^^^^^ TI-101 ^^^^^

OUR 4/A UNIVERSITY

by Jack Sughrue
Box 459
E.Douglas MA 01516

#6 THE COURSE TEXTS

In order for you to pass this course, Class, you have to have a decent working knowledge of the texts. Now, here is where we practice lots of flexibility (which, you may have noticed, abounds in this classroom). There are so many wonderful texts (and a few dogs) available for our TI, even now, that you should consider at least three for essential reading and the final projects. These will be worth one-third of your entire grade.

By texts I mean textware: the printed materials for your TI's. These would include your very best source, of course: the newsletters that come with club membership. This newsletter networking is THE BEST SOURCE of all because you are part of a group, even by long-distance mail.

Another essential source of educational and survival materials is the magazine devoted to your specific computer. There have been many, but there is only one left: *MICROpendium*. It's the only international source for all things (including advertisements) TI or Geneve. To own and use your computer to the fullest extent and not subscribe to *MICROpendium* is like owning a marvelous pair of eyeglass frames but not getting around to putting the lenses in so you can see properly. The subscription (from P.O. Box 1343, Round Rock, TX 78680) is only $25 per year. Tiny price to keep your great computer great.

The third source is what we're discussing today, Class. And, yes, Ms. Bronte, this will include references to adult learners as well as to children.

The third source is the texts available. Note the word "available," Class. Availability of text written a decade ago may seem impossible, but not so. Most user groups have extensive libraries of texts for long-term loan. Individuals within groups sell off their text materials often.' Such text materials can be found very inexpensively at every TI fair in America and Canada. And, again, *MICROpendium* lists agents and individuals from whom you may purchase lots of printed materials. For example, if you all look up here for a moment. I am holding the latest copy of THE magazine. Those up back can't see it, so I'll read it to you. "99-cent Book Blowout!" It's a publisher's clearance. You can buy books at 99 cents each. These include the following: THE ELEMENTARY TI, GAMES TI's PLAY, COMPUTER PLAYGROUND, PROGRAMS FOR THE TI COMPUTER, USING & PROGRAMMING THE TI, INTRO TO ASSEMBLY LANGUAGE, and Volumes I & II of GAME WRITERS PACK and STARTER PACK, both packs from England. Some come with disks and/or cassettes at additional charges. But the point is this: these are still new books that cost on the average $12-plus when they first came out. They are still the same good books, still new to anyone who has not read and used them, and still available at almost giveaway prices. Not all ten of these books will suit every learner, but there is certainly something for everyone included in this collection. And that's just from the TEXCOMP ad (P.O.Box 33084, Granada Hills CA 91344; Phone: 818 366-6631). There are other advertisers in

classifieds that also offer all kinds of text materials. So, Class, the stuff is available to anyone who wants it.

And did I mention Barry Traver? No? Wow! How is it possible that we are practically through this semester on such an important topic as the TI-99/4a computer and I did not mention its greatest advocate, the man who has done as much for the TI as any spokesperson for any organization that I know of. Rather than list all Barry's writing and speaking and programming efforts, his work on BBS's and for various magazines, his appearances at numerous fairs, I will just mention the relationship he has to the present topic, though I'm not sure it might be better saved until we discuss things next class. Ah, well, what the hell. Barry has a wonderful educational tool in the form of a diskazine. This diskazine is called GENIAL TRAVELER. The "zines" have been nothing short of remarkable. Volume after volume have been rich with educational wonders and remarkable, big bonuses. You have to experience GT to appreciate the jam-packed series. There's nothing like it.

Though I've given you Barry's address in a previous class, I'm well aware that some of you have not taken careful notes. Barry can be reached at 835 Green Valley Drive, Philadelphia PA 19128. I think when you send off for your magazine subscription today, you should also send a note to Barry asking about the cost of the numerous disks in his volumes of great materials.

What made me think of Barry is that one of the disks includes the second most complete list of all TI publications there ever was. A little synopsis goes with each. The first most complete list was in a series called NEW-AGE/99 by some old geezer from Massachusetts. I can't think of his name now, but I'm sure you can find references to him when you go searching through newsletters. He did a lot of reviews, too, but I think he approached things more from a visceral level than an intellectual one. Anyway, Barry's your better source here because of all the other things contained in the GT disks.

Now where was I, Class? Oh, yes, texts and tests. You'll be having some of this on the final, so wake up in the back, stick your gum behind your ears, and listen up! With your pencils.

Before I go any further, TI-ing or otherwising, I have to insist you get your hands on a copy - any copy - of THE SECRET GUIDE TO COMPUTERS by a bizarre and hilarious genius by the name of Russ Walter (22 Ashland Street #2, Somerville MA 02144-3202). This 8X11, 600-plus page, mindbogglingly wonderful book for layman or technowhiz is bursting at the seams with all you'll ever need to know about computers. Now in its 16th edition, it's a steal at $15, but if you order two they are $12 each. Four or more are $9 each, shipping and taxes (except Massachusetts) included, so hook on with some friends or your user group. Once you have this remarkable and remarkably readable book in your hand (considered the world's top-rated tutorial by a list of experts and novices as long as your arm and a lot longer than mine), you will never again be the same; nor will your computer. You simply have to experience "Russy-poo" in order to understand that his is THE essential book for any person who owns a computer.

That aside, let's look at some very specific educational text materials for our TI.

First, there's a problem. When we talk educational text material, we must eliminate the modules and anything related to LOGO, as we will deal with these educational items on a particular basis in future classes.

Second, we must define educational in the specific context with which we have been structuring these classes. That means we really have to eliminate the "learning" that comes only from learning about the computer. Texts that teach us how to write programs, for instance, or

texts that teach us how to balance a budget using our wonderful machine, are not really appropriate here, but typing in programs that specifically deal with education DO fit our class requirements. For example, the C.W.Engel book STIMULATING SIMULATIONS FOR THE TI-99/4A published in various forms from 1977 to 1984 by Hayden Publishers, was the first important educational tool for me. Not only did it have detailed instructions and flowcharts for each of the type-in programs, but it had enough errors from translating the programs from other computers to the TI that the intellectual puzzles of figuring out what went wrong and correcting it were wonderful educational opportunities. Many of the programs were "intellectual" games, rather than arcade. Later, with toots and whistles, they became more "arcadey" in other people's books, including other Hayden books. Hayden published the most TI-specific books. I wish they still did.

Ah, well. STIMULATING SIMULATIONS, though, is not the kind of text we need to locate and use for educating our youngsters and new oldsters.

The best book of the learning to program type is KIDS AND THE TI-99/4A, which also leads to some eduational programs, too. It's the clearest, most direct, easiest "programming" book. Done in 33 lessons, it is still used in many classrooms today; not just with TI's but with other computers, also, and at very young grade levels. THE ELEMENTARY TI, mentioned in that TEXCOMP list earlier, is probably the best of that sort for adults.

But the kind of books which best exemplify the educational aspects of the programs typed in - in other words, the LEARNING FUNCTION - are the ones I'll hold up now and give a say a word or two about. Another one from the TEXCOMP 99-cent list is COMPUTER PLAYGROUND (Datamost's TI version). Although this can be classified as a beginner programming book, it is so unusual that the logic it teaches in an incredibly entertaining way makes it a real winner for any learner. (It's geared for Grades 2-7.) The book is a combination workbook/coloring book and deals with BASIC in such a puzzle-solving way that it becomes, itself, a complete course in thinking. It stands alone.

This is not to be confused with TI PLAYGROUND by Fred D'Ignazio, another Hayden book. Fred also wrote a similar book called TI WONDERLAND. Both books include programs written by students, and ALL the 40-plus programs are written FOR students. Each chapter is an educational game that is introduced with a section for parents and teachers and another for kids. Each game has one educational feature (such as subtraction) and follow-up activities that allow an almost unlimited number of modifications. The programs aren't just alphabet and number programs. Because it's a TI, the computer is able to have programs that teach color and music and drawing and hand/eye coordination very readily through these marvelously childlike and truly sophisticated programs. Very easy to type in and change all along the way. Lots of very positive rewarding, too.

Lest you think Hayden was the only publisher for TI - though I'd recommend you seek out their other books - there were many others, as you can see from these piles on my desk. Take, for example, TI GAMES FOR KIDS put out by COMPUTE!, one of the very best publishers of TI stuff, including Regena's two classics.

TI GAMES FOR KIDS proclaims its purpose right on the cover here: "Turn your TI into a teacher. Thirty-two games that teach and entertain, ready to type in and run."

This book takes the trouble to identify each of its activities by age level (3 to 17/adult) and educational function and subject: strategy, logic, memory, coordination, language arts, social studies, math, etc. (Strategy games include an excellent version of Fox and Geese, by the

way.)

As I look over these other books, it's hard for me to say which I would recommend the most for educational use. I have a public school teacher friend who still uses the TI in his class. His favorite educational book is TERRIFIC GAMES FOR THE TI99/4A by Hal Renko and Sam Edwards. Let's see, this is published by Addison-Wesley. It has small size and type but is plenty thick. It contains some neat little drawings, to which you people up front can attest, for the 30-plus games. There are some unusual ones here: Genius at Work, Escher, Rainbow Square Dance, The Wolf and the Five Little Goats, Shakespearian Shuffle, Mini Mancala, and so on. Good stuff.

Remember, now, most of these are not made to be super arcade games. They are made to be typed in - usually with lots of explanations and helpful hints - and are meant to teach something while entertaining. And they do that well.

One of the most popular books ever of this sort was (and is) Steve Davis's PROGRAMS FOR THE TI HOME COMPUTER, self published in large 8X11 format. Although the type-in programs included many utility programs, the majority were educational or verging on the educational (like "Bar Graph Printer" and "Talking Calculator," for examples). There are very few people who did not get hung up on the probability games like "Ten-Up" and "Lucky Seven" or the maddening "Echo" of Simon fame. He even has a "French Teacher" program and a "Speed Reader." One of the best.

Scholastic book publishers released a pile of multi-computer large format books, each containing about 40 programs for elementary school children to type in and use. Here's an example: COMPUTER OLYMPICS. Each of the books has a theme. This one is all Olympics. It opens with the torch that lights the Olympic Flame. There are programs that teach words in various languages; some that let you recall other records; some that require some math skill (like the weightlifting one) or word skill (like the rowing one). All are simple to type in, no matter what your computer, if you follow the rules on the various basics, and all are simple to execute. Actually, with the built-in motivation of the Olympics in Spain, this book on summer Olympics is perfect for teaching at home or school. Look into the other Scholastic books, too. They also teach you lots about the various basics, if you want to do some explorations. With the TI, though, you can easily slip in some color and sound not readily available on other computers listed in these books.

Because our time is running short and we only have a couple more classes this semester, I've got to finish off with these last two books, but, as you can see, I haven't even been able to talk about all these others in these piles. At the last TI fair I went to, I picked up additional copies of 19 different titles of TI books! (And paid a grand total of $15, by the way.) Now everyone will be able to take two and give a two-minute review of each next time. You may come up and sign out two (or three, if you're interested in extra credit) to do for your reviews and your end-of-term projects.

Meanwhile, let me just mention these last two books. If you can get yourself a copy of Richard Mowe and Ron Mumaw's ACADEMIC TI, do it; even if you have to pay the full $12.95. It's worth it. Published by Reston in 1984 (one of the newer books mentioned today), it was one of the few books totally devoted to TI as an educational tool. It told parents and teachers how to get the very most out of our computer EDUCATIONALLY. When you read the articles and do the worksheets and explore the options from modules to disks to LOGO to word processing and so on, you will be astounded at how magnificent our machine is and how out-of-date it isn't. It still does all the educational things it was

geared up to do better than any other machine out there. Which is not to say it is as sophisticated as some of the biggies, but what it does educationally it still does better than anyone.

And the last book. Remember this, anyone? Ah, it's good to see so many hands up. THE BEST OF 99er is still around a lot of fairs and user groups. It's 368 8X11 pages are jampacked with all the goodies that made the 4a the prize goodie of all. And Regena was really kicking up her heels in this one. Do you recall "Name That Bone" and all the Homework Helpers? The Computer Assisted Instruction was so popular with the TI then, the term was simply CAI. And everyone understood it. There was so much of it around for parents, teachers, and other kinds of humans. The BEST OF 99er is one of the very best.

So, Class, unless there are any questions, come and sign up for your books and ... Yes, Mr. Shakespeare? What do I consider the Mother of All TI Books? Hmmm. Well, my personal favorite for more reasons than I can begin to list, including some super educational reasons, has got to be Paul Garrison's THE LAST WHOLE TI99/4A BOOK: PROGRAMS AND POSSIBILITIES, published by Wiley Press in 1984. Even after all these years it is still my preferred TI piece of textware. It's 460 pages are lucid, witty, intelligent, relevant, and very worthwhile. The tutorials and the programs are excellent. They DO provide unlimited possibilities for growth. I never loan out my only copy of that one. Sorry.

Until next time then. Don't forget to bring your book reviews and all of your modules.

6

What is AMS, and Why is it Here? A monologue by Chris Bobbitt

The Introduction

A wise man once said true wisdom can be attained by learning from your mistakes. Considering the number of mistakes I've made over the years, at least I can say I've had ample opportunity to become wise.

The largest "mistake" I made in the 10 years Asgard Software has been in existence was **Press**. It wasn't a complete mistake - I'm not going to apologize for the vision Charles Earl and I shared of what a word processor should be. I will readily admit, however, that the way we went about developing it and marketing it was all wrong.

Lots of little mistakes became apparent in my quest to discover What Went Wrong. Soul-searching aside, one of the biggest reasons was actually technical. Contemplation of the technical problem led to both a fundamental realization of, and appreciation for, The Problem.

The Problem

Press was designed to be the ultimate in modular code - the program would literally load individual functions into memory as needed, and reuse the space when the functions were no longer needed. In this way, it had more in common with mainframe programs than software for home computers. This level of modularity is what would have made **Press** possible - and without it **Press** was impossible.

As you may have already guessed, the fact we couldn't get this "Memory Manager" to work was the reason that **Press** didn't work. A large part of this was due to the fact that memory is scarce on the 99/4A. In order to work, **Press** needed all the memory it could get - so the Memory Manager was written to take advantage of all sorts of other types of memory: supercarts, the Mini-Memory and even some RAM-disks (such as a **Rambo**-equipped **Horizon RAM-disk**). While this was fine in theory, in practice it was a mess.

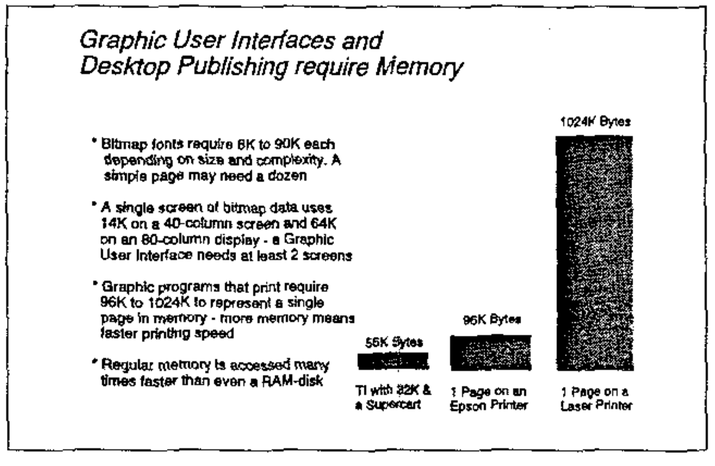
Because of the complexity of using some of these devices as memory for programs and data, more code was devoted to accessing memory beyond the standard 32K than all of the other code in the Memory Manager combined. Because only a small number of features could be implemented on a standard 32K 99/4A, and taking advantage of memory beyond the 32K resulted only in reams of buggy program code, **Press** collapsed under its own weight.

While it would have been possible to make a version of the program that ran in the standard 32K, we had promised the program would do much more, and neither Charles nor I really wanted to release a program that only did half of what we told everyone it would do for 2 years - so the project died, and The Problem first slapped us in the face:

"The 99/4A desperately needs more memory accessible to programs"

Our experience proved that the only real program-accessible memory for the 99/4A is the 32K card, and to a degree a supercart/Mini-Memory. You have to face it, while 32K was a lot of memory in 1979, not a single PC program today will run in it. Heck, 64K became standard in 1982 when the Commodore 64 was released. The average PC or Mac sold today is equipped with 4096K of RAM. The average PC word processor requires 640K to run minimally, and the next generation of word processors will no longer run in less than 1024K.

To illustrate how memory requirements have ballooned in the last decade, take a look at the following chart:



Desktop publishing was chosen as an example because, with the advent of programs like **Page Pro 99**, it has become one of the most popular things the 99/4A is used for. As someone who develops graphics software, I've been painfully aware of the 99/4A's memory limitations for years - especially as we've tried to expand the capabilities of our programs.

Why do 99/4A programs need extra memory? Very simply: to allow more data and more program functions to be in memory at once.

What is the difference between extra memory and a RAM-disk? While this is discussed in more depth below, the answer is also simple: RAM-disks are really only faster disk drives. *While they can hold programs and data, this information is only accessible when a program physically loads a piece of it from the disk. With more real memory, the program itself can be much larger, and more data can be stored in memory where it is accessed quickly before it has to go to the relatively slower disk drive.* As fast as they are, RAM-disks are slow compared to storing data in memory. More memory also allows programmers to write large programs that contain lots of data (such as a Graphics User Interface) without frequently referring to a disk drive.

In essence, more memory is needed so that more complex things can be done - and a RAM-disk is only a partial substitute at best for more memory.

So, now that we knew The Problem (the lack of real memory for 99/4A programs), The Search was on for The Solution.

The Search

After discovering The Problem we decided to evaluate all of the memory devices available for the TI-99/4A to see if any of them were The Solution. Our experience helped us to define the criteria any memory system should meet before it was usable as "real memory":

1. It had to offer memory within the normal programming area. This is so that it would be easier to adapt existing programs to take advantage of it.
2. It had to offer memory in usable "chunks" that were large enough to store a significant amount of program code and data, yet small enough where a single bank of RAM didn't take all of the standard memory area.
3. It had to offer a lot more memory than the standard 32K - the more the better.
4. It had to be invisible to, or at least not conflict with standard hardware and software.
5. It had to usable by average programmers - and not just super hackers.
6. It had to be inexpensive.

In our opinion, the ideal memory system would also:

7. Be invisible to the programmer - he or she would simply write a large program and let the memory card figure out how to fit it into memory.

To make a long story short - none of the memory devices on the market met these conditions. Supercarts and the Mini-Memory were limited to a certain amount of memory at a certain location. The only device that even offered a glimmer of the kind of memory needed was the Rambo, and it was both inflexible and a programmers nightmare to work with. All other RAM-disks also failed on one or more points.

After an exhaustive search, we decided that if we wanted to write more sophisticated software, we had no choice but to build a device that offered the capabilities we needed.

Using the Geneve and the un-released TI-99/8 as models, which both could be expanded to 2048K, Asgard Peripherals (the hardware division of Asgard Software) began a two-year odyssey of exploration, frustrating dead-ends, and back-tracking before we were able to construct a memory system that met the six criteria above, and could (with some work) even meet the last condition - a memory system that was invisible to the programmer.

The Solution?!

It's funny how sometimes the hardest questions have the simplest answers. In searching for a solution we almost entirely re-invented the wheel before we realized that TI had already done it for us. We discovered that they had built, and continued to produce, a single chip that did most of the work of expanding the memory of the 99/4A.

Variations of this device (known as a "Memory Mapper") are found in virtually every 9900-family computer product TI ever built and sold, with the exception of the 99/4A. A variation is even used in the Geneve and the TI-99/8.

The 99610 memory mapper (its original designation) is elegant in its simplicity. It takes the 16-bit address space of the 9900 processor and turns it into a 24-bit address space. In other words, it makes the 9900 think it has up to 16Mb (or 16384K) of memory instead of 64K. It does this by allowing a programmer to put 4K banks (or blocks) of memory anywhere within the normal address space of the 9900 processor.

The jist of this is that at a stroke this single chip met all of our first six conditions.

Besides offering memory within the normal programming space, it also offers it in 4K blocks that are easily manipulated. Further, it turns out, most software written by TI was also designed to work with blocks of the same size - and so it would be a lot easier to adapt existing TI software (Extended BASIC, etc.) to take advantage of mapper memory than any other kind of memory.

The mapper obviously allows for a lot more memory than 32K, but just as importantly, to the computer a memory card using the mapper is no more non-standard than a 32K card - and won't conflict with any device except those that try to provide 32K to the computer.

A final advantage of the mapper is that, for programmers, it is about the simplest way to use memory beyond 32K. Assembly programmers only need a few lines of code and other programmers a single command to push blocks of memory in and out of the 32K area - or even potentially have it done for them automatically (depending on the language).

After discovering the mapper it was only a matter of time before we built a prototype, the AMS, that would provide up to 512K of RAM to the 99/4A accessible through the mapper.

What is the Asgard Memory System?

- A family of fully compatible memory cards for the Peripheral Expansion Box
- Fully compatible with all existing TI software - works as a 32K card to regular programs
- Transparent to all disk controllers and some RAM-disks, as well as all other devices
- Provides memory to 99/4A in 4K banks - exactly what TI specified for the 99/4A & 99/8
- For AMS aware programs, provides additional memory on demand up to 16Mb in a few machine cycles
- Easy to adapt existing software and languages to take advantage of it - if it works with a Supercart it can work with AMS

AMS 128K-512K
Available now

AMS 1Mb-16Mb
Available Q2 '93

The AMS also met our sixth criteria - it is a cheap way to add 128K to 512K of usable program space to the 99/4A. Considering an 8K supercart costs $25 or so, $120 for 16 times as much RAM is a bargain. The AMS also allowed us to prove our concept, and get the technology embodied by the mapper quickly into the hands of programmers so that they could become familiar with the technology. Further, you can do real things for the 99/4A, even with "only" 128K.

Finally, the AMS has allowed us to begin work on a software system that brings our last goal - a memory system that is transparant to programmers - within reach. Several programming languages in development will take advantage of the memory without the programmer specifically writing program code to do so.

The Comparisons

Inevitably, there will be comparisons between AMS and other memory systems. While you can compare AMS to other devices in terms of the amount of memory, it is impossible to compare it in terms of how the memory is used. The only memory device that comes close is a supercart. Like a supercart, the AMS provides real memory for programs, and not disk storage. Unlike a supercart, it offers much more than 8K of RAM.

Comparing the AMS to a RAM-disk, as mentioned above, is like comparing apples and oranges. RAM-disks are "solid-state disk drives", meaning memory chips that are controlled by software to emulate a disk drive. The AMS, by comparison, is memory that can be directly used by programs to store data or program code which can be used or acted on without "loading" it first.

To illustrate the difference between RAM-disk memory and AMS memory, a good example of would be a database program. With TI-Base, for instance, if you have a 2000 record database you may be able to store (perhaps) 50 records in memory at once. Whenever TI-Base sorts the database, it has to physically load each of the 2000 records, 50 at a time, into the same space in memory. It creates an "index" of those records, which it then saves to disk after it has completed loading all of those records. This "index" tells the program the order the 2000 records are in. With a database like Personal Record Keeping, you would be limited to maybe 100 records altogether, since it doesn't allow you to have more records than you can fit into memory.

If you had a database designed to work with AMS, *all 2000 records could be in memory at once*. A sort would be instantaneous as the records could be put into sorted order within memory - or an index built in memory without loading anything first. The same database could be sorted in a tiny fraction of the time, and searching the database would be instantaneous. This same comparison would apply in any situation where you have a lot of information to store - a word processor, graphics program, etc. Because all of the data physically resides in memory, it can be accessed and used much faster.

In addition to providing memory for data, AMS also provides memory for larger programs. Because all of the program can be loaded into memory at once, managing memory with AMS is less difficult than loading parts of the program from disk - and takes much less code. The AMS memory card functions almost exactly like memory beyond 640K does on PC compatibles. Programming a 99/4A equipped with the AMS (or its sibling) is no more difficult than writing PC programs larger than 640K.

While technically, AMS is fundamentally different from other cards - some people have been tempted to compare it with other memory systems on the basis of the amount of memory offered. While the AMS offers as much memory as some other memory systems (the Foundation has been mentioned), again, there is no comparison. *The AMS is only memory card for the 99/4A that offers a memory mapper - and a memory mapper is the only way to truly expand the amount of memory available for programs and data on the TI-99/4A.*

The Future

The AMS is currently in the classic "chicken-before-the-egg" dilemma. If you were to buy one and plug it in, it only works as a 32K card until you run a program designed to use it. Why would anyone want to buy a card where there is currently no software designed for it? Conversely, while would anyone write a program for a device nobody owns?

These are legitimate questions, and they deserve honest answers.

From a programmer's perspective, if AMS was just another RAM-disk for the 99/4A, I would agree. However, the AMS is a real technical breakthrough in 99/4A memory expansion. We have good reason to believe that every programmer that has hit the 32K memory barrier in the past is (and if they aren't, should be) interested in writing programs for AMS. There are really no other options if you want to make larger and more powerful programs.

For programmers, the AMS is the first and only true memory-mapped memory device for the 99/4A. *There is no easier way to write programs larger than 32K.* Further, the AMS was designed to the only real standard that ever existed for expanding 99/4A memory - the one TI specified for the unreleased TI-99/8. The AMS provides memory to the 99/4A the same way memory is provided to the 99/8. The AMS is the only true memory expansion system for the 99/4A aside from a 32K card or a supercart.

While we can't make programmers take advantage of the card, its potential to make so many projects that were impossible in the past possible, or even easy, will interest any programmer who wants to make better programs.

From an average person's standpoint, the issue is more complicated.

AMS was designed and implemented by a software company. Therefore, its guaranteed support from at least one software company - one that is in the process of writing a considerable amount of software designed to take advantage of it. While initially most of our software will be AMS versions of our other products, these versions will not only be faster and more capable, but ultimately, may evolve into much bigger, more powerful programs. Further, programming languages designed to make writing software for AMS even easier are also in development - including a full assembler package and a new Extended BASIC cartridge. While we can't speak for other software developers, virtually every product in our future will take advantage of AMS one way or another.

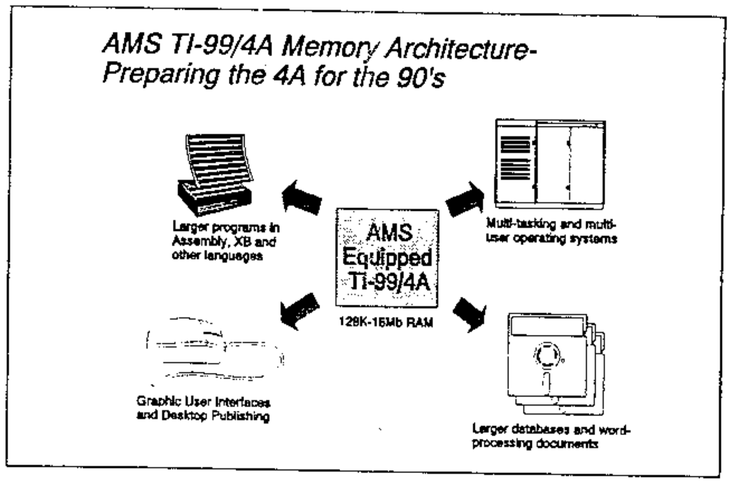
Also as a software company, we have a better idea of what programmers need in order to write programs for the AMS. In order to allow other developers to work with AMS, we've placed all programming specifications and source code into the public domain, and we are providing AMS systems with documentation at cost to any programmer that wants one. We already have mailed documentation to most prominent TI software developers.

From the point of view of a customer, the AMS is being produced by a company with an almost 10-year track record in the TI community - and a solid record of keeping our promises (well, all but 1 or 2 of them). We are in it for the long run, and so we will do whatever we can to support and develop this technology.

Examples? We are currently working on the next generation AMS card which allows 1024K to 16384K of memory for the 99/4A - potentially 8 times the memory of the Geneve. A wide array of development software will be included with the device - which will be fully compatible with its predecessor. To protect buyers of the current card - they will be able to trade it in on the next card when its available.

Finally, we can offer one last form of "buyer protection". We are committed to making this memory device the standard extended memory system for the 99/4A. Asgard intends to license the design for a low, one-time fee to any company, user group or individual that would like to make AMS-compatible cards. *We designed this card because we needed it to do the software we wanted to do - not to get into the hardware business.* We are negotiating with several people right now, and if all goes well, in the future you should be able to buy compatible cards from a number of vendors.

If all of these reasons don't convince you, the best reason of all is the potential of what can be done with more memory on the 99/4A. More memory will open up a wide variety of new applications that are difficult to impossible on the 99/4A:



As the chart illustrates, AMS makes lots of things possible that weren't before, including:

- Multi-tasking and multi-user operating systems
- Full-scale business packages
- Graphical user interfaces and true desktop publishing
- Full-size word processors, databases and spreadsheets
- Modern, full-size programming languages such as C and ever more capable Extended BASICs
- Great advances in graphics, speech and music software - including multi-media, digitizing, fax software, and more

All in all, AMS will let the 99/4A live up to its full potential as a computer by eliminating its most serious problem - the lack of memory available for programs and data.

NEXT MEETING TUESDAY, FEBRUARY 9, 1993 HAPPY PRESIDENT'S DAY!!!

MUNCH OFFICERS AND NUMBERS (all in 508 area unless noted)

President	W.C. Wyman		865-1213
Vice President	Bruce Willard		852-3250
Secretary	Jim Cox		
Treasurer	Jim Cox		869-2704
Acting Editor	Jim Cox		
Adv.Prog. Chair	Dan Rogers		248-5502
Library	OPEN		
Disk Librarian	Lou Holmes	617	965/3584
Tape Librarian	Walter Nowak	413	436/7675
NEW-AGE/99	Jack Sughrue		476/7630

MUNCH DUES	
NEW MEMBERSHIP	$25.00
RENEWAL MEMBERSHIP	$15.00
NEWSLETTER ONLY SUBSCRIPTION	$13.00

JANUARY MEETING. There were 12 members and 2 guests at this meeting, and considering the bad weather it was a great turn/out. Those who came were not disappointed. mike Wright gSdemoed his program which allows TI emulation to run on an IBM PC or a Mac. That's right a TI on a PC. Mike has done an amazing job on this program. He is still working on it but I expect it will be ready for the BCS Fair. I look forward to getting it.

FEBRUARY MEETING. This month Tony Falco will do a demo on how to program sprites and how they work. Tony has done some outstanding work with sprites. There will probably be some other surprises, as usual.

RAFFLE. Every month we have a raffle to help defer the rental cost of our meeting hall. A typical raffle will have game and utility programs T-Shirts, books, bumper stickers, blank discs and all sorts of odds and ends for the T.I.

LIBRARY NOTICE. Please return any items borrowed from our library. If you can not come to a meeting or give these items to someone who will be at the meeting.

REPRINTS. Reprints are permitted as long as credit is given to M.U.N.C.H.

ARTICLES. I am always looking for articles for this newsletter, anything which interests you will probably interest other members of the TI community, so please share your ideas and opinions with all of us.

DISK LIBRARY. The disk library will be at the meetings from now on. We have copies of all disks in the library and they are available to members for just $1.50 each for single discs, $2.00 flippies, $3.00 double discs and $4.00 double flippie.

FOR SALE. The group has a TI Count Business Software package available for sale. If interested contact Jim Cox at the above numer or the club address. We also have blank disks for sale. The price is $6.00 for a package of 25 disks. Bruce still has his sister-in-law's system for sale, call him for details.

DISK OF THE MONTH. This month Disk #114 is a games disk from our English cousins. Programs include 3-D Tic Tac Toe; Blackjack; Poker; Impirium Romanium and Texas Frogs. Most of these programs include speach.

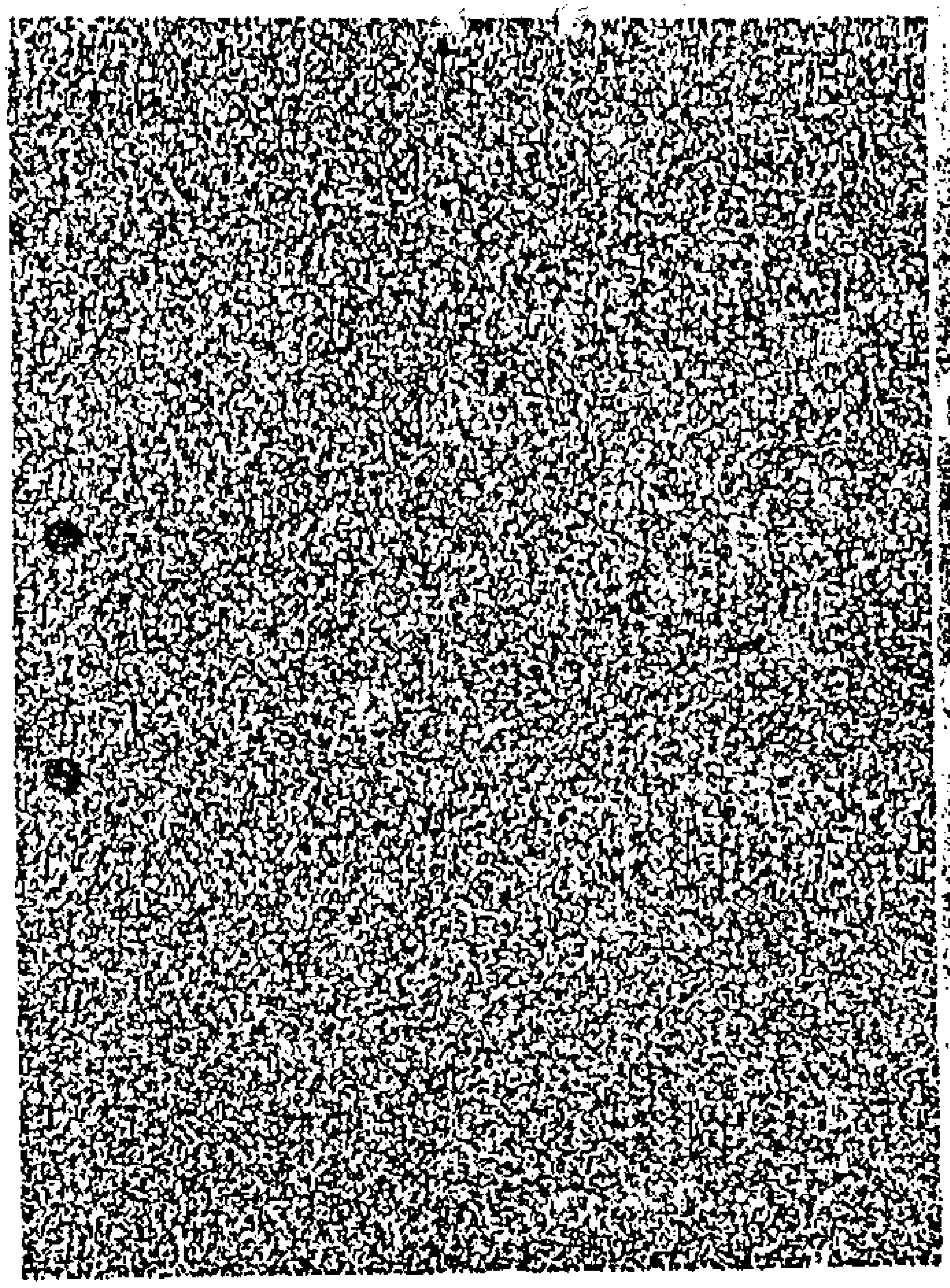
THE TONY FALCO CLASSICS will be our fundraiser for 1993. The cost to members is $5.00 add $3.00 for First Class postage. The regular price is $8.95 post paid, but add $3.00 for First Class postage. I expect to have some of these five disk sets ready for the February meeting.

WELCOME NEW MEMBERS. Bryant Krause of Mira Loma, California.

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 *     *   #     #   +      +    %%%    $     $
 **   **   #     #   ++     +   %    %  $     $
 *  *  *   #     #   +  +   +   %       $     $
 *     *   #     #   +    + +   %       $$$$$
 *     *   #     #   +     ++   %    %  $     $
 *     *    ###      +      +    %%%    $     $
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Mass Users of the Ninety-nine and Computer Hobbyist

FEBRUARY 1993 Monthly Newsletter Version 12.02



Can You See N.E. THING?

*To see the image above, diverge your eyes, as if looking at a faraway object. The two dots will fuse, forming a third central dot. When the divergence is correct, slight, controlled variations in the placement of the random dots are perceived by the brain as depth cues. A shape will appear to float above a textured background. Some see the image in seconds. Others find it more difficult. If you can't see it, let someone else try. If they succeed, perhaps they can help you. Be sure to get the full explanation and more examples by writing for free information!

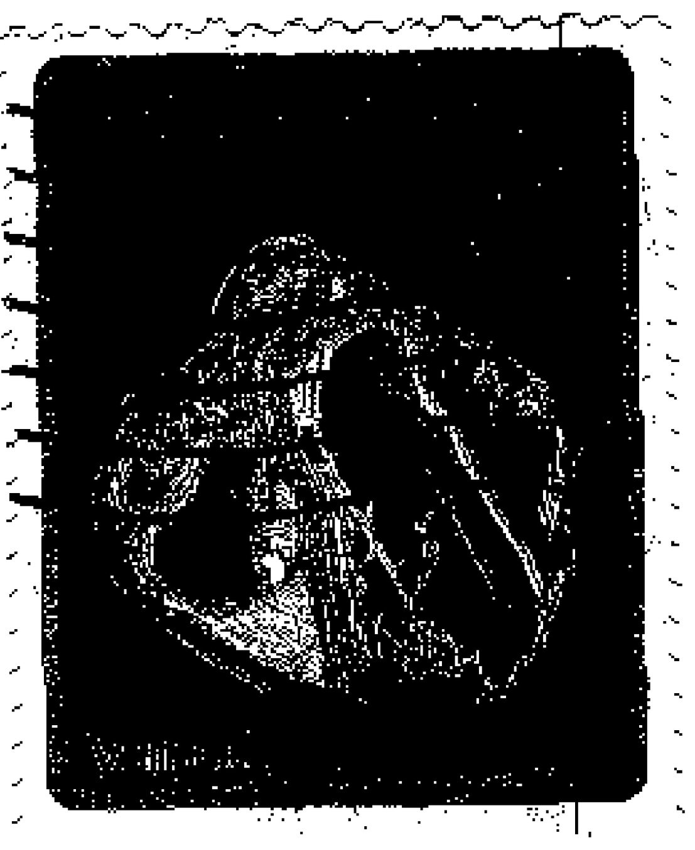
FROM CHICAGO U.G. TIMES

NEW ADDRESS:

M.U.N.C.H.
C/O J.W. COX
905 EDGEBROOK DR.
BOYLSTON, MA. 01505

3 FEB 1993

FIRST CLASS



Next Meeting FEBRUARY 9th.

POSTMASTER: Forwarding and Address Correction Requested.