

* A second area of weakness relates to the general lack of provision for newer members, who may need help in more basic areas. We need to make help available, showing all members how to get the best out of their computers, and the software they use. Perhaps we need to revise some of the more common software in use. Some members find our monthly meetings too technical, and a number of our T.N.D. articles hard to read. Our News Digest does not cater well for novice users, but this can be addressed. It was further suggested that we should ask members clearly what they want in software, and we should try to meet those needs.

In summary, the needs of the hardware buffs are being met but not the needs of the predominately software users.

The major issue of attracting new members was discussed, and the following suggestions were made:

-We could offer incentives for new members to join the club.

-We could reward members who introduce a new member to the club.

-We could make a serious effort to publicize the club's activities in a number of ways, for example:

1. Taking advantage of shopping centre notice boards.
2. Setting up stalls on market days in the Ryde area.
3. Putting notices in computer dealers' windows.
4. Advertising club meetings in computer magazines or the Trading Post.
5. Writing feature articles for newspaper publication.
6. Ringing anyone who advertises a TI for sale, advising them of the club and how it can help them or the person they sell to.

There is no doubt that there are many other things we can do to improve our situation, and we can all help. It was generally agreed that we should make a positive effort to grow stronger, rather than simply let our club wind down.

The second discussion point naturally follows on. As Ben Takash points out so clearly, the TI only has a limited life left for most users. Some members (like myself) will hang on to the TI, because of their investment, and because of their enjoyment and challenge with this machine. Others will choose another computer which needs their needs for updating and better performance. Already at least 70% of Tishug members own or use another computer, mainly IBM clones, but they include all types. Ben made a very strong point when he indicated that there are other computer orphans out there with no one to care for them. Other clubs do not provide the support or help our members get in Tishug. We could adopt the orphans. While most Tishug members own other computers, they still attend Tishug meetings, because of social needs, to buy accessories or peripherals, or simply to find out what is new. It was suggested that Tishug should gradually change its membership base, by opening its membership to the owners of ALL COMPUTERS. In time the club should become a computer club, not simply a TI club. Ben sees us as hackers, with a genuine and abiding interest in computers generally. John Paine points out that new members with diverse backgrounds would bring fresh ideas both in software and hardware, which could be adapted to the TI. One might even see a competitive spirit emerge, with upgraded TI systems. There would most likely be a resurgence of club activity and enthusiasm. It was not proposed that the club should be swamped by IBM users, but rather that memberships could be extended to a limited number of interested people who have something to contribute to the club. Outside membership could be limited to those people introduced by Tishug members. There is much which is common to computer usage. Most computers use disk drives or hard drives. Accessories and peripherals are often the same. Basic principles apply across the board. A computer game is as much a game on the TI as on the Amiga. A wider membership base would probably give us a wider contact with the computer industry, and the people we need to complete our projects, e.g. burning eproms. There was general agreement that this issue must be faced in the near future, and clearly a majority of members surveyed at the July meeting agreed that we should move cautiously in this direction. It is proposed that we should determine this matter at either an Extraordinary General Meeting, or at the Annual General Meeting in 1991, and if carried, adopt a strategy to implement the changeover. Those who discussed these matters feel that:

-It is only a matter of time (a short time!) before we will be forced to make the move, and that we should act early rather than later.

-There are a significant number of computer users out there who would feel very much at home in Tishug.

continued on page 6

Letter to the Editor

Does TISHUG have a Future?

The discussion at the July meeting on whether TISHUG should be opened up to computers other than the TI99/4A had plenty of arguments both for and against. Some believe that the future of TISHUG can only be secured by opening up the club to cater for other types of computers.

TISHUG is currently faced with falling membership and, consequently, rising costs. The end result is that the club will fold once our cash and assets are run down.

TISHUG is unique among computer clubs in providing a top class magazine, excellent shop facilities, a vast range of inexpensive software, hardware support and, equally important, invaluable social contact. However, the unique attributes of this club were brought about by a combination of events, including the orphaning of this computer by Texas Instruments in late 1983. TI99/4A users were unwilling to let their computer die and, in conjunction with third party hardware and software suppliers, worked towards securing the future. The result was a combination of hardware and software expertise that not even TI ever dreamed of. This combination of people and events will probably never happen again in the history of computing.

What we are trying to do at TISHUG is to prolong that dream.

Last year there was talk of a membership drive but, to date, there has not been much activity in this area. We understand that TISHUG did not even apply for a listing in the telephone directory as was agreed previously.

Would anyone purchase a 10 year old computer and join our club anyway? Who is the typical TI99/4A user? Look around you at your regional or TISHUG meetings to answer that question. It is something like the chicken and egg situation. You need to know how good the TI99/4A is before you buy it; but you need to buy it to find out how good it is. We do not have the luxury of reviews in commercial magazines to promote our product.

So, is the club slowly dying? Can we do anything about it? Perhaps we could look at what has happened with other user groups for some guidance.

There has been some talk in the overseas (USA) newsletters about the problems facing TI99/4A user groups. Some groups have tried to keep members by catering for other types of computer but it appears that this has not yet worked. There was an article by Jack Sughrue (I think) which was quite scathing in its condemnation of this approach. There is no evidence in the newsletters that we get, that groups are surviving better by following this path. Other types of computers are mentioned but I have not seen an article based on anything other than the TI99/4A or Geneve. Of course other computers are used by some (including us) to do jobs like prepare the newsletter or hold the membership list. Perhaps we can succeed where others have failed, but if we are to do that, we need to consider and plan carefully how this would all work together or we will only achieve a speedier decline, which is the opposite of the desired effect.

We are basically seeking to maintain our membership numbers in the face of a natural decline due to loss of interest or change of interest to another type of computer. Once members lose interest they will be lost to the group for ever. If members change to a different type of computer they will first look for a user group which caters for that type of computer. From our experience (limited to third hand reports), these groups are not very interesting or very organised or very friendly. These members then would like a user group

like the one they have grown up with and grown to like. We see nothing wrong with these members remaining with this group and using the facilities like the shop and even the centralized mailing facilities of the BBS and perhaps forming their own special interest group; but not at the expense of the members who still are with the TI99/4A. That means that they should not expect that the people who devote their time, free of charge, to the running of the group should have to shoulder additional burdens because of these extra activities not related to the TI99/4A.

After all, it is most unlikely that a hybrid (or heinz) group would be able to attract members from outside. Any individuals who did join would have difficulty in understanding why sane and sensible people could still want to use a computer that is 10(?) years old. Most computers have a lifetime of about 5 years. How many people would still have an original PC type of computer I wonder? The interesting thing about the TI99/4A is that it still provides more facilities than most of the currently available computers (speech, sound, colour, sprites) at a very friendly entry price. And in some form or other most of the popular languages and packages are available. If we did manage to attract a large number of users of other types of computers, they would be able to use their numbers to change the orientation of the club and squeeze out the TI99/4A users. If we were not able to attract members of this type then the decline in numbers is inevitable so we should just learn to live within our income. It reminds me a bit of the current problem in the financial world. An analyst comments that certain institutions may be a problem and the subsequent rush by depositors to remove their funds ensures that they are then in deep trouble. If we keep saying that we are in trouble then we will all believe it and rush out to buy another type of computer.

The members of the Illawarra Regional Group believe that the club can function for several more years, but for this to happen, we must cut costs. One of our biggest costs is maintaining TISHUG (Australia) Ltd (the company) for approximately \$700 p.a. The company was formed to protect individual club members from legal action. The probability of that happening now is very remote. We just do not need this millstone around our necks dragging us down.

What are our options with respect to TISHUG (Australia) Ltd?

- 1) Sell the company (less assets) to another user group better able to use the benefits of incorporation.
- 2) Legally wind the company down. This is very expensive and would probably lead to a sudden death for this club.
- 3) Let the company die. The directors would have to resign. Would we lose all of our assets?

It seems that option (1) is the most viable. It would relieve us of a financial burden and allow the club to continue as a user group.

The present membership (approximately 150) can sustain the magazine at its current quality and size.

We in the Illawarra area believe that any attempt to prolong the life of the club by diversification is doomed to fail and that the losers will be the dedicated TI99/4A users in TISHUG. If this happens, the Illawarra Regional Group will cease to be affiliated with the Sydney group.

Many of us own and use other brands of computers besides the TI99/4A but we do not see any advantage in promoting these within TISHUG.

Some Problems with Diversification

Sharing of limited resources means that TI99/4A coverage and support will decrease.

TI99/4A members will soon become the minority group and lose control of TISHUG's future.

Those members that are currently giving of their time for TISHUG are dedicated TI99/4A users. These members may not be interested in working in and for a diversified club. Who would then fill this void?

Who would produce and print the club's magazine?

Who would develop hardware and repair faulty equipment?

Who would look after the shop?

Who would maintain the BBS?

In answering these questions you must think from the point of view of a club which may be supporting a number of different brands of computers.

In addition, club fees and costs will need to be examined. How would the money be budgeted amongst the various brands of computers? Will bona fide TI99/4A users be supporting other computers? Should these SIGs pay their own way? This could be an accounting nightmare and not the sort of thing for volunteers to get involved in.

In summary, we believe that TISHUG has a healthy future and that this future lies with the TI99/4A. There is still plenty of software and hardware development going on, especially in the USA where the challenge of extracting more performance out of our home computer has encouraged companies such as Asgard, Inscebot, Myarc, Texaments, CorComp and MICROpendium to continue in their exclusive support of the TI99/4A.

History has shown that we are capable of surviving on our own. We must, however, make some important decisions on the future of the incorporated company "TISHUG (Australia) Ltd".

This is what we believe is the best course of action for TISHUG:

- 1) Sell TISHUG (Australia) Ltd to a user group better able to use the legal protection offered by incorporation and saving us about \$700 p.a.
- 2) Actively promote membership. There were about 20,000 TI99/4As sold in Australia. Surely we can attract 1% of these (200).
- 3) Provide for users needs as best we can, for example affordable disk based expansion including hard disks, 80 columns with colour monitors, memory expansion beyond 32K, better user written software and access to the latest commercial software through our shop.
- 4) Seek amalgamation with the Hunter Valley group. Our combined newsletter would be unbeatable - synergy in action!

With 200 active and supportive members we can survive for another 7 years. That is the challenge.

If some TISHUG members feel strongly about supporting an Amiga or PC group, for example, they should set about starting their own club. We cannot see any advantage to TISHUG and the TI99/4A in sharing our club. These members should also be willing to be pro-active in their new club to ensure its success.

How do the other TISHUG regional groups feel about these proposals?

Lou Amadio
Rolf Schreiber
Geoff Trott
George Meldrum
Illawarra Regional Group

Secretary's Notebook

by Terry Phillips

Members are reminded that at the August meeting several proposed changes to our Articles of Association shall be put to the vote. If possible make sure that you are there to register your vote on these important changes. A full description of the proposed changes was published in the July edition of the TND.

Also at the August meeting there will be demonstrations of Funnelweb with 80 column card and a demonstration of a revised faster Multiplan.

Despite my optimistic hopes for membership renewals, they have not drifted in as fast as I would have liked. In fact we have 42 members, whose membership expired either at the end of April or May, who have, for various reasons, not renewed. All of these members have been contacted but it would seem we have lost the majority of them. Total membership now stands at 164, with the majority of these paid up for a full 12 months.

On the credit side we are still attracting enquiries for membership, and indeed even new members. The latest new member to give a warm welcome to is Mr C J Bagley of Algester QLD. Welcome to TISHUG, Mr Bagley.

Ross Mudie's latest report on BBS operation for the month of June sadly reflects the drop of in usage of this facility. During that month only 91 calls were recorded, and 22 of these were by the SYSOP, while another 12 were recorded from VISITOR. In fact only 24 BBS subscribers out of a current membership of 58 used the BBS during June. Has the re-location of the service affected its usage? Is it now too costly to call bearing in mind that it is STD rates for many members? On current information, the BBS will attract somewhere in the vicinity of \$300 in access fees this year, while its operating expenditure on phone rental and electricity will be around \$280. This will be OK provided there are no major equipment failures that will require replacement.

That is it for this month. Hope to see as many as possible at the August meeting. o

continued from page 9

(if you play back). The magazine also mentioned the verb "flame" which refers to a simple innocent comment generating a massive overkill in verbal response - a destruction (verbally) of the poor offender. This form of somewhat anti-social behaviour is apparently common on bulletin boards - possibly as a result of the rather tenuous link between individuals.

I have commented lightly on some rather yucky comments made in various user group newsletters in the past, and checking back I find that those who have lashed out have in ALMOST every case been heavy bulletin board users. So perhaps using a bulletin board creates conditions liable to generate undue touchiness and once generated this form of behaviour spills out into the "real" world... perhaps bulletin boards should carry health warnings....

There is an interesting correlation - the nicest, friendliest, helpfulest TI users I have come across have absolutely nothing to do with bulletin boards.

Before I get flamed may I add that I am not knocking bulletin boards merely, commenting on something which IS happening, and your comments are welcome, especially if you use bulletin boards and have been flamed - or have flamed someone else (did you realise how severe you were being?). o

TISHUG Software

Column by Rolf Schreiber

Before I mention this month's software, I would like to quickly list all the software that has been released this year to date, and try to fill in some gaps as best I can. Some programs originally mentioned in this column have not ended up being distributed because of bugs or other difficulties with the programs. The following list shows the disk ID number, the disk format and the disk name/description.

A001 DSSD Infocom Rapid Loader/Demo game
 A014 SSSD Ringwraith's Lair
 A022 SSSD Return to Pirate's Isle
 A034 SSSD Speed Reading
 A052 SSSD Nibbler V1.2
 A078 SSSD Extended Business Graphs V1.0
 A091 DSSD Bridge Bidding I, II and III
 A131 DSSD Milton Bradley Arcade Plus Series
 A145 SSSD Scott Adam's Adventure Series
 A325 SSSD Teach Yourself Extended BASIC
 A346 SSSD Games of Wit Series
 A358 DSSD Funnelweb V4.21
 A361 SSSD Teach Yourself BASIC
 A362 SSSD Beginner's BASIC Tutor
 A363 SSSD Programming Aids I
 A364 SSSD Programming Aids II
 A366 SSSD TI Base Tutorial Disk #1
 A367 SSSD TI Base Tutorial Disk #2
 A368 SSSD TI Base Tutorial Disk #3
 A369 SSSD TI Base Tutorial Disk #4
 A370 SSSD H.C.M. On Disk Vol 4 No 1
 A371 SSSD H.C.M. On Disk Vol 4 No 2
 A372 SSSD H.C.M. On Disk Vol 4 No 3
 A373 SSSD H.C.M. On Disk Vol 4 No 4
 A374 SSSD H.C.M. On Disk Vol 4 No 5
 A375 SSSD H.C.M. On Disk Vol 5 No 1
 A376 SSSD H.C.M. On Disk Vol 5 No 2
 A377 SSSD H.C.M. On Disk Vol 5 No 3
 A378 SSSD H.C.M. On Disk Vol 5 No 4
 A379 SSSD H.C.M. On Disk Vol 5 No 5
 A381 SSSD Thorn-EMI Arcade Games
 A382 SSSD BOOT
 A383 DSSD Best of 99'er Volume 1
 A384 DSSD Funnelweb V4.30
 A385 SSSD ROS V8.14
 A387 SSSD Myarc HFDC disk manager MDM5 V1.30
 A388 SSSD Assembly Program Loader (APL)
 A389 SSSD TI Multiplan V4.0 upgrade
 A400 SSSD The Missing Link demo
 A401 SSSD Pix V1.2

Some releases from previous months

DISK A052 is Nibbler V1.2, originally a commercial release by Tarik Isani. This program is a sector-based disk copying utility which I find extremely useful for a variety of reasons. It is the only disk copying utility that I know of that will work on a damaged disk. Not only will this program allow you to easily copy a disk with a damaged sector zero (or a damaged sector 1), but you can sector copy from one disk format to another, and even copy the sectors to a different starting sector on the destination disk. This can be very handy if you need to do some non-routine copying, or need to recover some information from a blown disk. The program loads from Extended BASIC, or from TI BASIC if the Editor Assembler or Minimemory module is inserted. A help file is loaded if you hold down the '?' key during loading.

DISK A381 contains the Thorn-EMI Arcade games series of Submarine Commander, Computer War and River Rescue. These games were originally scheduled for release at about the time that TI pulled the plug on the TI-99/4A. As a result, these excellent games were withheld until only fairly recently. Submarine Commander is a war simulation which takes you on a submarine patrol mission to hunt down and destroy enemy shipping. Watch out for enemy

destroyers and depth charges on the realistic instrumentation display. Use sonar and your periscope to hunt down your prey.

Computer War is another well-programmed war simulation. This time you must defend North America from nuclear attack. Fast action, coupled with decision making have made this game as realistic as is possible.

River Rescue involves rescuing a group of people trapped between a raging river and a fierce battle. Rescue as many as you can, while avoiding hazards such as electric eels, hostile troops and tanks. Photocopied documentation is available for \$1.00 extra.

DISK A389 is Art Green's excellent upgrade to TI Multiplan. He has somehow been able to increase the speed of Multiplan by around 50%, which is no mean feat. He has also greatly speeded up the initial loading of the system files, which can be configured to load from various devices, including RAMdisk or Hard disk. If you use Multiplan at all, you will soon appreciate the difference that this upgrade makes towards overcoming the slowness of the original version.

Releases for August 1990

DISK A365 is Programming Aids III, the third programming utility disk released by TI back in 1982. This SSSD disk contains some useful tools for the advanced programmer who wants a list of all variables, arrays, keywords, and functions cross referenced to all the line numbers in which they occur. Also on the disk is a very useful program which lets you resequence, delete or merge only a part of a program. In my opinion, this program alone is worth the price. A photocopied manual is available on request for \$2.00 extra.

DISK A386, which I am calling BOOT HD80 in order to differentiate it from the BOOT menu loader on disk A382, differs from the other version in some important aspects. Firstly, this version allows you to toggle between 40 and 80 column display, if an 80 column card is attached. If your system has a hard disk attached, you can then also catalog any subdirectory and select to view a text file or run a program file from that subdirectory. As well as handling both Extended BASIC and Editor Assembler type program files, the loader can also directly load GRAM files into a P-GRAM card, if such a card is attached. The numerous other enhancements are all explained in the accompanying docs file. This version can reside on a floppy disk, a RAMdisk, or in GRAM (such as the Geneve, GRAM Kracker, or P-Gram card). It allows 24 User-defined options, which can be loaded from/saved to a configure file (in Dis/Var 80 format).

DISK A390 is called TI Print Shop V1.5 and is a freeware offering from Ron Wolcott. This program has been described as "the ultimate graphics utility for the 99/4A" (MICROpendium ad, Feb. 1990, p3), and allows you to create your own custom greeting cards, banners, flyers, posters and much more. There is a vast TIPS library of over 2000 graphics, including fonts, borders, margins and pictures, plus you can readily convert artwork to TI-Artist format.

DISK A391 is the first in a series of four educational disks to be released over the next few months. This particular SSSD disk is called "Don Shorock's Commonwealth Collection", and is aimed at the student who wants to improve his/her knowledge in specific areas. The author is a talented programmer called, obviously, Don Shorock, who has written quite a lot of other non-commercial educational software. All the programs on the disk require the Extended BASIC module, with the exception of one program, which needs the TEII module and the Speech Synthesizer. A menu makes program selection child's play:

continued on page 23

by Rolf Schreiber

Hi, the good news for the month is for those members who missed out on getting a coloured monitor. Contrary to my previous advice, I have now been informed that we have a good chance of acquiring a few more monitors still at the very good price of \$122.00 including the Interface PC Board. For those still wanting a monitor please contact me as soon as possible and get your name on the ordering list as we do not know how long this offer will last. When we have ten definite takers we will place an order for the monitors and I will get back to you for payment in advance. If you want to know just how good these monitors are just read Ross Mudie's "Letter to the Editor" in last month's TND or better still talk to Ross at the next meeting - I can assure you his views were completely unsolicited.

Unfortunately we have now depleted our supply of 5.25 DSDD Disk Drives but do not despair, we have a good supply of 3.5 Disk Drives which will work equally as well on your TI99/4A system and priced at \$20.00 each are a very good buy.

I would like comments from our members as to what they think of the quality of the shop software and hardware, how they would like to see it improved and whether there are any other items that they would like to see on the shop "shelf".

In addition to items listed in the May issue of TND the following are also available:

New release software disks -----	\$2.00
(See "Software Column" for details)	
Asgard News Vol. 1 No.4 -----	\$3.25
Asgard News Vol. 2 No.2 -----	\$3.25
5.25 in. DSDD Disks-Box of 10 -----	\$6.50
5.25 in. HD Disks-Box of 10 -----	\$16.00
3.5 in. DSDD Disks-Box of 10 -----	\$15.00
Horizon Ramdisk EProms -----	\$20.00
Multifunction PC Board -----	\$30.00
Multifunction Dsk Controller Kit ---	\$102.50
Two-way Interface PC Board-----	\$23.00
Monitor Interface Kits (no Jiffy Box)	\$21.00
Music Kit (including PC Board) -----	\$65.00
Lithium Batteries - 3.6V -----	\$4.00
Mini Speakers - 8 Ohm -----	\$1.50
Power Leads (need std 3 pin plug) ----	\$1.00
Transformers - 60VA -----	\$16.00
Transformers - 13V Arlec -----	\$12.00
Transformers - 12V ac -----	\$3.00
32K Printed Circuit Boards Only -----	\$7.00
Module PC Boards (Single chip vers.)--	\$9.00
32K Memory Chips -----	\$17.00
7 Way PCB DIL Switch -----	\$0.50
TI Technical Data Manual -----	\$15.00
TI Sort -----	\$15.00
Display Master -----	\$15.00
Adventure Hints Booklet -----	\$2.00
Pro-Mod Booklet -----	\$1.00
TI Writer for Novices Booklet -----	\$1.00
Extended Business Graphs Booklet ----	\$1.00

Packaging and Postage Charges:

Up to 2 Disks-----	\$1.60
Over 2 up to 5 Disks-----	\$2.10
Over 5 up to 12 Disks-----	\$3.00
3.5 Drives -----	\$10.50
Other items at actual postage cost.	

continued from page 2

Further discussion and comments on this important topic are essential. Your thoughts will be welcome.

See you at the next meeting.

Dick Warburton

Disk A382, released as part of the June 1990 software release, has two vexing bugs which would have caused anybody who bought it quite a few headaches. If you purchased this disk at either the June or the July meeting, please return it to the Shop for a free, modified replacement disk, or else read on and make the necessary changes yourself.

The first bug came about because of a filename change. The original library version was called "ROOT", but because there was a writeup that month about that particular program, I changed the filename to be the same as that used in the writeup, in order to avoid any confusion. Unfortunately, this introduced a bug, since the original filename was still embedded in the program, and this resulted in the "program not found" error if you tried to save any changes back to the disk. The easiest way to squash this bug is to rename the file to "ROOT", making sure that you also make the same name change to the LOAD file. Alternatively, you can sector edit the file to change the embedded name from ROOT to BOOT. The change to be made is on the 6th sector of BOOT (sector >27, byte >17, if BOOT is the first file on the disk). Use the "find string" feature of Disk Utilities in ASCII mode to locate "DSK1.ROOT" and change it to "DSK1.BOOT". Do not forget to write the changed sector back to disk, in order to make the change permanent.

The second bug only shows up if BOOT is on a floppy disk, but does not show itself if BOOT resides on a RAMdisk. Basically, the bug prevents you from changing any of the menu selections, or editing the menu in any way, and then saving the changes back to the floppy disk, in order to make the change permanent. Since the whole idea of BOOT is to allow you to customize it to your own requirements, the unmodified version sold at the June and July meetings is virtually useless.

The same sector is involved in fixing the second bug also. To make the necessary change, again use the "find string" feature of Disk Utilities. This time make the change in hex mode; change 02213000 to 02013100 and write the change back to the disk. Bytes >D0 to >D3 are the bytes changed. The patch changes the code from AI R1,>3000 to LI R1,>3100, in case you are interested. If you want to run BOOT from a floppy drive other than DSK1, change the 02013100 to 02013X00 where X=drive #.

continued from page 16

notices any "." not part of a text string and lets you know - as in LI R8,6 for instance. There are a couple of minor bits of anomalous behaviour, but they are only incidental and very occasional, and a fix is not a high priority in assigning code space. It bleeps on the filenames in COPY directives, which in standard TI Assembler syntax are indicated by normal double quote marks and usually contain one or more "."s not isolated within single quotes. Some opcodes do not have an operand field, and if these also carry comments the first word of the comment, if any, will be treated as if it were the operand field and converted to upper case. Most commonly these are the RTWP opcode and the R1 pseudo-instruction. Usually the function of these is so limited and obvious that they do not deserve a specific comment, but no harm is done anyway. In practice neither problem is a real bother, but if they do grate on any sensitivity just turn ASMode off, or never turn it on in the first place.

This new ASMode code is written over the Reformat code on the way in so that it will no longer be reasonable to have immediate re-entry to the Editor interchangeable between Word Processor and Program Editor. In fact instant reentry is a very marginal feature at best, and may end up being omitted altogether if it gets in the road of more desirable features.

Techo Time

by Lou Amadio

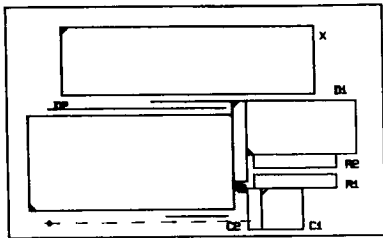
The July meeting provided a very spirited debate on the future of TISHUG. Most of the members present indicated a reserved willingness to open the club up to users of other computers. Personally I cannot see how this would prolong the future of the TI/99/4A - in fact it will probably do just the opposite. I hope that the members think about this proposition very seriously before giving it their blessing or we may see the end of the TI-99 before its time.

Techo Articles

The feedback from Eric Ockeden last month was very welcome. I would like to hear from others who have been tinkering with hardware. You never know if you can help someone else with a similar problem. Geoff received a letter from Jim Banfield showing the design for a floppy disk controller. Jim designed and built the hardware as well as writing the DSR. Unfortunately he is not able to contribute articles for the TND but we hope to receive more details of his experiments in the near future.

Errata 32K Memory Expansion

Some feedback from Tony Bell (Bomaderry) pointed out some omissions in the overlay diagram for the single chip 32K memory expansion project published in the May? 1990 issue of the TND. Please check the overlay diagram in this issue for the corrections.



Errata MFC Project

More problems and errors have surfaced on the MFC project.

- 1) The MFC printed circuit board is covered with a green varnish. I found that the varnish prevented electrical contact on some MFC's when the card was pressed all the way home into the edge connector. Pulling the card out slightly from the connector allowed correct operation. The same was true for the disk drive edge connector at the back of the card. A better long term solution is to carefully scrape away a further 2 mm of varnish from both sides of the board at both edge connectors. Take care that you do not damage the gold contacts. Use very fine sand paper of a very sharp knife.
- 2) Due to a design fault, the MFC cannot be used with a disk drive cable edge connector within a PEB with the top on and clipped down, as the top of the PEB fouls the connector. In this case use an IDC pin connector at the disk controller end of the floppy cable and the socket on the board.
- 3) The lower front and back edges of the MFC need to be cut back in order for the card to fit into a PEB slot. This should be done prior to fitting components if possible. The areas are marked on the PCB and correspond to approximately 41 x 7 mm at the front and 10 x 3 mm at the rear (near the voltage regulator). A small amount may also be taken off the top rear of the card near the drive connector to allow a better fit in the PE box. How much to take off will depend on the card is used.

- 4) The card that I am using would not fit into a 60 way edge connector. I had to file the end near pin one. Check fit and alignment of the gold contacts with your edge connector before attempting any modifications.
- 5) Some of the the PCB tracks on the MFC are very close to others. Make sure that you do not accidentally bridge any tracks during soldering.
- 6) A 0.01uF greencap capacitor must be used with the 74LS122 chip. This capacitor is connected to pins 4 and 7.
- 7) The 74LS245 chip should be an "ALS" type, that is a 74ALS245.
- 8) Be careful with the orientation of the SIP resistor packs. Pin 1 is marked with a dot and must face towards the top of the board, that means away from the I/O gold contacts.
- 9) If using Shugart or equivalent vintage drives, leave all of the dip switches off, otherwise set to suit your drives as per previous instructions.
- 10) The 27 pF capacitor may need adjusting during final setup. This can be done at the monthly meetings. I tried my card, prior to alignment with an oscilloscope, by including the 27pF cap and setting the pots to midpoint. Most functions seemed to work as intended.
- 11) The capacitor associated with U19 (74LS04) can be 220 pF to 330 pF.
- 12) To include the circuitry which makes the disk controller work correctly with all other peripherals, locate the 74LS32 in the middle of the PCB next to the 74LS30. If you have not soldered this in yet or if it is in a socket, bend out pin 11 so that it does not make a connection. If the 74LS32 is already soldered in then cut the track that runs on the top of the board from pin 11 to a hole through the board close to pin 1 of the TMS9901 (not the closest to pin 1, however). Solder a wire into that same hole and long enough to reach back to the 74LS32. Solder a short wire to pin 11 of the 74LS32. Then take a 74LS00 and bend out all pins except pins 7 and 14. Solder this 74LS00 to the top of the 74LS32 using pins 7 and 14 to the corresponding pins of the 74LS32. Solder the other end of the short wire on pin 11 of the 74LS32 to pins 12 and 13 of the 74LS00. Solder pin 10 to pin 11 of the 74LS00. Solder the wire from the hole through the PCB near pin 1 of the TMS9901 to pin 8 of the 74LS00. Take another length of wire and connect pin 9 of the 74LS00 to the hole in the PCB at about 4 o'clock from pin 21 of the 2793 IC (where 12 o'clock points to pin 40 of the 2793). The hole to use is the one just below the '+' sign on the board with a track on the top of the board running to the SIP resistor pack and nowhere else.

Apart with the difficulties mentioned above, in relation to the depth of insertion with edge connectors, I have found that the MFC will not format single density disks (I was also using a single sided drive at the time). Using the same setup, but with a TI disk controller, I found no problems whatsoever. On the other hand, when I transferred the MFC to my PEB I found that it would format single density! We are working on the problem.

Printed Circuit Boards

The shop still has a number of PCBs available for the Direct I/O Interface and the Multi Function Card.

RGB Monitors

We are going to try for one more order of Wang RGB monitors. These are high quality 12" units which offer a superb picture quality on a TI99/4A.

Setting up Hard Disk

by Derek Wilkinson, (02)5254111(W), (02)5259121(H)

As I use a RAMdisk as well, my CRU settings are:

HDFC 1100
RAM 1400

Connect the Hard Drive as per page 6 of the manual for the card and DIP switch settings.

Load the Myarc Disk Manager (MDM5). The easiest way is to boot the disk from drive 1 in Extended BASIC or use BASIC or Extended BASIC and type in CALL MDM.

When MDM5 has loaded the options on screen are Exit or Set. Set allows you to update the time and date on the internal clock. As the card is not battery backed the clock requires setting each session if time and date on file updates will be required. Exit brings up the first screen of the disk manager proper, giving the following options:

Catalog
Directory
Find
Quit
Setup
Utility

If we select Setup, this option will allow us to set the screen colours, printer type, type and number of hard and floppy drives (see page 20). To save selections press S.

Return to the main menu, select Utility and options given are:

Clone DSK1 Exit
Format RAMdisk Test

As we will need to format the Hard Disk, select 'Format': the options then are:

Exit
Floppy
Hard

Select 'Hard', the screen will prompt for a drive number. If you connected the ribbon cable to the J3 then the number will be 1. In my case I used a 10Meg Drive connected to J3.

The screen now asks you for the information it requires to format the drive. See page 11 of the manual. WDS*. gives you the opportunity to name the drive volume.

Page 11 and 12 in the manual help to explain the procedure. The selection I used to format my disk is as follows:

Sectors per track	32
Heads	4
Cylinders	305
(0 to 305 = 306 cylinders)	
Reserved dir/file sectors	1024
Write pre-compensation	240
Reduce write current	240
Sector verification retry	8
Interlace	4
Step rate	1

To check the best settings test your drive at various interlace and step rates. When test formatting set the number of cylinders to a low value,

Cylinders=100

This will speed up the time to format. Choose step and interface values which allow the fastest formatting time.

Now the disk has been formatted you are ready to use the drive. First try cataloguing the drive. FCTN[9] (Back) will return to the menu to allow this to be done. Choose Catalog, Hard, this will show the number of sectors and your volume name.

To set up a directory tree on your drive, press Exit to return to main menu and select from this screen Directory (see page 24 of the manual). This selection gives options:

Create
Delete
Exit
Rename
Utility

You can rename your directories at any time without losing your files but remember if a program uses the old directory to address a file, that needs to be changed as well. Select Create and you will be asked Hard or Floppy, choose Hard and you will be shown a directory of the drive. As you have not designated any sub-directories only the root directory will be indicated, press Continue and you will be prompted for a directory name. Call the first one DSK1 press enter and you will be returned to the first list of options after it has created the sub directory.

You may create up to 127 directories, with 127 sub-directories off each directory. In my own system I created DSK1 and DSK for emulation useage, Art, Games, FWB, Music, RAM5, Temp/Work, etc.

In the root directory I have only loaded a couple of files, LOADR and ROOT along with the head parking files for the Hard Disk. LOADR and ROOT are my first level Menu on the drive. In the sub-directories I have menus that are called up from this first one. If, for example, I want to call up a game, from my RAMdisk menu I would select the Hard Drive Menu. From the HD menu select Games that brings up a menu that has several menus of games. Similar to having several floppy drives on line and loaded. One option on all menus is to return to the previous menu. Therefore if the menu selected does not have your required program you do not have to use the quit function and start from scratch in your search.

In the DSK1 directory I have loaded the MDM5 files but first I changed the load file to LOADMDM to prevent auto loading of the program. I made provision to select MDM from both the RAM and HD menu. My RAMdisk now only contains the CHARA1 and BOOT files as permanent, leaving the rest as temporary storage area.

Note the program ROOT used as a menu for disks will not save changes back to any drive but the RAMdisk when I am using the HDFC card in my system. This requires use of the disk manager to transfer it from the RAMdisk to where you require it to reside, each time you alter the menu and save it. ○

Treasurer's Report

by Geoff Trot

The current position shows a healthy balance because we have just had our major membership renewal. This has to last for the next year.

Current assets at 31/05/90	\$7131.48
Add deposits in June	\$2762.70
	\$9894.18
Less expenses in June	-\$493.67
Current assets at 30/06/90	\$9400.51 ○

continued from page 19

* While the directory is visible, pressing any other key besides a direction key causes the worksheet previously on the screen to reappear. This other key has the same effect as it does while editing responses in command fields that need to be filled in, as described in the "Editing" section of chapter 8. ○

The Communicators

At the meeting on 7th July, two members brought modems along to find out how to connect them to their TI99/4A computers. One of the members also brought along his computer. After some discussion, steps were taken to connect the modems to both computer and telephone line. After a few of the usual problems on both the phone line and the connections between the modem and the RS232 (it is amazing how many combinations are possible here), successful connection was made to the BBS. One of the new users was promptly enabled on the BBS using the remote Sysop facilities. Later the other modem was found to work as well and both new users (and a couple of others) enjoyed a guided tour of some of the BBS facilities. My thanks to Peter Schubert and Ben Von Takach for their help, it is great to have assistance so willingly given. This spirit of helping each other with both hardware and software for the computer is just what the club is all about.

It is usual for the usage of the BBS to fall away in the winter months, but this June the callers were very sparse. Twenty members were deleted off the system this evening, due to their non renewal, leaving a total of 68 registered users and this number may fall by another 4 who are in the unfinancial category. (BBS membership is only \$5 on top of the standard TISHUG membership).

Each month a selection of files on different topics about the TI99/4A are loaded, some repeated information and some new. Members can use the MAIL system on the BBS to ask for help with something that they are having trouble with, or even ask the SYSOP if information on a particular topic is available. A wide selection of Basic, extended Basic and implanted assembly programs are available. These days repeats are occurring which is great for new members and those who have lost a program or file. A selection of up to 40 programs remains on the BBS for 2 months and requests for particular files or programs is possible. The BBS provides facilities for members to share programs by uploading and a members advertisement service is available, all for the cost of a phone call. (Even this article will be sent via the mail system of the BBS to the TND Editor who will download it at a time which is convenient to him, rather than me ringing him now, 12.45am, to do a modem transfer). I appreciate that Ross! ED.

For those who have forgotten the BBS can support 2400, 1200, 1200/75 or 300 baud operation with 8 data bits, 1 stop bit and No parity. If using TI Terminal Emulator 2 then just use the defaults. The BBS phone number is (02) 456 4606 and the system is available 24 hours per day 7 days per week. The system is menu driven and it is very easy to find your way around.

For more information contact the SYStem OPERator, Ross Mudie, at meetings or in the evenings (7pm to 9pm) on (02) 456 2122.

Like all of the club activities, the more people who participate, the better it becomes for us all. o

Bulletin Board Users

by Stephen Shaw, England

What follows is a rather humorous (but sobering?) article on the occasional conduct of Bulletin Board Users. I suppose that Ross will want equal time. ED

A magazine recently commented on crossover of technical words into general use, and mentioned a usage my own 6 year old makes use of... the concept of "rewinding" your memory - rather more effective and positive than simply forgetting (if you do not play back) and an interesting way of remembering something

continued on page 4

Program Information

from Ross Mudie

This information was taken from the BBS.

Program	System	Comments
ADDITMAGIC	Basic	Add up both ways
AMBULANCE	XB&32K Game	pick up patients
ASM2XBLOAD	XB&32K	Create CALL LOADS from XB linked assembly
BERT&ERNIE	XB Speech	Excellent speech
BINDECHEX	XB	Number conversion
BOUNCER	XB	Jumping game
BUCK2RG1XB BUCK2RG2XB	XB&32K	Space game with speech
CARWARS	XB&32K	TI module car race
COMPRESS	XB	Utility to remove remarks from basic or XB programs. Uses MERGE format in and out.
DIGDUG1XB DIGDUG2XB	XB&32K	Tunneling game
DRIVEDEMON	XB&32K	Driving game
FASTTRM1XB FASTTRM2XB	XB&32K emulator	Fast Term terminal
FREDDY1XB FREDDY2XB	XB&32K	Explore the tunnels
FT_DEFAULT	XB	Writes defaults for Fast Term
HOWIE	Basic TE2 + Speech	Speech demonstration
MOSAICPUZL	XB	Rearrange numbers or letters
PINBALL	XB&32K	Pinball game
POLEPOS1XB	XB&32K	Driving game
POPEYE1XB POPEYE2XB	XB&32K	Catch the hearts, watch out for Bluto
Q*BERT1XB Q*BERT2XB	XB&32K	Jumping game
RUNNER1XB RUNNER2XB RUNNER3XB	XB&32K	TI Runner game
SCORE_FOUR	Basic	2 player strategic game
SEALORDS	XB RS232 Modem	2 player modem game
SENDMAIL5	XB&32K	BBS utility sends files
SHAMUS	XB&32K	Shooting game
SLYMOID1XB SLYMOID2XB	XB&32K	Multi screen shooting and exploration game
TOMBSTONE	XB&32K	Tombstone City shooting
TOWER_SIAM	XB	Strategic game

Cutthroats part 1

Copyright 1984 Infocom

This walk through is by Scorpia, Copyright 1984.

Welcome aboard, matey! Dust off your scuba tank, shake out your flippers, and prepare to go treasure hunting. But first, a word from our sponsor. Cutthroats, like most Infocom games, has several solutions. This walkthru will show you one way of completing the adventure. However, there are others, so when you have finished, you might want re-play the game, doing different things, to see if you can come up with another way of recovering the treasures successfully.

Also, you should be aware that you can only recover treasures from 2 of the ships, the Sao Vera and the S.S. Leviathan. The other wrecks are only red herrings, and you do not have to bother with them. Which of the two real wrecks you will dive for depends on the item you are shown by Johnny Red. If he shows you the gold coin, it is the Sao Vera; if it is the dinner plate, then the ship is the Leviathan.

Further, most of your actions up to the dive itself will be pretty much the same, so this section of the walkthru will take you up almost to the dive itself. After that, consult either the Sao Vera section, or the Leviathan section, depending on which ship you are investigating.

Ok! The game starts with a long lead in, explaining how you came by the book of shipwrecks. You will have to sit through this on each boot-up; no way around it. After that, the game really begins, with you lying in bed in your scruffy room at the Red Boar Inn. The first thing to do is stand up, then wind your watch (time is important in the game, and if your watch runs down, you cannot keep track of the time).

There is a note on the floor. Read that, then open your dresser. Inside are the shipwreck book, your bankbook, and a room key. Get the key, open the door, go out, and lock the door again. You do not want to leave the door open, or the Weasel will come by later and steal the shipwreck book. If that happens, the game is over before it even starts. You do not need to take the book with you, so locking to door is effective here.

Now, go downstairs and out to the Wharf Road. Follow the road East until you get to the Shanty. Enter the Shanty, and you will see Johnny Red and Pete the Rat already there. down and order breakfast, then wait for Weasel to show. Order a glass of water when you get thirsty. While you are waiting, you might want to listen to the parrot. He does not have anything important to say, but you might get a chuckle out of him.

Eventually, Weasel will arrive, and Johnny will ask if you are interested in doing some treasure hunting. Say yes, and then Johnny will have you all meet again a little later at the lighthouse, in order to keep McGinty from finding out what you are up to. After that, leave the Shanty, go back West to the end of Wharf Road, and from there Southwest twice and Northwest once, which brings you to the lighthouse. Now, wait for Pete, who will be the last person to arrive.

Once Pete gets there, Johnny will show an object, either the coin or the plate. This indicates which wreck to dive for. After that, he will give further instructions, which you should read carefully. When he is finished, go back to your room at the Red Boar. Get your passbook. If you are diving for the Leviathan, also get your scuba gear from the closet (scub gear not needed for the Sao Vera).

Leave the room (lock the door behind you!), and go back out. Walk East along Wharf road to the end, and go Southeast to the Ocean Road. If you are going to use your scuba gear for the dive, go Southwest into the alley, and drop your scuba gear there. You do not want McGinty to see you lugging it around.

Follow the Ocean Road south to the end, then go Southwest to the Ocean Road, and North into the bank. Make your withdrawal, then leave and return to Ocean Road, where you go Southeast to Point Lookout. Drop your passbook here (that McGinty has sharp eyes, and you do not want him to see you with that, either), and wait for Johnny.

When Johnny arrives, show him the money you just took from the bank. He will be satisfied, and then ask if the wreck is more than 200 feet underwater. Answer yes if it is the Sao Vera, no if it is the Leviathan. The two of you will then head back to International Outfitters to rent a ship and purchase supplies and equipment. McGinty will be in the store when you get there. However, just wait, and he will leave eventually.

Johnny will make his purchases first, and you will have to chip in some of the cash you are carrying. However, you will have plenty of money left over to buy whatever you need. When it is your turn, buy the flashlight and the shark repellent. If you are diving for the Sao Vera, that is all you need. However, if you are diving for the Leviathan, also buy the following items: C battery, putty, and electromagnet, and also rent the small air compressor (so you can fill your tank). All these items will be delivered to the ship for you, so you do not have to take them with you.

Now, it is time to uncover a little double-dealing. Leave Outfitters, and go back East along Wharf Road to the end, then Southeast again to Ocean Road. Go along Ocean Road to the end, then Southwest to Shore Road, and continue West along Shore Road until you reach the Ferry dock. Wait around.

Soon McGinty will appear, and a short while later, Weasel. The two men will go off to a corner and talk. Then Weasel will hand something to McGinty, and board the Ferry (you cannot get on it yourself, but you have other things to do, anyway). Ok, now you have seen that, go back to Ocean Road, and then into the alleyway.

The alley runs behind all the buildings, and it will come in very handy! Go West along the alley (pick up your scuba gear if you dropped it here earlier), until you are standing behind the vacant lot, which is next door to McGinty's. Wait here, and McGinty will come by, heading from East to West. Continue waiting, and he will soon re-appear, going from West to East (he is walking along Wharf Road, of course).

Once you see him the second time, go West once, and you are behind his store. The door is locked, but you can open the window and get through into the place. Here you will find an envelope that proves the Weasel is out to double-cross you all. Get the envelope, then leave by the window.

Go back along the alley to the Vacant lot, then go straight North until you come to the dock where the rental ships are moored. Both ships have approximately the same layout; they are slightly different on the top deck, but below they are exactly the same. Enter whichever ship has been rented for the dive, and go below deck. Then go north until you reach the crews quarters, and hide your envelope under the bed. You do not want Weasel to know you have it (he will kill you), and if you show it to Johnny now, you will cancel the expedition.

Now you have to do some more waiting. The delivery boy will come around, and drop off the items you have bought. Then the others will start to arrive. When Johnny comes, go to the Captain's Cabin, and tell him the longitude and latitude of the wreck, which you can easily get by looking at the shipwreck book that came in the game package. Then go back to the crews quarters, and wait some more. Eventually, you will reach the dive site. At this point, you should now read either the Sao Vera section or the Leviathan section, whichever is applicable

Rambles

by Stephen Shaw, England

Welcome to another Rambles, which includes bits and pieces for all owners. Even a bit that at first sight you may feel is not appropriate to you is still worth a quick scan...

Mark Wills is interested in hearing from all our members to find out WHY they still use such a vintage computer. Me first:

Way way back, when I first thought of buying a computer, I had a very good think about what I wanted one for, what I might need one for, and so on, prioritising everything. There was only one choice then available - the TI99/4, which at that time had not yet been released in a UK version! I waited for the UK version, and bought a fully expanded system - one of my prime uses was for data, and I saw no future in trying to use a cassette for data!

Many years later, looking back to my original requirements, I still find the TI99/4 (now with an A) to be the best suited to my requirements -

- *A friendly operating system
- *Good educational software
- *Very rugged construction
- *Good choice of languages to learn and use
- *Capable of everything I want! in games and utilities

The competition is actually very thin on the ground in 1990 - Atari ST, Amiga and PC seems to be about it. The PC in many ways would be a very retrograde step, and the ST and Amiga require a dedicated monitor for worthwhile results, and are by no means so friendly or easy to program as the TI.

I really do not fancy trying to write a 500k program, or prepare 256 colour graphics! And a slow machine with limited RAM makes you learn to program effectively, to do in 16k what other people might take 200k for!

I have a stack of things still to do with this vintage computer, books to read, programs to convert or write, languages to learn and experiment with, many commercial utilities to come to grips with. About another twenty years worth I guess...

Why move on to fresh pastures when this one still has so much to offer!

Congratulations TI on winning a patent on the microchip after all these years, giving a hefty boost to corporate income...

IBM got it wrong again, after really faulty maths in the first IBM PC we now have a problem with PS/2 Model 30 machines, a temporary cure for which is to plug the keyboard into the mouse port... and still IBM get to set the standards...

The TI99 was established in 1979, and "killed off" in 1983, yet here we all are in 1990, and still with new software - including new modules.

The UK version of Computer Shopper gave such a lovely description of the fourth Alternative Micro Show (held in Nov 89) that I must share it with you... "...should have been a hit but it was poorly publicised and held somewhere between Valhalla and Oblivion, Staffs. ... the crowds stayed away... well off the beaten track". And yes, the fifth one is scheduled for the same venue. What started life as a very good idea has been lost- and the next AMS will have the machine it started off as being an alternative to! Yes the IBM PC will be there... pity. It WAS a good idea, just badly put together and terribly publicised.

RED AND GREEN TEST... page 18 issue 26... John Seager has kindly submitted a c99 answer, which works swiftly and well. c99 owners will be aware that c99 code can be lengthy, AND you need a disk system to use it, and as space remains a little cramped even in this issue, I cannot print Johns solution here, but many thanks anyway, at least one reader had a go! Good patterns to start with are lines at screen top and bottom or screen left and right, or a frame all around the screen. (Disk owners- send for a copy of Johns solution, just send one disk and return postage to me. sjs.).

ERROR: ISSUE 26: PAGE 29 : MANDELBROT ALGORITHM.
Sorry 'bout this. For step H please read:
TEMP=XTEMP-YTEMP+XCONST
(or alternative) then
YY=2*XX*YY+YCONST; XX=TEMP; C=C+1;
=====

MYARC HDFC-from Jan Alexandersson I hear one of our members hit a snag with this expansion card- he has only ONE physical disk drive, and having activated "DSK1 EMULATION" found that he no longer had access to the Myarc Disk Manager which he ONLY had on disk, in a form which REQUIRED Drive One- he was entirely locked out of his HDFC! Jan makes the comment that "it is thus VERY IMPORTANT to have several copies of DMS with DIFFERENT RELOAD paths. A subdirectory on the hard disk is OK as a load path". Before you deactivate drive one, make sure you can carry on with another drive!

M H Robinson is asking for articles on the use of Extended Basic sprites, games graphics, CALL JOYST, sprite movement and coincidence checking. We have the Miller Smart Programming Guide in the publications library. Do we have any XB programmers out there to take this a step further?

Back in issue 27, on page 20 I set two TESTS. Not a single response to either of them, however, I have now discovered a whole series of questions of the type posed by Test 4, in "The Master Book of Mathematical Recreations" by Fred Schuh, published by Dover, ISBN 486-22134-2. Refer to puzzles 298 to 301.

The same author, in the same book, claims the creation of (and describes) the puzzle Giants and Dwarfs - a program of this can be found on library disk Games-3.

No feedback at all on the module reviews (in fact unexpanded owners remain the great unknown, as they do not seem to want me to write anything for them - no letters anyway! Hello?) - so in this issue a smaller number of module reviews. And remember that ALMOST everything can be obtained from SOMEWHERE, even if you do have to try several US sources. Martin Blythe at Database is worth checking on - the poor chap had some extremely bad health recently, so it was nice to see him at Chester. Martin has now had a good look at the original F Parrish prices, which were his original guide, and has made some nice reductions, so if you were previously deterred, have another look! There are ample stocks of some quite nice modules, and at Chester Martin had stocks on sale of some modules which were both very good AND pretty rare!

There is a disk program called ELECTRIC HARRY bombing around - please be aware that it is a pirated version of ESPIAL with the pirates name none too well disguised!

The puzzle set in the last issue (Page 19) brought in two entries and a query. The first clue is in the first hint - the answer has more than fifteen digits. If you refer to NUMERIC CONSTANTS in your Users Reference Guide, you will see "...numbers will have 13 or 14 digits depending on the value of the number."

So how can a Basic program handle a problem which requires an answer with more than fourteen digits???

Simple! We allow the program to deal with numbers with less than fourteen digits - and create our larger numbers from these smaller numbers. We do this all the time, and learn it in our first or second year of schooling. $2+2=4$ is easy. Now try $8+8=...$ the answer needs more than one number! We could call it sixteen, but to write every number with a separate character would slow learning down a mite. In writing numbers we use a decimal system, with the rightmost number representing a number of units, the number to the left of it the number of tens of units and so on.

To solve this problem we must be much more specific in our instructions to the computer, which for our purposes is a dunce! (Though fortunately a fast one!).

We need our program to deal with the "carry"s! Then of course we need to work out how to actually do all the multiplications - how to store the little numbers making up the big number, how to get them back together for testing for the six sevens...

We can test six digits from every character position, and we can speed things up by only testing every sixth character (or digit) and if - and only if - that is a seven, testing the numbers in front of and behind it. This short cut brings a useful increase in speed.

Here are the only listings sent to me up to the time this issue went to press:

```

99 REM John Seager
100 !ANSWER TO TEST5/A
110 CALL CLEAR :: NUM$="117649" :: DISPLAY AT(1,1):
"7 TO THE POWER OF"
120 FOR POWER=7 TO 300 :: DIS$=NUM$ :: NUM$="" ::
CARRY,COUNT,SEVENS=0
130 FOR J=LEN(DIS$)TO 1 STEP -1 :: NEWNUM=(VAL(SEG$
(DIS$,J,1))*7)+CARRY :: CARRY =INT(NEWNUM/10)
140 IF NEWNUM-CARRY*10<7 THEN COUNT=0 ELSE COUNT=
COUNT+1
150 SEVENS=MAX(COUNT,SEVENS):: NUM$=STR$(NEWNUM-
CARRY)*10&NUM$
160 NEXT J
170 IF CARRY>0 THEN NUM$=STR$(CARRY)&NUM$ :: IF
CARRY=7 THEN COUNT=COUNT+1
180 SEVENS=MAX(COUNT,SEVENS):: DISPLAY AT(1,19):STR$
(POWER);"=":NUM$
190 IF SEVENS<6 THEN 210 ELSE DISPLAY AT(24,1):"ANY
KEY TO CONTINUE"
200 CALL KEY(O,K,S):: IF S=0 THEN 200 :: DISPLAY AT
(24,1)
210 NEXT POWER
220 REM WHY DOES THIS CRASH WHEN THE NUMBER OF
DIGITS EXCEEDS 254....
=====
99 REM FASTER WAY John Seager
100 ! ANSWER TEST5/B
110 CALL CLEAR :: DIM ELEM(26):: ELEM(0)=7 :: POWER,
SS=0 :: DISPLAY AT(1,1):"7 TO THE POWER OF"
120 ELM=SS :: SS,CARRY=0 :: POWER=POWER+1
130 DIS$=STR$(ELEM(ELM)):: FOR I=ELEM-1 TO 0 STEP -1
::DIS$=DIS$&RPT$("0",10-LEN (STR$(ELEM(I))))&
STR$(ELEM(I)):: NEXT I
140 DISPLAY AT(1,19):STR$(POWER);"=": :DIS$
150 FOR I=6 TO LEN(DIS$)STEP 6 :: IF SEG$(DIS$,I,1)
<"7" THEN 190
160 FOR J=I-5 TO I :: IF SEG$(DIS$,J,6)<>"777777"
THEN 180 ELSE DISPLAY AT(24,1) : "ANY KEY TO
CONTINUE"
170 CALL KEY(O,K,S):: IF S=0 THEN 170 :: DISPLAY AT
(24,1):: J=I
180 NEXT J
190 NEXT I
200 ELEM(SS)=ELEM(SS)+CARRY :: IF ELEM(SS+1)=0 AND
ELEM(SS)<1.E+10 THEN 120
210 CARRY=INT(ELEM(SS)/1.E+10):: ELEM(SS)=ELEM(SS)-
CARRY*1.E+10
220 SS=SS+1 :: GOTO 200
=====

```

The Horse Demo on p. 31 of the last issue caused a few problems - unfortunately as we have already proven, consoles run at different speeds, and it does become a

little obvious in programs like this one, in which internal running time is so important to decide how long the horse is going to be in the air, or even how often and how accurate the coincidence check is to be. With only a little tampering to adjust for your own console speed you should make this one work. If the horse fails to jump, then increase the number 10 in line 580 to say 13 or 14. If the horse jumps too late, instead of increasing the number in line 580, try instead reducing the loop in line 650, or even remove the loop entirely with a REM. To bring the horse down faster reduce the loop in line 620 or remove the loop entirely. (And vice versa in all cases!).

PLEASE NOTE that the TWEENING PROGRAMS on pages 71 to 76 of the last issue are entirely suitable to conversion to run with TI XB and THE MISSING LINK, as TML also operates in bit map mode with the ability to use sprites.

Now for some graphics programs for our latest addition to the TI Graphic family, THE MISSING LINK:

```

98 ! GRAPHICS PROGRAM FOR TI XB + THE MISSING LINK
S Shaw from JBM103
100 CALL LINK("CLEAR")
110 CALL LINK("WINDOW",0,10,193,241)
120 CALL LINK("PRINT",2,9,"When ANY KEY is
displayed")
130 CALL LINK("PRINT",19,1,"press: to:")
140 CALL LINK("PRINT",30,1,"CTRL FCTN....PRINT TO
PIO")
150 CALL LINK("PRINT",66,1,"R run this program
again")
160 CALL LINK("PRINT",78,1,"A save to artist file")
170 CALL LINK("PRINT",90,1,"ANY OTHER for different
random picture")
180 CALL LINK("PRINT",131,1,"right now...")
190 CALL LINK("PRINT",142,1,"PRESS R-RANDOM PIC")
200 CALL LINK("PRINT",153,1,"press I-INPUT
VARIABLE")
210 CALL KEY(5,X,Y)
220 IF Y<1 THEN 210
230 RANDOMIZE
240 CALL LINK("CLEAR")
250 DIM A(30,1):: P=2*PI
260 IF X=73 THEN CALL LINK("INPUT",160,200,N,2,"5")
:: IF N>29 OR N<3 THEN 260 ELSE 280
270 N=INT(RND*26)+4
280 CALL LINK("PRINT",50,50,"ONE MOMENT...")
290 FOR X=0 TO N-1 :: A(X,0)=SIN(P/N*X)*96+128 ::
A(X,1)=COS(P/N*X)*96+95 :: NEXT X
300 CALL LINK("CLEAR")
310 CALL LINK("PRINT",9,20,"N=&STR$(N)")
320 FOR I=0 TO N-1 :: FOR II=I+1 TO N-1 :: CALL
LINK("LINE",A(I,1),A(I,0),A(II,1),A(II,0))::
NEXT II :: NEXT I
330 CALL LINK("PRINT",1,1,"ANY KEY")
340 CALL KEY(5,X,Y)
350 IF Y<1 THEN 340
360 IF X=82 THEN RUN
370 IF X<65 THEN 230
380 CALL LINK("INPUT",1,1,A$,14,"DSK1.PICTURE")
390 CALL LINK("PRINT",1,1,"")
400 CALL LINK("SAVEP",A$)
410 GOTO 330
=====
100 REM TML XB APRIL 1990 STEPHEN SHAW ENGLAND
110 REM FOR TI XB + THE MISSING LINK
120 REM
130 REM after john corbitt
140 CALL LINK("CLEAR")
150 CALL LINK("WINDOW",0,10,193,241)
160 CALL LINK("PRINT",2,9,"When ANY KEY is
displayed")
170 CALL LINK("PRINT",19,1,"press: to:")
180 CALL LINK("PRINT",30,1,"CTRL FCTN....PRINT TO
PIO")
190 CALL LINK("PRINT",66,1,"R run this program
again")
200 CALL LINK("PRINT",78,1,"A save to artist file")
210 CALL LINK("PRINT",131,1,"right now...")
220 CALL LINK("PRINT",142,1,"PRESS SOMETHING...")
230 CALL KEY(5,A,B)

```

```

240 IF B<1 THEN 230
250 REM IKEDA MAP
260 REM from fractal report No 7
270 REM from program by John Corbit 16Aug89 FROM
    Myarc xb+ti99/4a s shaw feb 90
280 REM chaotic attractor map
290 REM
300 X,Y=0 :: P=7.70
310 REM
320 CALL LINK("CLEAR")
330 FOR N=1 TO 4399
340 COSTHETA=COS(.4-(P/(1+(X*X+Y*Y))))
350 SINHETA=SIN(.4-(P/(1+(X*X+Y*Y))))
360 X1=.85+.9*X*COSTHETA-.9*Y*SINHETA
370 Y1=.9*X*SINHETA+.9*Y*COSTHETA
380 CALL LINK("PIXEL",(X1+1)*63+1,(Y1+1.7)*85+1)
390 X=X1 :: Y=Y1
400 NEXT N
410 CALL LINK("PRINT",1,1,"ANY KEY")
420 CALL KEY(5,A,B)
430 IF B<1 THEN 420
440 IF A=82 THEN RUN
450 IF A>65 THEN 420
460 CALL LINK("INPUT",1,1,A$,14,"DSK1.PICTURE")
470 CALL LINK("PRINT",1,1,"")
480 CALL LINK("SAVEP",A$)
490 GOTO 410

```

```

=====
100 ! TI XB plus THE MISSING LINK Turtle type
    graphics
110 REM TML EX BAS MARCH 1990 STEPHEN SHAW ENGLAND
120 REM
130 CALL LINK("CLEAR")
140 CALL LINK("WINDOW",0,10,193,241)
150 CALL LINK("PRINT",2,9,"When ANY KEY is
    displayed")
160 CALL LINK("PRINT",19,1,"press: to:")
170 CALL LINK("PRINT",30,1,"CTRL FCTN....PRINT TO
    PIO")
180 CALL LINK("PRINT",66,1,"R run this program
    again")
190 CALL LINK("PRINT",78,1,"A save to artist file")
200 CALL LINK("PRINT",131,1,"right now...")
210 CALL LINK("PRINT",142,1,"PRESS SOMETHING...")
220 CALL KEY(5,A,B)
230 IF B<1 THEN 220
240 CALL LINK("PUTPEN",142,80,-30)
250 CALL LINK("CLEAR")
260 FOR I=1 TO 3
270 CALL LLL :: CALL LINK("TURN",60):: NEXT I
280 REM
290 REM
300 CALL XT
310 SUB LLL
320 CALL F(15)
330 CALL T(60)
340 CALL F(100)
350 CALL T(120)
360 CALL F(85)
370 CALL T(120)
380 CALL F(15)
390 CALL T(60)
400 CALL F(55)
410 CALL T(-120)
420 CALL F(85)
430 CALL T(120)
440 CALL F(15)
450 CALL T(60)
460 CALL F(100)
470 SUBEND
480 SUB F(X)
490 CALL LINK("FWD",X):: SUBEND
500 SUB T(X)
510 CALL LINK("TURN",X):: SUBEND
520 SUB XT
530 CALL LINK("PRINT",178,20,"ANY KEY ")
540 CALL KEY(5,A,B)
550 IF B<1 THEN 540
560 IF A=82 THEN RUN
570 IF A>65 THEN 540
580 CALL LINK("INPUT",178,20,A$,14,"DSK1.PICTURE")
590 CALL LINK("PRINT",178,20,"")
600 CALL LINK("SAVEP",A$)
610 GOTO 530
620 SUBEND

```

Games Modules

review by Stephen Shaw, England

4A FLYER - GAME-DATABIOTIC

The author was told to write a flight simulator within a tight time limit, in machine code, to fit entirely within 8k. Under the circumstances it is probably not a bad job, but it is instruments ONLY, and if you point your plane at the ground, there comes a point where you reach a maximum speed even with the throttle wide open - and it is not that fast a speed. There are various anomalies and the instrument-only-flying gets rather boring quite quickly. For very little more (in dollar terms anyway) you could have the so very much superior Spad XIII, so get that one and enjoy some scenery!

FROGGER - PARKER BROS

This module is an excellent reproduction of the very popular (well, some years ago...) classic arcade game. It plays very much like the arcade machine. What more praise can there be! If you like Frogger, buy it.

HMMM... do not remember Frogger eh! You must guide a frog over several lanes of busy road without being squashed, THEN across a river, without falling in, being eaten by crocs, or dumped by submerging turtles... and so on.

D STATION 1- DATABIOTICS

This started off life as a disk program from the Bethany based International User Group, and is a VERY simple program, although well written. From a fixed base at screen centre you must shoot down flyers and parachutists. That is about it. Graphics are extremely barren but the game can be difficult. You aim by rotating your gun (180 degree capability) which makes a change from moving left and right at screen bottom.

DEFENDER - ATARISOFT

This is possibly the best Atarisoft module, and is a first rate emulation of the classic arcade game, with a small wide range radar scan at screen top showing where all aliens are, and a playing area to move your ship around while being attacked from all sides from all sorts. The play can become really frantic. This module should be in your game collection. Supplies are generally good but you may need to pay a slight premium price.

TI INVADERS - TI MODULE

Yet another classic arcade game, which may start off a little too slowly for todays hardened games player, but a quick check of the museum piece arcade machines shows that they too start very slowly and take a while to speed up. This does NOT make it easy to clear the first screen incidentally!

This TI Module is almost certainly the very best version of Invaders on ANY home computer. The graphics are, if anything, better than on the arcade machines, and all the arcade play is there.

This is one of only two modules that TI officially released on disk - the disk version had a TEST mode not in the module which allowed you to really slow the game down, and to start at any level UP TO LEVEL 50 (and it went on after level 50 too).

Another module which you really should have in your collection if you do not have it. You may not play it too much, but it is a lovely example of what TI could do. Not hard to find.

JUMPY - TI MODULE

Possibly the ONLY TI Module actually WRITTEN on a TI99/4A system- the others were developed and assembled on larger 9900 computers. You control a cute little Kangaroo who pushes boxes around trying to trap monsters - trapped monsters die and score points. The game becomes surprisingly difficult at level ten. Nice game, fairly difficult, good game action. Reasonable graphics.

MS PACMAN - ATARISOFT

Remarkably like PacMan... Ms PacMan if you forgot is the one with the ribbons! This version has a faster screen set up (PacMan as a module for the TI had a very slow screen set up!) and more varied screen layouts. Also introducing the bouncing strawberry - so much more difficult to collect than a static bonus fruit. You eat dots and avoid monsters in a maze, clear all the dots for next maze, go over power points to temporarily gain the strength to shoo the monsters back home. Better than PacMan, comparable to MunchMan, which seems to play faster.

MUNCHMAN - TI MODULE

Way back when, when PacMan was all the rage and PacMan Fever failed to make the pop charts, TI were thinking of modules for their new Home Computer. A variant of PacMan came out, and Atari jumped on it hard. Gobbling dots was OUT! So a quick rewrite and we instead eat up a trail while the requisite monsters chase us around the maze! This one can get incredibly fast, and you need some pretty good joysticks to play well. If you do not have it, well worth trying.

MUNCHMAN II. A Triton/Databiotics module. 1987

Source: Try offering Database a suitable bribe to import some or try TexComp. Supplies scarce.

It has been brought to my attention that some TI owners, especially disk owners, have entirely the wrong impression of what this module is! You will see from the above review that the original Munchman module was actually version 2, which makes this one Version 3!

This is a complete rewrite by (former?) TI employee John Phillips, responsible for so much module work. Here he has married the two connected screens of Sewermania with the first Munchman variant in which instead of laying a trail you remove one! Yes there is a screen to the right of the first one which you enter through two passages, OR you can use a roving teleport machine! And you must clear BOTH screens to go on! The hoonos are FASTER than you are so you cannot outrun them - except after going over a power dot when you can outrun them and gain good points! You must use skill and tactics in this version. A worthwhile addition to any collection of PacMan variants!

Disk Reviews

by Stephen Shaw, England

COLUMN ATTACK. By Chris Bobbitt.

Written in FORTRAN 99. Asgard Software. US\$10 plus shipping. PO Box 10306, Rockville, MD USA 20850.

Loads with usual Editor Assembler or Extended BASIC loaders.

You may recall a game featured in this magazine, called Spider Bop, which we gave you first in Extended BASIC then in C99. Well this program is very similar but with less graphics, less playability and at least one bug! The defined graphic is your spaceship at the base of the screen, shooting solid lines dropping from the top of the screen. They grow in several columns, at random, and you must shoot them all the way to the top of the screen to remove them. If they reach a lower part of the screen they start shooting at you. If you are underneath when they shoot you lose a ship. There are two columns on my version which cannot be destroyed once they reach the screen bottom, and the only exit is suicide! As an introduction to the capabilities of FORTRAN this is not a good product. It is similar in many ways to 99VADEKS written in TI BASIC by Not Polyoptics, but that had excellent graphics, and high playability, which the slightly faster speed of this game does not replace.

I have wiped my version of this program, I do not think it is worth cluttering up my disk collection even as a novelty item. It is not novel! Not recommended.

=====

TI Artist Companions.

No 1 (5 disks) US\$16. No 2 to 9 (2 disks each) US\$10 per set. Add US\$8 postage per order for overseas insured airmail. TEXANENT'S. 244 Mill Road, Yaphank, NY USA 11980.

These disks are filled with fonts and instances for TI-Artist. Altogether over 100 fonts and more than 400 instances (mostly small but some larger). Quite a lot of artwork ready for you to use, even if you are unlikely ever to use half of it! How do you review fonts? There are lots here of all sorts of shapes and sizes.

It is perhaps worth mentioning that the documentation for Companion #7 contains information not exactly relevant to the disks that go with it. Instances are illustrated which do not appear on the disk and vice versa, and the file list differs from the disk contents. This does not reduce the value of the disks in any way.

TI-Artist itself is \$20 from the same source.

Jenny's Younger Set

Dear Jenny,

I have devised a program that will simulate a rugby league goal kick. To change the speed of the ball, adjust the call sprite and/or the call motion statements. Full instructions on how to do this is contained in the Extended Basic hand book.

I hope people will like this program. Now you are able to see your team kick a goal on computer!

VINCENT MAKER

```
100 CALL CLEAR
110 INPUT "ENTER THE FOOTBALL TEAM YOU GO FOR OR ANOTHER
    NAME-":TEAM$
120 CALL MAGNIFY(2)
130 CALL CLEAR
140 PRINT "*"
150 CALL VCHAR(13,10,30,11)
160 CALL HCHAR(19,10,30,11)
170 CALL VCHAR(13,20,30,11)
180 DISPLAY AT(1,1):"ATTEMPTED GOAL SHOT FOR";TEAM$
190 CALL SPRITE(#1,48,2,176,111,0,0)
200 CALL COLOR(#1,16)
210 FOR T=0 TO 9
220 CALL COLOR(T,2,4)
230 NEXT T
240 CALL MOTION(#1,-4,0)
250 CALL COLOR(0,16,16)
260 CALL COINC(#1,80,111,10,R)
```

continued on page 20

TI-Writer Once Again

by Tony McGovern
Courtesy Hunter Valley 99ers

Several years ago I wrote a HV99 Newsletter article entitled "TI-Writer under the Hood". As far as I know, this general overview of its Editor's internal operation, as distinct from how to use it articles, remains the only published discussion seen here of the Editor's internal workings. This is not a very good reflection on the state of information sharing in the TI-99 community, as quite a few people must have studied its workings over the years, even if only a few have ever seriously attacked it.

Recently I resumed work on a substantial rewrite of the Editor in preparation for updating Funnelweb to Vn 4.20, and while the details are fresh in mind I will write about the operation of the buffer manager in the Editor. For a gentle introduction dig out your old HV99 and read the original article again. This time we will go into a lot more detail, at the level of talking about the routines used, even if we still stay clear of detailed coding.

To remind you of how it goes, the hi-mem segment of CPU RAM contains some buffers, the buffer manager code itself, and starting at >A410 the pile of squished lines growing upwards and the line table growing downwards from >FFCS. When these meet up you have the dreaded "Text Buffer Full" condition. Each entry in the line table is a pointer to the squished text record in the Edit buffer pile corresponding to that line number used as an index into the line table. Records are usually added to the top of the Edit buffer, and if a line is deleted it is removed and the pile collapsed down so there are no gaps left.

The buffer manager code as bequeathed to us by TI contains 4 major functions called as BLWP procedures and a small cast of supporting routines. The main procedures are, to give them TI's labelling, GETLN, INSTLN, UPDTLN, and DELTLN. As you can see from the names these are line oriented. The minor routines are

BINDEC - converts a line number in binary form to the 4 digit decimal you see down the side of your screen.

CKLIN - given the line number it returns the line table and edit buffer pointers and the length of the record.

CKROOM - this is the one that returns the 'Text Buffer Full' if it is not satisfied.

MOVMEM - removes a record from the edit buffer pile by dropping down all the records above it in a mass move of the Edit buffer contents.

SQUISH - this is the one we talked about in last month's Assembly programming contest, that does run length encoding of text lines before they are placed on the top of the Edit buffer pile. SQUISH, and the unsquisher in GETLN, have had a functional update in that the encoding has been improved slightly to give greater apparent buffer capacity. The improvement varies depending on the document file from none at all, through about 3% for a file like this one, to the 5-10% range for assembly source files. Buffer capacity is never decreased. As before buffer encoding is purely internal to the Editor.

These services are called on by the main routines as needed. I will start from the simplest function (not necessarily the simplest code) in looking at how these work. All the buffer management functions in the standard TI-Writer are performed by these routines, and as we shall see, in very inefficient fashion because the basic services are all line oriented.

GETLN - this procedure fetches a specified line from the Edit buffer for purposes such as refreshing the

screen buffer, for Find String etc, and for Saving to disk. It uses CKLIN to find the record in the Edit buffer and then unsquishes the record as part of the service. At the level of detail coding this was the sloppiest effort by TI on any of the routines. I have left the function alone but have refined the code extensively. At the intermediate level the original code is inefficient in that it copies the line from the Edit buffer into the squish buffer before unsquishing to the destination buffer, where it could have been done directly from buffer to destination and now is of course.

INSTLN -- whenever a new line has to be added to the buffer, the INSertLiNe procedure is called. This uses SQUISH to encode the line into the squish buffer, and it is then copied at the top of the Edit buffer pile, with CKROOM testing for "Buffer Full" before this is done. Then the new pointer is inserted into the line table and all pointers above it shuffled along to make room. Nothing too much needed updating in this routine.

UPDTLN -- when you edit a line and changes are made, it has to be replaced in the Edit buffer by the UPDateLiNe procedure. This fetches the data for the existing line with CKLIN, SQUISHes the new line, and then checks the length against that of the existing line. If this is unchanged the record is written back over the earlier version without further ado. If the lengths do not match then it has to delete the existing line and insert the new line. A subtle improvement on the original is that it now uses only the increase in length when it calls CKROOM.

DELTN -- when you use <Fctn-3> or <Ctrl-N> to delete a single line DELeTeLiNe does the job. For normal documents this is the most expensive operation of all, because when a line is deleted from the Edit buffer the whole buffer above where that line was stored must also be deleted and the rest of the table moved along, with pointers to any record that has been moved adjusted appropriately. Yes, all that goes on every time you delete or even just edit a line. Since the line table has only a 2-byte word aligned entry for each line (to give the CPU memory address of the actual record) it is usually much shorter than the Edit buffer. However just how much of the Edit buffer has to be moved depends on past history. A well ordered buffer is one in which the records start from the bottom at >A410 in order of their line numbers. A file freshly loaded from disk is in this state. Editing any line enough to change its length will, as we have seen already, end up with it pulled out of sequence and placed at the top of the pile.

TI-Writer uses these buffer manager routines in the form that we have described them, always in a line at a time fashion. The most annoyingly slow function of all in TI-Writer is Delete Lines and this was the one I decided to attack first as it is also called by other functions. Why is it so slow? The reason is that each time a line is deleted, it is done as a separate task and a complete mass move of the Edit buffer and update of the line table is performed for deletion of that line. Code polishing improved this noticeably, but a new approach is clearly needed. So a new procedure was added.

DLBLOC -- when given a range of line numbers to delete this keeps checking successive lines to see if they form a contiguous block in the Edit buffer. Each such block is deleted as a single unit from both Edit buffer and line table. So with a well ordered buffer even large blocks are now deleted almost instantaneously. You could imagine a large buffer in the opposite of a well ordered state - then the process would proceed a line at a time, but usually there is some degree of block ordering. What you can see is that TI's original method is guaranteed to give the worst possible result every time. Oh well, they invented GPL too! DLBLOC is now also called from Reformat when it has finished making a reformatted copy of a paragraph to get rid of the original paragraph, and so Reformat is now speedier.

Move Lines and Copy Lines are also very much slower in TI-Writer than one might reasonably expect them to be. Once again this is because line by line services are used to construct block related functions. Apart from speed a bad failing of Move Lines is that it can give rise to a "Text Buffer Full" condition. If you think about it this should not occur as no text is being added. It happens because TI-Writer implements Move Lines by first doing a Copy Lines - which is where the full buffer disaster may strike, and then deletes the original lines, adding insult to injury by doing it one at a time always. At this stage DLBLOC had already speeded things up very noticeably, but clearly something much better is needed. In principle all that is necessary is to shuffle around blocks of the line table, and the Edit buffer need not be altered at all. So a new general purpose support routine, MVLINS, was written that could be called as whole or part of buffer functions.

MVLINS -- this is a routine that does the line table block shuffle. It is written as a generic routine, and the block limits are pre-ordered by the calling procedures, William came off his Amiga for long enough to make an initial suggestion on how to do it in linear time in block size, and I think the resulting code is the most elegant piece of 9900 code I have yet written.

TI-Writer did Copy Lines by using GETLN to fetch each line and INSTLN to add a copy to the Edit buffer and line table. Once again this is about as inefficient a method as could be dreamed up, and made worse by all the unsquishing and resquishing involved. A further bad feature is that if the buffer fills during the process you are left with a partially copied mess.

CPBLOC -- the obvious solution here is to do direct copying of the lines from their existing position in the Edit buffer to append them to the end of the buffer, and to add new pointers to the end of the line table. In this process a new entry point CKCOPY to CKROOM is invoked which checks for room in the buffer before copying each line. At this stage the copy process can still be abandoned leaving everything as it was since nothing has yet been done inside the previous Edit buffer and line table limits - only additions to the ends. If all is satisfactory then the end pointers for the buffer and line table are updated. The Edit buffer is now in fine shape but we have a whole block of line pointers out of position, unless they were originally meant to be added at the end. But all it takes now is to call MVLINS with the right pointers and the job is all done.

An aspect of Copy Lines that bears a little thinking about is that the new approach would make it possible to do a Copy Lines with the "After" line number in between the "From" and "To" line numbers. This could be done because the copy of the line is made before any reinsertions are made, but it is not implemented because the Copy Lines and Move Lines call a common routine to take down the line numbers and do range checking. The conditions for Move Lines are more restrictive of course.

A further modification to the buffer manager code which has not yet been incorporated, is to abandon the fixed squish buffer and float it at the top of the Edit buffer, with appropriate modifications to the code. There is no net gain in buffer capacity, but since lines are always added to the top of the Edit buffer (except for the direct replacement in UPDTLN) it will mean that the squished line is already in place without having to be copied into position again. Strictly speaking it is only the direct replacement path in UPDTLN that forces provision of an explicit squish buffer, since you cannot tell that a replacement line is the same length until it has been squished. The other major area besides the buffer manager that will bear looking at is the Load/Save module. The change just discussed will help this along by reducing overhead in the file reading process. Another way to help it along was discussed as an aside in the Living with Spiders series on interfacing to Funnelweb published in the HV99

newsletter many months ago. This is the technique used in the E/A Utilities object loader, and extensively exploited in the LINEHUNTER utility program that has been part of the Funnelweb package for what seems like very long time now. The idea is to use the full DSRLNK only for the initial access to a device, usually a disk DSR, saving the CRU base address and the entry address, and using these directly for future accesses to the same file. Linehunter gets its speed by keeping track of these separately for both MAIN and COPY files.

The revised buffer manager has been incorporated in the 40 column version also, and the speed-up is such that it now seems impossible to lose a keystroke on end of line automatic wordwrap. It is still just possible to lose a repeated character in the AVPC version, but remember that this has to write up twice as much on the screen in between each keystroke, and do it via the 8-bit bus to the PE-Box as well. I must compare this against the Mechatronics 80-col unit sometime. Another upgrade to the 40 column version is that <Ctrl-Y> now functions as a full margin release as has been the case in the AVPC version.

Another new feature which is proving its worth right now in writing this document is the provision of dual tab settings. The command line entry "DF" is now obsolete as SD does the job better, and has been replaced by "ST" for Swap Tabs, which swaps the current tab set with an alternate set. So when I want to write inset paragraphs all I have to do is go to command mode, ST, and Enter. Both tab sets are now saved with the document and if an old document with only one tab set is read in this is made the initial alternate set also. In the AVPC version the current tabs show on the line 26 ruler when called for editing by T or ST. Pressing <Ctrl-> cleans the ruler line to numbers only. I have also added a NTSC/PAL toggle from <Ctrl-N> in the command mode. The control key may change but it seems a worthwhile inclusion (assuming your monitor will handle it) as I find it easier on the eyes to read the less elongated screen characters with less "lininess" to the display. Interlace mode becomes pretty horrendous though at 50 Hz frame rate. I still have to sort out why the output from the 9938 seems so much worse than standard TV.

Users of the Programmer's Editor will find a whole new mode of operation. In the Word Processor pressing <Ctrl-O> toggles the cursor between the solid word-wrap shape and the hollow cursor to indicate fixed mode. Previously the Pgm/Ed allowed only a modified form of fixed mode to give E/A compatibility of saved files and to make sure no reformat could ever be done on program source files. Now <Ctrl-O> toggles in a diamond cursor to indicate that you are in assembly source mode (ASMode from now on). It has always been a nuisance switching between upper case for the assembly code itself and to lower case for the comments. In ASMode you may leave the alpha lock off, and type both code and comments on the same line in lower case, with <shift> needed only for capitals in comments as in normal typing. It does not matter that the assembly instructions are in lower case because ASMode converts them to upper case before updating the line in the Edit buffer. I recall from the brief look we had at My-Word for the Geneva a year or more ago that a "C" could be added to the tab line to control conversion to upper case. Atrax Robustus would never be satisfied with anything so inflexible. What ASMode does is to parse partially each line as an assembler source line. Comment lines are ignored, and the line parsed for the first three fields (two if no label starts in the first column) separated by blanks. Some error checking is done to catch a couple of common typing errors that can be difficult to find by eye. It indicates these by giving a little audible error bloop as the line is converted, and ceases case conversion at that point to make it more obvious. In the Opcode field only alpha characters are allowed as all legal opcodes and directives contain only these - it dislikes things like Ll or SOCB. In the Operand field ASMode leaves any entry between single quotes unaltered - it is not allowed to touch your text data strings. It also checks for an uneven number of single quotes. Further it

continued on page 6

Living with Spiders part 1

by Tony McGovern
Courtesy Hunter Valley 99ers

What does it take to coexist with the infamous Atrax Robustus, the funnelweb spider? Mainly it needs a certain amount of discretion about walking out in the yard at night in bare feet, care in gardening, and remember always to shake out your rubber boots before putting them on. Less seriously, though, what we want to do here is give a workthrough on how to write assembly programs to live in spiderland. The Funnelweb system I suppose can be categorized as something between a new operating system for the TI-99 and a shell for what already exists there in various forms. TI was trying to sell lots of cartridges and either did not care to tie it all together at the user level, or even took definite steps to prevent it. The expanded TI-99 system has lasted so well since Orphan Day 5 years ago because it was thoroughly engineered behind the scenes, and outside programmers have been able to draw on this underlying strength. The Funnelweb system is an attempt to gather those strengths in a form convenient for the general user. Perhaps the heading on the XB LOAD program of "Utility Environment" sums it up best.

Whatever you call it though, it can provide improved user convenience for programs written to the normal rules. If you further respect its own internal rules, just as you would do with the DOS on any other system, then better programs still can be written. The Funnelweb package contains several utility programs which do just this, some usable only with Funnelweb, and some which can run freestanding. This article is intended to give guidance on writing dual-mode programs which can function with or without FWB. Two examples are the DM-1000 files MG/MG and Linehunter LH which will be used as a detailed example in the Vn 4.12 update.

The requirements for dual-mode programs are that they

- (1) Observe FWB style
- (2) Know the load path
- (3) Do not tread on Atrax R.
- (4) Make a graceful exit

Let us look at these in more detail. The first item is not essential but is nice to have. A good example to follow is the way FWB has in its turn followed the TI-Writer Editor in making <Ctrl-C> an alternative to <Fctn-9> for BACK or <escape>. Once using <Ctrl-C> has become familiar, <Fctn-9> seems like the imposition of an awkward stretch. A FWB convention to save more awkward stretching is to use <Ctrl-A> as alternative to <Fctn-6> for PROCEED. Think of A for Action as mnemonic.

The next hurdle is for the program to know whether it has been loaded from FWB so that it can call on details or routines from FWB. What sort of details? One is to know what the screen colours were so that these can be retained and a jarring transition to a standard colour set avoided. Remember that FWB gives great flexibility for you to choose your favorite colours of the moment. Another is to know what drive the program was loaded from, and if complete reload of FWB is needed, to know where FWB was lurking also. The nature of a dual-mode application means that the special FWB information will be used only at entry and exit, the main exception here being the availability of QD from <Fctn-7> AID as a "hot key" consistent with normal FWB usage as per item #1.

Next we want to avoid conflict with FWB, or if this is unavoidable to know how to restore FWB before exit. The primary requirement is for FWB itself to remain unmolested. This means that the program must leave the top 5K or so of hi-mem alone, more precisely from >EBC8

to >FFD7. The UL file resides at >E9B0 when it is in memory and is allowed a length of >218 bytes. FWB does not define XOP 1 but the E/A manual cautions that this is not necessarily available on all consoles in any event. If QD is to be called then this extends down to >CE00 to cover QD and its RAM buffers. QD also keeps various information buffers in VDP on a temporary basis during use which might affect your program. VDP is used from >17FE through >19FF and from >2B70 through >3359. QD's memory usage may seem strange but remember that it has to live with XB LOAD, FWB itself, and the Assembler and Formatter too. The final area of memory that needs to be preserved is the FWB "mailbox" from >A000 to >A04F which is used to preserve and transfer the DV/80 Workfile name. For program files this is most easily handled by AORging your code suitably. XML addresses at >2000, >2002, and >8300 may be used with various modules and may need to be preserved depending on the details of your exit code. FWB programs always follows TI-Writer practice and preserve the contents of >8300.

A dual-mode program will most likely be an E/A Option 5 Run Program File or its FWB equivalent, Loaders Option 2 GPL Environment, which does not assume E/A utilities are present. As an alternative it could be a DF/80 Object File for Load & Run, but we will discuss the program file format here. The code examples will be drawn from Linehunter so that they reflect an actual working program.

The extracts, with nonessential minor changes, from the LH source which follow cover a number of complexities which may not always be found. The chief of these is that FWB, E/A, and QD work in Graphics mode while LH is in Text mode. Some of the details are of course specific to LH.

* Funnelweb system equates

```
QDLEN EQU >A40
MLBOX EQU >A000
QDPOS EQU >CE00
QDSAV EQU >FF20
QDLOAD EQU >FF3C
QDXFL EQU >FF52
CMSRET EQU >FF5C
FWREGS EQU >FF7C
LDR11 EQU >FF9C
```

* Loader entry point

```
AORG >A050
B @START
```

* Data block

```
MAINWS BSS >20
SAVRET DATA 0
QDFL DATA 0
FWEB DATA 0
VDP7 DATA >F487
DSID TEXT 'DS'
K TEXT 'K.'
```

```
B20 BYTE >20
AID BYTE >01
EVEN
```

*

* Real entry

```
START LIM1 0
MOV R11,@SAVRET
C R11,@LDR11
JNE NOFWEB

SET0 @FWEB
LWPI MAINWS
MOV @QDSAV,@QDFL
JNE STARU
```

```

MOV @QDLOAD,R1
BL *R1
DATA 'QD'
DATA QDPOS,QDLEN
JMP STARU
SETO @QDFL
STARU MOV @QDFL,@QDXFL
LI R9,>380
BL @VAD
MOVB @VDPRD,@VDPR7
BL @LOADNB

```

* Continue all paths

```
NOFWEB EQU $
```

First thing to note is that the code is AORGed above the mailbox area, and the first word is an executable instruction as required by generic program file loaders. Here it branches to the real start of the program. Just to be sure we turn off interrupts, and the interrupt hook should also be cleared as you can not guarantee that all loader programs clear this properly, though if you have to clear the hook, it is probably too late. This is not necessary for E/A or FWB. The next thing to do is to save the return address in R11. This is not strictly necessary if your final return does not use it, but here we do in order to cover as many exit paths as possible. Now we determine if the program was loaded from FWB. The FWB program file loaders always hand over with R11 loaded with the re-entry address also stored as a data item at LDR11 (>FF9C). This is good enough to tell FWB from anything else, such as an E/A load while the FWB code is still all or partly in memory. If the comparison fails the FWEB flag is left at its load time initialized reset state, and the code jumps to the common continuation at NOFWEB. If it is FWB, the flag is set, and the program's own register set loaded. At this stage no registers have been written to, so even if the program had been loaded as FWB Opt 3 with workspace at >20BA and had loaded over that, it would still function correctly.

When FWB loads a program it has no way of knowing if the QD code has been overwritten, so it always resets its internal QD flag at QDXFL (>FF52), but preserves it at QDSAV (>FF20) just in case the program loaded can use the information. The flag is saved and tested. If not null then QD was already loaded and we can jump to the next item. If null then we try to load it using the special utility file loader in FWB (used for things like QD, EA, UL, LL etc). A pointer to this routine is stored at QDLOAD (>FF3C) and it requires as inline data the 2 letter filename, the CPU load address, and the file length. This routine looks on both the primary and secondary FWB drives before giving up. The 6 bytes worth of file header details are ignored and the code is not executed as in a normal loader. The following JMP is the error return, and here it just steps over setting the QDFL flag. At label STARU the flag is returned to FWB so that if QD was already loaded or has been reloaded, then FWB knows it immediately on return and does not bother to reload QD if it is summoned.

Now the FWB colour is read. This particular program happens not to have an explicit VSBR utility, so the first byte of the colour table in normal E/A position is read and stored for use in setting text mode later, and for rewriting the E/A colour table on final exit. Non-FWB loads will use the default value. This little bit of code helps give that seamless effect which we have been trying to achieve in the FWB system software.

The final subroutine call to LOADNB checks the mailbox and loads a valid name into various file name buffers. The actual code is further on. It checks that "DSK" and "." are there and if so transfers a block containing the name into various buffers. The drive number itself is not checked, and the form RD. for Myarc RAMdisks has not been explicitly catered for, but could easily be added. Length byte derivation is done elsewhere in this particular program. The exact details are up to you.

```

* Key returned in RO
* AID call for QD

```

```

CB RO,@AID
JNE NXKEY

```

```

ABS @QDFL
JBQ NXKEY

```

```

BL @DPREP
STWP RO
BL @QDPOS
BL @RESUME
JMP WARM

```

```
NXKEY CB RO,...
```

This next piece of code is the call sequence for QD disk directory. The key routine has returned the key value in RO with Keyscan 5 assumed. If QDFL is not set then the QD code is ignored. The next routine DPREP, code listed later, is necessary only to set up graphics mode for QD because LH is in text mode. Such a routine would not be necessary in a graphics mode program but if sprites are defined it would be necessary to shut them off and restore them on return. QD saves and restores the graphics screen >0->2FF itself but more is needed for text screens. Sprites are shut off, graphics mode set, and the E/A colour table rewritten. The next instruction STWP RO is a special one for QD, to render harmless an action set up for some special load paths. Then BL to the QD code at QDPOS (>CE00). QD does not interfere with the caller's workspace. The RESUME routine restores the full text mode screen, resets text mode, and in this application writes the marked filename into the buffers. After return it branches where needed.

* Exits

```

EXIT ABS @FWEB
JNE FWEX

BL @CLRSCR
BL @EXITEA
LWPI GPLWS
MOV @SAVRET,R11
B @>6A

```

* FWB exit path

```

FWEX LI R1,>50
FWEX05 MOVB @B20,@MLBOX-1(R1)
DEC R1
JGT FWEX05

LI R1,25
FWEX10 MOVB @FIL2-1(R1),@MLBOX-1(R1)
DEC R1
JGT FWEX10

LWPI FWREGS
MOV @CMSRET,R11
SETO R13
SETO R4
RT

```

This code excerpt shows the exits. If the FWEB flag is not set then it executes a normal return. There are enough different loaders around now which do not handle standard exits gracefully that the simplest thing might just be a return to the title screen. The first instructions set graphics mode again to make the return to E/A look smoother. The FWB return first sets the FWB workspace, and to leave fetches the central menu screen return, and does some housekeeping in the funnelweb's burrow. SETO of R13 is necessary and reminds FWB that it was a utility load and not a return from the Editor for example. SETO of R4 flags the E/A side of the Central Menu screens and CLR of R4 would re-enter on the TI-Wr side.

The other code before the exit writes spaces in the mailbox out to >A050, and then returns a filename from a buffer. In this case the filename is already padded out with spaces, otherwise the length byte would be used. Again, this is specific to this particular program, and you would write code to suit your own.

* VDP address set

```
VAD EQU $
    SWPB R9
VAE MOV B,R9,@VDPWA
    SWPB R9
    MOV B,R9,@VDPWA
    RT
```

* Directory prepare

```
DPREP EQU $
    BLWP @VMBRD
    DATA >300,HIBUF,>1C0
EXITEA MOV R11,R10
    LI R0,>D000
    BLWP @VFILL
    DATA >300,1
    LI R9,>E081
    MOV B,R9,@>83D4
    BL @VAE
    MOV B,@VDPR7,R0
    BLWP @VFILL
    DATA >380,>20
    B *R10
```

* Resume after QD

```
RESUME EQU $
    MOV R11,R10
    BLWP @VMBWD
    DATA >300,HIBUF,>1C0
    LI R9,>F081
    MOV B,R9,@>83D4
    BL @VAE
    BL @LOADNB
    B *R10
```

* Load filename buffers

```
LOADNB EQU $
    C @MLBOX,@DSID
    JNE NOFN

    CB @MLBOX+2,@K
    JNE NOFN

    CB @MLBOX+4,@K+1
    JNE NOFN

    LI R1,25
BOXNM MOV B,@MLBOX-1(R1),@FIL1-1(R1)
    MOV B,@MLBOX-1(R1),@FIL2-1(R1)
    DEC R1
    JNE BOXNM

NOFN RT
```

This has been an example where FWB remains in memory. The next article will look at code for reloading FWB when you know your code will destroy FWB in memory. ○

For Sale

Lou Newhouse still has three (3) new MiniMemory modules for sale at \$40 each. Phone Lou on (02)639 8888 during business hours.

For Sale

A Daisy Wheel printer including tractor feed, 15 inch carriage with superb quality print. \$350. Phone (042)284906 after 6.30pm.

Cataloguing with Multiplan

by Bob Relyea

The other night I had one of those experiences where I thought I had accidentally made an original discovery. I was placing classlists on a Multiplan spreadsheet and was periodically using the Transfer Load option, of course, to carry out the process when it happened. I accidentally pushed the 'wrong' keys and lo behold a series of titles appeared on the screen which I eventually recognised as a catalogue of the disk in drive 2. I excitedly experimented for a while and afterwards looked at the table of contents in the Multiplan manual for any reference to cataloguing or getting a directory on a disk. I searched in vain. The next day I decided to look in the section in the manual under Transfer Load and there it was right before my eyes - clear directions on how to get a catalogue of a disk. Somehow I had never seen this before and began reading to see what I had missed. There were a few points that I picked up from the reading and subsequent experimenting that I had not been able to work out on my own. What follows is an edited portion of the manual on page 153 dealing with the matter. I trust that this is of use to somebody.

* After you have pressed T for (T)ransfer and then L for (L)oad, pressing any one of the direction keys (e.g. Fctn S) causes Multiplan to display a directory of files on the diskette. The direction keys may be used whether the "filename" field is empty or has a filename filled in.

* When the "filename" field is empty, the whole directory is displayed.

* When you enter a filename in the "filename" field and then press a direction key, Multiplan uses the filename you enter as a pattern and displays only those filenames on the default diskette that match the pattern. Frankly I do not understand this part possibly because I am not sure as to what is meant by 'pattern'. Does it mean file type such as DV 80? At any rate I had a disk in drive 2 that had both the Multiplan files as well as the DV 80 files and it always displayed both types no matter what I did. If somebody out there has worked out what they mean please enlighten me! Ed.

* Once the directory is on the screen, use the direction keys to move the highlight among the filenames. As you do, the highlighted filename also appears as a proposed response in the "filename" field in the command line. Press the ENTER key to load the highlighted file.

* Multiplan can display up to 18 filenames on one screen. To scroll to additional filenames:

1. Select the Transfer Load command and press one of the direction keys to display the first 18 files filenames.

2. Type any filename that appears near the bottom of the display, then again press one of the direction keys. The filename you typed will be the first filename at the top of the next screen of filenames. Editor's Note - the word type is the key word to get this to work as they indicate. There will be a filename already in the "filename" field that matches the one the 'cursor' is on, i.e. the highlighted one. If you simply press a direction key straight off you will not get the remaining part of the filenames. You actually have to type right over what is already there to get a favourable response. After you have typed the first letter of the filename, pause briefly because the previous directory will disappear and the normal screen (blank or otherwise) appears. After this brief pause finish typing the filename and press a direction key and the remaining files will appear (or at least the next 18).

continued on page 8

Cracked! part 2

by Cracker Jack

Ahoy there shipmates!

After the anti-LIST item we overcame last month, I felt it worth mentioning a similar protection nasty you may encounter. This one has a remarkably identical effect to that of last months. You remember that when we typed LIST we got a flash of colours on the screen then a computer lock-up. Well matey, if the last fix did not work then we have another you can try.

To give a little background as to how this protection works we will construct an example of how it could be set up. Set your looking glass on the following program.

```
100 GOTO 120
110 REM
120 [program code] :: GOTO 140
130 REM
140 [program code] :: GOTO 160
150 REM
160 [program code] :: etc.
```

You will notice that the REM statements are never actually in the program flow however they are scattered throughout the program listing. The GOTO's skillfully steer around the REM's meaning the BASIC interpreter will never see them. If we now take a peek below deck into the actual memory ships we find a table. The navigation points of this table can be found at latitude >8330 and longitude >8332. Upon this table we find a treasure map of line numbers followed by location points of buried booty. Such a treasure map for our fictitious program above might look something like this :

```
:
00A0 = line number 160 (decimal)
E945 = pointer to program code
0096 = line number 150 (decimal)
FF22 = pointer to program code (REM)
008C = line number 140 (decimal)
FF26 = pointer to program code
0082 = line number 130 (decimal)
F29A = pointer to program code (REM)
:
etc.
```

It is at this point where the tampering of the map takes place. Imagine if the pointers to the REM statements were altered to point to the deadly waters of the Bermuda Triangle.

```
:
00A0 = line number 160
E945 = pointer to program code
0096 = line number 150
---> 6000 = pointer to Never Never Land
008C = line number 140
FF26 = pointer to program code
0082 = line number 130
---> 6000 = pointer to Never Never Land
:
etc.
```

Remember line number 150 and line number 130 were REM statements that were never executed when the program was RUN. Now the pointers for those lines are changed to point to the ROM area of the Extended BASIC module. Now when we type LIST the pointers send the LIST routine into the Never Never Land of the Extended BASIC ROM to look for the program code. What happens then is history.

When confronted with a program protected in this way then the task is to remove the unnecessary line numbers and pointers from the line number table. The cure for this problem is the following code (my suggested object code name "AHOY") which swabs out the unwanted line numbers. Have your Extended BASIC module

inserted and your console and expansion memory ready and fired up. Load that barnacle of a program that refuses to LIST. Now also load the "AHOY" code thus : CALL INIT :: CALL LOAD("DSK1.AHOY") and then CALL LINK("AHOY").

That should swab the crustacea out of that program and enable it to be LISTed in the normal way. I am afraid the only way you can tell which form of protection was used on a program is to start peeking at memory. Alternately try each Cracker Jack virtuous vaccine method until you hit on one that works.

```
* This routine searches the line
* number table for pointers which
* are outside the high expansion
* memory area. These false line
* numbers and pointers are then
* removed from the BASIC program.
* Requirements : XB + 32K.
* Use : Load XB program and this
* routine. To use type -
* CALL LINK("AHOY")
*
```

```
DEF AHOY
*
AHOY LWPI DECK
MOV @>8330,R1
MOV @>8332,R2
MOV R1,R4
CLR R3
*
THERE INCT R1
C R1,R2
JHE MATES
MOVB *R1,R3
INCT R1
CI R3,>A000
JHE THERE
MOV R1,R5
AI R5,-5
MOV R1,R6
DEC R6
*
SHIP MOVB *R5,*R6
DEC R5
DEC R6
C R5,R4
JHE SHIP
AI R4,4
JMP THERE
*
MATES MOV R4,@>8330
CLR R3
MOVB R3,@>837C
LWPI >83E0
RT
*
DECK BSS 32
END
```

continued from page 22

I am sure that some young genius will soon take advantage of this technique to create some truly mind boggling graphics on our TI screen.

continued from page 14

```
270 IF R=-1 THEN 290
280 GOTO 260
290 CALL MOTION(#1,5,0)
300 CALL COINC(#1,192,111,20,S)
310 IF S=-1 THEN 330
320 GOTO 300
330 CALL DELSPRITE(#1)
340 A$="8898FF3D3C3C0404"
350 B$="8898FF3D3C3C0404"
360 CALL CHAR(96,A$)
370 CALL CHAR(97,B$)
380 CALL VCHAR(16,13,96,1)
390 CALL VCHAR(16,17,96,1)
400 DISPLAY AT(5,7):" A SUCCESSFUL GOALKICK.
TWO POINTS."
410 GOTO 410
```

Assembly Squish

by Tony McGovern
Courtesy Hunter Valley 99ers

This article poses a little puzzle and challenge for those HV99 members who know or are learning assembly language (the only way to fly on the TI-99). Perhaps there will be a small prize for the best successful attempt, though the main prize will be the eternal glory from being recorded in these pages as the winner. Well ... it may not be precisely eternal, but it will do for as long as the TI-99 keeps going enough to have a HV99 UG interested in it.

The background to the problem is found in the SQUISH routine of the buffer manager in TI-Writer. With all the ROM space assigned in the TI-99's memory map, there is not all that much RAM available, only 32K in the normal machine, and the Editor and all its text has to be squeezed in. One way TI-Writer does this is by squishing the text held in the memory buffer into a run length encoded form. A line number table grows down from the top of high memory, each entry being a pointer to the start of a squished record kept in a pile growing upwards from above the buffer manager code at the bottom of hi-mem (>A410 is the usual start). The records may be in any physical order in this pile and no gaps are ever left in it. Records are always added at the top, and if any are deleted the whole pile above that is dropped down.

We are not going to attack all of that problem, even though TI's code could stand a lot of improvements if only there were a bunch more bytes to spare. Atrax R. is working on it. The immediate challenge or exercise is for you to see if you can do better than TI did in writing the SQUISH routine itself.

So what is it and what does it do? It is a BL routine called from a couple of places that is given a pointer to a 80 character raw record (no length byte, just the 80 chars including all the blanks out to the end) in CPU RAM somewhere, and returns the squished version in a standard buffer area. The raw record may be the result of keyboard entry, or else a record read in from disk by LF. Records are stored in unsquished form on disk in DV/80 files and so may be shorter than 80 characters, in which case the raw record buffer is padded out with blanks to the full 80 bytes when the record is read back in.

A squished record consists of a length byte for the record which includes the length byte itself, followed by the line in an RLE form. Characters which are not repeated are stored just as they are, but repeated characters are followed by a byte containing the number of characters in that particular run of repeats. How does the unsquisher know a character is to be repeated? SQUISH flags this by setting the most significant bit in repeated characters. As an example a repeated space becomes >A0 instead of >20 for a single space. This works because the normal ASCII characters accepted by the editor do not have the MS bit set. The SQUISH routine has to guard against full 8-bit bytes being slipped in from external disk records by stripping off the MS bit from all characters, otherwise the unsquisher could get dangerously confused.

A totally blank line encodes to the hex bytes

```
>03,>A0,>50
```

A line consisting of A BB .. starting at the far left in column zero is stored in the text buffer as

```
>07,>41,>20,>C2,>02,>A0,>4C
```

You can see the sort of thing that happens from these examples. The squish buffer can accommodate the maximum possible total length of 81 bytes. This is not necessarily the only way to do things, but it is the TI-Writer way so we will stick to it for present purposes.

The challenge then is to write a SQUISH routine that can replace TI's original version, but to do it better than TI did by making it shorter. SQUISH expects the pointer to the raw record 80 character buffer to be held at address BUFADR, and that the squish buffer starts at SQSBUF. To match the TI routine the following restrictions MUST be observed.

- (1) The routine is called by BL @SQUISH
- (2) On exit R3 must point to SQSBUF.
- (3) On exit R6 must contain the length byte of the squished record, right adjusted as a word.
- (4) Registers R0,R3,R6,R7,R8,R9,R10,R12 are available for use. Contents of other registers must be preserved.
- (5) If needed the data word >8000 is available at address H8000.
- (6) The MS bit must be stripped from all characters from the raw record.
- (7) The maximum length allowed for the body of the routine is 45 words (>5A bytes).

As a matter of convenience for evaluation and comparison use R8 for your index into the raw record buffer and R9 as your index into the squish buffer. The assembly source will then have the form

```
SFIRST EQU >A000
SLAST EQU see text below
DEF SFIRST,SLAST
BUFADR EQU >205E
SQSBUF EQU >A198
H8000 EQU >A1EE

AORG >A22A

SQUISH EQU $
RT
```

Your code should be fully commented of course so that the judges do not have to strain their little brains too hard. Entries should also include a pseudo-code description of the program logic, and in the event of a tie on length achieved the decider will be quality of commenting in the source code. As a goal to shoot for, Atrax R. had it down to 38 words with no external data needed.

The SFIRST and SLAST EQUates and DEFs are included to allow you to test your code. To determine SLAST use DPATCH or other sector reader to inspect the first sector of the EE program file in your Funnelweb system. The second word is the file length and should be added to >A000 to get a value for SLAST. For instance on the AVPC version of EE I am using now this length is >1422 so that SLAST would be >B422. The normal 40-col version is a bit shorter. So the suggested procedure is

- (1) Make up a working disk containing FW (or as UTIL1) configured for the drive you will have it in, and Editor files ED,EE. Your Funnelweb system disk itself should have LL as part of the usual system files, and FMSAVE should be available. It might be a good idea to put file protection on your ED,EE system files on your Funnelweb disk to prevent possible accidents.
- (2) Write the source code for your SQUISH routine using the Funnelweb Programmer's Editor, and check its length by counting off the words in the source code.
- (3) Assemble this code. If there are any errors go back to (2) and make the necessary corrections.
- (4) Go to the Funnelweb Loaders screen and choose program file loader, Option 2.
- (5) Load the EE file from your Funnelweb system

disk. It will return immediately to the Loaders screen.

(6) Immediately select Option 6, Low-Loader and load the object file you created in part(2). If the name of your object file does not come up automatically, then it is high time to update your Funnelweb system. Then load FMSAVE. If you make any blunders start at (5) again.

(7) <Enter> to go to the DEF display and select the SAVE entry. SAVE as DSKx.EE where "x" is the working drive #. If you have only a single drive remember to switch disks.

(8) Go to Loaders Option 2 and load FW (or as UTIL1) from your working disk. Then load the Editor

If all has been successful it will run normally. If not - back to the drawing board. This article will appear in the August 89 issue, and closing date for entries will be the last Tuesday in October to allow time for out of town members to participate. The winner will be announced at the November meeting, the last of the year. Atrax R. will be the judge. If there is at least a reasonable degree of interest in this kind of activity there will be future problems posed as little contests in the future. Perhaps one of our XB programmers might care to put up a similar kind of challenge/puzzle, say to write a sub-program to do some clearly delineated task with a host program to merge it into for test, so that a wider range of members can participate.

Happy coding !!!

Assembly Squeezing part 1

by Tony McGovern
Courtesy Hunter Valley 99ers

When you first start writing assembly code you find that initially there is a vast arena to work in. As projects get more ambitious the crunch eventually comes, or you may have to squeeze some code into a gap that just seems too small at first. Then it is time to scrunch down your code to fit. In this series of short articles we will look at various little tricks that can be used to save the bytes or words here or there. There are larger issues to be considered, of program organization and identification of repetitive code segments, but we will just play around on the microscopic scale.

For our first topic let us look at the problem of advancing register pointers. This is often handled directly by auto-increment addressing

```
MOV *RO+,*R1+
MOVB *RO+,*R1+
```

Now suppose you want to advance a pointer by 4. The obvious solution is to use

```
AI RO,4
```

This takes 2 words or 4 bytes. A shorter solution is to use

```
C *RO+,*RO+
```

This gets the job done in only 1 word, saving 2 bytes on each use. The comparison C only reads the data pointed to, and you just ignore any setting of comparison flags. Like most things that shorten code, it takes more machine cycles to execute, but this would hardly matter except perhaps in the innermost heart of a frequently executed loop. One little trap in the TI-99/4a is that some memory mapped devices respond even to a read from their address, so you should be sure that RO does not point to such an address. This is rarely a problem though.

It is possible to come up with some more little byte-savers along these lines. Sometimes you have to keep up with incrementing two register pointers simultaneously, and the code is such that auto-incrementing the register is not done. This often occurs when maintaining pointers to addresses in auto-incrementing devices such as GROM or VDP RAM. Suppose the pointers are in registers R8 and R9 and both must be incremented. The obvious coding is

```
INC R8
INC R9
```

which takes 2 words. Alert readers will already have guessed a shorter method

```
CB *R8+,*R9+X
```

to increment both in one word of code. So if you come across strange comparisons being made, remember that comparison may not have been the intention.

4-D graphics

by Jim Peterson, Tigercub Software

Those of you who remember your first lesson in geometry are aware that a straight line has only one dimension, that of length. Ignoring the necessary breadth of one pixel, this can be programmed on the TI by CALL HCHAR(12,1,95,32).

Now, if you fix that one-dimensional line at one end and rotate the other, you will describe a circle, which is of course a two-dimensional figure having length and breadth. This too is easily programmed on the TI using its built-in SGN function.

Proceeding in logical sequence, if you fix that two-dimensional circle at two points and rotate it, you will describe a three-dimensional globe having length, width and breadth. The programming of this will require a slightly more complex algorithm and the radius should be limited to 14 units, since the TI-99/4A screen has only 29 planes.

Proceeding further in logical sequence, if you fix this 3-dimensional globe at three points and rotate it, you will obviously describe a four-dimensional figure. The algorithm required here is somewhat beyond the limits of my high-school geometry, so I will leave it to some other programmer. The first one to publish this routine will have performed a valuable service to the TI community.

The more observant among you will have detected an apparent fallacy in my line of reasoning. It is impossible, you say, to fix an object at three points and still be able to rotate it. That is a valid argument, and it is perhaps theoretically impossible to describe a 4-dimensional object having perfect symmetry in all four dimensions.

However, it is not necessary to fix one point of a line in order to rotate the other. You may vary the point of fixing during rotation, alternately fix one point and then the other, move both points simultaneously, etc., and thereby create an infinite variety of two-dimensional objects. You might even rotate both points in a third plane, in either the same or opposite directions, and thereby convert a single-dimensional line into a three-dimensional cylinder or opposing cones.

Similarly, it is not necessary to maintain the two points on a circle in a fixed position while rotating it. Note that it is not even necessary that the points be opposite, nor that they be moved only in a two-dimensional plane. It is only necessary that they maintain their relative distance from each other.

Therefore, the same obviously holds true for the rotation of an object having three dimensions.

continued on page 20

Regional Group Reports

Meeting Summary For AUGUST

Banana Coast 12/08/90 Sawtell
Carlingford 15/08/90 Carlingford
Central Coast 11/08/90 Saratoga
Glebe 09/08/90 Glebe
Illawarra 13/08/90 Keiraville
Liverpool 10/08/90
Northern Suburbs 23/08/90
Sutherland 17/08/90 Jannali

BANANA COAST Regional Group (Coffs Harbour area)

Regular meetings are held in the Sawtell Tennis Club on the second Sunday of the month at 2 pm sharp. For information on meetings of the Banana Coast group, contact Kevin Cox at 7 Dewing Close, Bayldon, telephone (066)53 2649, or John Ryan of Mullaway via the BBS, user name SARA, or telephone (066)54 1451.

CARLINGFORD Regional Group

Regular meetings are normally on the third Wednesday of each month at 7.30pm. Contact Chris Buttner, 79 Jenkins Rd, Carlingford, (02)871 7753, for more information.

CENTRAL COAST Regional Group

Regular meetings are now normally held on the second Saturday of each month, 6.30pm at the home of John Goulton, 34 Mimosa Ave., Saratoga, (043)69 3990. Contact Russell Welham (043)92 4000.

GLEBE Regional Group

Regular meetings are normally on the Thursday evening following the first Saturday of the month, at 8pm at 43 Boyce St, Glebe. Contact Mike Slattery, (02)692 0559.

ILLAWARRA Regional Group

Regular meetings are normally on the second Monday of each month, except January, at 7.30pm, Keiraville Public School, Gipps Rd, Keiraville, opposite the Keiraville shopping centre. A variety of activities accompany our meetings. Contact Lou Amadio on (042)28 4906 for more information.

LIVERPOOL Regional Group

Regular meeting date is the Friday following the TISHUG Sydney meeting at 7.30 pm. Contact Larry Saunders (02)644 7377 (home) or (02)642 7418 (work) for more information. The proposed activities for the coming months are as follows:

10th August 1990 - The Missing Link (Hopefully)

7th September 1990 - Getting the best out of word processors and their utilities.

12th October 1990 - New programs from Asgard Software

All Welcome. Larry Saunders

NORTHERN SUBURBS Regional Group

Regular meetings are held on the fourth Thursday of the month. If you want any information please ring Dennis Norman on (02)452 3920, or Dick Warburton on (02)918 8132.

Come and join in our fun. Dick Warburton.

SUTHERLAND Regional Group

Regular meetings are held on the third Friday of each month at the home of Peter Young, 51 Jannali Avenue, Jannali at 7.30pm. Group co-ordinator is Peter Young, (02) 528 8775. BBS Contact is Gary Wilson, user name VK2YGW on this BBS. The format of each meeting is quite informal, with topics ranging from software reviews to hardware modifications with a fair sprinkling of purely social chatter in between. The June meeting was again well attended with all the regular members being there. Some time was spent in becoming familiar with the features of Page Pro including the merging of DV80 files, pictures and graphics. Comparisons were

made with similar products such as Picasso. Derek Wilkinson brought along a collection of Library disks which proved to be quite interesting. Unfortunately my May Regional Report did not leave the disk box. Well everybody is entitled to one mistake.

Peter Young Regional Co-ordinator

TISHUG in Sydney

Monthly meetings start promptly at 2pm (except for full day tutorials) on the first Saturday of the month that is not part of a long weekend. They are now held at the RYDE INFANTS SCHOOL, Tucker Street (Post Office end), Ryde. Regular items include news from the directors, the publications library, the shop, and demonstrations of monthly software.

AUGUST MEETING - 4th August

The meeting this month will involve a demonstration of Version 3 of TI-Base, FWB 4.3 in 80 column mode, and the new, faster Multi-plan update. A Special General Meeting will be held (see p. 3 of the July TND for details). A games room with Shane will be revived for the younger set. Percy will be available with the shop gear, Alf with the library and Geoff should be on hand for assistance with repairs.

Other meeting dates for 1990 will be as follows.

SEPTEMBER 1
OCTOBER 6
NOVEMBER 3
DECEMBER 1

September Meeting Notes

Saturday 1st September 1990 meeting will start at 2-00PM.

BUY, SELL AND SWAP DAY

Remember to bring your spare parts that you want to sell and money to buy other members spares.

October Meeting Notes

Saturday 6th October 1990 meeting will start at 10-00AM.

FULL DAY LECTURE TYPE WORKSHOP.

Alf Ruggeri Larry Saunders - Page Pro

Other items to be added as they come available with volunteers to run them.

The cut-off dates for submitting articles to the Editor for the TND are:

September	12 August
October	9 September
November	7 October
December	11 November

Russell Welham (Meeting coordinator).

continued from page 5

- | | |
|---------------------|------------------------------|
| a) English Monarchs | b) Factoring |
| c) Gaggles of Geese | d) Ancient Greeks and Romans |
| e) Mirror Game | f) More or Less |
| g) Patterns | h) Philosopher's Quotes |
| i) Solar System | j) Subject/Verb Agreement |
| k) Word Math | l) World Capitals (m/c) |
| m) World Miles | n) Audio Math |
| o) Catalog | p) Documentation |
| q) Quit | |

The documentation on the disk is not extensive, but is sufficient to get you "up and running".