CLEVELAND AREA 99-4A USERS GROUPS NEWSLETTER

MARCH, 1986

****

OFFICERS	SOLON	NORTHCOAST	GOLDEN CRESCENT	TI-CHIPS
PRESIDENT	STEVE WEINKAMER (1-274-2544)	RON HINADED (943-2791)	CHUCK MARENO (324-4388)	RICH POLIVKA (238-3971)
VICE PRESIDENT	WALT RYDER (921-8223)	MARTIN SMOLEY (257-1661)	DICK BURGER (967-1317)	RUSS SHINANDLE (237-3001)
TREASURER	FRANK JENKINS (283-8526)	JIM MEKEEL (486-7848)	DALE SCHIEFERSTEIN (988-5150)	DAN KELLER (225-0658)
SECRETARY	RUSS OSBEN (1-626-2622)	JOHN BLACKNER (257-2672)	HAROLD SKIPWORTH (631-4529)	JAN FEDOR (524-6192)
LIBRARIANS	KIM JONES(1-274-8004)	DEANNA SHERIDAN (333-5986)	BOB BARTO (1-282-7037)	MARK McCAULEY (886-1119)
	LADD PATA (662-9253)			
BULLETIN BOARD	----------(24 HOURS) (944-1072)			

****

EDITORIAL COMMENTS

Bruce Young and Steve Weinkamer of Solon joined the newsletter group at Tom Wellis' home this month. The is an enthusiastic group and it is hoped that enthusiasm will show in an informative newsletter.

Jim Peterson of Tigercub Software pointed out that our article on software for cassettes neglected to mention that he has well over 100 programs which run on cassette for the affordable price of $3.00 each. Perhaps since Mr. Peterson just lives down the road apiece, he could come up sometime and demonstrate his wares. I have his Nuts+Bolts #1 and find that it contains many subprograms that will save you hours of writing if you are doing your own programming.

Another followup to last month, Marty Smoley had an enhanced version of his 132PRINT perfected almost before we went to print with the first version. If you used it, you may have discovered that you needed a CR after every line for it to work and that both sides had to have the same number of lines. The enhanced version does not require the CR's, the columns can be lopsided if you wish, and you can choose the amount of space you want between columns. The new version is already in NorthCoast's library and will be published in the newsletter soon.

Is there anyone who makes extensive use of the various bulletin boards who would like to write an occasional article on what is going on and what is hot on the boards?

Look inside for....A review of the Horizon Ramdisk, How to convert assembly programs to program format, using your new disk, creating Tunnels of Doom adventures without the TOD editor, learn to program music using several methods, MS/LABELS for return addresses and disk labels, news and views.

HORIZON RAMDISK

By Dwayne Hughes - Golden Crescent

I received my RAMDISK on January 22, only 2 weeks after placing my order.

The package contained a doubled-sided (720 sectors/192K) RAMDISK card, a disk with the operating system and utilities, the source code, development software and the manual.

The card emulates a floppy drive at an incredible speed of 20 times faster than a floppy drive!!!

The card is battery backed up with 3 rechargeable Radio Shack ni-cad batteries. The batteries are charged when the system is in use and retains its contents when the system is off. These batteries are expected to last several years with a minimum use of 20 hours a month. The card should retain data for around 6 months from a full charge.

The card makes use of the DISK MANAGER 1000 V.2.3, which is covered in detail in the manual. If the DM-1000 files are on the RAMDISK, it can be loaded with a "CALL DM" statement from BASIC or EX-BASIC.

The card also contains several "CALL" statements that can be executed from BASIC or EX-BASIC. They include:

CALL DN(n) set drive number. "DSKn"
CALL MS set max. number or sectors. (360/720)
CALL WO set write protection.
CALL WF remove write protection.
CALL EX(a) execute machine code at address "a".
CALL CO enable DSR ram.
CALL CF disable DSR ram.
CALL DM load DM-1000 from RAMDISK.
CALL NF(n) set number of floppies in system.
CALL ? capability to add your own call statements.

(Continued on back page)

EXECUTIVE NOTES SOLON MARCH

Our February meeting got off to a start with a few techincal problems, but once fixed everything moved along smoothly. I passed out a questionaire/survey form to get a feel of what members have and want in the way of hardware, software, and expertise. Since our group has changed since the survey taken last year, I felt that it was time to update our information. This will help us better serve the group's needs in the months to come, and allow us to help eachother grow. I will summarize the results at the next meeting. For those of you who missed the meeting, I will make available copies of the survey for you to fill out.

Bruce Young reported on some of the new programs he has gotten from the expanded disk library. He informed us that the disk library has been greatly expanded and that the catalog is contained on seven disks.

Frank Jenkins gave the monthly presentation, demonstrating some of the programs in his personal library. Some of the most notable were a home budget, a mail list, and an autoboot program. It is always interesting to see the favorite programs that members have in their arsenal. Next month's presentation will be given by Kim Jones and Barbara Butler.

We unfortunatly ran short of time at the end of the meeting this month. I have therefore requested the library to extend the use of the meeting room to 2:00. So when planning your presentations, try to keep the material to about 30 to 40 minutes in length. That way we can still have time to copy programs and socialize after the session.

We have been requested by the newsletter editors to send a representitive to the newsletter planning sessions each month. As my Saturdays are sometimes tied up due to business and other commitments, Dick Ptak has agreed to help me out as corepresentitive to the newsletter committee. Between the two of us, we should be able to cover or commitment to the newsleter. If anyone else would like to help represent Solon, please feel free to volunteer your services. This could be a great opportunity to give personal input into what you want to see in the newsletter.

One final note of interest. I got a copy of DISK MANAGER 1000 at the last meeting. This is a freeware offering that the author invites others to freely share. The disk based utility contains all of the usual features of the cartridge, with the addition of a few extras. One of the most lifesaving extras is one that allows you to recover a file that you may have accidently deleted. The only catch is that you must recover the file before writing any other information to that particular disk. There have been a few times that I mistakenly (more to the point STUPIDLY), deleted a wanted file and realized too late that I goofed and could have really used this feature.

See you March 8, Solon Public Library! Bring a friend.

Steve Weinkauer, President

EXECUTIVE NOTES
TI-CHIPS

We had a good membership turnout at the February meeting. This is the time of year when the weather encourages us to spend more time with our computers.

Rich Polivka gave an interesting presentation of the "Adventure Editor" which allows you to edit and write your own adventure games. There will be a raffle for the program with the winner chosen at the March meeting.

Jan Domzal gave an informative presentation on copyrights. It was interesting to learn what can and can't legally be done with software, both here and internationally.

Oscar Predmetsky prepared a booklet to show all the things TI Artist can do. Thanks, Oscar, for a great job.

Thanks also go to Jan and Rich for a great meeting.

Mark McCauley has been printing out a catalog of the new programs for the library. It should be ready for the next meeting. There will be presentations in the future to let you know what is available.

Thanks also to Tom and Judy Thalner of Edu-Comp for making their software available at the meetings.

Next month there will be a presentation on graphics and the control and use of your printer. Due to the unavailability of the library, the April meeting date will be changed to the SECOND Saturday. Mark your calendar now for April 12. The March meteting will be March 15 at the North Royalton Public Library at 10:00 a.m. See you there. Bring your friends with TI computers too!!

JAN FEDOR

EXECUTIVE NOTES - GOLDEN CRESCENT

The Golden Crescent Users Group met on Thursday, February 20, at 7:30 pm at RAM Enterprises, new expanded showroom in Vermilion, Ohio. Dick Burger is hard at work enlarging his stock of TI merchandise and will be displaying the new Miller Graphics GramKracker very soon. (he should have it available by the time you read this.)

Ten members attended the meeting and six members renewed their memberships. I would like to remind all of our Golden Crescent members that there will be an increase in dues starting April 1, so renew this month and beat the price increase.

Dwayne Hughes displayed the New Horizons Ram Disk (see Dwayne's article elsewhere in this issue). His version is 720 sectors, and Dwayne is very pleased with this fine product. Our next meeting will be held on March 20, a Thursday, at 7:30 pm at RAM Enterprises, 967-1317. We will again hold a copying session as we did for the February meeting. Hope to see you all there and bring a friend to keep up with the latest info in the TI World.

CHUCK MARENO

EXECUTIVE NOTES - NORTHCOAST

The February 15 meeting went well with a crowd of about 50 people. Six new members signed up and there were 13 guests, with 2 from Youngstown and several from TI-Chips.

We had a demonstration by Tom Thalner of EDU-COMP of the GramKraker module. It opens the machine to many new possibilities. Changes made to the operating system and TI BASIC can be saved to disk or cassette and run at anytime. The door is open for exciting new innovations.

This year we are establishing several new committees. They would take very little of your time, but would make things easier for all of us. Please consider doing some small part and be a part of the action. We need someone to coordinate an education committee. You wouldn't have to be a teacher yourself.

Howard Winkler is now waiting on the parts from TI for the TRIPLE-SUPER CARTRIDGE he is building. It contains TI-WRITER, EDITOR/ASSEMBLER, and DISK MANAGER2. A selector switch and a reset switch allow it to work like a widgit.

Thanks to the efforts of the library committee, there are now 45 disks of FREEWARE and miscellaneous programs in our library. Be sure to take advantage of these fine programs and also send a few dollars to the FREEWARE authors.

We will be having a survey of hardware with the specific equipment you have, and also the programming knowledge you may have.

At the March 15 meeting, we will have a demonstration by Jim Mekeel of a morse code generator from Dynamic Data Devices. The first part teaches the morse code language, and the second part is a programmable Morse Code drill which also sends a file.

This will be raffled, along with a couple of basic games.

See you there.

RON MINADEO

ADDENDUM:

Ron asked that the following be added to the NorthCoast news:

As announced at the February meeting, dues will be raised to $15 for an indvidual and $20 for a family at the March meeting. Beat the price increase by sending your dues in now!

Also, as mentioned last month, NorthCoast has made its entire library available to the librarians of the sister clubs of the Cleveland Area users groups. This library consists of 2000+ programs, plus FAIRWARE, all of which have been sorted and catalogued with a description of each program. This is being offered for a share (one-fourth) of the cost of obtaining each.

To encourage active participation by our sister groups in expanding the library, ALL future acquisitions of FAIRWARE can be obtained by trade or for the FULL cost of obtaining each.

All of the material is available to any group outside the Cleveland area only through trade. Please contact Deanna Sheridan at 20311 Lake Road, Rocky River, OH 44116 if you would like to trade programs with our group.

EDITED FROM....

THE MUSIC CORNER
BY JEFF GATLIN

NORTHEAST TARRANT HOME COMPUTER USERS'GROUP, HURST, TX.
January, 1986

Whenever I think about programming music, I think of the time it takes to key in all the CALL SOUND statements. My next concern is which programming trick to use to make the desired song 'sing' through the computer instead of spit and hiccup its way through. So far I have encountered 4 distinctive methods: 1) the CALL SOUND method, 2) the DATA method, 3) the GOSUB method, and 4) the ARRAY method.

CALL SOUND is simple: A series of CALL SOUND statements, each with different information, such as:

```
CALL SOUND(800,10,220,10,370,10)
CALL SOUND(800,165,5,247,5,415,5)
CALL SOUND(1600,110,0,227,0,440,0)
```

This produces smooth, precise sounds, but gets very tiresome for the nontypist. My first variation was setting up variables for all the notes to avoid constantly referring to the manual for the appropriate frequency. It worked, but didn't save any typing time or memory.

The DATA method involves setting up DATA statements which contain the notes, a READ statement to assign the notes to variables, a single CALL SOUND statement with variables read from data, and a clever FOR TO NEXT statement.
Example:

```
100 FOR REP=1 TO 4
110 READ A,B,C
120 CALL SOUND(400,A,0,B,0,C,0)
130 NEXT REP
140 RESTORE
150 GOTO 100
DATA 100,40000,40000,139,330,400,
147,40000,40000,165,277,440
```

The notes with frequencies of 40000 are used to create silence without having to key in a separate CALL SOUND statement with only one note. The disadvantage of this method is when you have a program with oodles of data (lots of notes), you occasionally get hiccups. The cause lies somewhere in the BASIC language. I've been told that basic generates garbage that has to be taken out occasionally. When the garbage is dumped, the computer hiccups causing the flow of the music to be interrupted. However, it usually takes quite a bit of data or an extreme tempo (speed) to cause hiccups.

The GOSUB method was introduced to me by Gerry Myers. It involves keying in one or more CALL SOUND statements (as needed) and follow them with a RETURN statement. Once done, your programming consists of redefining the CALL SOUND variables and adding a GOSUB statement to initiate the sound. Just to make things easier, set up variables for each note within an octave (see lines 100-110 below). Now instead of having a variable for every note, you alter the base variable (is this making sense). For example: BF is B flat, BF#2 is B flat 1 octave higher, BF#4 is 2 octaves higher, BF#8 is 3 octaves higher and so on. BF/2 is 1 octave lower, and BF=4 is 2 octaves lower. Explanation: well there is one, its just that if I try to explain it correctly, you'll probably put down the article and grab the TV guide. Very simply: double the frequency of any note and you'll have a note one octave higher, halve the frequency of any note and you'll have a note one octave lower. Example:

```
100 B=493.88 :: AS,BF=466.16 :: A=440.00 :: GS,AF=415.30
    :: G=392.00 ::FS,GF=369.99 :: F=349.23
110 E=329.63 :: DS,EF=311.13 :: D=293.66 :: CS,DF=277.18
    :: C=261.63 :: R=40000 :: L=250
120 FOR REP=1 TO 3
130 X=G#2 :: Y=E#2 :: Z=C :: GOSUB 500
140 X=F#2 :: Y=D#2 :: GOSUB 500 :: X=G#2 :: Y=E#2 ::
    GOSUB 500
150 Z=G/2 :: GOSUB 500 :: X=A#2 :: Y=F#2 :: Z=C ::     GOSUB
    500 :: GOSUB 500
160 X=BF#2 :: Y=G#2 :: Z=G/2 :: GOSUB 500 :: X=A#2 :: Y=F#2
    GOSUB 500
170 X=BF#2 :: Y=G#2 :: GOSUB 500 :: GOSUB 500 :: GOSUB 500
180 X=A#2 :: Y=F#2 :: GOSUB 500
190 NEXT REP
200 X=G#2 :: Y=E#2 :: Z=C/2 :: GOSUB 510
499 END
500 CALL SOUND(L,X,0,Y,0,Z,0):: RETURN
510 CALL SOUND(L#7,X,0,Y,0,Z,0)::RETURN
```

In the above example, "L" is the defined length of the note. Notes longer than "L" can be lengthened with successive GOSUBs or altered in a separate SOUND statement (as in line 510). The "voices" are defined as X,Y,Z. X is the highest voice, Y is the middle, and Z is the bass. Although it is not necessary to keep them in that order, it does help make the editing of mistakes easier. Once a voice is defined, it will remain until you change it. This causes the illusion of sustained notes behind moving notes(lines 130, 140). Once again 40000 can be used for silence (defined as R but not used in the example).

The last method is the ARRAY method. It is similar to DATA in that all the notes are in DATA statements. However, instead of reading each voice and then playing them, the notes are read into ARRAYS, then played via a FOR TO NEXT statement. Even better, each array can be a musical line. The melody can be one array, countermelody in another, and bass line in a third.

Combining this method with a negative "duration" within

the SOUND statement creates a remarkably smooth and incredibly fast musical line. When a negative duration is specified, the previous sound is stopped and the new sound is started immediately. The first question that comes to mind is how do you use negative values without getting ridiculously fast music? Easy! Just put some sort of delay between th SOUND statements. In the Bach Invention example, I've used a math function that I saw used in a program by Robert Gagle. The statement "P=2 50" causes the computer to think for a few extra milliseconds before it plays the next sound. A higher number than fifty creates a longer delay and thus a lower number creates a shorter delay. Why? It's a mystery to me but it works like a charm so I don't complain. (To get an idea of how fast the 4A can play, change Line 240 to read "for N=1 to 104 :: CALL SOUND(X,A(N),V1,B(N),V2):: NEXT N", remove line 270 and run the program.)

Once you've completed your data statements, you can write the data to a disk file to conserve program memory. This can allow you to create programs that execute extremely long songs without running out of memory while in the middle of programming (it happened to me, really!)

In the example program, I've added the option of changing the volume of either voice while the program is running and without sacrificing the smoothness of execution (well, maybe a little, occasionally). Hope you enjoy the program, and I hope this article has helped someone.

```
10 ! THIS PROGRAM USES
20 !ONE SOUND STATEMENT!!!
30 !AND TWO SIMPLE ARRAYS!!
40 !
50 !PROGRAMMED BY
60 !JEFF GATLIN
100 CALL CLEAR
110 PRINT TAB(5):"INVENTION NO.13"
120 PRINT
130 PRINT TAB(7):"BY BACH"
140 FOR T=1 TO 5 :: PRINT :: NEXT T
150 PRINT "PROGRAMMED BY JEFF GATLIN"
160 PRINT
170 PRINT "CONTROL VOLUME OF VOICES   USING '1'&'2' FOR LOUDER   AND 'Q'&'W' FOR SOFTER"
180 DIM A(104):: DIM B(104)
190 FOR N=1 TO 104 :: READ A(N):: NEXT N
200 FOR N=1 TO 104 :: READ B(N):: NEXT N
210 X=-999 :: V1=10 :: V2=10 :: P=0
220 INPUT "READY! PRESS ENTER.":UU$ :: GOTO 240
230 INPUT "PLAY AGAIN? PRESS ENTER.":UU$
240 FOR N=1 TO 104 :: CALL SOUND(X,A(N),V1,B(N),V2):: P=2^50
250 CALL KEY(0,K,S):: IF K=49 THEN V1=V1-1 ELSE IF K=50 THEN V2=V2-1 ELSE IF K=81 THEN V1=V1+1 ELSE IF K=87 THEN V2=V2+1
260 IF V1<0 THEN V1=1 ELSE IF V1>30 THEN V1=29 ELSE IF V2<0 THEN V2=1 ELSE IF V2>38 THEN V2=29
270 NEXT N
280 GOTO 230
290 DATA 40000,659,880,1047,988,659,988,1175,1047,1047,880,880,831,831,659,659
300 DATA 880,1047,1319,1047,880,1047,740,880,1047,880,740,880,622,1047,988,880
310 DATA 831,988,1175,988,831,988,587,698,831,698,587,698,494,698,659,587
320 DATA 523,659,880,659,523,659,440,523,622,523,440,523,370,523,494,440
330 DATA 415,415,988,988,831,831,659,659,40000,659,880,1047,988,659,988,1175
340 DATA 1047,880,1047,1319,1175,988,1175,1397,1319,1047,1319,1568,1397,1319,1175,1047
350 DATA 988,1047,1175,1319,1397,1175,1661,1175,1976,1175,1047,1760,1397,1175,988,1175
360 DATA 831,988,1047,880,659,880,988,831,880,659,523,659,440,440,440,440
370 DATA 523,523,440,440,415,415,330,330,440,330,440,523,494,330,494,587
380 DATA 523,659,880,659,523,659,440,523,370,440,523,440,370,440,311,370
390 DATA 330,330,415,415,494,494,415,415,330,330,247,247,208,208,165,165
400 DATA 220,220,262,262,330,330,262,262,220,220,262,262,156,156,40000,40000
410 DATA 40000,494,415,330,294,494,415,294,262,262,330,330,208,208,330,330
```

TIPs

To speed up loading infocom games, don't use Extended Basic. Use Minimemory or EditorAssembler instead. To use these, select the load and run option and type DSK1.BOOT. When this is finished loading, press ENTER until you get the program name, then type START.

On the Minimemory, you will get an error after BOOT loads, but keep pushing enter and proceed as above.

(FROM HUGers, June, 1985)

$$$$$$$$$$$$$$$$$$$$

FOR SALE: Stand-alone Disk Controller - $115
RS232 serial interface for Epson-type printers - $40

PHONE: 261-4540

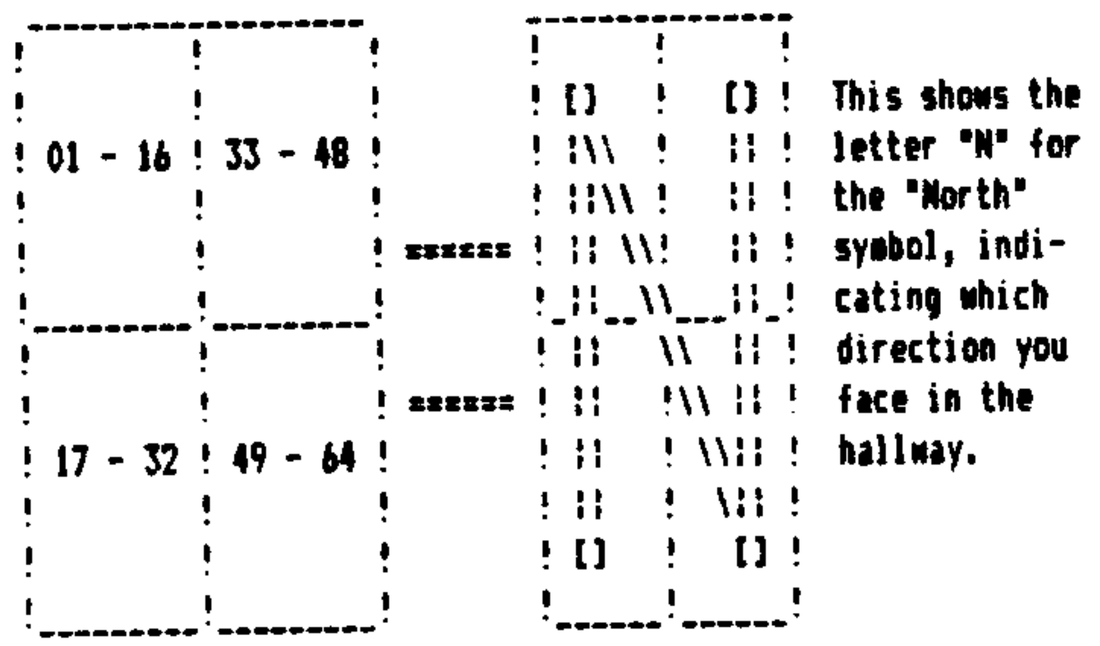
CREATING YOUR OWN TUNNELS OF DOOM ADVENTURES WITHOUT TOD EDITOR

By Dave Talan, with help from Walt Dollard (Pittsburg U6)

To create your own adventures, you will need: a disk drive, a sector editing program (preferrably DISK0), a blank initialized disk, and the TOD cartridge to test and play your game. The first step is to copy the original game, "Quest for the King" onto the blank disk, then load in DISK0. Go to sector 22 which is the start of the file. Following is the sector layout.

SECTOR #	FUNCTION OR PURPOSE
0022	Has Graphics for (in order with their respective place values): N=1,S=2,E=3,W=4,Red Map Square=5+6, Shoot Cursor=7,Fireball part 1 and 2=8
0023	Fireball part 3 and 4=1+2,Arrow 1,2,3=3,4,5, Last three segments?

An example. Sector 22 begins 00784848743A, etc...Map the first 64 numbers in a plot (if you are unsure how to map, refer to "CALL CHAR" in the TI Manual):



This shows the letter "N" for the "North" symbol, indicating which direction you face in the hallway.

0026	Contains : Fighter 1,2=1&2,Fighter 1,2=3&4, Wizard 1,2=5&6 Rogue 1,2=7&8.
0027+0028	Contains all the letters, numbers, symbols used in the game (all of the ASCII codes)
0029	Contains the Map Graphic Segments (horz. hall, vert. hall, Room, steps down & up, etc,...)
002A-002D	Too complex to explain...
002E	The Adventurer's name (Fighter, Wizard, etc.) followed by Hitpoints, Wounds, Armor, Armor Prot.Shield Prot. Weapon #1 Code, Weapon #1 Damage, Weapon #2 Code, Weapon #2 Damage, # of projectiles for 2nd weapon, ,..etc...,Level,Class ___?___, ___?___,Magic Items,ETC...

Hopefully you can convert Hex to Decimal and understand negative numbers in Hex (FF is -1, 01 is 1, 0C is 12, F4 is-12). To convert Hex to decimal take the left digit and multiply that by 16 (since hex is base 16) and then add the right digit on, i.e. 15 (hex) = 21 (decimal).

002F	In 00CA and onward are stored Fighter pictures, then into the next sector are Wizard, Rogue and Hero pictures.
0031-0035	Contains Monster data in this order: Creature Name,H.P./6(divided by 6), Defense, Attack,Maximum Damage, Special Attack,amount to change, Character ref., Sound, Mobility/Negotiation, Magic Resistance, Speed.
0036	Should be obvious
0037-003A	Graphic Characters for the monsters
0045-0047	You can figure out these.
004C	From 0090 on are the Fountain, Living Statue, Up/Downstairs
004D	Enterstore symbol, same symbol.
004E-004F	Quest objects and things that use color are stored here. Try to graph them out

---

Other things are very difficult to do. It would be like trying to teach a foreign language in one article, but I've given you a place to start. Good luck!!

If you're interested in making your own Tunnels of Doom Adventure, I have 2 utilities to design your own graphics. The program has some new monsters already defined, which may give you some ideas. The other program defines items such as Statues, Weapons, etc. which have some new graphics and names. These utilities are also in NorthCoast's library.

A D V E N T U R E S P E C I A L I N T E R E S T G R O U P

If anyone from any group is interested in starting an Adventure SIG where we could discuss a wide range of topics from solving adventures to designing your own, call Dave Talan at (216) 333-5829. I think this is a wonderful opportunity to expand your knowledge and meet new people who are also interested in adventuring. Thanks.

USING YOUR NEW DISK

By Darren Leonard
PUG, FEBRUARY, 1986

It has been drawn to my attention that a considerable number of our members are encountering some difficulty getting the programs on the Disks of the Month to run. In this column I will attempt to provide you with a concise and foolproof method to get any program running so long as it is not written in Forth or Pascal.

The first and most logical thing to do is catalog the disk using a disk manager.

Your catalog should look like the following:

```
Disknane: PUG DOM9/85
Used 340 Free 20
```

Program	Size	Format	P
MYFILE	22	PROGRAM	
MYFILE2	69	INT/VAR 254 (EXT BASIC)	
MYFILE3	49	DIS/FIX 80	
STARGAZER	33	PROGRAM (E/A #5)	
STARGAZERS	33	PROGRAM (E/A #5)	
STARGAZET	9	PROGRAM	
SPACEGAME	47	PROGRAM	
DATAFILE	87	DIS/VAR 80 (TI-WRITER)	
SCRABBLE#1	57	INT/FIX 123	
GAME#1	60	DIS/FIX 80 (E/A #3)	
GAME#2	46	DIS/FIX 80 (E/A #3)	

These are most of the possible file types you might receive.

1) Any DIS/VAR 80 type file like "DATAFILE" above are usually text or documentation files that should be read with an editor such as TI-WRITER or the Editor/Assembler package. Alternately, you can use this short BASIC program:

```
10 Open #1: "DSK1.DATAFILE"
20 OPEN #2: "PIO" (or your printer's commands.)
30 IF EOF(1) THEN 70
40 INPUT #1:X$
50 PRINT #2:X$
60 GOTO 30
70 CLOSE #1
80 CLOSE #2
90 END
```

2) Any file with a size greater than 50 and saved INT/VAR 254 is a long Extended Basic program that requires memory expansion. To use this program, you would OLD DSK1.MYFILE2 RUN from XBasic.

3) Any file in the format DIS/FIX 80 is an Assembly language program that can be loaded several ways. The surefire way is to use the Load and Run option of an E/A cartridge or a MiniMemory Cartridge. You might need the Program name to start it. If it is not given to you, search the last 5 sectors of the program with DISK0 to look for it. Usual program names include: START, BEGIN, BOARD, GAME, LOAD, RUN, etc. If you do not have an E/A or a MMM then you MAY be able to load it with Extended Basic. Memory Expansion is a MUST. You also need the program name to do this. Here is an example of an Extended basic program to run MYFILE3.

```
10 CALL INIT ----initializes memory expansion and
             prepares for loading
20 CALL LOAD ("DSK1.MYFILE3") --Loads the program into
             memory
30 CALL LINK ("START")  --Starts the program IF start is
             the program name.
```

This will work most of the time, but not always. Therefore, your best bet is to use an E/A or MMM if you have one.

4) Any file that is in program format and is EXACTLY 33 sectors in size is most likely an assembly program file that will load out of option #5 of the Editor/Assembler module or the Utility option of TI-Writer.

5) Files that are program format and are longer than 33 sectors are either Basic or Extended Basic. Usually try Extended Basic first. If you get an error, try Basic. It might be necessary to free extra memory in console basic if you get a memory full error. To do this type:

```
CALL FILES (1)
NEW
OLD DSK1.SPACEGAME
RUN
```

If you still get a memory full error, then it should be an extended basic program, or will only run on tape without the disk drive attached.

6) Any file in a program format that is less than 33 sectors could be a Basic, Extended Basic, or Assembly program. Try them in that order.

7) Any file that is in Program format and is 52 sectors long could be a Tunnels of Doom file. If you have the Tunnels of Doom Module, try it. If it is 54 sectors long, it could be an adventure game: if you have the Adventure module, try it.

8) When you have 2 consecutively named DIS/FIX 80 files, you must load the second file BEFORE you give the program name.

9) Any other type of format is most likely a data file that is used by a program on the disk. They will not run, unless it is a merge format, but in genral should be left on the disk with the other programs.

HOW TO CONVERT ASSEMBLY PROGRAMS TO PROGRAM FORM FOR FASTER LOADING AND LESS DISK SPACE

By Darren Leonard, PUG (Pittsburg Users Group)
-- on an idea by Marty Kroll, Jr.
February, 1986

If you have ever loaded an assembly program with editor/assembler Option #3, you may have noticed that it takes quite a while to load. With some programs this can take over 3 minutes. These are Display/Fixed 80 format programs which are going to change to PROGRAM format to load with OPTION #5. In addition to loading 3 to 5 times faster, programs stored in program format, i.e., Memory Image, take as little as 1/4 the disk space of D/F 80 files.

The method outlined in this article will work on 95% of all Assembly D/F 80 programs. Prior to writing this, I tried it on 20 programs and it worked on 19 of them. It will even allow you to save an ASSEMBLY program to cassette. Thus, people with an E/A and 32K can run assembly programs!

To begin with, read page 420 of the Editor/Assembler manual. Try your program the way they outline it. If you get an error, then read on, and I will explain in detail how to get around it.

This section describes the procedure for D/F 80 files that DO NOT AUTOSTART!. If your program does autostart, read down a few paragraphs on how to remove it with DISK0.

1) Plug in your E/A and call up TI-BASIC, your E/A must be plugged in!

2) Type CALL INIT
   CALL LOAD("DSK1.FILENAME")

3) If your program has more than one file, type in all the remaining files in order as follows:
   CALL LOAD("DSK1.GAME#1")
   CALL LOAD("DSK1.GAME#2")
   CALL LOAD("DSK1.GAME#3")
   get the idea?

4) Type CALL PEEK(8228,A,B)
   PRINT A,B

5) Now 2 numbers will appear on the screen, one on the left and one in the middle. This number corresponds to the first free address in the memory which is also the last address of your program.

6) Convert these numbers to hex and add A+B to come up with a 4-digit hexadecimal number. Since your program is normally loaded in memory from address >A000->FFD7, if you get A000 for A+B, then your program has an Absolute Origin Statement (AORG) and you you will not be able to convert it with this method. Similarly if A+B is A780 or smaller, then the program is loaded in an unusual manner since it cannot fit in in the small area from >A000-A780. But, if you come up with A+B=B000 or greater, then this method will work 99% of the time.

7) Type "BYE" and call up the editor. Now type in the small assembly program listed here:

```
      DEF SFIRST,SLAST,SLOAD
SFIRST EQU >A000
SLOAD  EQU >A000
SLAST  EQU >A780 (the value of A+b)
END
```

NOTE !! PUT THE HEX NUMBER OF A+B IN THE PLACE WHERE A780 IS

Hit FCTN 9 twice and save to disk.

8) Load the Assembler.

   For source file enter what your saved in step 7.
   For object file type DSK1.GAME#4 or what you want.
   Hit return for the printout output.
   Type "RC" when it prompts for assembler directives.
   It will then assemble the program. You shouldn't get any errors.

9) Now load E/A option 3.
   Enter your filename: DSK1.GAME#1
   DSK1.GAME#2
   Then enter the
   assembled file name DSK1.GAME#4, step 8.

10) Insert E/A Disk #2 into drive one and load file DSK1.SAVE. Hit enter and type "SAVE" for the program name. Follow the screen input prompts.

11) Now hit FCTN + and call up E/A option #5 and type DSK1.YOURFILE and wala!

TROUBLESHOOTING THE PROCEDURE

If you encountered an error in steps 1-11 above there is still hope!

If you received an error in Step 9 when you attempted to load your assembled program, and that error was a "DUPLICATE DEF", you can attempt to figure which is the duplicate: SFIRST, SLAST OR SLOAD by 2 ways.

1) If you have DISK0 load it up and search your program file for SFIRST, SLAST OR SLOAD on your disk and change them to TLOAD, TLAST OR TFIRST AT EVERY PLACE they occur!! BE SURE TO CHECK THE LAST 3 SECTORS OF THE PROGRAM THOROUGHLY!!!
   Then go back and try STEP 9 AGAIN.

2) Change the Assembly program in 7 to allow all COMBINATIONS

```
      DEF SFIRST, SLAST      Try it eliminating
SFIRST EQU >A000             all three one at a
SLAST  EQU >A+B              time        END
```

If this doesn't work, you will have to wait until part 2 of this article comes out.

HOW TO ELIMINATE AUTOSTART FUNCTION ON D/F 80 PROGRAMS

If your program autostarts, you cannot use the above procedure because it will take over control of the machine. You can remove that feature if you have DISKO.

Load up DISKO and examine the last 3 sectors of your program for the following: (in hex mode) 20314523462020
** **

The thing to look for is the 31 and the 46 with the address between them. Change the 31 to a 40 or change the entire sequence to 20 After change it should look like this:

20202020202020 or 20404552462020
** ** ** **

HOW TO CONVERT FROM DECIMAL TO HEXADECIMAL

This might appear quite intimidating, but I assure you that it is very simple. I will not go over the principles of HEXADECIMAL numbering systems, but I will show you how to convert it.

Decimal	Hexadecimal	Binary	Octal
0	0	0	0
1	1	1	1
2	2	10	2
3	3	11	3
4	4	100	4
5	5	101	5
6	6	110	6
7	7	111	7
8	8	1000	10
9	9	1001	11
10	A	1010	12
11	B	1011	13
12	C	1100	14
13	D	1101	15
14	E	1110	16
15	F	1111	17
16	10	10000	20
17	11	10001	21
18	12	10010	22

AD INFINTUM...

The number in the left colum represents the numbers you are familiar with. In the second, third and fourth colums are the equivalent numbering systems.

Take A from step 4 above. Say it is 213 which is in decimal. Divide by 16 213/16=13.3125. Take the part to the left of the decimal point, which in this case is 13 and convert to Hex from above chart...13=D.

Now take 213- (13)=5 and this =5 in hex. Therefore, your hex number is D5 which equals 213 decimal.

Do the same for B and add the D5 to what you obtain for B. If the first digit is not an A,B,C,D,E, or F, you have an invalid address, or you have incorrectly converted to hex.

By doing the exact reverse of the above, you can go from HEX to DECimal.

HARDWARE HINTS

EXTERNAL DISK DRIVE POWER SUPPLY UPDATE

By Ken Gladyszewski
(NorthCoast 99ers)

The 12-volt requirement is 1 1/2 amps, not 1/2 as previously reported. Since the publication of this article, Martin Smoley has built the power supply described with success. He cautions that EXTREME care be exercised regarding the polarity of the electrolytic capacitors. The negative side of each attaches to the common line!! Improper installation results in overheating and possible leakage. He, therefore, suggests bench testing the supply for an hour before installing it in a case with a disk drive.

As previously suggested, a complete ready-made power supply might be purchased inexpensively. It should be noted that one having a single or dual voltage other than the required +5 and +12 volts can be used along with just one or two additional voltage regulators.

FROM TIGERCUB TIPS #30

A useful tip from Stephen Shaw in England - if you have a long program which will run only in Basic, and which will load from disk with CALL FILES(1), but runs out of memory when you try to run it; and if you have the MiniMemory module - Insert MiniMemory module, select Basic, enter SAVE EXPMEM2. When SAVEd, enter CALL LOAD(-31888,63,244), enter NEW, enter OLD EXPMEM2, and enter RUN. That is still a lot faster than loading along program from tape!

If you are printing out of TI-Writer Editor, finish your letter with CTRL U, SHIFT L, CTRL U and when it is printed, the paper will automatically feed to the top of the next sheet.

```
100 ! ***** MS/LABELS ***** By: Martin A. Saoley ***** For EPSON Printer *****
110 !                 ***** NorthCoast 99er's UG *****
120 OPEN #9:"P1O" !  OPEN PRINTER (Could be RS232)    *** Extended Basic ***
130 PRINT #9:CHR$(27);"O";CHR$(27);"8";!
       "O"=STOP skip over perf,"8"=STOP paper end detector
140 CALL CLEAR :: CALL SCREEN(13)
150 PRINT "     ** MS/LABELS **": :"        PRINTS": :"    3-1/2in BY 15/16in":
 :"         LABELS": : :
160 PRINT " Enter Data at Prompts!": :" You will have 4 line per": :" label.  Li
ne #1 = 15 Cols.": :"         Line #2 = 28 Cols.": :
170 PRINT " Lines #3 and #4 = 49 Cols.": : :
180 GOSUB 190 :: GOSUB 210 :: GOSUB 220 :: GOSUB 230 :: GOTO 240
190 PRINT :: PRINT "              ^^^^^^^^^^^^^^^^"
200 INPUT "ENTER LINE 1 ":A$ :: RETURN
210 PRINT :: PRINT "  ENTER LINE #2" :: INPUT "^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^":B$
:: RETURN
220 PRINT :: PRINT "  ENTER LINE #3" :: INPUT "0^^^^^^^^1^^^^^^^^^2^^^^^^^^^3^^^
^^^^^4^^^^^^^49       ":C$ :: RETURN
230 PRINT :: PRINT "  ENTER LINE #4" :: INPUT "0^^^^^^^^1^^^^^^^^^2^^^^^^^^^3^^^
^^^^^4^^^^^^^49       ":D$ :: RETURN
240 PRINT :: INPUT "HOW MANY COPYS ":X
250 CALL CLEAR :: PRINT "  Hold >Q< to Quit Printing": : : : : :
260 FOR I=1 TO X !   ******** PRINTOUT LOOP ********
270 ! PRINT #9:CHR$(27);"G";!            START DOUBLE STRIKE OPTIONAL
280 PRINT #9:CHR$(27);"E";!               START EMPHASIZED
290 ! PRINT #9:CHR$(27);"M";!           Start Elite-size(makes #1=18 characters)
300 PRINT #9:CHR$(27);"W";CHR$(1);!        START ENLARGED
310 PRINT #9:A$
320 PRINT #9:CHR$(27);"W";CHR$(0);!        STOP ENLARGED
330 ! PRINT #9:CHR$(27);"P";!           Stop Elite-size(Needed if 290 is used)
340 PRINT #9:" ";B$;CHR$(27);"F" !         STOP EMPHASIZED
350 PRINT #9:CHR$(27);CHR$(15);" ";C$;:;" ";D$;CHR$(18);CHR$(27);"H";!
CHR$(15)=START CONDENSED+CHR$(18)=STOP,"H"=STOP DOUBLE STRK.
360 FOR K=1 TO 3 :: PRINT #9 :: NEXT K
370 CALL KEY(0,K,S):: IF K=81 OR K=113 THEN 390
380 NEXT I
390 CALL CLEAR :: CALL SCREEN(6)!     ******** Beginning of TASK SCREEN ********
400 PRINT " Enter M for More labels": :"       N for New labels": :"       L to
Change a line": :
410 PRINT "       Q to Quit the program": :
420 INPUT " Enter your chioce: ":DO$
430 IF DO$="M" OR DO$="m" THEN CALL CLEAR :: GOTO 240
440 IF DO$="N" OR DO$="n" THEN 140
450 IF DO$="L" OR DO$="l" THEN 480
460 IF DO$="Q" OR DO$="q" THEN 520
470 GOTO 420
480 CALL CLEAR !                ******** Beginning of LINE CHANGE SCREEN ********
490 INPUT "  Enter line number to be          changed  1 to 4 ":L :: IF L<1 OR
L>4 THEN 490
500 ON L GOSUB 190,210,220,230
510 GOTO 390
520 PRINT #9:CHR$(27);"@";! Initialize Printer = Wipe out any leftover commands
530 CLOSE #9
540 ! *** MS/LABELS ***
550 END
```

MS/LABELS-DOC

"MS/LABELS" started out to be a small, simple program to print 3-1/2 in X 15/16 in. labels for return addresses and disk labels, but it evolved into the program you see at the left.

THE USER INSTRUCTIONS FOLLOW

(1) Load the program (Don't run it yet).

(2) Align your labels in the printer then turn the printer on.

(3) Now RUN the program.

(4) Enter the data as prompted by the program. There is one circuaflex (^) for each space on the entry line. Do not use any commas.

(5) After you have entered (4) lines the program will ask how many labels you want. If you want to see one enter 1. After the label is printed you will see a screen which will let you print (M)ore if you like what you see.

(6) If you don't like them enter L to change a line and then the line number you would like changed. You can repeat the L for as many lines as you need, or you can use M for more and print one at any time until you like th. label you have. At this point you use More, then type in the quantity you want and the printer will start running them off.
If you change your mind, HOLD >Q< until the printer stops and you will return to the task screen.

(7) At the task screen you can also enter an (N) if you want a coopletely New label or (Q)uit to exit the program.

NOTE: If your ribbon is not dark enough you can edit the prograe and delete the (!) and the space from the beginning of line 270 This will give you Double Strike throughout. Also! Doing the same thing to line Nos. 290 and 330 will give you 18 characters in line #1 if your printer is capable of Elite Print (You will have to reaember that you have (3) characters past the last (^) in line one.)

If you do not like to type, my programs are in the NorthCoast 99er's Library.
Good Luck! Marty

-- MS/LABELS --
TI99/4A Extended Basic
This label was made by the program listed above.
Ln.#1=ENLARGED #2=Std. size #3&#4=Condensed

Welcome to
St. Silicon's Hospital
and
Information Dispensary

somewhat affiliated with

The Department of Family Medicine
Case Western Reserve University
School of Medicine

THE Administrator (Sysop):
T.M. Grundner, Ed.D

Chief of Staff:
Robert E. Garrett, M.D.

ST. SILICON'S HOSPITAL
DIRECTORY

1. THE MAIN LOBBY
   - General Message Board & E-Mail
2. THE INFORMATION DESK
   - Patient Records, Visitors Guide
3. THE WAITING ROOM
   - Hosp. Newspaper & FREE Software
4. THE MEDICAL CLINIC
   - Doc-In-The-Box - Medical Q & A
5. THE DENTAL CLINIC
   - Dent-in-the-Box - Dental Q & A
6. THE PSYCHIATRIC CLINIC
   - Shrink-in-the-Box??? <== NEW
7. THE STAFF LOUNGE
   - CME, Grants, Seminar Informatio
8. THE LOADING DOCK
   - Uploading Files to St. Silicon
0. DISCHARGE

Those of you who have modems, might dial up (216) 368-3888 for a short or long stay at St. Silicoon's Hospital and Information Dispensary. No doubt many will recognize the Administrator's name from the excellent articles which appeared in the Plain Dealer a couple of years ago.

After registering, you can visit any of the above departments you wish. At the information desk you can sign up for a free users guide which walks you through the system and describes each department in detail. This should be an informative and educational way to get some use from your modem.

DEANNA SHERIDAN, NORTHCOAST 99'ers

NEW PROGRAMS AT NORTHCOAST'S LIBRARY, ETC.

As reported at Saturday's meeting, new programs are coming in from exchanges with other users groups.

Additions to FREEWARE are: FAST-TRANS, a complete checkbook and budget recordkeeping system. Instructions and sample files are on one disk, and the program on another; TEII ENCODER, allows the Bulletin Board to do graphics, transmit graphics, etc., HOMEPLAN2, a spreadsheet program; EASYSPRITE, another spritebuilding program, and an Extended Basic loader for A/L programs from Option 85. There will be several more either ordered or here by the next meeting.

Others of interest are: a public-domain TI-WRITER which loads from Extended Basic, and is very fast. This disk also contains an Editor/Assembler and DM1000 so that you can use these 3 popular programs without changing disks or cartridges. Note, those of you who have CorComp controllers probably will not be able to use the E/A feature. This E/A is from Italy and has certain "blank" sectors which the CorComp will not recognize. Also, any backup must be made with a "clone" copier to pass on the empty sectors.

Thanks to Dave Talan's help in cataloging, these will probably be ready at the next meeting. There are 2 or 3 disks of utilities to help in programming, a screen dump which rivals the commercial one by Quality 99, public domain disassemblers, printing routines, 2 disks of Sam Moore, Jr. creations, 2 disks which will print out posters, wrapping paper, labels, etc. There are 3 disks of LOGO programs, so if you were putting off buying it at the current $19.95 price because you didn't think there would be anything to do with it, be assured there is because there are more where these came from. Also, if anyone who has LOGO would like the job of cataloging the disks we have, please let me know. Since no one has asked for them, they have been given low priority. There is also a disk of FORTH programs which should be cataloged. Any volunteers?

Saturday, a voluntary donation box for contributions was placed in the copying area. It was explained that if everyone would contribute a small amount when having disks copied, it would not be necessary to impose a copying fee at some later time. Also, ALL of the money donated would be reinvested in postage, mailers, disks, etc. for additional public domain and freeware software. It was reported there were several dollars in the box after this announcement. Thank you for your support. The really good stuff is just starting to come in, and there seems to be no end to it in reading other groups newsletters.

DEANNA SHERIDAN, NORTHCOAST LIBRARIAN

RAMDISK (CONTINUED from page 1)

The manual details how to write your own call statements that can be loaded on the card and can be executed from BASIC or EX-BASIC just like "CALL CLEAR or CALL COLOR()".

The RAMDISK is compatible with all applications using a standard DSRLNK, including: disk managers (except the CorComp disk manager, which will not recognize the RAMDISK), sector copy programs and disk editors. BASIC programs can be saved or loaded and files can be opened and accessed with normal I/O operations. Assembly files can be loaded on the card and executed with the normal "CALL LOAD("DSKn.filename")".

This card is also available for storage of machine code similar to 32K expansion memory, by turning off the card's DSR. The card emulates a floppy drive only because that's what the operating system emulates. So you could write your own operating system to emulate a print spooler, or whatever you can imagine. The possibilities are limited only by your need and imagination.

The RAMDISK is being offered in 3 packages:

1. As a schematic with parts and assembly instructions.
2. As a bare printed circuit board with parts list and intructions.
3. As a ready-to-run card, fully assembled and tested with a 90-day warranty.

All packages come with the operating system with commented source code, documentation and development software. For more information contact:

HORIZON COMPUTER PERIPHERALS
213 EARL STREET
WALBRIDGE, OH 43465

NEW ADDRESS
CLEVELAND AREA 99/4A USERS GROUPS
P.O. BOX 23283
EUCLID, OHIO 44123

CHECK YOUR EXPIRATION DATE.
THIS MAY BE YOUR LAST ISSUE!

Pushcart 1880s
125
USA

04/86

!! TIME DATED MATERIAL !!