OUR NEXT MEETING will be on Friday,

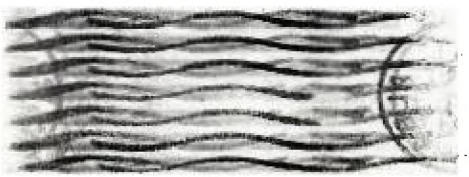
JANUARY 27, 1984 at 7:30 pm

PLACE: KEY BANK BLDG.

SW corner of Rt. 20 and Rt. 155

THE FEBRUARY MEETING will be on Friday,

?



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

VOL I, NO. 11 JANUARY, 1984

Edmonton Users Group
P.O. Box 11933
Edmonton, Alberta T5J 3L1

M I N I - M E M O R Y

Since my last column, I have acquired and started to play with the EDITOR/ASSEMBLER and now know why Jon Daggett (the previous author of this column) was so anxious to get out of the MINI-MEMORY and into the full blown EDITOR/ASSEMBLER. The main advantage of the MINI-MEMORY is that, although it is limited in capability, it does not require MEMORY EXPANSION and a DISK SYSTEM to operate. Should we change this column to a general one on assembly language? What's your opinion? Is any one out there actually using the MINI-MEMORY? To participate in this decision, you have to let me know what you want.

Remember that there are things that you can use your MINI-MEMORY for other than writing and executing assembly language programs. When you are in TI-BASIC with the module in place, you have available 7 new subprograms:

INIT LOAD LINK PEEK PEEKV POKEV CHARPAT

These subprograms can be CALL'ed in TI-BASIC programs to speed up some of your basic programs, to get information from the machine, to activate sprites, or even to do a little bit of hacking that you might have thought was denied to you if you did not have a full-blown system (see Item 3 in the Hints From Henry column). LOAD and PEEK operate on CPU and, POKEV and PEEKV operate on VDP RAM. To wet your appetite and maybe even motivate you to blow the dust off your module, try the following TI-basic program (from Bill Gronos/Int'l Users-group newsletter) with MINI-MEMORY in-place:

```
100 CALL CLEAR
110 CALL POKEV(768,98,128,161,1,208)
120 CALL POKEV(1920,50,50)
130 CALL LOAD(-31878,1)
140 GOTO 140
```

M i k e H e n r y

B O O K R E V I E W

The SMART PROGRAMMING GUIDE FOR SPRITES by Craig G. Miller is a short book that is a must for any TI-99 owner who has an EXTENDED BASIC module and is serious about using sprites. This was another mail order purchase made with fear and trepidation from one of those small ads in 99'ER MAGAZINE. It was a great success. The price was $5.95 plus $1.50 shipping & handling. Before even getting serious about sprites, Mr. Miller treats us to 27 pages on general tips, conversion formulas to take you back and forth between text and graphic rows and columns, and the use of CALL CHAR, CALL JOYST, CALL KEY, and CALL PEEK.

Miller then shows us some terrific concepts on sprites. He makes the learning process easy by the use of a short, dramatic example for each concept. He shows us how to have one sprite chase another, how to overcome difficulties we often have with CALL COINC by using CALL PEEK and CALL SOUND instead, and how to have a sprite pick up or lay down things as it moves. As a bonus he finishes with a program called MAZE PUZZLE consisting of only 7 lines, and a GENERAL BAR GRAPHER of only 18 lines (both have multiple statements per line).

This is clearly the best buy I have ever seen in a book for the TI-99. If you want to improve your ability to get the most out of graphics with EXTENDED BASIC on you 99/4, then order this book from Miller Graphics or nag your dealer to stock it.

M i k e H e n r y

FORMAT	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	OP CODE		B		Td			D			Ts		S			
2	OP CODE										RELATIVE JUMP					
3	OP CODE							D			Ts		S			
4	OP CODE							C			Ts		S			
5	OP CODE								C				W			
6	OP CODE										Ts		S			
7	OP CODE												N			
8	OP CODE											N	W			
	IMMEDIATE VALUE															
9	OP CODE							D			Ts		S			

KEY		Td/Ts FIELD CODES	
B	1=byte 0=word	00	Register RX
Td	destination address mode	01	Indirect *RX
D	destination address	10	with R0,symbolic @>XXXX
Ts	source address mode	10	with R1-R15,indexed @>XXXX(RX)
S	source address	11	Indirect with increment *RX+
C	counter		
W	register number		
N	unused		

RELATIVE JUMP from +127 to -128

RELATIVE JUMP CHART

lsd / msd	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32
1	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62	64
2	66	68	70	72	74	76	78	80	82	84	86	88	90	92	94	96
3	98	100	102	104	106	108	110	112	114	116	118	120	122	124	126	128
4	130	132	134	136	138	140	142	144	146	148	150	152	154	156	158	160
5	162	164	166	168	170	172	174	176	178	180	182	184	186	188	190	192
6	194	196	198	200	202	204	206	208	210	212	214	216	218	220	222	224
7	226	228	230	232	234	236	238	240	242	244	246	248	250	252	254	256
8	-254	-252	-250	-248	-246	-244	-242	-240	-238	-236	-234	-232	-230	-228	-226	-224
9	-222	-220	-218	-216	-214	-212	-210	-208	-206	-204	-202	-200	-198	-196	-194	-192
A	-190	-188	-186	-184	-182	-180	-178	-176	-174	-172	-170	-168	-166	-164	-162	-160
B	-158	-156	-154	-152	-150	-148	-146	-144	-142	-140	-138	-136	-134	-132	-130	-128
C	-126	-124	-122	-120	-118	-116	-114	-112	-110	-108	-106	-104	-102	-100	-98	-96
D	-94	-92	-90	-88	-86	-84	-82	-80	-78	-76	-74	-72	-70	-68	-66	-64
E	-62	-60	-58	-56	-54	-52	-50	-48	-46	-44	-42	-40	-38	-36	-34	-32
F	-30	-28	-26	-24	-22	-20	-18	-16	-14	-12	-10	-8	-6	-4	-2	0

PATTERN DESCRIPTOR TABLE MEMORY LOCATION

CHAR	LOC	CHAR	LOC	CHAR	LOC	CHAR	LOC	CHAR	LOC	CHAR	LOC
0	>0800	1	>0808	2	>0810	3	>0818	4	>0820	5	>0828
6	>0830	7	>0838	8	>0840	9	>0848	10	>0850	11	>0858
12	>0860	13	>0868	14	>0870	15	>0878	16	>0880	17	>0888
18	>0890	19	>0898	20	>08A0	21	>08A8	22	>08B0	23	>08B8
24	>08C0	25	>08C8	26	>08D0	27	>08D8	28	>08E0	29	>08E8
30	>08F0	31	>08F8	32	>0800	33	>0808	34	>0810	35	>0818
36	>0920	37	>0928	38	>0930	39	>0938	40	>0940	41	>0948
42	>0950	43	>0958	44	>0960	45	>0968	46	>0970	47	>0978
48	>0980	49	>0988	50	>0990	51	>0998	52	>09A0	53	>09A8
54	>09B0	55	>09B8	56	>09C0	57	>09C8	58	>09D0	59	>09D8
60	>09E0	61	>09E8	62	>09F0	63	>09F8	64	>0900	65	>0908
66	>0A10	67	>0A18	68	>0A20	69	>0A28	70	>0A30	71	>0A38
72	>0A40	73	>0A48	74	>0A50	75	>0A58	76	>0A60	77	>0A68
78	>0A70	79	>0A78	80	>0A80	81	>0A88	82	>0A90	83	>0A98
84	>0AA0	85	>0AA8	86	>0AB0	87	>0AB8	88	>0AC0	89	>0AC8
90	>0AD0	91	>0AD8	92	>0AE0	93	>0AE8	94	>0AF0	95	>0AF8
96	>0B00	97	>0B08	98	>0B10	99	>0B18	100	>0B20	101	>0B28
102	>0B30	103	>0B38	104	>0B40	105	>0B48	106	>0B50	107	>0B58
108	>0B60	109	>0B68	110	>0B70	111	>0B78	112	>0B80	113	>0B88
114	>0B90	115	>0B98	116	>0BA0	117	>0BA8	118	>0BB0	119	>0BB8
120	>0BC0	121	>0BC8	122	>0BD0	123	>0BD8	124	>0BE0	125	>0BE8
126	>0BF0	127	>0BF8	128	>0800	129	>0808	130	>0810	131	>0818
132	>0C20	133	>0C28	134	>0C30	135	>0C38	136	>0C40	137	>0C48
138	>0C50	139	>0C58	140	>0C60	141	>0C68	142	>0C70	143	>0C78
144	>0C80	145	>0C88	146	>0C90	147	>0C98	148	>0CA0	149	>0CA8
150	>0C80	151	>0CB8	152	>0CC0	153	>0CC8	154	>0CD0	155	>0CD8
156	>0CE0	157	>0CE8	158	>0CF0	159	>0CF8	160	>0C00	161	>0C08
162	>0D10	163	>0D18	164	>0D20	165	>0D28	166	>0D30	167	>0D38
168	>0D40	169	>0D48	170	>0D50	171	>0D58	172	>0D60	173	>0D68
174	>0D70	175	>0D78	176	>0D80	177	>0D88	178	>0D90	179	>0D98
180	>0DA0	181	>0DA8	182	>0DB0	183	>0DB8	184	>0DC0	185	>0DC8
186	>0DD0	187	>0DD8	188	>0DE0	189	>0DE8	190	>0DF0	191	>0DF8
192	>0E00	193	>0E08	194	>0E10	195	>0E18	196	>0E20	197	>0E28
198	>0E30	199	>0E38	200	>0E40	201	>0E48	202	>0E50	203	>0E58
204	>0E60	205	>0E68	206	>0E70	207	>0E78	208	>0E80	209	>0E88
210	>0E90	211	>0E98	212	>0EA0	213	>0EA8	214	>0EB0	215	>0EB8
216	>0EC0	217	>0EC8	218	>0ED0	219	>0ED8	220	>0EE0	221	>0EE8
222	>0EF0	223	>0EF8	224	>0E00	225	>0E08	226	>0E10	227	>0E18
228	>0F20	229	>0F28	230	>0F30	231	>0F38	232	>0F40	233	>0F48
234	>0F50	235	>0F58	236	>0F60	237	>0F68	238	>0F70	239	>0F78
240	>0F80	241	>0F88	242	>0F90	243	>0F98	244	>0FA0	245	>0FA8
246	>0F80	247	>0F88	248	>0FC0	249	>0FC8	250	>0FD0	251	>0FD8
252	>0FE0	253	>0FE8	254	>0FF0	255	>0FF8				

Note: Values are for Editor/Assembler Defaults.

COLOR TABLE

Address	Char (dec)	Char (hex)	Set #	TI Basic Equiv.
>380	0 - 7	>0 - >7	>0	NA
>381	8 - 15	>8 - >F	>1	NA
>382	16 - 23	>10 - >17	>2	NA
>383	24 - 31	>18 - >1F	>3	NA
>384	32 - 39	>20 - >27	>4	1
>385	40 - 47	>28 - >2F	>5	2
>386	48 - 55	>30 - >37	>6	3
>387	56 - 63	>38 - >3F	>7	4
>388	64 - 71	>40 - >47	>8	5
>389	72 - 79	>48 - >4F	>9	6
>38A	80 - 87	>50 - >57	>A	7
>38B	88 - 95	>58 - >5F	>B	8
>38C	96 - 103	>60 - >67	>C	9
>38D	104 - 111	>68 - >6F	>D	10
>38E	112 - 119	>70 - >77	>E	11
>38F	120 - 127	>78 - >7F	>F	12
>390	128 - 135	>80 - >87	>10	13
>391	136 - 143	>88 - >8F	>11	14
>392	144 - 151	>90 - >97	>12	15
>393	152 - 159	>98 - >9F	>13	16
>394	160 - 167	>A0 - >A7	>14	NA
>395	168 - 175	>A8 - >AF	>15	NA
>396	176 - 183	>B0 - >B7	>16	NA
>397	184 - 191	>B8 - >BF	>17	NA
>398	192 - 199	>C0 - >C7	>18	NA
>399	200 - 207	>C8 - >CF	>19	NA
>39A	208 - 215	>D0 - >D7	>1A	NA
>39B	216 - 223	>D8 - >DF	>1B	NA
>39C	224 - 231	>E0 - >E7	>1C	NA
>39D	232 - 239	>E8 - >EF	>1D	NA
>39E	240 - 247	>F0 - >F7	>1E	NA
>39F	248 - 255	>F7 - >FF	>1F	NA

Note: Values are for Editor/Assembler Defaults.

Hexadecimal Code	Mnemonic Code	Name	Format	Section
0200	LI	Load Immediate	VIII	10.1
0220	AI	Add Immediate	VIII	6.4
0240	ANDI	AND Immediate	VIII	11.1
0260	ORI	OR Immediate	VIII	11.2
02B0	CI	Compare Immediate	VIII	8.3
02A0	STWP	STore Workspace Pointer	VIII	10.7
02C0	STST	STore STatus	VIII	10.6
02E0	LWPI	Load Workspace Pointer Immediate	VIII	10.3
0300	LIMI	Load Interrupt Mask Immediate	VIII	10.2
0380	RTWP	ReTurn with Workspace Pointer	VII	7.17
0400	BLWP	Branch And Load Workspace Pointer	VI	7.3
0440	B	Branch	VI	7.1
0480	X	EXecute	VI	7.18
04C0	CLR	CLeaR operand	VI	11.5
0500	NEG	NEGate	VI	6.11
0540	INV	INVert	VI	11.4
0580	INC	INCrement	VI	6.8
05C0	INCT	INCrement by Two	VI	6.9
0600	DEC	DECrement	VI	6.5
0640	DECT	DECrement by Two	VI	6.6
0680	BL	Branch and Link	VI	7.2
06C0	SWPB	SWaP Bytes	VI	10.8
0700	SETO	SET to One	VI	11.6
0740	ABS	ABSolute value	VI	6.3
0800	SRA	Shift Right Arithmetic	V	12.1
0900	SRL	Shift Right Logical	V	12.2
0A00	SLA	Shift Left Arithmetic	V	12.3
0B00	SRC	Shift Right Circular	V	12.4
1000	JMP	Unconditional JuMP	II	7.11
1100	JLT	Jump Less Than	II	7.10
1200	JLE	Jump if Low or Equal	II	7.9
1300	JEQ	Jump EQual	II	7.4
1400	JHE	Jump High or Equal	II	7.6
1500	JGT	Jump Greater Than	II	7.5
1600	JNE	Jump Not Equal	II	7.13
1700	JNC	Jump No Carry	II	7.12
1800	JOC	Jump On Carry	II	7.16
1900	JNO	Jump No Overflow	II	7.14
1A00	JL	Jump if logical Low	II	7.8
1B00	JH	Jump if logical High	II	7.7
1C00	JOP	Jump Odd Parity	II	7.15
1D00	SBO	Set CRU Bit to One	II	9.2
1E00	SBZ	Set CRU Bit to Zero	II	9.3
1F00	TB	Test Bit	II	9.5
2000	COC	Compare Ones Corresponding	III	8.4
2400	CZC	Compare Zeros Corresponding	III	8.5
2800	XOR	EXclusive OR	III	11.3
2C00	XOP	EXtended OPeration	IX	7.19
3000	LDCR	LoaD CRU	IV	9.1
3400	STCR	STore CRU	IV	9.4
3800	MPY	MultiPlY	IX	6.10
3C00	DIV	DIVide	IX	6.7
4000	SZC	Set Zeros Corresponding	I	11.9
5000	SZCB	Set Zeros Corresponding, Byte	I	11.10
6000	S	Subtract words	I	6.12
7000	SB	Subtract Bytes	I	6.13
8000	C	Compare words	I	8.1
9000	CB	Compare Bytes	I	8.2
A000	A	Add words	I	6.1
B000	AB	Add Bytes	I	6.2
C000	MOV	MOVe words	I	10.4
D000	MOVB	MOVe Bytes	I	10.5
E000	SOC	Set Ones Corresponding	I	11.7
F000	SOCB	Set Ones Corresponding, Bytes	I	11.8

H I N T S F R O M H E N R Y

We still do not know what will happen to 99'ER MAGAZINE. We hope that it will continue to be published, but there is concern that if it starts to lose support from its advertisers then they may cease to be a source for us of the many useful little hints that it has provided in the past. We are still getting good ideas from "ENTHUSIAST'99" and the TIGERCUB SOFTWARE newsletter, and we will pass them on to you.

The time is right to share your ideas with your fellow 99/4A users. If you have hints and/or tricks that you think are worth sharing with your fellow club members, send them to my home at 734 Wright Avenue, Schenectady, New York 12309, to me c/o the Users Group at P.O. Box 13522 in Albany, or talk to me at our monthly meetings. If you pick up an idea someplace else that you have not seen appear in this column, send that along. Just include a note as to where you found it and we can give an appropriate credit when we use it.

Item 1:
Don Wemple found a low cost source for an adapter to let you use a low cost Atari joystick as joystick #1 with the TI-99 computer. It is the CHAMP ADAPTER (No.2 PC-310) made by Championship Electronics, Inc. It only lets you plug in one joystick, and some canned software (even some that use only one joystick) accesses joystick #2, but it may help you where appropriate. I paid $4.99 for the adapter at a local discount store and have seen the simple Atari joystick on sale from $4.99 to $6.99 at similar stores.

Item 2:
Having trouble beating the CHESS module? I only play at the NOVICE level and I have discovered (by painful trial and error) a fact of life. We make mistakes and the CHESS module doesn't. Therefore take all the even swaps that you can get as early as you can. As the total number of men left on the board decreases, we make fewer mistakes and thus play a better game. The computer plays its normal consistent game and therefore our game gets better relative to the computer's. Next month I'll tell you about a fascinating game I evolved using the CHESS module that provides short strategy filled games.

Item 3:
Did you ever lose what you were working on by resetting the computer back to the title screen accidently by pressing FCTN-QUIT when you meant to press SHIFT-+ (or any other key for that matter)? I got a soultion from Rich Lane and from Miller Graphics' SMART PROGRAMMER on the same day. You can totally disable the FCTN-QUIT function if you have any one of the following setups:

a) Extended Basic & Memory Expansion
b) Editor/Assembler & Memory Expansion
c) Mini-Memory with or without Memory Expansion

Before you begin a working session simply type CALL INIT, press ENTER, then type CALL LOAD(-31806,16), and press ENTER. Now FUNCTION-QUIT will no longer work. You still can get back to the title screen by typing BYE.

Item 4:
If you are familiar with some of the other microcomputers, or have not read the TI manual carefully you may not be using the full power of the INSERT key. In some other home computer, you have to press the INSERT key for each character that you want to insert. In the TI-99 once you have pressed the INSERT key, you stay in the "insert" mode until you press any other special editing key (such as DELETE, one of the arrow keys, etc.) and can insert multiple characters.

Mike Henry

A REVIEW OF ARCADE GAMES BY ATARISOFT

I was very excited when I saw the first advertisement for the Atarisoft arcade games which were going to be released by Atari for the TI. I had heard rumors that Atari had done wonders with the TI. Finally, I got to see some of the new releases.

I have played 4 of the 5 games which are presently out. I have seen Donkey Kong, Pac Man, Centipede and Defender. Also currently available locally is Dig Dug, although I haven't seen it yet.

Overall, I rate most of their games fairly well for the limitations that TI put in front of them. If you wonder what I mean about limitations, this refers to the licensing regulations which TI had on their GROM chips prior to them announcing the discontinuation of the TI-99/4A.

Some of the modules have more than the normal 8K of RAM inside. Donkey Kong for one is like this. This is a complicated type of memory mapping in which they actually bank select the 2 different groups of 8K RAM. So, in actuallity, the 2 blocks have the same memory locations, but they are "talked to" separately.

The play of the games are very good, and have an extremely fast response. Most play pretty much like that of the arcade versions, with small differences in graphics and music.

Donkey Kong plays very well and the only complaint that I have is the music. They must have been tight for space, so what they sacrificed was the sound processing. It's not quite as impressive as the arcade version, but it can be lived with. The only other difference is that there are no "pogo sticks" on the elevator screen. I am not complaining though, because this makes that screen a little easier.

Pac Man has very impressive graphics and music, but the initial speed of the game is a little slow. It does speed up later in the game though. But, it still is enjoyable to play if you are a Pac Man Buff. Plus, it doesn't "blink" like some of the Atari versions.

Centipede has a very nice response, if you can get used to not having a "Trac Ball" as the arcade version does. Graphics are acceptable and a nice job done for only being under 8K. Defender as well has good graphics and response, but again has minor differences with the arcade version. I find this an enjoyable yet frustrating version of the game.

WARNING!!! Donkey Kong (and possibly others) has not properly worked on some consoles. This doesn't seem to affect the game play, but does leave a few letters looking garbled. We have no way at present to determine whether it will work on your console or not.

Overall, I rate the new Atari games highly. I'm sure Atari could have done a little better if they had the cooperation of TI, but they still squeezed alot into a little space. Although the list price of these games is very high, they can be found in some local stores for around $21.95. Have fun with these games, and I look forward to some of the future releases. Six more games have been announced for the TI-99/4A.

Jon Daggett