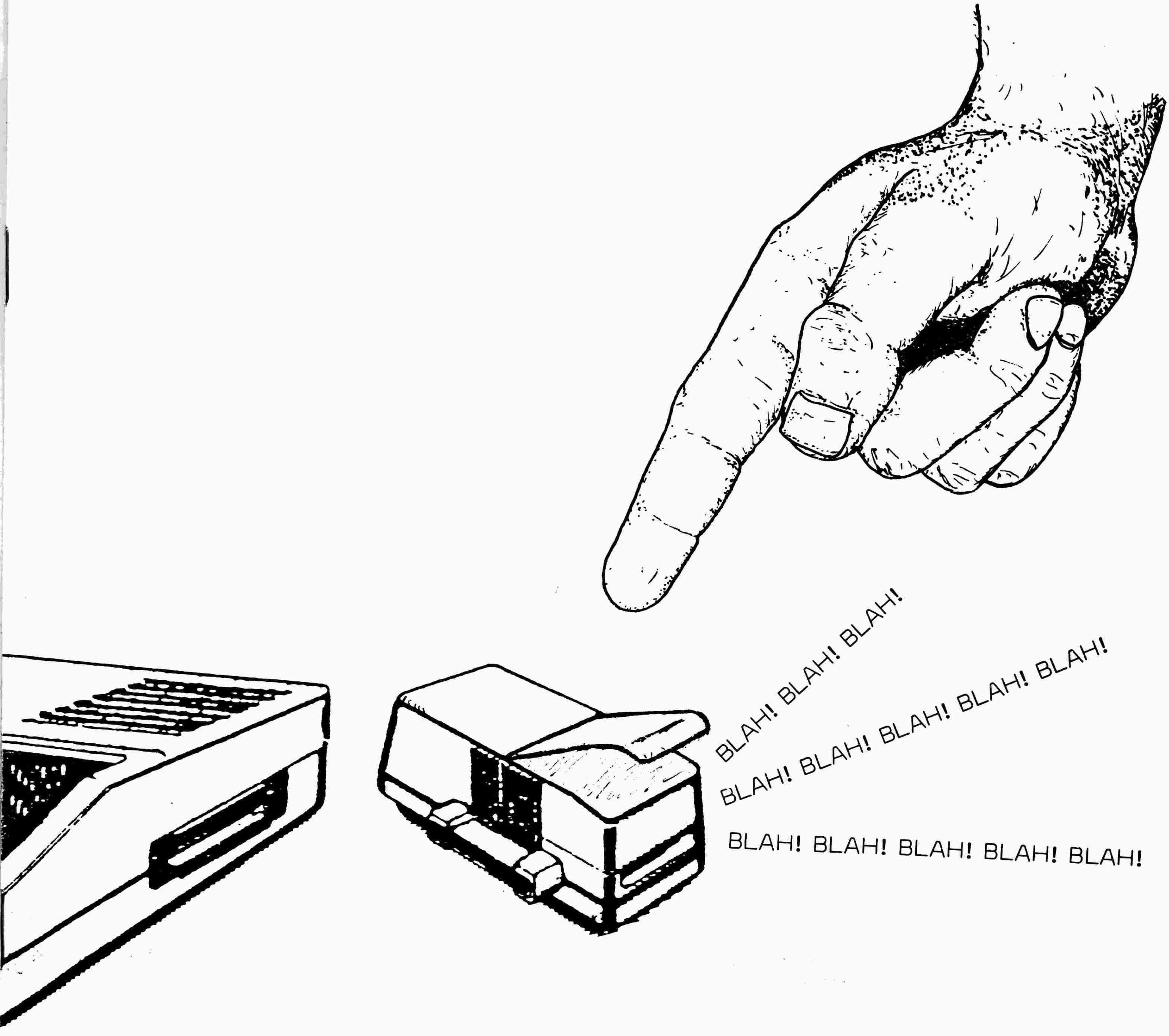
VOLUME 1
ISSUE 6

LOOK WHO'S TALKING



99/4A

INDEPENDENT BI-MONTHLY MAGAZINE FOR
THE USERS OF THE T.I. 99/4a HOME
COMPUTER

PARCO
Electrics

LETTERS LETTERS LETTERS LETTERS LETTERS LETTERS LETTERS

Dear Sirs,

I am writing to you about, among other things, your excellent magazine. One reader thinks it is a bit expensive, but I totally disagree with this view and think it is cheap at the price when you compare it with other magazines which cater to the TI99/4a. Other mags have on average one program and possibly one or two reviews, whereas 99/4a is totally devoted to items, programs, reviews etc about the Texas. By the way, I especially liked Beagle Hike and Jungle Fever. I also liked the section ´Sound Advice´.

Yours sincerely,

Scott Clelland (13)

(Well WE can´t help it if people keep sending in these nice letters)

Dear Sir/Madam,

- congratulations on the production of a value-for-money, informative magazine. Keep up the good work.

Yours faithfully,

Martin Pond

(It´s Sir, actually - hope that clears up any confusion...)

Dear Sir,

As a new subscriber to 99/4a, I have just received issues 3 and 4, and I must say that I am very pleased with them, they are very good value for my money.

I am a new starter to computing, and have only recently started using EB, and found your article on ´Expanding the TI´ very informative.

Yours sincerely,

Mr R.W.Potts

(of course we agree, sir)

Dear Sir,

I have got the first two issues of your magazine and have found them to be helpful and informative.

Yours faithfully,

P.W.Nicholson

(we wish to affirm that, contrary to popular belief, these letters are NOT written by fictitious characters)

Dear Sirs,

A rather belated thankyou for Issue 4 of the ´99/4a´ magazine. It gets better all the time, and I hope that you receive sufficient support for its continued production. As a tachnical writer in an entirely different field, I can appreciate the sweat that goes into producing it.

(did everyone get that? - Ed)

The program listings are excellent and no wonder that Spontaneous Reaction won a prize. Pity about the loss of the instructions with the original listing, but after three months they were a bit sperfluous!

Is it possible to give me any hints on how to negotiate screen 3 of Miner 2049er?

Yours faithfully,

Mr G.D.Denton

Dear Sir,

I am writing to thank you for my coy of 99/4a magazine. It is one of the pest I have read, and the program listings are excellent (Keep it up).

Yours faithfully

Mr K.F.Hughes

Dear Sir,

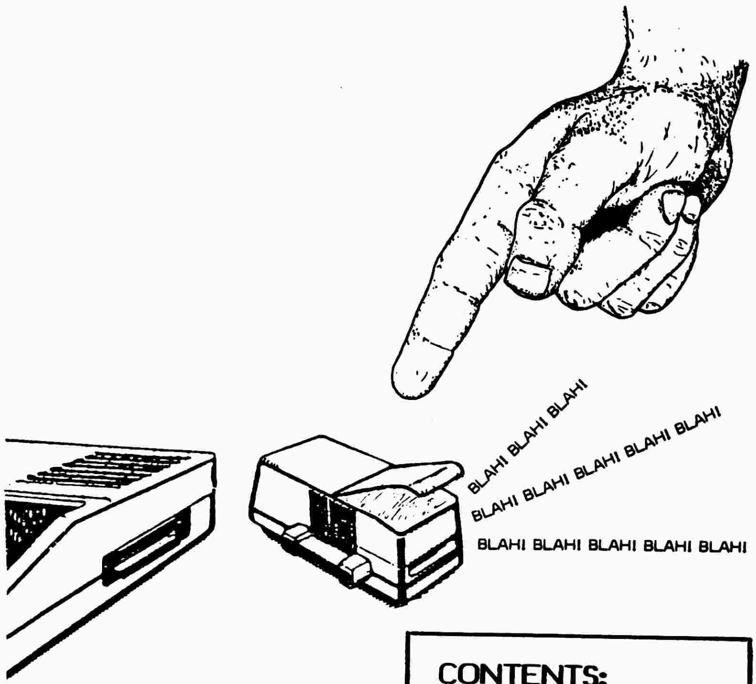
Please find enclosed a cheque for the fifth edition of your brilliant 99/4a magazine,

Yours faithfully,

S.King

(thankyou Auntie)

LOOK WHO'S TALKING



I find it simultaneously exciting and depressing when I reflect on some of the capabilities of the TI99/4a Home Computer that sits on my desk.

Exciting to think that it has graphic, sound, and many technical and programming features that were years ahead of its time, and that have given you and I many hours of fun, education and practical use.

Depressing to think that, as a computer, it has been underrated and overlooked, and that time has now passed, allowing other machines to steal the limelight and catch up in certain areas.

But hang on - there are still ways in which my TI is unique. There are aspects that I would be bound to miss if I had another computer. THERE ARE AREAS IN WHICH THE TI IS STILL TECHNOLOGICALLY AHEAD. One of these is in its SPEECH capability, and we will include a look at this in our feature 'Look Who's Talking'.

Can you believe we have completed Vol.1 of TI99/4a? Time has passed very quickly, and we can only thank YOU for the support you have shown, even when we let you down with delays and missed deadlines. What is pleasing is that we are still on an upward trend, even though others are struggling. It is only because we reckon that YOU still won't give up on the TI and 99/4a (and why should you?) that we don't hesitate to press on with Vol.2. Watch the PARCO mailshots for news, as we intend to make it worth your while subscribing again, and we also want to hear your views on '99/4a' on cassette.

In the first issue of Volume 2 we hope to review a brand new Hardware System from Germany, where plenty of exciting things are happening still for the TI. We'll say no more here, but just keep those tongues hanging out for more news soon!

OK - you're free to get stuck into the rest of the magazine now . . . .

-H-A-R-R-Y-> function <-P-R-I-D-M-O-R-E-

for PARCO ELECTRICS

CONTENTS:

2. Letters
3. Editorial
4. LOOK WHO's TALKING
9. Soap Opera
11.Keyboard Krunchers
19.Bright Sparks
22.T.I.Tbits
24.Bugbusters
26-28 Reviews
29.Book Review
30.Can't Do That
32.Scores

99/4A MAGAZINE
is produced by

PARCO ELECTRICS
4 DORSET PLACE
NEW STREET
HONITON
DEVON
EX14 8QS

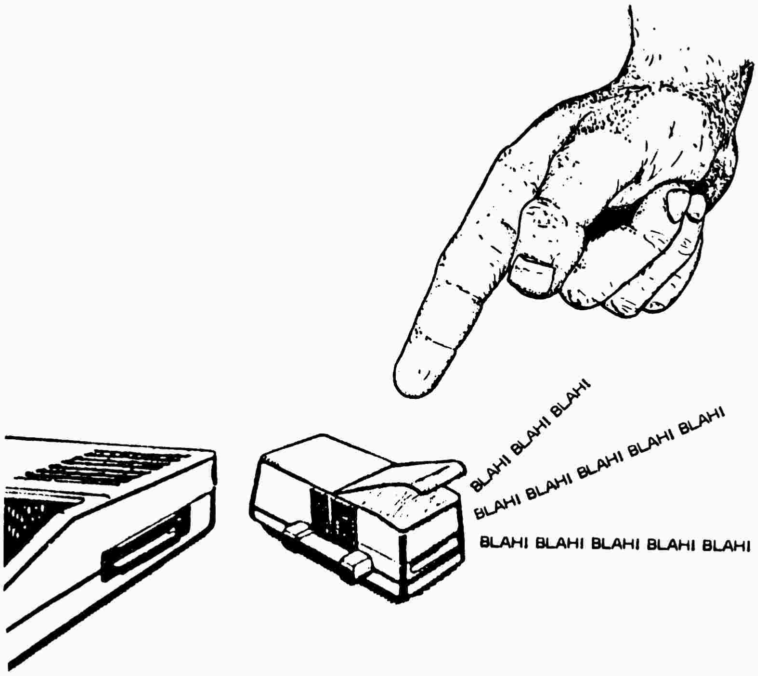
Telephone: (0404)44425

published by Parco Electrics as a Bi-Monthly Magazine for the users of the Texas Instruments TI99/4A Home Computer

price: £2.25 per issue, annual subscription £12.00

Articles and programs from readers are welcome, and the publishers reserve the right to edit submissions at their own discretion, without the permission of contributors, although through courtesy they would normally be advised.

LOOK WHO'S TALKING



To follow last issue's feature on graphics packages, we promised you an article on SOUND. There are a number of ways in which your computer excels through the sounds it is capable of producing. As well as the sound chip itself, (capable of emitting 3 tone and one noise channel), there is the speech capability of the TI99/4a. Even now, after the passing of several years, the TI is way ahead of the opposition in this area. What follows here is an outline of the potential of your 99/4a in sound, music and speech. It is not meant to be comprehensive, but hopefully I will answer some of the questions frequently posed, as well as providing some listings that illustrate the point.

We all like to hear the splat of an alien biting the dust, and I guess it is no surprise that SOUND EFFECTS probably constitute the major part of the sound chip's work. This is not a tutorial, so I won't stop to explore the whys and wherefores, but here is a little program to amuse you:-

```
100 FOR I=1 TO 2
110 FOR J=1 TO 6
120 CALL SOUND(1,1730,0)
130 NEXT J
140 FOR K=1 TO 3
150 NEXT K
160 FOR L=1 TO 50
170 NEXT L
180 NEXT I
190 FOR M=1 TO 400
200 NEXT M
210 GOTO 100
```

You will find that little program most effective in Extended Basic, and the following even more so:-

```
100 FOR I=1 TO 2
110 FOR J=1 TO 5
120 CALL SOUND(1,1270,0)
130 CALL SOUND(1,1100,0)
140 NEXT J
150 FOR L=1 TO 50
160 NEXT L
170 NEXT I
180 FOR M=1 TO 400
190 NEXT M
200 GOTO 100
```

Ring a bell? Well we'll leave sound effects there - we have concentrated on them before, and doubtless will again.

Music has been featured much in these pages before, so you will be familiar with TI capabilities and how to make use of them. In case you have the slightest doubt about your computer's musical ability, get a load of this. The following program will treat you to a rendition of 'TEXAS TRAFFIC', a piece written specially for the TI99/4a by Paul Templar, who recently contributed an excellent tutorial on music for the TI99/4a.

```
10 CALL CLEAR
20 CALL SCREEN(11)
30 RESTORE
40 A$="TEXAS TRAFFIC"
50 J=5
60 K=6
70 GOSUB 340
80 A$="A MUSICAL FANTASY"
90 J=7
100 K=6
110 GOSUB 340
120 A$="FOR THE TI-99/4A"
130 J=9
140 K=6
150 GOSUB 340
160 A$="COMPOSED BY"
170 J=11
180 K=6
190 GOSUB 340
200 A$="PAUL TEMPLAR"
210 J=13
220 K=6
230 GOSUB 340
240 CALL SCREEN(7)
250 A$="PLEASE WAIT"
260 J=19
270 K=6
```

```
280 GOSUB 340
290 A$="LIGHTS ON RED"
300 J=21
310 K=6
320 GOSUB 340
330 GOTO 380
340 FOR N=1 TO LEN(A$)
350 CALL HCHAR(J,K+N,ASC(SEG
$(A$,N,1)),1)
360 NEXT N
370 RETURN
380 REM
390 JJ=1
400 JAJ=1
410 DIM A(16),B(16),AB(16),C
(120),E(120),F(120),AA(8),BB
(8),BA(8),CC(8),EE(8),CE(8)
420 DIM ZA(60),AZ(60),ZZ(60)
,AP(12),PA(12),PQ(12)
430 D=-1000
440 FOR I=1 TO 16
450 READ A(I)
460 NEXT I
470 FOR I=1 TO 16
480 READ B(I)
490 NEXT I
500 FOR I=1 TO 16
510 READ AB(I)
520 NEXT I
530 FOR I=1 TO 120
540 READ C(I)
550 NEXT I
560 FOR I=1 TO 120
570 READ E(I)
580 NEXT I
590 FOR I=1 TO 120
600 READ F(I)
610 NEXT I
620 FOR I=1 TO 8
630 READ AA(I)
640 NEXT I
650 FOR I=1 TO 8
660 READ BB(I)
670 NEXT I
680 FOR I=1 TO 8
690 READ BA(I)
700 NEXT I
710 FOR I=1 TO 8
720 READ CC(I)
730 NEXT I
740 FOR I=1 TO 8
750 READ EE(I)
760 NEXT I
770 FOR I=1 TO 8
780 READ CE(I)
790 NEXT I
800 FOR I=1 TO 60
810 READ ZA(I)
820 NEXT I
830 FOR I=1 TO 60
840 READ AZ(I)
850 NEXT I
860 FOR I=1 TO 60
870 READ ZZ(I)
880 NEXT I
890 FOR I=1 TO 12
900 READ AP(I)
910 NEXT I
920 FOR I=1 TO 12
930 READ PA(I)
940 NEXT I
950 FOR I=1 TO 12
960 READ PQ(I)
970 NEXT I
980 CALL CLEAR
990 CALL SCREEN(13)
1000 A$="ALL CLEAR"
1010 J=12
1020 K=11
1030 GOSUB 340
1040 FOR I=1 TO 16
1050 CALL SOUND(D,A(I),0,B(I
),2,AB(I),2)
1060 FOR V=.01 TO .01
1070 NEXT V
1080 NEXT I
1090 DATA 698,880,880,523,88
0,880,880,880,1047,932,880,7
84,698,659,587,523
1100 DATA 175,220,220,131,22
0,220,220,220,659,932,880,78
4,698,659,587,523
1110 DATA 698,880,880,523,88
0,880,880,880,1047,932,880,7
84,698,659,587,523
1120 FOR MM=1 TO 2
1130 FOR Q=1 TO 2
1140 FOR I=1 TO 120
1150 CALL SOUND(D,C(I),0,E(I
),9,F(I),12)
1160 FOR V=.01 TO .01
1170 NEXT V
1180 NEXT I
1190 DATA 698,523,698,784,88
0,698,880,932,1047,880,1047,
1175,1047,932,880,932,784,93
2,932,587
1200 DATA 932,932,932,932,78
4,932,932,587,932,587,659,74
0,784,587,784,880,932,784,93
2,1047
1210 DATA 1175,1047,932,880,
784,698,659,784,698,698,698,
880,523,523,523,523,698,880,
880,523
1220 DATA 880,880,880,523,69
8,523,698,784,880,698,880,93
2,1047,880,1047,1175,1047,93
2,880,932
1230 DATA 784,932,932,587,93
2,932,932,932,784,932,932,58
7,932,932,932,1047,988,1047,
1319,1175
1240 DATA 1047,932,880,784,7
40,784,880,784,698,659,587,6
59,698,1397,1397,1397,1319,1
319,1319,1319
1250 DATA 1397,1397,1397,139
7,1319,1319,1319,1319,1175,1
175,1175,1175,1047,1047,1047
,1047,1175,1175,1175,1175
1260 DATA 587,587,587,587,78
4,932,932,784,932,587,659,74
0,784,784,784,784,740,740,74
0,740
1270 DATA 1397,1397,1397,139
7,1319,1319,1319,1319,1397,1
397,1397,1397,1397,1397,1397
,1397,698,880,880,262
1280 DATA 523,523,523,523,69
8,698,698,698,659,659,659,65
9,587,587,587,587,523,523,52
3,523
1290 DATA 587,587,587,587,29
4,294,294,294,370,466,466,29
4,466,294,277,262,330,392,34
9,415
1300 DATA 330,392,262,392,19
6,330,131,330,196,392,131,39
2,349,698,698,349,330,165,33
0,165
1310 DATA 175,349,131,349,17
5,349,131,349,175,349,131,34
9,185,370,185,370,196,392,14
7,370
1320 DATA 196,392,147,185,19
6,233,147,185,233,147,165,18
5,196,392,147,392,196,392,14
7,392
1330 DATA 196,392,147,392,19
6,392,131,392,175,349,131,34
9,175,349,131,349,175,220,22
0,131
1340 DATA 698,1047,1047,523,
175,349,131,349,175,349,131,
349,175,349,131,349,185,370,
185,370
1350 DATA 196,392,147,370,19
6,392,147,392,392,466,466,29
4,392,294,139,131,165,392,17
5,415
1360 DATA 165,392,131,392,19
6,330,131,330,196,392,131,39
2,349,440,440,262,349,349,26
2,440
1370 IF JJ=2 THEN 1490
1380 FOR I=1 TO 8
1390 CALL SOUND(D,AA(I),0,BB
(I),9,BA(I),9)
1400 FOR V=.01 TO .01
1410 NEXT V
1420 NEXT I
1430 DATA 1397,1319,1245,104
7,1397,1319,1245,1047
1440 DATA 349,330,311,262,22
0,196,175,523
1450 DATA 698,659,622,523,44
0,392,349,262
1460 JJ=JJ+1
1470 IF JAJ=2 THEN 1760
1480 NEXT Q
1490 FOR I=1 TO 8
```

```
1500 CALL SOUND(D,CC(I),0,EE
(I),9,CE(I),9)
1510 FOR V=.01 TO .01
1520 NEXT V
1530 NEXT I
1540 DATA 698,659,587,659,69
8,523,440,349
1550 DATA 175,349,175,349,17
5,175,349,349
1560 DATA 349,330,294,330,34
9,262,220,175
1570 FOR QQ=1 TO 2
1580 FOR I=1 TO 60
1590 CALL SOUND(D,ZA(I),0,AZ
(I),14,ZZ(I),4)
1600 FOR V=.01 TO .01
1610 NEXT V
1620 NEXT I
1630 DATA 932,1175,1175,698,
175,1175,698,1175,1047,1175,
1245,1047,932,880,784,831,88
0,1047,1047,698
1640 DATA 1047,1047,523,1047
,988,1047,1175,1047,932,880,
784,880,932,1175,1175,698,11
75,1175,698,1175
1650 DATA 1047,1175,1245,104
7,932,880,784,880,932,698,93
2,1047,1175,1047,932,880,932
,698,587,698
1660 DATA 698,698,698,698,13
97,1397,1397,1397,1245,1245,
1245,1245,622,622,622,622,44
0,440,440,440
1670 DATA 880,880,880,880,46
6,587,587,349,587,587,587,58
7,622,622,622,622,1245,1245,
1245,1245
1680 DATA 698,698,698,698,13
97,1397,1397,1397,698,698,69
8,698,698,698,698,698,349,34
9,349,349
1690 DATA 233,233,349,349,23
3,175,196,220,262,262,349,34
9,262,262,349,349,262,262,34
9,349
1700 DATA 262,175,196,220,26
2,262,349,349,262,262,349,34
9,262,262,349,349,233,175,19
6,220
1710 DATA 262,262,349,349,26
2,262,349,349,233,233,349,34
9,262,262,349,349,233,233,34
9,349
1720 NEXT QQ
1730 JAJ=JAJ+1
1740 JJ=1
1750 NEXT MM
1760 FOR RR=1 TO 2
1770 FOR I=1 TO 12
1780 CALL SOUND(D,AP(I),0,PA
(I),9,PQ(I),9)
1790 FOR V=.01 TO .01
1800 NEXT V
1810 NEXT I
1820 DATA 932,1175,1175,698,
1175,698,784,880,466,466,466
,466
1830 DATA 466,587,587,349,58
7,349,392,440,370,370,370,37
0
1840 DATA 233,294,294,175,29
4,175,196,220,330,330,330,33
0
1850 NEXT RR
1860 CALL CLEAR
1870 A$="PLAY IT AGAIN Y/N"
1880 J=10
1890 K=6
1900 GOSUB 340
1910 CALL KEY(0,K,KK)
1920 IF K=89 THEN 10
1930 IF K<>78 THEN 1910 ELSE
 1940
1940 STOP
```

Before we leave music, you may find this routine interesting:-

```
100 CALL CLEAR
110 FOR A=1 TO 108 STEP 2
120 READ B,C
130 FOR D=1 TO B
140 CALL SOUND(-1000,C,0)
150 CALL SOUND(-1000,C*1.015
,0)
160 NEXT D
170 NEXT A
180 CALL SOUND(-1,10000,30)
190 DATA 1,587,1,523,2,494,2
,587,2,587,2,587,2,659,4,587
,1,587,1,523,2,494,2,587
200 DATA 2,784,2,880,10,988,
2,587,2,988,2,988,2,988,2,98
8,4,880,1,880,1,784,2,740,2,
784,2,880,2,988
210 DATA 10,880,2,494,2,587,
2,587,2,587,2,659,4,587,1,58
7,1,523,2,494,2,587
220 DATA 2,784,2,880,10,988,
2,587,2,1047,2,1047,2,1047,2
,1047,4,988
230 DATA 1,988,1,880,2,784,2
,587,2,988,2,880,8,784
```

You can experiment with the value by which the variable C is multiplied in line 150, (set here at 1.015).

Maybe you are one of the many people who lashed out and got a Speech Synthesizer after witnessing PARSEC for the first time at your mate's place. Impressed with the thought of writing your own 'talking programs', you were perhaps a little disappointed to find that it could only be achieved with the purchase of more equipment. But now you're skint of course! Well, now you've had a chance to save up again, but you want to know what the options are before you part with any more readies. So here goes:-

a) EXTENDED BASIC gives you, amongst many other features, immediate ability to access the synthesizer's built-in vocabulary. What this means is that the range of frequently-used words (373 in all) that have been programmed into the synth already can be 'spoken' by your computer through using the command CALL SAY.

e.g. CALL SAY("HELLO")

As I have said, this is NOT a tutorial, so I will not go deeper into the mechanics, suffice to say that with this option you are more or less restricted to that built-in vocabulary. It is possible to split words and add suffixes ('ed','ing',etc), but it gets a bit cumbersome and complex. If you already have EB, 32k, and disk-system, however, there is another option available to you:-

b) TEXT-TO-SPEECH is available from Parco, and comes on disk. Although it runs with Extended Basic, it contains Assembly Language routines that give true 'Text-to-Speech' facilities. That is, your vocabulary is not restricted. Whatever you ask the computer to say, it will do so. In the main, words can be spelt as normal, and pronounciation will be correct. Should this not be the case, then it is simply a matter of altering spelling to suit. Hence you may have to substitute SAIRAH for SARAH and so on. Not so very complicated, eh? What the subroutines in fact do for you, once loaded IS rather complicated. Text-to-speech is "a type of speech technology in which words are broken down into parts of speech called allophones. An allophone is one two or more articulatory and accoustically different forms of the same phoneme. A phoneme is the smallest unit of speech that distinguishes one utterance from another." Told you it was complicated, didn't I? But the hard work is done by the software. XLAT translates the text string of your choice into allophones, and the SPEAK subroutine 'speaks' the text out through the synthesizer. In addition you can use various symbols to alter the pitch and slope of the voice. Thus you have the ability to emphasize certain syllables, give inflection to a phrase (as in asking a question), and changing the overall pitch of the voice lower or higher. The instructions with this package are very full and informative, and the beauty of it is that sophisticated speech can be incorporated easily into EXTENDED BASIC programs. Please note that 32k expansion and disk system are required, as well as Extended Basic and Speech Synthesizer.

c) TERMINAL EMULATOR II is a command module with two linked, but individually useful applications. One is the world of communications that is possible through linking your computer via a modem to a telephone line. The other is the true 'Text-to-speech' capabilities that concern us here. As with the TEXT-TO-SPEECH package discussed above, the conventions here are the same. The advantage is that no memory expansion or disk system is required, just the TE2 module itself, plus synthesizer of course. It is hard to describe the effect of using TE2 for the first time. It is so powerful, yet so easy. Actually, it is even easier to use than the 'TEXT-TO-SPEECH' above, since the subroutines are built into the module, and once you have opened a file after the fashion:

```
10 OPEN #1:"SPEECH",OUTPUT
```

All you need to do then is PRINT to that file, e.g.:

```
20 PRINT #1:"I AM MAGIC"
```

With virtually no delay, the text will be spoken. As with TEXT-TO-SPEECH, symbols and commands can be used to affect the voice quite easily. The instructions with the module are not as comprehensive as with TEXT-TO-SPEECH, but we will run a tutorial on the subject if enough of you write in. In the meantime, have a look at the following program. It starts up with an impression of Norman Colyer (he of the faulty microphone), then announces some of the latecomers to the TI Users Convention.

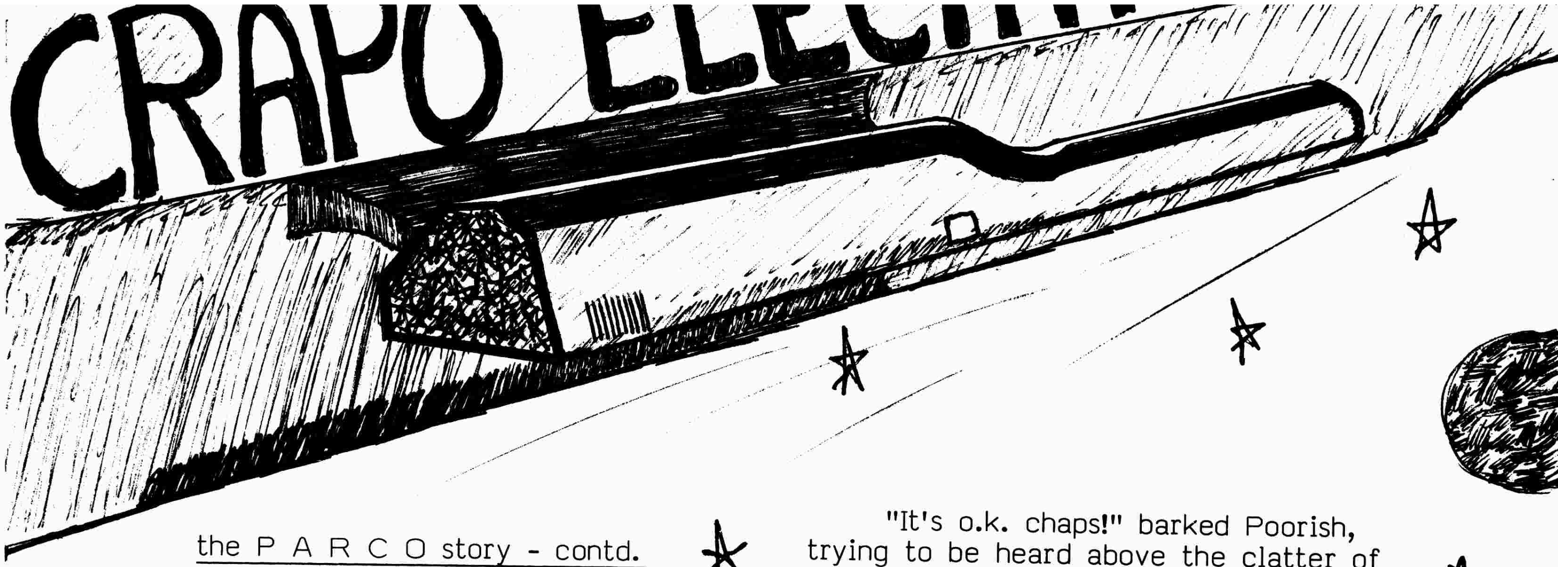
```
100 DIM WORD$(16)
110 DIM PRINT$(16)
120 CALL CLEAR
130 CALL SCREEN(2)
140 FOR I=9 TO 16
150 CALL COLOR(I,8,1)
160 NEXT I
170 FOR I=3 TO 8
180 CALL COLOR(I,7,1)
190 NEXT I
200 CALL COLOR(2,7,1)
```

```
210 N=16
220 OPEN #1:"SPEECH",OUTPUT
230 PRINT  1:"//36 120"
240 FOR I=30 TO 0 STEP -1
250 CALL SOUND(-500,4555,I)
260 NEXT I
270 PRINT #1:"_HELL O ,. TES
TING TESTING "
280 GOSUB 1470
290 PRINT #1:"_1 2 . KK KK _
1 2 . _1 2"
300 FOR I=0 TO 15
310 CALL SOUND(-1000,6039,I)
320 NEXT I
330 GOSUB 1470
340 GOSUB 1470
350 PRINT #1:"PF .PF _1 2 PF
360 FOR I=30 TO 10 STEP -1
370 CALL SOUND(-500,5200,I)
380 NEXT I
390 PRINT #1:"KK .KK "
400 GOSUB 1470
410 PRINT #1:"_IS THIS MY"
420 PRINT #1:""
430 GOSUB 1470
440 PRINT #1:""
450 PRINT #1:"FONE WORKING"
460 PRINT #1:"IS TH "
470 PRINT #1:""
480 PRINT #1:",RUFFOAN SWITC
HED ON"
490 PRINT #1:""
500 GOSUB 1470
510 GOSUB 1470
520 GOSUB 1470
530 PRINT #1:""
540 PRINT #1:"TESTING "
550 PRINT #1:"KK"
560 CALL SOUND(-80,-4,0)
570 PRINT #1:"TING . TESTING
"
580 PRINT # 1:"CAN YOU"
590 PRINT #1:""
600 PRINT #1:""
610 GOSUB 1470
620 GOSUB 1470
630 PRINT #1:"ME O K?"
640 CALL SOUND(200,3000,0)
650 FOR I=30 TO 0 STEP -1
660 CALL SOUND(-100,2999,I)
670 NEXT I
680 GOSUB 1470
690 FOR I=0 TO 30 STEP 6
700 CALL SOUND(-100,2999,I)
710 NEXT I
720 PRINT #1:"_1 . 2"
730 GOSUB 1470
740 GOSUB 1470
750 GOSUB 1470
760 GOSUB 1470
770 FOR I=20 TO 0 STEP -1
780 CALL SOUND(-500,4987,I)
790 NEXT I
800 PRINT #1:"PF. PF. KK KK"
```

```
810 GOSUB 1470
820 GOSUB 1470
830 FOR D=1 TO 20
840 CALL SCREEN(9)
850 NEXT D
860 GOSUB 1470
870 CALL SCREEN(9)
880 PRINT #1:"THANK YOU .LAD
IES AND GENTLE MEN"
890 FOR I=1 TO N
900 READ WORD$(I)
910 NEXT I
920 FOR I=1 TO N
930 READ PRINT$(I)
940 NEXT I
950 FOR I=1 TO N
960 PRINT "-----------------
-----------":
970 PRINT PRINT$(I):
980 PRINT "-----------------
-----------"::::::::::
990 CALL VCHAR(1,2,46,24)
1000 CALL HCHAR(1,2,46,30)
1010 CALL VCHAR(1,31,46,24)
1020 CALL HCHAR(24,2,46,30)
1030 FOR C=3 TO 8
1040 CALL COLOR(C,16,1)
1050 NEXT C
1060 PRINT #1:WORD$(I)
1070 FOR D=1 TO 14
1080 CALL COLOR(2,D,1)
1090 NEXT D
1100 CALL COLOR(2,7,1)
1110 FOR C=3 TO 8
1120 CALL COLOR(C,7,1)
1130 NEXT C
1140 NEXT I
1150 DATA "ANNOWNCING LATE C
UMMERS","TO THE_  T.I. USERS
 CONVENTION","MISTER AND MIS
SIS FURRALL BOX"
1160 DATA "AND THEIR SON ._
PERREE FURRALL BOX","MISTER
AND MISSIS STIX","AND THEIR
DOORTER._ JOI . STIX"
1170 DATA "MISTER AND MISSIS
 RAMMIXPANSION","AND THERE F
REND WHO IS THIRTY 2 , . KAY
"
1180 DATA "MISSIS ROW PRO SE
SSER","AND HER HUZBAND MIKE
, . ROW PRO SESSER"
1190 DATA "ROD HULL ","AND H
IS EE MEW LATOR, TOO ","THE
MEMRY FAMILY","AND THERE NEW
 BABY . MINNEE . MEMRY"
1200 DATA "THE, ETRICORDER F
AMILY","AND THEIR MAMA . CAS
S. ETRICORDER"
1210 DATA "   ANNOUNCING LAT
ECOMERS    ","TO THE  TI  US
ERS CONVENTION"
1220 DATA "mr and mrs FURRAL
BOCKS      ","and their son
                PERRY FURRALBOC
```

```
KS"
1230 DATA "mr and mrs STICKS
            ","and their daugh
ter            JOY STICKS"
1240 DATA "mr and mrs rammic
k-spanshan   ","and their fr
iend who is 32  KAY"
1250 DATA "mrs ROWPROCESSER"
,"and her husband
  MIKE ROWPROCESSER"
1260 DATA "rod hull","and hi
s EMU LAITA TOO"
1270 DATA "the MEMRY family
","and their new baby
   MINI MEMRY"
1280 DATA "the etricauda fam
ily       ","and their mama
                CASS ETRICAUDA"
1290 CALL CLEAR
1300 CALL SCREEN(2)
1310 PRINT #1:"//25 80"
1320 PRINT #1:"_HAY . YOU"
1330 PRINT #1:"//50 150"
1340 PRINT #1:", YEP?"
1350 PRINT #1:"//25 80"
1360 PRINT #1:"  WOTS ALL TH
IS ,FOUR?"
1370 PRINT #1:"//50 150"
1380 PRINT #1:"  ITS A DEMON
STRAISHUN OF _ TERMINAL EMUL
ATOR TOO , . YOU MOOR ON"
1390 PRINT #1:"//25 80"
1400 PRINT #1:"FAR . OUT ,MA
N , . I MUST RUSH TO PARCO A
ND BY ONE"
1410 PRINT #1:"//50 150"
1420 PRINT #1:"NOW THATS WOT
 I CALL A WISE MAN"
1430 PRINT #1:"//36 120"
1440 FOR D=1 TO 2000
1450 NEXT D
1460 STOP
1470 CALL SCREEN(9)
1480 CALL SCREEN(2)
1490 RETURN
```

Continued on page 31



the P A R C O story - contd.

It is probably worth reflecting at this point that Frankly's father, Big Ed, had always wanted his son to tackle something exciting and extravagant - like working down the Mace - but no, Frankly was always one of those unambitious types who are content just to conquer the universe and anything else that happens to be around at the time. And so it was, with as much inevitability as Jimmy Tarbuck laughing at his own jokes, that Frankly's ship CRAPO ELECTRICA had become the number one outfit in Tecasus Galaxy.

Frankly paused to look at himself in the mirror.

"Hell's Teeth!", he blurted.

Petty Gripe turned his tranny up, caught the remainder of that over, then rushed in to discover the reason for Poorish's outburst.

"What is it, Sir?", he panted, pretending to be out of breath and terribly concerned.

"I must be the Smartest Alec in history!" chuckled Poorish with a fat, satisfied grin.

"I couldn't agree more" smiled Gripe, wishing he could agree more. Gripe wanted Poorish to think that he was his biggest idol, but in fact that honour strictly speaking belonged to Hardly Didmore, who was totally idol.

At that precise lightyear something very strange happened. Whether it was the effects of a cataclysmic megastorm in some nearby field of cosmic turbulence, or Gripe opening his sandwich-box is hard to tell, but whatever it was sent the ship into violent tremors.

"It's o.k. chaps!" barked Poorish, trying to be heard above the clatter of equipment and personnel being hurled around the room. "We're going through a price war!"

Salad Rosebowl's expression was a mixture of relief and panic - sort of pralif.

"Just got the banking done in time," Salad struggled to keep her desk in order as the ship lurched again. "but how long will this war last? I've got orders to sort out, you know!" she complained, although no-one could actually make out what she was saying since she had the bankbook clenched firmly between her teeth; having amply occupied her arms already with folders, ring-binders, box-files, wadges of A4, a calculator, stapler, Access machine, paper-clips, Bulldog clips, scissors, another pair of scissors that everybody wondered what had happened to, a prawn wolemeal sandwich and 23 biros in assorted colours.

CRAPO ELECTRICA was going through what might justifiably be called a warp. Everyone clung to their seats just wishing it would soon be over. Then, as suddenly as it had begun, it just carried on. But then, a stroke of luck - a void appeared on the horizon. Frankly grappled with the controls in a desperate attempt to guide the ship into the void. Unfortunately the void was empty, and the ship plummeted straight through the void floor and into who knows where.

In a side street near The Restaurant at the End of the Universe sat the Chippy at the Back of the Cosmos. And that was precisely where our chums found themselves. As they sipped a peculiar brew not unlike the odd mixture in Gripe's flask, a cosmic combo serenaded them at the end of their table. The horrendous melodies were all too familiar - a selection from the Call Sound of Music. One of the creatures paused and held out a green drumstick-like limb.

"Hi, We're the Symbolics, "He grinned a smarmy alien grin. "I'm Sym, can I introduce my friend B...."

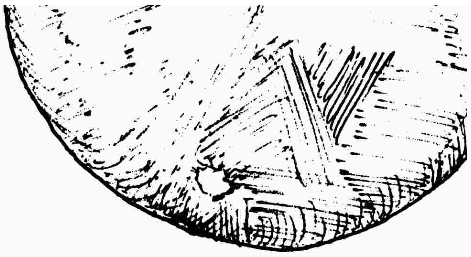
His words were lost as the withered frame of the other alien, who was slumped over an out-of-tune keyboard, ploughed carelessly into the next piece, apparently oblivious to anything going on around him, or for that matter what notes he was hitting. They had tortured our friends already with Do Re Mi and If Not Else, and were just destroying These are a Few of my Favourite Subroutines when Gripe could take no more.

"You carry on breathing garlic at the dozy one Salad, while I keep my gun on the creep!" rasped Gripe with calm authority. The party edged slowly back out of the room. It was a tense moment. Salad was thirty yards away from her target, and the effect was now becoming marginally reduced. Gripe wondered why the creep wasn't at least making a move. The creep wondered why this humanoid was waving a staple gun around.

They made a bolt for the door. On finding that it didn't fit, they gave it up as a bad job, and although not actually being pursued by anyone, the group took off like a bat out of Gripe's sandwich-box. Luckily they were able to grab a ride in an open-topped interstellar taxi.

"Phew! That was a close thing" gasped Poorish. "Never mind, we'll soon be back in the comfort and safety of our own ship."

It was at that moment, as Hardly woke up long enough to stretch, yawn, and peer skywards from his seat, that some idiot passing overhead must have decided to heave his pudding overboard. I conclude this, since just as Hardly opened his eyes the full force of a banana trifle met him smack in the laughing tackle. On noticing his plight, the rest of the crew were moved to come to his aid quickly with tissues, in between wetting their corporate selves.



Anyway, to cut an incredibly long story just about bearably long, they were soon safely back on board the starship assuming normal duties.

"Got a bloke on the phone called Parsnips or something, sir. He wants Thermal paper."

"Thermal paper?" chuffed Frankly as he polished his cheque-book. "The silly man must have one of those damn thermal printers from the 20th century. What's the idiot doing with one of those?"

"You sold it to him last Thursday, sir" replied Gripe.

"Okay, no problem Petty, leave it to me." he smiled, wondering what the hell he was going to do about it.

The silent night sky was split by the roar and flash of the Starship charging up its engines on its way toward the next mailshot.....

- WILL THEY GET THE THERMAL PAPER???

- WHAT WILL GRIPE HAVE NEXT IN HIS SANDWICH BOX???

- WILL SALAD GET THE BANKING DONE???

- WHO IS RICHARD TWYNING???

- HOW CAN I FINISH THIS STORY???

The answers to these and many of the other questions that have been baffling nobody will probably not be found in the next issue of 99/4a, but get your copy anyway.

KEYBOARD KRUNCHERS

We make no apologies for the fact that our programs are often long. Sure it leaves your fingers feeling like they´ve been run over by Cyril Smith on his bike, but we want to give you programs with some substance. We aren´t silly enough to equate quantity with quality (in fact the reverse is often true) but we do think you´d rather spend many minutes on a program you will come back to rather than a few on one that never gets loaded again.

This time round we have two TI Basic games (in case you thought we had forgotten you), and a game and utility in Extended Basic.

KEYBOARD KRUNCHERS
99/4a Magazine
Parco Electrics
2 Devonshire Court
Heathpark
Honiton
Devon

****
* ZOMBIES and POTHOLES ** David Martin *
****

TI Basic
Joystick optional

You´ve been trapped on Zombie island with some hungry Zombies. These Zombies are very clever, and home in on your position - but they´re also rather short-sighted, and the island is littered with bottomless pits. Using the arrow keys or a joystick, you must move around the island so that when the Zombies walk towards you they will fall down a hole. If you clear a screen of Zombies you must then get to the quay at the bottom right of the screen, and you will then go on to another, more difficult screen. If you manage to clear four screens you will be rescued. If you fall down a hole or get caught by a Zombie you will die. At the end of the game you will be able to type in your name if you have a new high-score, and you will be given the option to play again.

ZOMBIES & POTHOLES

```
10 REM ZOMBIE POT-HOLES
      BY DAVID MARTIN
20 CALL CLEAR
30 CALL SCREEN(2)
40 HI=0
50 HI$="?????????"
60 DIM MEM(24,32),RZ(32),CZ(
32)
70 RESTORE
75 REM GRAPHICS
80 FOR I=1 TO 10
90 READ C,M$
100 CALL CHAR(C,M$)
110 NEXT I
115 REM FILL ARRAYS
120 FOR C=1 TO 32
130 MEM(2,C)=1
140 MEM(23,C)=1
150 NEXT C
160 FOR R=1 TO 24
170 MEM(R,1)=1
180 MEM(R,32)=1
190 NEXT R
200 CALL CLEAR
210 DEAD=0
220 SK=1
230 SN=1
```

```
240 CALL SCREEN(2)
250 PRINT TAB(7);"SETTING UP
 NOW": : : : : : : : : : : :
 : : : : : : : : : : :
260 FOR S=1 TO 8
270 CALL COLOR(S,16,1)
280 NEXT S
290 FOR S=9 TO 14
300 CALL COLOR(S,1,1)
310 NEXT S
320 RESTORE
325 REM SET UP SCREEN
330 CALL HCHAR(3,1,128,640)
340 CALL VCHAR(1,1,32,24)
350 CALL VCHAR(1,32,32,24)
360 ZMBI=SK*2
370 FOR R=3 TO 22
380 FOR C=2 TO 31
390 MEM(R,C)=0
400 NEXT C
410 NEXT R
420 RANDOMIZE
430 FOR I=1 TO 10
440 C=INT(RND*30)+2
450 CALL HCHAR(3,C,32)
460 MEM(3,C)=1
470 C=INT(RND*28)+2
```

```
480 CALL HCHAR(22,C,32)
490 MEM(22,C)=1
500 NEXT I
510 FOR I=1 TO 12
520 R=INT(RND*20)+3
530 CALL HCHAR(R,2,32)
540 MEM(R,2)=1
550 R=INT(RND*19)+3
560 CALL HCHAR(R,31,32)
570 MEM(R,31)=1
580 NEXT I
590 FOR I=1 TO 15
600 R=INT(RND*20)+3
610 C=INT(RND*30)+2
620 CALL GCHAR(R,C,G)
630 IF G<>128 THEN 600
640 CALL HCHAR(R,C,136)
650 MEM(R,C)=5
660 NEXT I
670 CALL VCHAR(22,30,96,3)
680 MEM(22,30)=10
690 MEM(23,30)=10
700 CALL SCREEN(5)
710 RESTORE 2090
720 FOR S=9 TO 14
730 READ F,B
740 CALL COLOR(S,F,B)
750 NEXT S
760 CALL HCHAR(1,1,32,32)
770 M$="LEVEL:"&STR$(SN)&"
SCORE:"&STR$(DEAD)
780 CW=8
790 RW=24
800 GOSUB 2020
810 M$="HI SCORE:"&STR$(HI)&
" BY:"&HI$
820 RW=1
830 CW=INT((32-LEN(M$))/2)
840 GOSUB 2020
850 MAN=112
860 R=INT(RND*20)+3
870 C=INT(RND*30)+2
880 CALL GCHAR(R,C,G)
890 IF G<>128 THEN 860
900 CALL HCHAR(R,C,112)
910 FOR I=1 TO SK*2
920 RW=INT(RND*20)+3
930 CW=INT(RND*30)+2
940 CALL GCHAR(RW,CW,G)
950 IF G<>128 THEN 920
960 CALL HCHAR(RW,CW,121)
970 RZ(I)=RW
980 CZ(I)=CW
990 NEXT I
1000 RW=24
1010 CW=23
1020 CALL GCHAR(R,C,G)
1030 IF (G=121)+(G=136)THEN
1720
1035 REM MAIN LOOP
1040 CALL HCHAR(R,C,MAN)
1050 IF (MEM(R,C)=10)*(ZMBI=
0)THEN 1900
1060 ZOMB=INT(RND*ZMBI)+SGN(
ZMBI)
1070 IF ZMBI=0 THEN 1230
1080 RO1=RZ(ZOMB)
1090 CO1=CZ(ZOMB)
1100 RZ(ZOMB)=RZ(ZOMB)+SGN(R
-RZ(ZOMB))
1110 CZ(ZOMB)=CZ(ZOMB)+SGN(C
-CZ(ZOMB))
1120 CALL GCHAR(RZ(ZOMB),CZ(
ZOMB),X)
1130 IF (X<>121)*(X<>32)THEN
 1160
1140 RZ(ZOMB)=RZ(ZOMB)-SGN(R
-RZ(ZOMB))
1141 CZ(ZOMB)=CZ(ZOMB)-SGN(C
-CZ(ZOMB))
1160 RO2=RZ(ZOMB)
1170 CO2=CZ(ZOMB)
1180 CALL HCHAR(RO1,CO1,128)
1190 CALL HCHAR(RO2,CO2,121)
1200 CALL SOUND(-1,330,0)
1210 IF MEM(RO2,CO2)=5 THEN
1540
1220 IF (RO2=R)*(CO2=C)THEN
1720
1230 IF MEM(R,C)=5 THEN 1720
1240 CALL KEY(3,K,S)
1250 CALL JOYST(1,A,U)
1260 IF (K=69)+(U=4)THEN 130
0
1270 IF (K=88)+(U=-4)THEN 13
60
1280 IF (K=83)+(A=-4)THEN 14
20
1290 IF (K=68)+(A=4)THEN 148
0 ELSE 1050
1300 X=MEM(R-1,C)
1310 IF (X=1)+((X=10)*(ZMBI>
0))THEN 1050
1320 CALL HCHAR(R,C,128)
1330 R=R-1
1340 MAN=112
1350 GOTO 1020
1360 X=MEM(R+1,C)
1370 IF (X=1)+((X=10)*(ZMBI>
0))THEN 1050
1380 CALL HCHAR(R,C,128)
1390 R=R+1
1400 MAN=112
1410 GOTO 1020
1420 X=MEM(R,C-1)
1430 IF (X=1)+((X=10)*(ZMBI>
0))THEN 1050
1440 CALL HCHAR(R,C,128)
1450 C=C-1
1460 MAN=114
1470 GOTO 1020
1480 X=MEM(R,C+1)
1490 IF (X=1)+((X=10)*(ZMBI>
0))THEN 1050
1500 CALL HCHAR(R,C,128)
1510 MAN=113
1520 C=C+1
1530 GOTO 1020
1535 REM DEAD ZOMBIE
1540 M$="0000000000000000081B
D5A3C186642C3"
1550 CALL CHAR(125,"81BD5A3C
186642C3")
1560 CALL HCHAR(RO2,CO2,125)
1570 FOR I=17 TO 1 STEP -2
1580 CALL CHAR(125,SEG$(M$,I
,16))
1590 T=I*10+220
1600 CALL SOUND(-500,T,30,T,
30,T,30,-4,0)
1610 NEXT I
1620 FOR I=ZOMB TO SK*2-1
1630 RZ(I)=RZ(I+1)
1640 CZ(I)=CZ(I+1)
1650 NEXT I
1660 ZMBI=ZMBI-1
1670 DEAD=DEAD+1
1680 M$=STR$(DEAD)
1690 GOSUB 2020
1700 CALL HCHAR(RO2,CO2,136)
1710 GOTO 1020
1715 REM DEAD MAN
1720 FOR I=1 TO 5
1730 CALL CHAR(121,"81BD5A3C
186642C3")
1740 CALL SOUND(-100,-5,0)
1750 FOR D=1 TO 50
1760 NEXT D
1770 CALL CHAR(121,"3C5ABD99
186642C3")
1780 CALL SOUND(-100,-7,0)
1790 FOR D=1 TO 45
1800 NEXT D
1810 NEXT I
1820 CALL CLEAR
1830 PRINT "YOU ARE DEAD": :
 : : : :
1840 IF HI>=DEAD THEN 2400
1850 PRINT "BUT YOU GOT A NE
W HI SCORE!": :"PLEASE TYPE
IN YOUR NAME":"(NOT MORE THA
N 15 LETTERS)": :"THEN PRESS
 ENTER": :
1860 INPUT "":HI$
1870 HI$=SEG$(HI$,1,15)
1880 HI=DEAD
1890 GOTO 2400
1895 REM NEXT SCREEN
1900 SN=SN+1
1910 SK=SK*2
1920 IF SN=5 THEN 2100
1930 CALL CLEAR
1940 PRINT "    WELL DONE  .
 . .": : :"  NEXT SCREEN FOL
LOWS SOON": : :
1950 CALL SOUND(100,262,0,33
0,0,392,0)
1960 CALL SOUND(1000,262,0,3
30,0,392,0)
1970 FOR S=9 TO 14
1980 CALL COLOR(S,1,1)
1990 NEXT S
2000 PRINT : : : : : : : : :
 : : : : : : : : : : : :
2010 GOTO 330
2015 REM PRINT AT
2020 FOR I=1 TO LEN(M$)
2030 CALL HCHAR(RW,CW+I,ASC(
SEG$(M$,I,1)))
2040 NEXT I
2050 RETURN
2055 REM DATA FOR UDG'S
2060 DATA 120,FFFFFFFFFFFFFF
FF,128,FFFFFFFFFFFFFFFF,136,
70FCFF7FFFFE783C,121,3C5ABD9
9186642C3,112,183C18FF183C24
66
2070 DATA 113,3038727C306C44
66,114,0C1C4E3E0C362266
2080 DATA 96,7E427E427E427E4
2,104,0307073F79C97F3F,105,C
0E0E0FC9E9CFEFC
2090 DATA 15,5,12,5,13,11,9,
11,11,11,2,11
2100 REM THE END!
2110 FOR L=2 TO 28 STEP 2
2120 M$="hi"
2130 RW=23
2140 CW=L
2150 GOSUB 2020
2160 CALL SOUND(10,550+L,14-
(L/2))
2170 FOR D=1 TO 70
2180 NEXT D
2190 CALL HCHAR(23,1,32,29)
2200 NEXT L
2210 CALL VCHAR(22,30,96,3)
2220 FOR L=26 TO 2 STEP -2
2230 M$="hi"
2240 RW=23
2250 CW=L
2260 GOSUB 2020
2270 CALL SOUND(10,550+L,14-
(L/2))
2280 FOR D=1 TO 70
2290 NEXT D
2300 CALL HCHAR(23,1,32,28)
2310 NEXT L
2320 FOR D=1 TO 200
2330 NEXT D
2340 CALL CLEAR
2350 PRINT "CONGRATULATIONS
. . .": :"YOU ESCAPED FROM T
HE ZOMBIES": :
2360 PRINT "PLEASE TYPE IN Y
OUR NAME":"(NOT MORE THAN 15
 LETTERS)": :"THEN PRESS ENT
ER": :
2370 INPUT "":HI$
2380 HI$=SEG$(HI$,1,15)
2390 HI=30
2400 REM GO AGAIN
2410 PRINT :"WOULD YOU LIKE
ANOTHER TRY ?": :
2420 CALL KEY(3,K,S)
2430 IF K=89 THEN 200
2440 IF K=78 THEN 2450 ELSE
2420
2450 CALL CLEAR
```

```
2460 PRINT "BYE . . .": :"HO
PE YOU ENJOYED YOURSELF!": :
: : : : : : : :
2470 END
```

****
* SENTINAL 1 ********** Adrian Bunting *
****

TI Basic

Welcome to ´splat-em-up´ Sentinal One. Your mission is to defend yourself against the horde of aliens that are homing in on you from all sides. The game is not that complex or original, but is really well presented and playable.

Note that Adrian has used Control Characters to display the aliens in the instruction phase. When you have typed up to line 420, RUN the program to initialize the characters, then when you come to line 4070, the aliens can be incorporated into the PRINT statements by pressing CTRL and the respective key as follows:-

T - MERCATOR
U - LIMPETOID
V - THE THING
W - STOMPER
X - TEEGAN
Y - ZOID BEAST
Z - ENERGY BLOB

SENTINAL1

```
100 REM ******************
110 REM *  SENTINAL  ONE  *

120 REM *       BY        *
130 REM * ADRIAN  BUNTING *
140 REM *    TI. BASIC    *
150 REM *     19/7/84     *
160 REM *  UNDER  STRICT  *
170 REM *    COPYRIGHT    *
180 REM ******************
190 REM
200 REM * INITIALISATION *
210 REM
220 CALL CLEAR
230 CALL SCREEN(14)
240 GOSUB 4290
250 PRINT "ADRIAN BUNTING PR
ESENTS.....": : : : : : : :
: : : :
260 GOSUB 4340
270 FOR I=1 TO 48
280 READ CC,HEX$
290 CALL CHAR(CC,HEX$)
300 NEXT I
310 DATA 128,"102828444482FE
82",132,"82FE824444282810",1
34,"071A6282621A0700",130,"E
05846414658E000"
320 DATA 129,"030D32C2442418
08",135,"C0B04C4322241810",1
33,"101B2422434CB0C0",131,"0
8182444C2320D03"
330 DATA 136,"80402010080402
01",137,"0102040810204080",1
38,"000000FF",139,"101010101
0101010"
340 DATA 148,"3CDB7E3C18FF81
E7",140,"0124005200201A20",1
49,"3810BAEEBA103800",150,"1
83C5AFF24428142"
350 DATA 151,"245AFFC3FF1818
FF",152,"82C6EED67C381010",1
53,"244299E799814224",154,"3
C4299A5A599423C"
360 DATA 155,"81423C42423C42
81",96,"1F3F60C0C0C0C0FF",97
,"E0F0180C000000FC",98,"FF00
0000C0603F1F"
370 DATA 99,"FC0C0C0C1830E0C
0",100,"1F3F60C0C0C0C0FF",10
1,"E0F0180C000000FC",102,"FF
C0C0C0C0603F1F"
380 DATA 103,"FC0000000C18F0
E0",104,"40C0E0F0DBCCC6C3",1
05,"30180C0C0C0C0C0C",106,"C
1C0C0C0C0C06030"
390 DATA 107,"8CCC6C3C1C0C0C
08",108,"1F3F63C303030303",1
09,"E0F0180C00000000",110,"0
303030303030303"
400 DATA 111,"FC783030303030
30",112,"30303030303078FC",1
13,"1F3F60C0C0C0C0FF",114,"E
0F0180C0C0C0CFC"
410 DATA 115,"FFC0C0C0C0C060
30",116,"FC0C0C0C0C0C1830",1
17,"3060C0C0C0C0C0C0",118,"C
0C0C0C0C0603F1F"
420 DATA 119,"000000000C18F0
E0",120,"1C3C6CCC0C0C0C0C",1
21,"0C0C0C0C0C0C0C0C",127,"0
07E7E7E7E7E7E00"
430 SC=0
440 LIVES=3
450 CALL SCREEN(7)
460 CALL CLEAR
470 CALL COLOR(12,5,1)
480 CALL COLOR(13,9,1)
490 CALL COLOR(14,8,1)
500 CALL COLOR(15,4,1)
510 CALL COLOR(16,11,1)
520 GOSUB 4290
530 PRINT TAB(5);"INSTRUCTIO
NS (Y/N)?": : : : : : : : :
:
540 GOSUB 4470
550 GOSUB 4620
560 GOSUB 4340
570 GOSUB 4420
580 CALL KEY(0,K,S)
590 IF S=0 THEN 580
600 IF K=89 THEN 3820
610 GOSUB 4420
620 CALL CLEAR
630 CALL SCREEN(2)
640 CALL COLOR(12,14,1)
650 PRINT TAB(10);"SENTINAL
1":
660 GOSUB 4620
670 A=2
680 AA=6
690 B=2
700 BB=16
710 C=2
720 CC=26
730 D=12
740 DD=26
750 E=22
760 EE=26
770 F=22
780 FF=16
790 G=22
800 GG=6
810 H=12
820 HH=6
830 Q=16
840 Z=128
850 FOR I=1 TO 8
860 FL(I)=1
870 NEXT I
880 FOR I=1 TO 8
890 RANDOMIZE
900 J(I)=INT(RND*7)+148
910 NEXT I
920 CALL HCHAR(A,AA,J(1))
930 CALL HCHAR(B,BB,J(2))
940 CALL HCHAR(C,CC,J(3))
950 CALL HCHAR(D,DD,J(4))
960 CALL HCHAR(E,EE,J(5))
970 CALL HCHAR(F,FF,J(6))
980 CALL HCHAR(G,GG,J(7))
990 CALL HCHAR(H,HH,J(8))
1000 REM
1010 REM * MAIN LOOP
1020 REM
1030 CALL HCHAR(12,16,Z)
1040 GOTO 2630
1050 CALL KEY(0,K,S)
1060 IF S=0 THEN 2630
1070 IF K=83 THEN 1110
1080 IF K=68 THEN 1170
1090 IF K=76 THEN 1240
1100 REM *ROTATE LEFT*
1110 IF Z<129 THEN 1120 ELSE
 1130
1120 Z=136
1130 Z=Z-1
1140 CALL SOUND(-99,-5,15)
1150 GOTO 1030
1160 REM *ROTATE RIGHT*
1170 IF Z<135 THEN 1190
1180 Z=127
1190 Z=Z+1
1200 CALL SOUND(-99,-5,15)
1210 GOTO 1030
1220 REM * LAZER FIRE *
1230 REM *UPWARDS*
1240 IF Z<>128 THEN 1370
1250 CALL SOUND(50,300,0,200
,0)
1260 CALL VCHAR(2,16,139,10)
1270 IF FL(2)=1 THEN 1280 EL
SE 1320
1280 SC=SC+10
1290 CALL SOUND(50,-7,0)
1300 FL(2)=0
1310 GOTO 1330
1320 SC=SC-10
1330 CALL VCHAR(2,16,32,10)
1340 B=7
1350 GOTO 2630
1360 REM *DOWNWARDS*
1370 IF Z<>132 THEN 1500
1380 CALL SOUND(50,300,0,200
,0)
1390 CALL VCHAR(13,16,139,10
)
1400 IF FL(6)=1 THEN 1410 EL
SE 1450
1410 SC=SC+10
1420 CALL SOUND(50,-7,0)
```

```
1430 FL(6)=0
1440 GOTO 1460
1450 SC=SC-10
1460 CALL VCHAR(13,16,32,10)
1470 F=17
1480 GOTO 2630
1490 REM *RIGHT*
1500 IF Z<>130 THEN 1630
1510 CALL SOUND(50,300,0,200
,0)
1520 CALL HCHAR(12,17,138,11
)
1530 IF FL(4)=1 THEN 1540 EL
SE 1580
1540 SC=SC+10
1550 CALL SOUND(50,-7,0)
1560 FL(4)=0
1570 GOTO 1590
1580 SC=SC-10
1590 CALL HCHAR(12,17,32,11)
1600 DD=21
1610 GOTO 2630
1620 REM *LEFT*
1630 IF Z<>134 THEN 1760
1640 CALL SOUND(50,300,0,200
,0)
1650 CALL HCHAR(12,6,138,10)
1660 IF FL(8)=1 THEN 1670 EL
SE 1710
1670 SC=SC+10
1680 CALL SOUND(50,-7,0)
1690 FL(8)=0
1700 GOTO 1720
1710 SC=SC-10
1720 CALL HCHAR(12,6,32,10)
1730 HH=11
1740 GOTO 2630
1750 REM  *NORTH-EAST*
1760 IF Z<>129 THEN 1980
1770 CALL SOUND(50,300,0,200
,0)
1780 FOR I=11 TO 2 STEP -1
1790 Q=Q+1
1800 CALL HCHAR(I,Q,137)
1810 NEXT I
1820 IF FL(3)=1 THEN 1830 EL
SE 1870
1830 SC=SC+10
1840 CALL SOUND(50,-7,0)
1850 FL(3)=0
1860 GOTO 1880
1870 SC=SC-10
1880 Q=16
1890 FOR I=11 TO 2 STEP -1
1900 Q=Q+1
1910 CALL HCHAR(I,Q,32)
1920 NEXT I
1930 Q=16
1940 C=7
1950 CC=21
1960 GOTO 2630
1970 REM *SOUTH-EAST*
1980 IF Z<>131 THEN 2200
1990 CALL SOUND(50,300,0,200
,0)
2000 FOR I=13 TO 22
2010 Q=Q+1
2020 CALL HCHAR(I,Q,136)
2030 NEXT I
2040 IF FL(5)=1 THEN 2050 EL
SE 2090
2050 SC=SC+10
2060 CALL SOUND(50,-7,0)
2070 FL(5)=0
2080 GOTO 2100
2090 SC=SC-10
2100 Q=16
2110 FOR I=13 TO 22
2120 Q=Q+1
2130 CALL HCHAR(I,Q,32)
2140 NEXT I
2150 Q=16
2160 E=17
2170 EE=21
2180 GOTO 2630
2190 REM *SOUTH-WEST*
2200 IF Z<>133 THEN 2420
2210 CALL SOUND(50,300,0,200
,0)
2220 FOR I=13 TO 22
2230 Q=Q-1
2240 CALL HCHAR(I,Q,137)
2250 NEXT I
2260 IF FL(7)=1 THEN 2270 EL
SE 2310
2270 SC=SC+10
2280 CALL SOUND(50,-7,0)
2290 FL(7)=0
2300 GOTO 2320
2310 SC=SC-10
2320 Q=16
2330 FOR I=13 TO 22
2340 Q=Q-1
2350 CALL HCHAR(I,Q,32)
2360 NEXT I
2370 Q=16
2380 G=17
2390 GG=11
2400 GOTO 2630
2410 REM *NORTH-WEST*
2420 CALL SOUND(50,300,0,200
,0)
2430 FOR I=11 TO 2 STEP -1
2440 Q=Q-1
2450 CALL HCHAR(I,Q,136)
2460 NEXT I
2470 IF FL(1)=1 THEN 2480 EL
SE 2520
2480 SC=SC+10
2490 CALL SOUND(50,-7,0)
2500 FL(1)=0
2510 GOTO 2530
2520 SC=SC-10
2530 Q=16
2540 FOR I=11 TO 2 STEP -1
2550 Q=Q-1
2560 CALL HCHAR(I,Q,32)
2570 NEXT I
2580 Q=16
2590 A=7
2600 AA=11
2610 GOTO 2630
2620 REM *ALIEN MOVEMENT*
2630 RANDOMIZE
2640 ON INT(RND*8)+1 GOTO 26
50,2730,2800,2880,2950,3030,
3100,3180
2650 CALL HCHAR(A,AA,32)
2660 A=A+1
2670 AA=AA+1
2680 CALL HCHAR(A,AA,J(1))
2690 FL(1)=1
2700 IF A=12 THEN 3280
2710 CALL SOUND(5,500,0)
2720 GOTO 1050
2730 CALL HCHAR(B,BB,32)
2740 B=B+1
2750 CALL HCHAR(B,BB,J(2))
2760 FL(2)=1
2770 IF B=12 THEN 3280
2780 CALL SOUND(5,600,0)
2790 GOTO 1050
2800 CALL HCHAR(C,CC,32)
2810 C=C+1
2820 CC=CC-1
2830 CALL HCHAR(C,CC,J(3))
2840 FL(3)=1
2850 IF C=12 THEN 3280
2860 CALL SOUND(5,500,0)
2870 GOTO 1050
2880 CALL HCHAR(D,DD,32)
2890 DD=DD-1
2900 CALL HCHAR(D,DD,J(4))
2910 FL(4)=1
2920 IF DD=16 THEN 3280
2930 CALL SOUND(5,600,0)
2940 GOTO 1050
2950 CALL HCHAR(E,EE,32)
2960 E=E-1
2970 EE=EE-1
2980 CALL HCHAR(E,EE,J(5))
2990 FL(5)=1
3000 IF E=12 THEN 3280
3010 CALL SOUND(5,500,0)
3020 GOTO 1050
3030 CALL HCHAR(F,FF,32)
3040 F=F-1
3050 CALL HCHAR(F,FF,J(6))
3060 FL(6)=1
3070 IF F=12 THEN 3280
3080 CALL SOUND(5,600,0)
3090 GOTO 1050
3100 CALL HCHAR(G,GG,32)
3110 G=G-1
3120 GG=GG+1
3130 CALL HCHAR(G,GG,J(7))
3140 FL(7)=1
3150 IF G=12 THEN 3280
3160 CALL SOUND(5,500,0)
3170 GOTO 1050
3180 CALL HCHAR(H,HH,32)
3190 HH=HH+1
3200 CALL HCHAR(H,HH,J(8))
3210 FL(8)=1
3220 IF HH=16 THEN 3280
3230 CALL SOUND(5,600,0)
3240 GOTO 1050
3250 REM
3260 REM  * ENDING *
3270 REM
3280 LIVES=LIVES-1
3290 CALL HCHAR(12,16,140)
3300 FOR V=0 TO 30 STEP 5
3310 CALL SCREEN(16)
3320 CALL SOUND(-170+VOL,-7,
V)
3330 CALL SCREEN(2)
3340 NEXT V
3350 IF LIVES<1 THEN 3550
3360 M$="SHIPS "
3370 ROW=10
3380 COL=12
3390 GOSUB 3720
3400 M$=STR$(LIVES)
3410 ROW=10
3420 COL=19
3430 GOSUB 3720
3440 M$="SCORE "
3450 ROW=12
3460 COL=12
3470 GOSUB 3720
3480 M$=STR$(SC)
3490 ROW=12
3500 COL=18
3510 GOSUB 3720
3520 FOR D=1 TO 500
3530 NEXT D
3540 GOTO 620
3550 M$="SCORE "
3560 ROW=12
3570 COL=12
3580 GOSUB 3720
3590 M$=STR$(SC)
3600 ROW=12
3610 COL=18
3620 GOSUB 3720
3630 M$="ANOTHER GO (Y/N)?
3640 ROW=14
3650 COL=8
3660 GOSUB 3720
3670 CALL KEY(0,K,S)
3680 CALL SOUND(5,(RND*1000)
+110,25)
3690 IF S=0 THEN 3670
3700 IF K=89 THEN 430
3710 IF K<>78 THEN 3670 ELSE
 3760
3720 FOR I=1 TO LEN(M$)
3730 CALL HCHAR(ROW,COL+I,AS
C(SEG$(M$,I,1)))
3740 NEXT I
3750 RETURN
3760 CALL CLEAR
3770 PRINT "BYE !!": : : : :
3780 END
3790 REM
3800 REM * TITLES *
3810 REM
```

```
3820 CALL CLEAR
3830 GOSUB 4420
3840 GOSUB 4290
3850 PRINT TAB(4);"SENTINAL
1-RED ALERT": : : : : : : :
: : : :
3860 GOSUB 4340
3870 FOR I=1 TO 9
3880 CALL SOUND(-200,200,0)
3890 CALL SOUND(-200,180,0)
3900 FOR D=1 TO 50
3910 NEXT D
3920 NEXT I
3930 CALL CLEAR
3940 GOSUB 4290
3950 PRINT "  YOU ARE COMMAN
DER OF THE": :"BATTLESHIP 'S
ENTINAL 1' AND": :"IT IS YOU
R MISSION TO KILL ": :
3960 PRINT "ALL ALIENS IN YO
UR SECTOR ": :"OF THE GALAXY
.": :"  YOU HAVE A FLEET OF
THREE": :
3970 PRINT "SHIPS BUT WHEN I
N BATTLE;": :"YOU ONLY HAVE
ENOUGH ENERGY": :"TO FIRE YO
UR BLASTER AND ": :
3980 PRINT "ROTATE YOUR SHIP
.": : :
3990 PRINT TAB(8);"HIT ANY K
EY"
4000 GOSUB 4620
4010 GOSUB 4340
4020 CALL KEY(0,K,S)
4030 IF S<1 THEN 4020
4040 CALL CLEAR
4050 GOSUB 4420
4060 GOSUB 4290
4070 PRINT "ALIENS YOU MIGHT
 ENCOUNTER": : :"        - ME
RCATOR": :"        - LIMPETOI
D": :"        - THE THING": :
4080 PRINT "        - STOMPER
": :"        - TEEGAN": :"
    - ZOID BEAST": :"
- ENERGY BLOB": :
4090 PRINT "        - SPACE W
EED": : :"        HIT ANY KE
Y": :
4100 GOSUB 4620
4110 GOSUB 4340
4120 CALL KEY(0,K,S)
4130 IF S<1 THEN 4120
4140 GOSUB 4420
4150 CALL CLEAR
4160 GOSUB 4290
4170 PRINT TAB(10);"CONTROLS
": : :
4180 PRINT TAB(6);"S - ROTAT
E LEFT": :
4190 PRINT TAB(6);"D - ROTAT
E RIGHT": :
4200 PRINT TAB(6);"L - LIQUI
DATE": : : : : : : :
```

```
4210 PRINT TAB(8);"HIT ANY K
EY": : : :
4220 GOSUB 4620
4230 GOSUB 4340
4240 CALL KEY(0,K,S)
4250 IF S<1 THEN 4240
4260 GOSUB 4420
4270 GOTO 620
4280 REM * MAKE CHARS DIS *
4290 FOR I=1 TO 12
4300 CALL COLOR(I,1,1)
4310 NEXT I
4320 RETURN
4330 REM * MAKE CHARS APP *
4340 FOR I=1 TO 8
4350 CALL COLOR(I,16,1)
4360 NEXT I
4370 FOR I=9 TO 12
4380 CALL COLOR(I,2,1)
4390 NEXT I
4400 RETURN
4410 REM * BELL *
4420 FOR I=0 TO 30 STEP 5
4430 CALL SOUND(-99,698,I,19
24,I)
4440 NEXT I
4450 RETURN
4460 REM * TITLE *
4470 ROW=11
4480 FOR I=1 TO 15
4490 READ COL,HEX
4500 CALL HCHAR(ROW,COL,HEX)
4510 NEXT I
4520 IF ROW=11 THEN 4530 ELS
E 4550
4530 ROW=12
4540 GOTO 4480
4550 RESTORE 4570
4560 RETURN
4570 DATA 8,96,9,97,10,100,1
1,101,12,104,13,105,14,108
4580 DATA 15,109,16,111,17,1
04,18,105,19,113,20,114,21,1
17,24,120
4590 DATA 8,98,9,99,10,102,1
1,103,12,106,13,107,14,110
4600 DATA 16;112,17,106,18,1
07,19,115,20,116,21,118,22,1
19,24,121
4610 REM * BOARDER *
4620 CALL HCHAR(1,1,127,32)
4630 CALL HCHAR(24,1,127,32)
4640 CALL VCHAR(2,1,127,22)
4650 CALL VCHAR(2,32,127,22)
4660 RETURN
```

****

* GORILLA *************** Andrew Lord *

****

Extended Basic

The object of the game is to climb the platforms and rescue the girl before the gorilla gets to her.

You begin on the bottom platform where a lever will appear at a random position. You must then pull the lever which will cause a ladder to appear, again at a random position. Walk over to the ladder and you will climb automatically up to the next platform, where another lever will appear at a random position. You must do this on four levels; when you reach the top level you must hurry past the gorilla and reach the girl.

Bonus points are awarded for collecting the umbrella and handbag which are placed on platforms two and four.

The gorilla is constantly dropping rocks down at you. If you are hit by one of the rocks you lose one of your three lives. You will also lose a life if the gorilla reaches the girl before you do.

The next screen is harder, as the girl is placed nearer the gorilla, giving you less time to reach her. If you lose a life a gravestone will be displayed at the position you were killed. If you lose all your lives, your ´spirit´ will float away from your gravestone!

At the end of the game your score will be displayed, along with the level that you reached, and you will be asked if you wish to play again.

Controls:-

´S´ = LEFT
´D´ = RIGHT
´P´ = PAUSE

Scoring:-

Reaching a lever	25 points
Climbing ladder	50 points
Taking umbrella	100 points
Taking bag	100 points
Saving girl	500 points

GORILLA

```
100 REM ***************
110 REM ****GORILLA****
120 REM ** BY A.LORD **
130 REM *** 17/4/85 ***
140 REM ***************
150 CALL CLEAR :: CALL CHAR(
88,"FFFFFFFFFFFFFFFF"):: CAL
L CHAR(46,"000000000F0F0F0F"
):: CALL COLOR(8,7,8,2,7,8)
160 DISPLAY AT(1,2):"  XXX X
XX XXX X X   X   XXX"
170 DISPLAY AT(2,2):"  X X X
 X X X X X   X   X X"
180 DISPLAY AT(3,2):"  X.  X
 X XXX X X   X   XXX"
190 DISPLAY AT(4,2):"  X X X
 X XX  X X   X   X X"
200 DISPLAY AT(5,2):"  XXX X
XX X X X XXX XXX X X"
210 DISPLAY AT(13,1):"
 BY  ANDREW LORD"
220 CALL CHAR(100,"003048808
0838F9FFF7F7F7FDFEFE7E700003
07CF4FEF6FAFCF8D89C0C008080"
)
230 CALL MAGNIFY(4):: CALL S
PRITE(#1,100,7,120,117)
240 CALL SOUND(100,392,0)
250 CALL SOUND(100,392,0)
260 CALL SOUND(200,392,0)
270 CALL SOUND(200,392,0)
280 CALL SOUND(100,392,0)
290 CALL SOUND(100,440,0)
300 CALL SOUND(100,494,0)
310 CALL SOUND(100,523,0)
320 CALL SOUND(200,587,0)
330 CALL SOUND(200,587,0)
340 CALL SOUND(200,587,0)
350 CALL SOUND(500,494,0)
360 DISPLAY AT(23,1):"    PR
ESS ANY KEY TO BEGIN"
370 CALL KEY(0,K,S):: IF S=0
 THEN 370
380 CALL DELSPRITE(#1)
390 CALL CLEAR
400 LIVE=3 :: L=1
410 RANDOMIZE
420 CALL MAGNIFY(4)
430 CALL CLEAR
440 CALL SCREEN(2)
450 CALL COLOR(8,9,2)
460 CALL CHAR(90,"00000000FF
A5FF")
470 CALL SOUND(330,880,0)::
CALL HCHAR(6,1,90,32)
480 CALL SOUND(330,988,0)::
CALL HCHAR(10,1,90,32)
490 CALL SOUND(330,784,0)::
CALL HCHAR(14,1,90,32)
500 CALL SOUND(330,392,0)::
CALL HCHAR(18,1,90,32)
510 CALL SOUND(990,587,0)::
CALL HCHAR(22,1,90,32)
520 REM ** DEFINE CHARS**
530 CALL CHAR(80,"1824D4C4AA
5A6C3000000000000000000000000
0000000000000000000000000000"
)
540 CALL CHAR(120,"000000000
0020204040808l0107CFEFE")
550 CALL CHAR(122,"000000000
0000000000000000000000000000"
)
560 CALL CHAR(32,"0000000001
070F1F1F3F00000000201000000
00C0F0F8FCFCFE8080808080000")
570 CALL CHAR(128,"0F1C3D3C3
C3D3F3E3E3E3F3C3D3C3D3DE0707
878F878F8F8F8F8F8787878F8F8"
)
580 CALL CHAR(36,"0F1020203F
797F7F3F000000000000000008040
40C0E0E0E0C0")
590 CALL CHAR(122,"000000000
0000000000000000000000")
600 CALL CHAR(125,"000000000
0000000000000000000000")
610 CALL CHAR(116,"083F08080
83F0808083F0808083F080810FC1
01010FC101010FC101010FC1010"
)
620 CALL CHAR(68,"0107020603
030207")
630 CALL CHAR(69,"0707070303
070E04")
640 CALL CHAR(70,"E0E0F03070
E03018")
650 CALL CHAR(71,"881838F8F0
703870")
660 CALL CHAR(76,"0107020603
030207")
670 CALL CHAR(77,"0707070301
000001")
680 CALL CHAR(78,"E0E0F03070
E03018")
690 CALL CHAR(79,"881838F8F0
E0E0E0")
700 CALL CHAR(60,"004063333F
1F1F0F0F070F0F1F1F0406000000
80008090E08080C0C0E0E080C0")
710 CALL CHAR(104,"011311090
70303010307070707070280C88
890E0C0C080C0E0E0E0E0E040"
)
720 CALL CHAR(108,"07070F0C0
E070C1811181C1F0F07070780E04
060C0C040E0E0E0E0C080000080"
)
730 CALL CHAR(112,"07070F0C0
E070C1811181C1F0F0E1C0E80E04
060C0C040E0E0E0E0C0C0E07020"
)
740 V=214
750 REM ** SPRITES **
760 CALL SPRITE(#1,100,7,12,
10,0,1):: CALL DELSPRITE(#2)
770 T=141
780 CALL SPRITE(#10,104,16,1
3,V,0,0)
790 CALL SPRITE(#13,33,14,11
0,9)
800 CALL SPRITE(#14,37,12,58
,200)
810 CALL SPRITE(#3,108,5,141
,10,0,6)
820 CALL SPRITE(#6,120,4,T,(
INT(RND*244)+1))
830 CALL POSITION(#1,A,B)::
CALL SPRITE(#20,80,15,A+17,B
+20,10,0)
840 REM  **KEY INPUT**
850 CALL KEY(0,K,S):: IF S=0
 THEN CALL MOTION(#3,0,0)::
GOTO 910
860 IF K=ASC("S")THEN CALL P
ATTERN(#3,70):: CALL MOTION(
#3,0,-8):: CALL SOUND(10,110
,3):: CALL PATTERN(#3,77)
870 IF K=ASC("D")THEN CALL P
ATTERN(#3,112):: CALL MOTION
(#3,0,8):: CALL SOUND(10,110
,3):: CALL PATTERN(#3,108)
880 IF K=ASC("P")THEN 1500
890 CALL POSITION(#3,X,Y)::
IF Y<12 THEN CALL MOTION(#3,
0,8)
900 IF Y>242 THEN CALL MOTIO
N(#3,0,-8)
910 CALL COINC(ALL,C):: IF C
=-1 THEN 1000
920 SD=SD+1 :: IF SD=23 THEN
 SD=20
930 CALL POSITION(#20,N,M)::
 IF N>143 THEN GOTO 830
940 GOTO 840
950 CALL COLOR(#3,16)
960 CALL MOTION(#3,0,0,#1,0,
0,#20,0,0):: CALL SOUND(1000
,3000,0,3190,0):: CALL PATTE
RN(#3,128):: LIVE=LIVE-1 ::
IF LIVE=0 THEN 1350
970 FOR DEL=1 TO 1000 :: NEX
T DEL
980 GOTO 760
990 REM **COINCS**
1000 CALL COINC(#1,#10,20,C)
:: IF C=-1 THEN 950
1010 CALL COINC(#3,#20,16,C)
:: IF C=-1 THEN 1300
1020 CALL COINC(#3,#6,14,C):
: IF C=-1 THEN 1160
1030 CALL COINC(#3,#2,12,C):
: IF C=-1 THEN 1080
1040 CALL COINC(#3,#10,25,C)
:: IF C=-1 THEN 1200
1050 CALL COINC(#3,#13,15,C)
:: IF C=-1 THEN 1550
1060 CALL COINC(#3,#14,15,C)
:: IF C=-1 THEN 1590
1070 GOTO 840
1080 SCORE=SCORE+50 :: CALL
POSITION(#3,X,Y)
1090 CALL SOUND(1500,1500,0,
1600,0)
1100 CALL MOTION(#3,0,0)
1110 FOR I=1 TO 32
1120 X=X-1 :: IF X<10 THEN X
=13
1130 CALL LOCATE(#3,X,Y):: N
EXT I
1140 T=T-32
1150 GOTO 820
1160 CALL MOTION(#3,0,0):: S
CORE=SCORE+25 :: CALL PATTER
N(#6,124)
1170 IF T<45 THEN 850
1180 CALL SPRITE(#2,116,11,T
,(INT(RND*235)+1))
1190 CALL DELSPRITE(#6):: CA
LL SOUND(50,440,0,550,0):: G
OTO 840
1200 SCORE=SCORE+500 :: CALL
 MOTION(#1,0,0,#3,0,0,#20,0,
0)
1210 FOR I=0 TO 30 STEP 2
1220 CALL SOUND(-10,110,15,1
10,15,220,15)
1230 CALL SOUND(-100,110,I,1
10,I,220,I)
1240 CALL SOUND(-100,110,I,2
20,I,330,I)
1250 CALL SOUND(-100,220,I,2
20,I,110,I)
1260 CALL SOUND(-100,110,I,1
10,I,220,I)
1270 NEXT I
1280 V=V-9 :: L=L+1 :: IF L=
6 THEN CALL SOUND(200,1000,0
,3250,0,6750,0):: LIVE=LIVE+
1
1290 GOTO 760
1300 CALL MOTION(#20,0,0,#1,
0,0,#3,0,0)
1310 CALL DELSPRITE(#20,#6,#
2):: CALL PATTERN(#3,128)::
CALL COLOR(#3,16)
1320 LIVE=LIVE-1 :: IF LIVE=
0 THEN 1350 :: CALL SOUND(30
00,-8,0)
1330 REM ** END OF GAME **
1340 FOR DEL=1 TO 1100 :: NE
XT DEL :: GOTO 760
1350 CALL POSITION(#3,AB,CD)
:: IF AB<69 THEN CALL SPRITE
(#15,60,16,AB,CD,10,5)
1360 IF AB>69 THEN CALL SPRI
TE(#15,60,16,AB,CD,-10,5)
1370 CALL POSITION(#15,S,D):
: IF S<10 THEN CALL DELSPRIT
E(#15):: GOTO 1400
1380 IF S>180 THEN CALL DELS
PRITE(#15):: GOTO 1400
1390 GOTO 1370
1400 CALL CLEAR :: CALL DELS
PRITE(ALL)
```

```
1410 CALL SCREEN(8):: CALL C
HARSET :: DISPLAY AT(10,1):"
YOU HAVE BEEN TOTALLY KILLED
"
1420 DISPLAY AT(12,1):"YOU S
CORED A TOTAL OF";SCORE
1430 DISPLAY AT(14,12):"POIN
TS"
1440 DISPLAY AT(16,3):"YOU R
EACHED LEVEL : ";L
1450 DISPLAY AT(20,1):"DO YO
U WANT TO PLAY AGAIN?" :: DI
SPLAY AT(22,13):"(Y/N)"
1460 CALL KEY(0,K,S):: IF S=
0 THEN 1460
1470 IF K=ASC("Y")THEN CALL
CLEAR :: GOTO 400
1480 IF K=ASC("N")THEN CALL
CLEAR :: STOP
1490 GOTO 1460
1500 CALL MOTION(#1,0,0,#3,0
,0,#20,0,0)
```

```
1510 FOR DEL=1 TO 120 :: NEX
T DEL
1520 CALL KEY(0,K,S):: IF S=
0 THEN 1520
1530 CALL MOTION(#1,0,1,#20,
10,0)
1540 GOTO 840
1550 CALL MOTION(#3,0,0):: C
ALL DELSPRITE(#13)
1560 FOR S=0 TO 30 STEP 10 :
: CALL SOUND(200,400,S):: NE
XT S
1570 SCORE=SCORE+100
1580 GOTO 840
1590 CALL MOTION(#3,0,0):: C
ALL DELSPRITE(#14)
1600 FOR S=0 TO 30 STEP 10 :
: CALL SOUND(200,400,S):: NE
XT S
1610 SCORE=SCORE+100
1620 GOTO 840
```

****
* PAPER SAVER *********** M.Rajyaguru *
****

Extended Basic
32k RAM Exp
Disk System
RS232
Printer

This is what Mr Rajyaguru has to say about his useful utility:-

"The program was written for the purpose of listing programs and saving paper, as I am a serious programmer who finds it useful to stop the wasting of paper.

I came across the idea from my tutor at school who had a similar program for the 380Z in RML Basic.

The program is written for the Memory Expansion, but can be made to run without it, although this obviously prevents it from holding as much. With the expanded system it will hold about 295 lines. At present I am trying to change it so that it can handle any program size, this will be slower, but it will access more - it will also let you paginate, allowing you to have more page numbers.

Once the program has been typed in the following will be requested by the computer:-

´LOG DRIVE(1-3)?´

Here the drive number is entered.

´FILENAME?´-(10 chars max)

Here the name of the file to be listed is entered. After this you will have to wait until the program has been loaded.

´DEVICE NAME?´

The name of the output port is entered here.

´DATE?´-(11 chars max)

The date is entered here, if ENTER is pressed then no date will appear with the title.

Once all these have been entered, you may wish to go back to re-enter certain data. You can do this by pressing ´FCTN/E´, this will allow the cursor to go up one line.

If you wish to stop the printing, then by pressing ´FCTN/4´ you will have the option of closing the file or carrying on."

N.B. THE PROGRAM THAT YOU WANT TO LIST WITH ´PAPER SAVER´ MUST PREVIOUSLY HAVE BEEN ´LISTED´ TO DISK, e.g.:

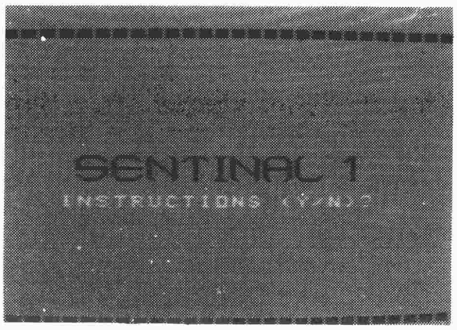
´LIST "DSK1.INVADERS"´

PAPER SAVER

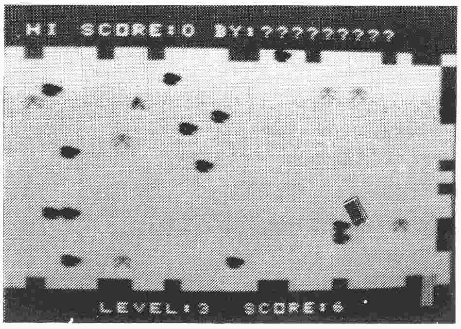
```
10 ON ERROR 900
20 ON BREAK NEXT
30 CALL KEY(3,K,S)
40 REM LISTER(SAVES PAPER)
50 REM
60 REM CREATED BY
        M. RAJYAGURU
70 REM
80 REM    VARIABLE LIST
90 REM
100 DIM LINE$(1000)
110 LET HLINEL=1
120 LET HLINER=0
130 LET DDRIVE$="1"
140 LET ARRAYLOC=-1
150 LET PRINT$=""
160 LET AUX$=""
170 REM
180 REM    DISPLAY
```

```
190 REM
200 CALL CLEAR
210 DISPLAY AT(1,10):"PAPER
SAVER"
220 DISPLAY AT(2,9):"______
      "
______
230 DISPLAY AT(4,1):"LOG DRI
VE(1-3)? "&DDRIVE$
240 ACCEPT AT(4,17)BEEP SIZE
(-1)VALIDATE("123"):DDRIVE$
250 IF DDRIVE$="" THEN LET D
DRIVE$="1" :: GOTO 230
260 DISPLAY AT(6,1):"FILENAM
E ? "
270 ACCEPT AT(6,12)BEEP SIZE
(-10)VALIDATE(UALPHA,DIGIT,"
,"):FILENAME$
280 REM LOAD DATA INTO
           MEMORY
```

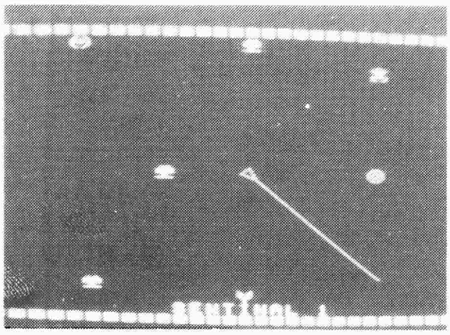
```
290 REM
300 OPEN #1:"DSK"&DDRIVE$&".
"&FILENAME$,INPUT
310 LET ARRAYLOC=ARRAYLOC+1
320 IF EOF(1)<>0 THEN 410
330 LINPUT #1:LINE$(ARRAYLOC
)
340 IF LINE$(ARRAYLOC)="" TH
EN 400
350 LET POSS=POS(LINE$(ARRAY
LOC)," ",1)
360 IF POSS>=6 OR POSS=0 OR
POSS=1 THEN 380
370 IF SEG$(LINE$(ARRAYLOC),
POSS-1,1)>="0" AND SEG$(LINE
$(ARRAYLOC),POSS-1,1)<="9" T
HEN 310
380 LET ARRAYLOC=ARRAYLOC-1
390 LET LINE$(ARRAYLOC)=LINE
$(ARRAYLOC)&LINE$(ARRAYLOC+1
)
400 GOTO 310
410 CLOSE #1
420 REM
430 REM PRINT LINE$ ON PARER
440 REM
450 DISPLAY AT(8,1):"DEVICE
NAME? "
460 ACCEPT AT(9,1)BEEP SIZE(
-28):DEVNAME$ :: CALL KEY(0,
K,S):: IF K=11 THEN ARRAYLOC
=-1 :: GOTO 270
470 DISPLAY AT(11,1):"DATE?"
480 ACCEPT AT(11,7)SIZE(11):
DATE$ :: CALL KEY(0,K,S):: I
F K=11 THEN 460
490 OPEN #1:DEVNAME$&".EC.LF
",VARIABLE 132
500 LET T=20-INT(LEN(FILENAM
E$&" "&DATE$)/2)
510 PRINT #1:CHR$(27);"-";CH
R$(1);CHR$(14);TAB(T);FILENA
ME$&" "&DATE$;CHR$(27);"-";C
HR$(0)
520 REM USE CONDENSED
        CHARECTER SET
530 PRINT #1:CHR$(15)
540 LET HLINER=INT(ARRAYLOC/
2)+1
550 LET LEN_LEFT=LEN(LINE$(H
LINEL))
560 LET LEN_RIGHT=LEN(LINE$(
HLINER))
570 CALL KEY(0,K,S)
580 IF S=0 THEN 600
590 IF K=2 THEN CALL ST_BREA
K
600 IF LEN(LINE$(HLINEL))<=6
5 AND LEN(LINE$(HLINER))<=65
 THEN 700
610 IF LEN(LINE$(HLINEL))>=6
5 AND LEN(LINE$(HLINER))>=65
 THEN 750
620 IF LEN(LINE$(HLINEL))<=6
5 AND LEN(LINE$(HLINER))>=65
 THEN 800
630 IF LEN(LINE$(HLINEL))>=6
5 AND LEN(LINE$(HLINER))<=65
 THEN 850
640 IF HLINEL>=INT(ARRAYLOC/
2)AND LINE$(HLINER)<>"" THEN
 680 ELSE IF HLINEL<=INT(ARR
AYLOC/2)THEN 550
650 PRINT #1:CHR$(18)
660 CLOSE #1
670 STOP
680 LET LINE$(HLINEL)=""
690 GOTO 550
700 REM IF THEY ARE BOTH UND
ER 70 CHARS.
710 PRINT #1:CHR$(10);TAB(0)
;LINE$(HLINEL);TAB(68);LINE$
(HLINER)
720 LET HLINEL=HLINEL+1
730 LET HLINER=HLINER+1
740 GOTO 640
750 REM IF BOTH BIGGER THEN
70 CHARS
760 PRINT #1:CHR$(10);TAB(0)
;SEG$(LINE$(HLINEL),1,65);TA
B(68);SEG$(LINE$(HLINER),1,6
5)
770 LET LINE$(HLINEL)="  "&S
EG$(LINE$(HLINEL),66,LEN_LEF
T-65)
780 LET LINE$(HLINER)="  "&S
EG$(LINE$(HLINER),66,LEN_RIG
HT-65)
790 GOTO 640
800 REM IF DIFFRENT SIZES
810 PRINT #1:CHR$(10);TAB(0)
;LINE$(HLINEL);TAB(68);SEG$(
LINE$(HLINER),1,65)
820 LET LINE$(HLINER)="  "&S
EG$(LINE$(HLINER),66,LEN_RIG
HT-65)
830 LET HLINEL=HLINEL+1
840 GOTO 640
850 REM X>60 Y<60
860 PRINT #1:CHR$(10);TAB(0)
;SEG$(LINE$(HLINEL),1,65);TA
B(68);LINE$(HLINER)
870 LET LINE$(HLINEL)="  "&S
EG$(LINE$(HLINEL),66,LEN_LEF
T-65)
880 LET HLINER=HLINER+1
890 GOTO 640
900 REM ERROR HANDLING
910 CALL ERR(ERR_C,ERR_T,ERR
_S,LN)
920 IF ERR_C<>130 THEN STOP
930 IF LN<>300 THEN 960
940 ON ERROR 900
950 PRINT LN :: RETURN 260
960 ON ERROR 900
970 CALL ST_BREAK
980 RETURN NEXT
990 SUB ST_BREAK
1000 DISPLAY AT(23,1):"
  LIST BREAK"
1010 DISPLAY AT(24,1):" S=RE
START   C=CLOSE & STOP"
1020 LET LOOP=LOOP+1
1030 CALL KEY(0,K,S)
1040 IF S=0 THEN 1090
1050 IF K=83 THEN 1210
1060 IF K<>67 THEN 1090
1070 PRINT #1:CHR$(18)
1080 CLOSE #1 :: STOP
1090 IF LOOP<=15 THEN 1020
1100 DISPLAY AT(23,1):"":""
1110 LET LOOP2=LOOP2+1
1120 CALL KEY(0,K,S)
1130 IF S=0 THEN 1180
1140 IF K=83 THEN 1210
1150 IF K<>67 THEN 1180
1160 PRINT #1:CHR$(18)
1170 CLOSE #1 :: STOP
1180 IF LOOP2<=5 THEN 1110
1190 LET LOOP,LOOP2=0
1200 GOTO 1000
1210 DISPLAY AT(23,1):"":""
1220 SUBEND
```



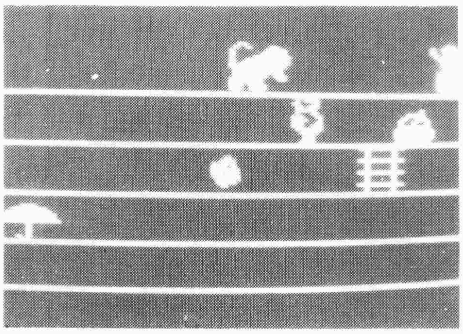
SENTINAL 1



ZOMBIES



SENTINAL 2



GORILLA

BRIGHT SPARKS

Well, we promised you cleverdicks some more Assembly meat to chew on, and thanks go to Graham Marshall for this article.

By the end of this tutorial you will be able to define, move, magnify and detect coincidences between sprites using the Minimemory and Line-by-Line Assembler.

Defining sprites in Assembly Language is completely different from doing so in Extended Basic. Instead of the single command ´CALL SPRITE´, many commands are required to load the information into a set of tables. These tables are only memory locations where the computer expects to find information about sprites. There is a table for the sprite attributes, one for patterns, and one for motion.

To define patterns, the appropriate values must be placed in the sprite pattern table. This table starts at address >0400, the character starting there is given the character number 128 (>80); since each character takes up 8 bytes these are the memory locations:-

>0400 to >0407	for character	>80 (128)
>0408 to >040F	" "	>81 (129)
>0410 to >0417	" "	>82 (130)
>0418 to >041F	" "	>83 (131)

The character codes are worked out in the same way as in the BASIC ´CALL CHAR´ statement. They are moved to the sprite pattern table in the VDP ram using a routine called VMBW (Video Multiple Byte Write). The following program segment will do this:-

```
  LI R0,>0400 - for VMBW routine, R0
                must contain the VDP
                ram address where we
                want to start writing
                (in this case >0400
                for character >80)

  LI R1,CH    - in the VMBW routine R1
                must contain the
                Label (like line no.)
                where the data to be
                written starts.

  LI R2,32    - R3 must contain the
                number of bytes we
                want written into the
                VDP ram (in this case
                32 bytes for 4
                characters)

  BLWP @>6028 - this command uses the
                VMBW routine to write
                the data into the VDP
                ram - the VMBW routine
                is located at address
                >6028

CH DATA >FFFF,>FFFF,>FFFF,>FFFF,
        >FFFF,>FFFF,>FFFF,>FFFF,
        >FFFF,>FFFF,>FFFF,>FFFF,
        >FFFF,>FFFF,>FFFF,>FFFF
```

To display the sprite on the screen, the location, pattern number, and colour must be entered into the sprite attribute table. This table starts at address >0300, each sprite taking up 4 bytes.

e.g. to display sprite #1 at point >7F,>5F (127,96) with character number >80 (128) and colour black (1), the following addresses must contain the following values:-

>0300 must contain	>5F	(y-location)
>0301	>7F	(x-location)
>0302	>80	(character no.)
>0303	>01	(colour: values in assembly are 1 less than in Basic)
>0304	>D0	(If no other sprites, D0 will cancel them. If other sprites are used it will contain the 1st value of sprite #2 attributes, its y-location.)

The next program segment will display sprite #1 at location (>7F,5F), coloured black, with a character number >80.

```
  LI R0,>0300    - R0 contains the
                   address in VDP ram
                   to be written to

  LI R1,AT       - R1 contains the
                   label where the data
                   to be written starts

  LI R2,5        - 5 bytes to be write

  BLWP @>6028    - this statement uses
                   the VMBW routine to
                   write the data into
                   the VDP ram, the
                   VMBW routine is
                   located at >6028

AT DATA >5F7F,>8001,>D000
                 - this statement has
                   the data to be
                   written in the order
                   it must be written
```

To magnify sprites there are two bits in a special register called a VDP write only register, and the two magnification bits are in VDP R1. There are 8 of these registers, which control various aspects of the screen display. To change the magnification, the last two bits of VDP R1 must be changed to:-

00	for unmagnified sprites/1 character
01	magnified sprites/1 character
10	unmagnified sprites/4 character
11	magnified sprites/4 character

The VDP register consists of 1 byte, to change the magnification only the last two bits of VDP R1 must be changed - if any other bits are changed then the program may crash. So to change it either >E0,>E1,>E2, or E3 must be entered into VDP R1. The default is >E0, giving 1 character unmagnified sprites. There is a routine to change the VDP registers, it is called VWTR (Video Write To Register) Address >83D4 must contain the value in VDP R1, since every time a key is pressed the value in this address is transferred to VDP R1 - therefore if it is not changed, the sprites will become 1 character and unmagnified if a key is pressed. The next program segment will make the sprites 4 characters and magnified.

```
LI R0,>01E3     - For VWTR routine
                  the left byte of R0
                  must contain the
                  number of the VDP
                  register to be
                  written into, the
                  right byte must
                  contain the value
                  to be written (>E3
                  gives enlarged
                  magnified sprites)

BLWP @>6034     - the VTWR is at
                  address >6034

LI R0,>E300

MOVB R0,@>83D4 - moves the left byte
                 of R0 into CPU ram
                 address >83D4
```

Sprite motion is slightly more difficult, since three things must be done. Firstly the values for the sprite velocity must be entered into the sprite motion table. This table starts at address >0780 - each sprite takes up four bytes in the motion table, one for y-velocity, one for x-velocity. Since the computer uses the last two they can be forgotten about.

Positive velocities range from:

>00 to >7F or 0 to 127

Negative velocities range from:

>FF to >80 or -1 to -128

e.g. to give sprite velocity of (6,-1) the sprite motion table must contain the following values:

```
>0780 must contain >FF (y-velocity)
>0781 must contain >06 (x-velocity)
>0782 and >0783 must be left alone
>0784 to >0787 used for #2, etc.
```

Secondly, the number of sprites moving must be put into location >837A so if sprites #1 and #2 are moving the address must contain 2, if only sprite #32 is moving the address must contain 32 (>20)

Lastly, to move the sprites you must enable interrupts to let the sprites move, and disable them to let the rest of the program run. It is not important to understand this, but only to know how to do it. This next program segment will set the sprite in motion:-

```
   LI R0,>0780    - The address in VDP
                    ram to be written
                    into

   LI R1,VE       - The label where the
                    value for velocity
                    is kept

   LI R2,2        - The number of bytes
                    to be written

   BLWP @>6028    - The address of VMBW


   LI R0,>0100    - Sprite #1 to be set
                    in motion

   MOVB R0,@>837A- loads no. of sprites
                    to move in >837A


LO LIMI 2         - enables interrupts

   LIMI 0         - disables interrupts

   JMP LO         - repeats by jumping
                    to LO


VE DATA >FF06     - the values for the
                    sprite velocity
                    (6,-1)
```

Here is the complete program for you to type in and run, you may experiment by changing the patterns, locations, magnification and velocities to get practice at using sprites. To run the program use EASYBUG and type ´E7D00´ to execute.

```
   LI R0,>0400   )
   LI R1,CH      )____  define
   LI R2,32      )     pattern
   BLWP @>6028   )

   LI R0,>0300   )
   LI R1,AT      )____   set
   LI R2,5       )     attributes
   BLWP @>6028   )

   LI R0,>01E3      )
   BLWP @>6034      )____   magnify and
   LI R0,>E300      )     enlarge sprite
   MOVB R0,@>83D4   )

   LI R0,>0780      )
   LI R1,VE         )
   LI R2,2          )
   BLWP @>6028      )
                    )____   set and
   LI R0,>0100      )     keep in motion
   MOVB R0,@>837A   )
                    )
LO LIMI 2           )
   LIMI 0           )
   JMP LO           )

CH DATA >FFFF,>FFFF,>FFFF,>FFFF,
        >FFFF,>FFFF,>FFFF,>FFFF,
        >FFFF,>FFFF,>FFFF,>FFFF,
        >FFFF,>FFFF,>FFFF,>FFFF

AT DATA >5F7F,>8001,>D000
VE DATA >FF06
```

All of the data statements are at the end of the program, since if the program were to run through them it would crash.

I have not enough space to explain in detail coincidence detecting, so here is a small routine which will jump (for example) to a line starting with a label ´CO´ if a coincidence between 2 or more sprites occurs. ´CO´ can be changed to a memory location to save memory:

```
   MOVB @>837B,R0
   ANDI R0,>2000
   CI R0,0
   JNE CO
```

Hopefully now you will be able to write your own assembly language using sprites.

T.I.TBITS

99/4a Magazine
Parco Electrics
2 Devonshire Court
Heathpark
Honiton
Devon

Here´s a few of contributions from readers who enjoy dabbling with their computers, and have come up with more TItbits for us all to share.

Dear Sir,

I hope you can find space in your mag for my ´one-liner´ in Extended Basic. Type in lines 100,200 (yes I know that´s two lines but we don´t count REMs do we?!), then LIST the program before RUNning it. I know this is only a ´one-liner´, but plenty of midnight oil was burnt writing it. As the seeds are RAND and not RANDOMIZE, the sequences are predictable, and I was able to watch and alter it so that it ran at its best, so when you have LISTed it and RUN it give it a couple of minutes to get in its best mood.

```
100 S=INT(RND)+1 :: CALL
SCREEN(S+3):: CALL COLOR(S,S
+3,S+3):: X=INT(RND3)+33
:: CALL HCHAR(S,S,X+3,X-7)::
 CALL VCHAR(S,S,X+S-8):: FOR
 D=1 TO 100+X :: NEXT D :: G
OTO 100
200 !  COLOR
         AND          eee: e:
       CONTRAST        e:: e:
         BY            e:  e:
     LEN  REEVES       e:  e:
```

I have also included here another near ´one-liner´ that puts a spider in the back of the monitor. Once again, let it run a few seconds.

```
90 !************************
            PET SPIDER
    ************************
100 CALL CLEAR :: PRINT "* R
EADY *" :: CALL CHAR(52,"005
4387C38440000",48,"001038383
8000000"):: DEF RND15=INT(RN
D-5):: CALL SPRITE(#1,52,
2,100,100)
110 QQ=RND :: LL=RND15 ::
 CALL MOTION(#1,QQ,LL):: FOR
 I=48 TO 52 STEP 4 :: CALL P
ATTERN(#1,I):: FOR D=1 TO 10
 :: NEXT D :: NEXT I :: GOTO
 110
```

Yours faithfully,

Len Reeves

ED´s note:

make sure they type these routines in accurately (you will need to re-enter some lines in edit mode to get the whole line in). The first routine is subtle - you may be tempted to think ´so what?´, but spare a minute - watch how colours are affected by being set against other colours. Good ´un, Len.

That Stephen Meadows bloke has been at it again - read what he has to say about Control codes:-

Dear Sir,

- reading the Texas manuals reveals that control keys are used to acess telecommunications devices. If so, what have I found? In Extended Basic, pressing the CTRL key and other keys, keywords (similar to Sin Klair´s machines, but not as convenient(?)) are available.
Using NUM or after a line number, pressing the appropriate key results in the cursor moving forward a space. Press ENTER, then take a look at the line. A keyword has appeared! Here is a list of those I have found (In QWERTY format).

1	TO
2	STEP
3	,
4	;
5	:
6	)
7	(
8	OPTION (type BASE 0/1 after)
9	OPEN

	0	THEN
	=	CALL
*	Q	UNTRACE
	W	READ
	E	GO
	R	INPUT
*	T	RESTORE
	Y	DELETE
*	U	RANDOMIZE
	I	DEF
*	O	UNBREAK
*	P	TRACE
	/	AND
	A	ELSE
	S	DATA (works with short strings of data after)
	D	IF
	F	GOTO
	G	GOSUB
*	H	RETURN
	J	DIM
	K	END
	L	FOR
	;	PRINT (will print data as nos. or in inverted commas, but not string, eg A$)
	Z	REM
*	X	STOP
	C	!
	V	NEXT
	B	::
*	N	BREAK
	M	LET
	,	02 "+
	.	ON (break,warning,error not line nos.)

* These words can be used direct, without extra editing, etc. Those with notes will work with some additions, but with limitations, eg. ON, which will take BREAK NEXT, ERROR, and WARNING, but will not accept ON X GOTO n,n,n! Other words can be entered, but not added onto until editing (I hope you understand all this!) when the extra parts can be added on. DATA (CTRL/S) can only take a few items directly, but when edited the full line can be used.

Also FCTN works!

Try FCTN 0 after line no. - XOR

FCTN /	- OR
FCTN Q	- ^
FCTN ;	- NOT

I´ll leave this subject in case I get confused myself - if you want to investigate further, I´ll leave it to you -

and more from meddling Meadows:

- while experimenting with my TI I found a useful method of erasing a program during its execution. It could be used in passwords and secret programs, or just to clear the memory without typing NEW. In Extended Basic it runs:

```
1000 ON ERROR 1003
1001 RUN ""
1003 CALL CLEAR
```

Someone might find it useful.

This produces a reasonably authentic series of chimes:

```
100 FOR T=1 TO 6
110 FOR I=0 TO 29
120 CALL SOUND(-1,444,I,110,
I,140,I,-1,I+1)
130 NEXT I
140 NEXT T
```

Here´s an alternative to the inaccurate CALL COINCidence statement. Try CALL PEEKing (-31877,X) in EB. If any two sprites have coincided, then 188 is returned in X, else 156 is given. This is the same as the ALL part of the COINC subprogram, but returns different values (normally 0 or -1).

Thankyou Stephen, we look forward to your next volume.

Richard Speed asks:

- ever got bored with the INPUT tone? Yes? Well simply precede the INPUT with a CALL SOUND(-dur,tone,etc). The minimum duration appears to be around -70, you could always have a silent INPUT by setting the volume to 30. Useful!

Lastly, here are a couple of enhancements to existing programs, as suggested by Richard Owen.

<u>MLIST/RPT</u> - TI Mailing List

```
1141 IF SEG$(DEV$,I,3)="PIO" THEN 1170
```

<u>SORCERORS CASTLE</u> - Apex Software

```
530 VC$="NSEWUDTI"
550 ON X+1 GOTO 1580,560,560,560,560,56
```

```
0,560,820,2670
860 FOR I=1 TO 6
890 PRINT "you can only carry six     o
bjects - leave one behind": :
1040 for I=1 to 3
1140 for I=1 TO RMN
1150 B(I,1)=0
1160 B(I,2)=0
1210 B(I,3)=0
1220 NEXT I
1221 OB(1)=0
1222 OB(2)=0
1223 OB(3)=0
1224 VL=0
1230 GOTO 320
1580 PRINT "use N S E W U D I or T plea
se": :
1640 DATA N (north) S (south) E (east)W
 (west)  U (up)    D (down)T (take)  I
(index)
2670 PRINT "1. ";A$(OB(1)):"2.  ";A$(OB
(2)):"3 ";A$(OB(3)):"4. ";A$(OB(4)):"5.
 ";A$(OB(5)):"6 ";A$(OB(6))
2680 GOTO 510
```

That just about wraps it up for this iss:ie, keep them coming!

BUGBUSTERS

Dear BUGBUSTERS,

There appears to be a problem with the DRAGON program on GAMESWRITER PACK 1. On running from the dragon, it printed "YOU DROP..." and BAD VALUE IN 6010.
Also if you hit the wall on right-hand side * BAD SUBSCRIPT IN 960.

Can you help?

D. Swingler

(We welcome comments, Ed)

Dear BUGBUSTERS,

GAMESWRITER PACK 2

1. DICERACE - Sometimes when playing this game it shows DATA ERROR and stops the program running. Is this a common fault or are we unlucky?

2. COMMANDO - After playing this game and losing your men, it shows STRING NUMBER MISMATCH IN 5050. Can you give me a correction to remove this?

per pro S.J.Davey

(We welcome readers' comments on DICERACE, as for COMMANDO, anyone struggling with this will find corrections in 'BUGBUSTERS' in issue 3 of 99/4a.)

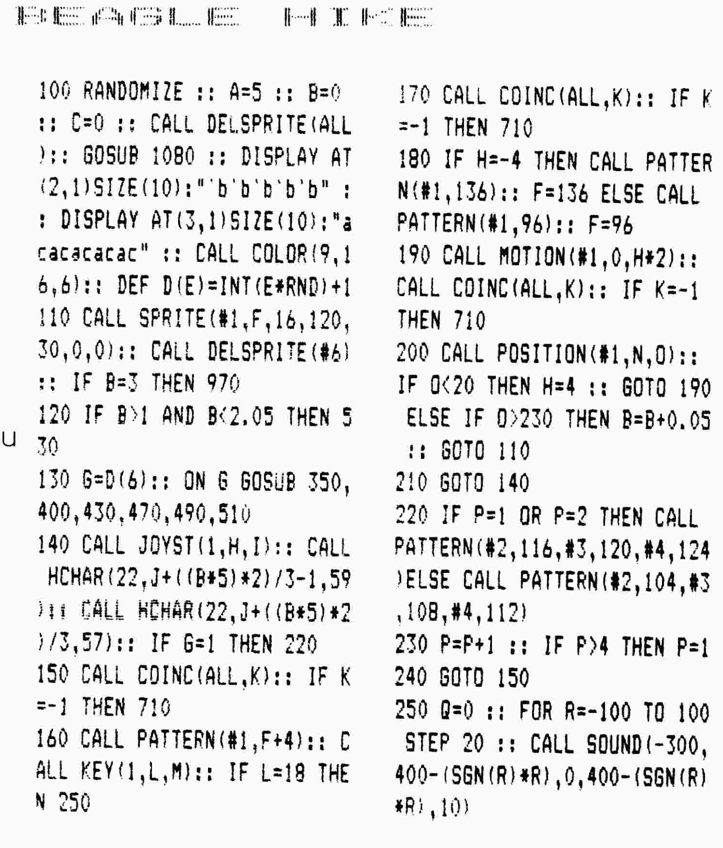
Dear BUGBUSTERS,

LUNAR RESCUE on page 15 - I get BAD VALUE on line 750, and DATA ERROR on line 910. Being very much a 'new girl' I do not know how to get over this so please HELP!! BEAGLE HIKE page 20 due to bad printing lines 100 to 380 are unreadable, so again can you help!!

(Keep up the good work, TI Users need you)

Mrs P.J.Bascombe

(We don't think any other reports of these LUNAR RESCUE bugs have been reported, so check and double check the whole listing Patricia; as for BEAGLE HIKE, here are the offending lines.)



BEAGLE HIKE

```
100 RANDOMIZE :: A=5 :: B=0
:: C=0 :: CALL DELSPRITE(ALL
):: GOSUB 1080 :: DISPLAY AT
(2,1)SIZE(10):"`b`b`b`b`b" :
: DISPLAY AT(3,1)SIZE(10):"a
cacacacac" :: CALL COLOR(9,1
6,6):: DEF D(E)=INT(E*RND)+1
110 CALL SPRITE(#1,F,16,120,
30,0,0):: CALL DELSPRITE(#6)
:: IF B=3 THEN 970
120 IF B>1 AND B<2.05 THEN 5
30
130 G=D(6):: ON G GOSUB 350,
400,430,470,490,510
140 CALL JOYST(1,H,I):: CALL
 HCHAR(22,J+((B*5)*2)/3-1,59
):: CALL HCHAR(22,J+((B*5)*2
)/3,57):: IF G=1 THEN 220
150 CALL COINC(ALL,K):: IF K
=-1 THEN 710
160 CALL PATTERN(#1,F+4):: C
ALL KEY(1,L,M):: IF L=18 THE
N 250
170 CALL COINC(ALL,K):: IF K
=-1 THEN 710
180 IF H=-4 THEN CALL PATTER
N(#1,136):: F=136 ELSE CALL
PATTERN(#1,96):: F=96
190 CALL MOTION(#1,0,H*2)::
CALL COINC(ALL,K):: IF K=-1
THEN 710
200 CALL POSITION(#1,N,O)::
IF O<20 THEN H=4 :: GOTO 190
 ELSE IF O>230 THEN B=B+0.05
 :: GOTO 110
210 GOTO 140
220 IF P=1 OR P=2 THEN CALL
PATTERN(#2,116,#3,120,#4,124
)ELSE CALL PATTERN(#2,104,#3
,108,#4,112)
230 P=P+1 :: IF P>4 THEN P=1
240 GOTO 150
250 Q=0 :: FOR R=-100 TO 100
 STEP 20 :: CALL SOUND(-300,
400-(SGN(R)*R),0,400-(SGN(R)
*R),10)
```

```
260 CALL MOTION(#1,R/4,SGN(H
)*(Q*6)):: CALL POSITION(#1,
N,O):: IF O<20 THEN H=4
:: GOTO 260 ELSE IF O>230 TH
EN B=B+0.05 :: GOTO 110
270 Q=Q-SGN(R)/2 :: CALL POS
ITION(#1,N,O):: IF N>120 THE
N CALL LOCATE(#1,121,O)
280 CALL COINC(ALL,K):: IF K
=-1 THEN 300
290 NEXT R :: CALL MOTION(#1
,0,0):: CALL LOCATE(#1,120,O
):: GOTO 140
300 IF G<>3 THEN 710
310 CALL POSITION(#1,N,O)::
IF N>90 THEN 710
320 CALL MOTION(#1,0,5)
330 CALL KEY(1,H,I):: IF I<>
0 THEN H=0 :: GOTO 290
340 GOTO 330
350 CALL DELSPRITE(#2,#3,#4,
#5):: CALL CHAR(104,"070F1F1
E0C1E0B01020303040F00010170F
830491224CC7B60C0C020F080E0E
0")
360 CALL CHAR(108,"1F6080"R
PT$("0",26)"F80601"RPT$("0
",26)):: CALL CHAR(112,"070F
07B44924321 90F03030407000303
F0F8FCFC7C7C78F0E0F0E010E080
C0C0")
370 CALL CHAR(116,"070F1F1E0
C1E0B01020303040F00010170F83
0481020C0786BC4C422F180E0E0"
):: CALL CHAR(120,RPT$("0",2
6)"80601F"RPT$("0",26)"01
06F8")
380 CALL CHAR(124,"070F07040
90402011F33234487000303F0F8F
CFC7C7C78F0E0F0E010E080C0C0"
):: CALL SPRITE(#2,104,2,123
,100):: CALL SPRITE(#3,108,2
,123,132)
```

Dear BUGBUSTERS,

I am writing to tell you about two 'BUGS'. One is in DATA-FILER (Dec.), and it is when you try to enter the data that it doesn't work.

I added this line and it worked:

2555 IF KEY=126 THEN 610

Also the program Spontaneous Reaction does not work when you ask for a 1 player game - it just goes on to an ordinary 2 player game. I have tried everything, but it is a little difficult for me because I am only 12 years old.

C.Jones

(Thankyou Master Jones - the DATA FILER has caused problems. We think that if it is typed correctly, making absolutely sure of the number of spaces in those long strings, that it works. I haven't tried your remedy yet, but we are grateful for your tip. Another way to get round those long strings is to convert them so that the program only looks for the first 3 chars: "EOF" instead of trying to identify the whole string. e.g.:

570 IF SEG$(NA$(I),1,3)="EOF" THEN 610

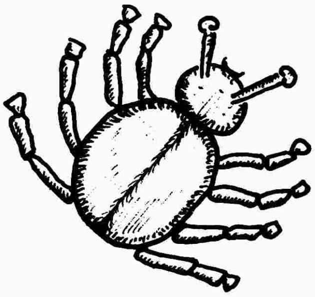
- we would like to hear from readers regarding Spontaneous Reaction, as to whether the same bug has been spotted and rectified)

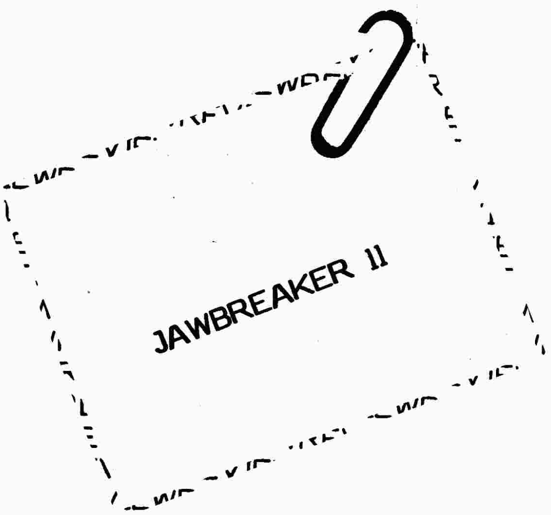
Dear BUGBUSTERS

Could you tell me what the answer is to the game TREASURES OF XEROX Vol.1 Issue 4, Line 4400 (ON SCREEN GOSUB 32767, etc.). What is the correct number, it is driving me mad after typing it in for several hours to find it won't work. I also have to congratulate you on a marvellous magazine. The game and graphics program are brilliant.

Mr K.F.Hughes

(We stand to be corrected, as usual, but we think that the offending line number (and the following comma) can simply be ommitted. Sorry about the bug - what has happened is that a line has been deleted, then evidently the program was RESequenced. That's ok as long as the line you are deleting isn't directly pointed to by any other line(s) in the program, as must have been the case here. When the RESequence was done, the line number in question couldn't be found, and what happens is that the number 32767 (the highest possible line number) is assigned by default. This is something to be more careful about than we evidently were!)





REVIEW
REVIEW
REVIEW
REVIEW

REVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEW

JAWBREAKER

If you don't like your games fast and furious, then leave JAWBREAKER on the shelf. If you want to spend your leisure time leisurely, forget it.
If you are looking for a game that will calm you down at the end of a long day, then don't bother. If you enjoy playing a game with one hand while unwrapping and eating a Kitkat with the other, then I'm afraid this is not the one. Jawbreaker doesn't even give you time to pick your nose.

It would not have been so out of place to call this one JOYbreaker; I can virtually hear the sound of snapping sticks all over the nation as I write.

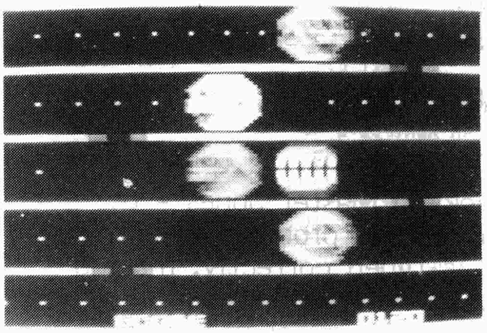
The screen is divided into five horizontal levels. You take the form of a set of gnashers in mid-screen. There is a door in each of the dividing walls, and the doors are continuously shifting horizontally. A row of dots lines each row, A la Pacman, with energizing capsules at each corner of the screen. Naturally you are being chased, and yes you guessed it, by Grinning Gobblers. In keeping with the Pacman guise, the Gobblers have the advantage until you reach an energizer, whereupon they turn into ghosts and become vulnerable to you for a limited time. Nifty shifting is required to move between the levels, and you can easily be trapped. Various treats such as lollipops appear from time to time for bonus points. When you clear a screen, a toothbrush appears to give you a quick brush-over. Whether this qualifies the Module as an educational one is not clear, but it certainly won't do any harm!

Okay, having set the scene, what is the appeal?

Firstly the title-screen and music are superb. A lot of thought has gone into this alone, and the options offered include choice of which keys to use for controls (If you aren't using joystick). Difficulty levels range from 'Teddy Bear' to 'Jawbreaker', with the latter being frantic to say the vary least. In fact, I can't think of many games that can compete with Jawbreaker for sheer speed. Breathtaking stuff this, for £12.50.

There is, of course a knack to the game. What it actually is evades me, as I keep getting hammered.

I'll have to leave Jawbreaker there, as I have an appointment at the dentist . . . .

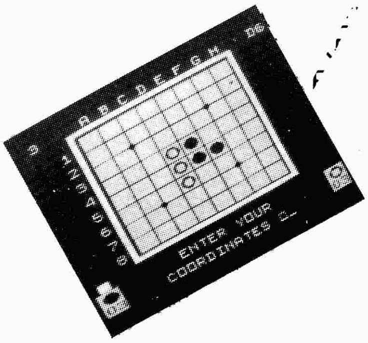


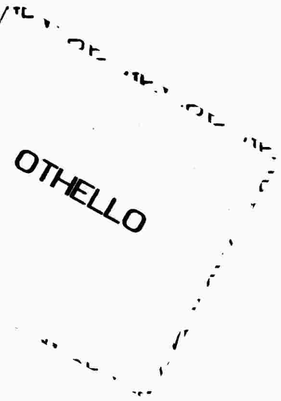
(taken at high speed)

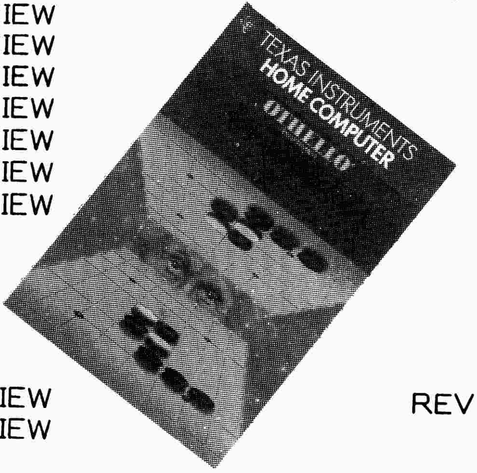
REVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEW
REVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEW
REVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEW

REVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEW

REVIEW
REVIEW
REVIEW
REVIEW
REVIEW
REVIEW
REVIEW







REVIEW
REVIEW

REVIEW

REVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEW

For those of you who are not aware of this strategy game, I thought it best to sit down and review it to prove how good a game it is.

I will take you through the game step by step, and hopefully persuade you to buy it because for a module it is remarkably cheap.

The game starts by displaying the grid numbered 1 to 8 vertically, and letered A to H horizontally. There is a lack of colour here, and throughout the game. Standard black screen, green board with white border and medium red for no. of moves and last move.

There is a prompt in cyan, asking whether you would like to create a game in progress. I suggest type 'N' or else you have to set up the board yourself! You are then asked if you would like to play against the computer (most people type 'Y' for ego purposes). You are then given the choice of 8 levels of play. If you have played before, then level 1 is a piece of cake. You are then asked if you want to move first. If you haven't played before let the computer go first.

The object of the game is to capture as many of the opponent's disks as possible by outflanking them. This simply means getting your disks either side of the opponent's (horizontally, vertically OR diagonally), and changing them to your colour.

To place a disk, all you do is type in the desired co-ordinates (column first) and press ENTER. The computer then changes the disks to the appropriate colour, alters the scores, displays the number of moves, and displays the last move. After a few games you begin to realize the importance of the edge squares, and more importantly the corners. If you have the good fortune to capture a corner first then you shouldn't have much trouble winning the game.

This game really is outstanding. Only two points to criticize - the colours as mentioned are a little dull, and unlike levels 1 to 5 which are very fast, levels 6 to 8 are much slower. The computer can take over 2 minutes making up its mind on the very hardest level, so it is possible to play for an hour and still be on the same game.

If you happen to lose, you are then subjected to the ubiquitous 'Death March', which gets on my nerves. (Always a shame, because when played properly this is a fine piece of music!) If you win, then a cheerful tune is played. The best music is reserved for a draw, which seldom happens.

Overall the program is well written, and has frequently induced me to smoke 20 cigs in an evening session (particularly level 6) and at £9.95 must be a bargain.

If you like to use your head instead of a fire-button (Does that explain those weird indentations in your face, Phil? - Ed) then Parco will certainly oblige.

MODULE review - Phil Donald

REVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEW
REVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEW

REVIEW

REVIEW

REVIEW

REVIEW

REVIEW

HOPPER

Command Module Review

(Reviewed on account of the fact that lots of people wonder if it is the same as Frogger.) (It isn't.)

Have I often heard it said, or is it just a convenient figment of my imagination that the most effective things are simple things done well? Wherever I got the notion, HOPPER fits the category anyway. I have always thought that when it comes to sitting down at the computer and choosing a game to play, a rather different set of rules comes into effect than the ones in operation when I want to gasp at someone's programming ability or choose to review a meaty package. Somehow I need to have a goal to achieve, yet relax at the same time. I don't want a long drawn out or complex issue at times like that, I want something pleasant to look at and convenient to manipulate. That's why I often revert to Munchman, Invaders, and lately HOPPER.

It was after playing Hopper for a cumulative total of about 6 hours over several days that I realized there is no music written into the module, not even on title screen. Somehow if the game 'works' such details have less import. Nor do the graphics win awards for variety, but who cares if the all important 'playability' factor is present?

You see it's like this. You are a kangaroo - oh yes you are - and you are being pursued around the room by trainers. Scattered about the place are packing crates. In order to evade your pursuers you duck and weave around the crates and 'kick' them into the paths of the trainers; either to trap them or squash them. And that's it. Mind you, with strategy and fast reflexes you can really pile on the points. If you just squash a trainer, so good, but if you can confine one or more and close in on them to hem them in completely, they die rather more dramatically and you get extra points. Sadistic aren't we? You get points also for knocking the crates around. In fact, as well as wiping out the trainers, you want to clear as many crates from the screen as possible. If a crate is free-standing and you kick it, it shoots across the room to the nearest obstacle, whether it be another crate or the wall. If the crate you kick is already up aginst something then the crate is crushed. Obviously, the aim is to trap trainers, smash crates to pieces all over the shop, then come back and polish off the poor trainers. Therein lies my ONLY moan, in that once you have trapped the trainers, it is a bit of a plod going round smashing crates without even a time limit or bonus scheme. Still it wouldn't be a real review without one gripe, would it?

Naturally the completion of a screen heralds the dawn of another, harder one. There are ten in all, with a nasty twist on the tenth. You have five lives to play with, and options include one or two player games, and keyboard or joystick control. There is also a useful pause facility in case you get caught short.

To sum up, the impressions that HOPPER leaves are simple but appealing graphics, compelling sound-effects, good joystick response, and fast, addictive animation.

And is that not, my friend, what all good arcade-style computer games are really about?

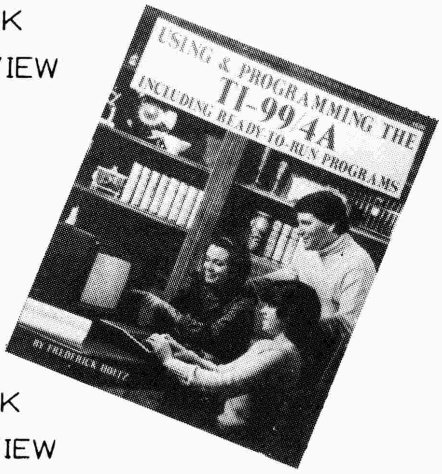
REVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEW
REVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEWREVIEW

USING AND PROGRAMMING THE TI-99/4A INCLUDING READY TO RUN PROGRAMS

by Frederick Holtz

a book review

BOOK REVIEW BOOK REVIEW



BOOK REVIEW BOOK REVIEW BOOK REVIEW

If a book were judged by the length of its title, then this one would need a lorry to carry away the awards. If on the other hand you could judge a book by its cover, then this one surely contains the secret to domestic bliss. You know the kind of thing I mean - the perfect All Americar happy family poised gleefully around their computer with not a hair, tooth, or rich-oak-bookcase-full-of-luxury-leather-bound-volumes-and-expensive-trinkets out of place.

The thing is - does the content of the book live up to all this? Well actually, it's not bad. I don't suggest that it will replace marriage guidance; or even grab the missus so wildly that she will come to so share your enthusiasm for the computer that she agrees to have a P.E. box instead of a week in Margate; but it's worth a look, all the same.

Best start by sorting out who it's aimed at. Like 'Getting Started' which I reviewed last issue, this book tends not to exhaust any particular subject, but introduces many. By definition then, it will suit the 'man in the street' (who IS he?) rather than the 'Expert'.

Having said that, there are features of considerable note here. One first impression that grabs is the number of photos, diagrams and charts in the book. It's surprising how helpful this can be. Included are illustrations of things that some folk have heard of but never seen, so it's a nice touch. There are a few bits that are either dated or inappropriate, like the fabled TI Thermal Printer and the Monitor, but it's interesting all the same. As for content? Well here again, what impresses is that in amongst the usual mixture of intro/first-steps/ examples and glossaries there are some juicy items that you'll be hard pressed to find in other 'basic' books. For instance, there is a whole chapter describing TI and 3rd party software that was available at time of going to print. Sadly some of the titles never reached these shores, but the info does make interesting reading nevertheless. Incidentally, it says 1983 inside the cover, yet have prices really changed so in such a short time? Do I hear $30-$40 for Scott Adams cassettes? (and I mean JUST the cassettes) Othello (reviewed above) is priced here at $40 - and how many of you paid $55 for Early Reading?

Well, that's as may be, but back to business. One section sticks out like a chapter on Himalayan Fruit-Bat Microsurgery in the Blue Peter Annual. Not that it's irrelevant, just that having been gently introduced to the rudiments of TI Basic, you are suddenly launched into a chapter on the TMS9900 Microprocessor Chip and its Assembly Language, without even a quiet warning to fasten your seatbelt! To be fair, I found it really useful, but I just feel for the absolute TI novice (me being just a relative novice), who is expected to come to terms with the fact that "...To perform double-precision multiply operations on unsigned 32-bit numbers with the 9900, a cross multiply technique can be used..." Oh, yeah?

I may have poked fun at various points here, but I actually like the book. It's quite big - over 200 pages and 9" by 8" cover-size, with good quality print and illustrations. If you are looking for an 'introductory' type book, then you will gain a lot from this one. If you already have 'Getting Started' type books, then this will duplicate some info for sure, but TI freaks will want it if only for its sheer enthusiasm and attention to items overlooked or forgotten by otherwise similar books. £7.85

cant do that

Hello again, time we took another look at problems and answers, don't you think?

Richard Owen has been busy as usual. He writes with suggested answers to various problems that we have printed in previous issues.

First off, that LINE 0 question. Richard says he experienced it once when he turned of the Peripheral Expansion Box. He got an error sound, a cursor on the screen (not flashing), and an error message:

```
* ERROR AT LINE 0
```

He then LISTed, but there was no program present.

Well, Richard, it's different to the one I got, since I had no P.E.B. at that time. Has ANYONE out there got any ideas?

Richard has help for Mark McGurn, who asked for a routine to detect if joystick fire-button had been pressed:

```
10 CALL KEY(1,K,S)
20 CALL KEY(2,K,S)
30 IF K=ASC("Q") etc.
40 IF K=ASC("Y") etc.
```

Ta, boyo.

Simon White has also offered a variation on a theme:

For joystick 1:

```
CALL KEY(1,K,S)::IF K=18 etc.
```

For joystick 2:

```
CALL KEY(2,X,S)::IF X=18 etc.
```

Thanks Simon.

Here's a response from Stephen Meadows to Dave Trevorrow's enquiry about DEF Statements:.

DEF can be used not only to set a variable to a figure, eg:

```
PI=3.149, DEF PI=3.149
```

but can be used as a quick and easy method of operating formulae, without using subroutines, (though DEF IS called in the same way as an EB subroutine), eg:

```
10 DEF TOTAL=COST1+COST2
20 INPUT COST1,COST2
30 PRINT TOTAL
```

uses less memory than the equivalent program:

```
10 INPUT COST1,COST2
20 TOTAL=COST1+COST2
30 PRINT TOTAL
```

and does not look so scruffy.

The User Guide does describe it adequately in TI's usual 'advanced beginner' style, with a complicated description comprising mostly of "function name...parameter...variables constants...parentheses...string results...etc..." This could be simplified a little (to say the least) but I severely doubt that TI will start reprinting their manuals now!

Now here's some new problems for you Bright Sparks out there to tackle.

Colin Jones wants to know if there is any way to:

a) stop people listing a program
b) make the program quit without pressing FCTN/=

<u>In TI Basic!</u>

I have my own ideas about one of Colin's questions, but the idea of this page is for YOU lot to get off yer butts. What do you think?

The relative speeds of TI Basic and Extended Basic have been a topic of some discussion lately. No-one will argue about the general performance of EB, and the fact that most things happen faster than with TI Basic, but it does seem than certain processes are actually SLOWER.

Mr. S.E.Hunt wants to know why, when running the short program below, it is quicker in TI Basic than EB:

```
100 FOR K=1 TO 20000
200 NEXT K
300 PRINT "COMPLETED"
400 END
```

He's right, you know. I just tried it and unless I'm mistaken it's about TWENTY SECONDS slower!

What's more, Stephen Meadows says:

- After testing the speed of Extended Basic against TI Basic, using the archaic 'Kilobaud' benchtests, it came as a surprise to see that TI BASIC was FASTER than EB! It beat EB in seven out of the eight tests with an average of being one and a half seconds ahead! The furthest ahead TI Basic got was 6.4 seconds, though compared with EB's one victory pulling almost 17 seconds ahead. This test used the functions (SIN, LOG, etc) and proved that EB is more suitable for some tasks than others. This lack of speed on EB's part is made up for with the use of multiple statement lines and fast-moving sprites, and only seems to show up in these (rather dated) benchtests -

We'd love to hear from anyone who has gone into this subject in depth.

Mr D.J.Armer requests an article on the use of Control Key (CTRL) and what it does, since " - the handbook does not cover this point very well - "

Good news, Mr Armer, have a peek at this issue's TItbits.

(I'm rapidly coming to the conclusion that the Reference Manual should be re-named the Bruiser Guide.)

Lastly, Richard Twyning writes:

"I have read through the first 99/4a magazine and I saw the piece about turning the cursor into a face for Minimemory, and I tried it with Editor/Assembler and it worked. The statement was:

```
CALL POKEV(1008,60,126,219,255,231,189,
195,126)
```

Could you please tell me how I could turn the cursor into a face using CALL LOAD in EB because there is no CALL POKEV in EB?

That's it for now, folks.

Continued from page 8

If you run that program you will see that interesting sound effects can be achieved with the speech chip, as well as speech. This gives you the opportunity of 'layering' sound effects by using this feature plus the noise generator of the sound chip together. Lastly, with a bit of imagination and experimenting, you can get your TI singing! What follows here is a fairly rough and ready attempt to combine both singing and background music.

```
100 OPEN #1:"SPEECH",OUTPUT
110 DIM W$(16),X(16),Y(16),Z
(16),SP$(16),P$(16)
120 FOR I=1 TO 16
130 READ F$
140 P$(I)=F$
160 FOR I=1 TO 16
170 READ E$
180 SP$(I)=E$
190 NEXT I
200 FOR I=1 TO 16
210 READ A,B,C,D$
220 X(I)=A
230 Y(I)=B
240 Z(I)=C
250 W$(I)=D$
260 NEXT I
270 FOR J=5 TO 1 STEP -1
280 FOR I=1 TO 16
290 PRINT #1:"//"&P$(I)&" 12
8"
300 CALL SOUND(J0,X(I),10
,Y(I),10,Z(I),10)
310 PRINT #1:SP$(I)
320 NEXT I
330 NEXT J
340 FOR I=1 TO 500
350 NEXT I
360 CALL CLEAR
370 END
380 DATA "12","12","22","22"
,"10","10","12","12","10","6
","12","22","14","16","18","
20"
390 DATA "FEED","","THE","",
"WOR","","LD","","LET","THEM
","KNOW","ITS","CHRISS","MAS
S","TIME",""
400 DATA 262,659,392,"FEED",
196,698,392,"",165,784,262,"
THE",196,523,262,""
410 DATA 175,880,440,"WORLD"
,220,698,440,"",196,784,392,
"",165,523,392,""
420 DATA 175,880,440,"LET",1
75,698,523,"THEM",165,784,39
2,"KNOW",165,523,330,"ITS"
430 DATA 147,880,349," chris
tmas",147,784,330,"
",196,698,294,"TIME",196,659
,262,"     "
440 END
```

*****JOYSTICK BENDERS*****

BUCK ROGERS	292,194
MOON MINE	101,990
MASH	4,077
BURGERTIME	500,000
HOPPER	31,250
STAR TREK	660,110
JAW BREAKER	6,435
PARSEC	10,842,800
ALPINER	92,186
TI INVADERS	54,121
CAR WARS	56,250
TOMBSTONE CITY	1,000,000
THE ATTACK	388,800
BLASTO	47,340
MUNCHMAN	204,000
CHISHOLM TRAIL	89,990
MUNCHMOBILE	27,290
PACMAN	43,180
MS PACMAN	151,610
MOON PATROL	133,150
JUNGLE HUNT	29,180
POLE POSITION	109,850
DEFENDER	104,150
DONKEY KONG	1,741,300
PROTECTOR II	125,764
PICNIC PARANOIA	1,750
FATHOM	5,881
SLYMOIDS	113,905
SNEGGIT	2,635
SUPERDEMON ATTACK	3,425
MOONSWEEPER	19,150

99/4a Magazine
Parco Electrics
2 Devonshire Court
Heathpark
Honiton
Devon

***** TOP FIVES *****

Games Five

Adv/Pirate
Dig Dug
Tunnels/Doom
Donkey Kong
Hunt/Wumpus

Tape Five

Virgin Funpacs
Mkt Simulation
Oldies 1
Oldies 2
Superhod

Serious Five

Artist
Collins Starters
Psnl Rec Keeping
Psnl Finance Aids
Graphx

Bookworm Five

TI Program book
Getting Started
Beginners Basic
Ed/Ass Manual
Using/Programming