

VIC COMPUTING

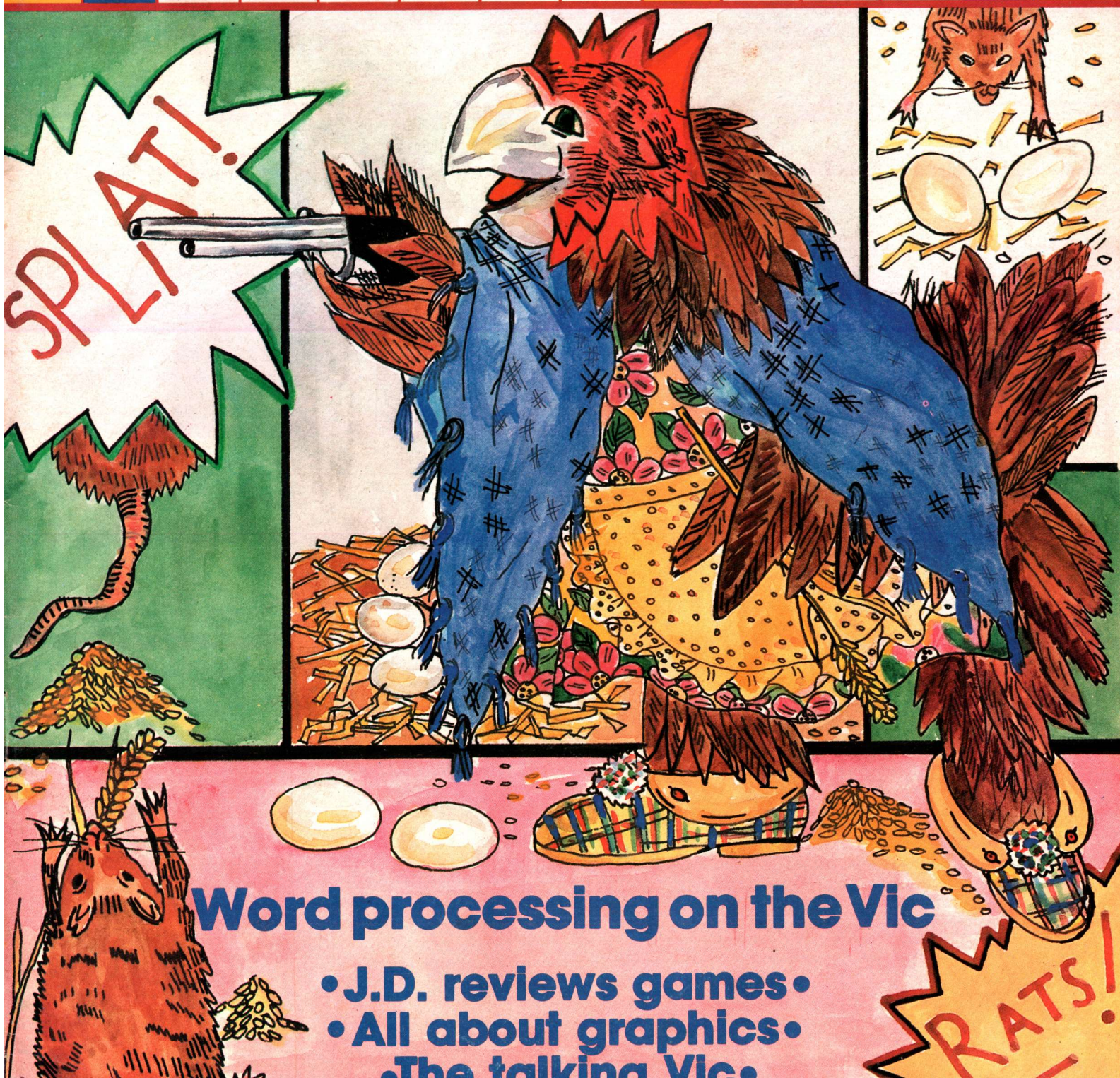
THE MAGAZINE FOR COMMODORE'S SMALL COMPUTERS

FEBRUARY 1983 Vol 2 ISSUE 3 95p

my's Tips
adders programs

MOTHER

HEN



Word processing on the Vic

- J.D. reviews games •
- All about graphics •
- The talking Vic •

Contents

Word processing on the Vic 8

This month we devote the In Business section to handling text — a useful tool for business, but by no means irrelevant to everyone else. An introduction to the subject and reviews of three packages on the Vic

Todd's Lore 17

Mike Todd starts into his 1983 column with a round-up of questions and answers on the Vic — the kind of things you wondered about but were never answered

Punter's Progress 20

We gave Average Reader a boxfull of Vic games and asked him to check them out. His no-holds-barred report includes Star Ratings — and five Best Buys

A Talking Vic 27

Machines that talk back to you are the stuff of science fiction, but the Vic can do it today — with an add-on speech synthesiser. Here's David Calderwood (who is blind) and his 'talking typewriter'

The Graphics Primer — Part One 31

Kevin Smart starts a series on Vic graphics, completed with routines, examples and demonstrations. Part One starts with the Magic Window — the screen and what you can do with it

Tinymon Vindicated 35

Remember Jim Butterfield's TINYMON machine code monitor? Here's a way of making it run on the Vic, complete with Basic hex loader, by Stephen Vine

Tommy's Tips 39

Two pages of insights, arguments, advice and Good Sense from our resident expert. This time: floating point, cassette funnies, altering error messages . . . and more

Mother Hen 44

Another great game by Tim Duncan, who gave you 'Rhino'. Just for a change you're the rat: and that's a very angry hen

Dear Vic 46

All your own work. A selection of letters to the Editor, with comments, queries and tips

Editor

Dennis Jarrett

Workers

Hywel David,

Subs Manager

Annabel Hunt

Ad Manager

Cathy Lane

Production

Rock Storm, Maggie Underwood

Art and design

Karen the Black Douglas, Mitch Guevara

Front Cover

Maggie Underwood

Old master

Julian Allason

Telephone: 01-607 9489 (all departments)

Vic Computing is published six times a year by Paradox Group, 39-41 North Road, London N7 9DP. Annual subscription: £6 per year (UK), £18.50 (Eire), £19 (Europe), £16 (rest of world). Printing by Thamesmouth Printing Co Ltd, Southend-on-Sea, Essex; Phototypesetting by Carlinpoint Ltd, London WC2E 9NW.

Material is accepted on an 'all rights' basis unless otherwise agreed. Enclose SAE if you want the return of original manuscripts and photographs. Please do not send listings: we only accept programs in cassette form.

All material © copyright Vic Computing Ltd 1983 and contributors. No part of this issue may be reproduced in any form without the prior consent in writing of the publisher, other than short excerpts quoted for the purposes of review and duly credited. The publishers do not necessarily agree with the views expressed by contributors, and assume no responsibility for errors in reproduction or interpretation in the subject matter of this magazine or any results arising therefrom.

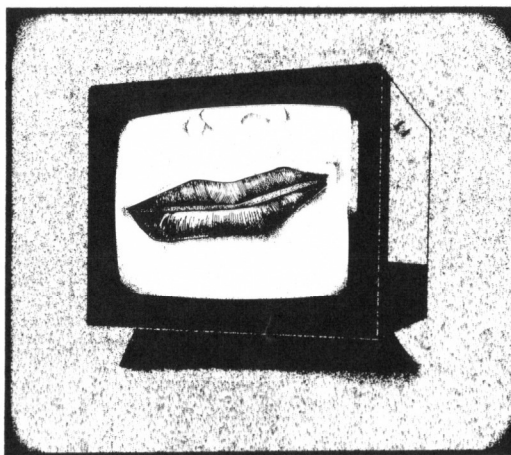
A Talking Vic by David Calderwood

```

10 PRINT "TALKING TYPEWRITER":PRINT:PRINT
20 PRINT"F1 - CARRIAGE RETURN ON SHORT LINES"
30 PRINT"F2 - RETYPES COMPLETE DOCUMENT"
40 PRINT"F3 - CHECKS IF LAST CHARACTER WAS A SPACE"
50 PRINT"F4 - OPENS A FILE FOR RECORDING ON CASSETTE"
60 PRINT"F5 - CLEARS SCREEN"
70 PRINT"F6 - READS FILE ON CASSETTE ONTO SCREEN AND INTO PRINTER"
80 PRINT"F7 - SPEAKS SCREEN"
90 PRINT"F8 - PREMATURE RETURN FROM SPEAKING ROUTINE"
100 PRINT:PRINT:HIT F5 TO CLEAR SCREEN"
110 REM COPYRIGHT DAVID CALDERWOOD
120 DIM SV$(200)
130 GOSUB 500
140 OPEN 123,4
150 CMD 123
160 GET A$:IF A$="" THEN 160
170 IFLC=1AND A$=" "THENPRINT#123,WB$:GOTO160
180 POKE 36878,15:POKE36876,200:POKE36876,0:POKE36878,0
190 IF A$=" " THEN GOSUB870:GOTO160
200 IF A$=" " THEN GOTO1040
210 IF A$=" " THEN PRINT "J": GOSUB360:GOSUB360:GOTO160
220 IFA$=" " THEN 1180
230 IF A$=" " THEN GOTO1070
240 IF A$=" " AND RIGHT$(WB$,1)=CHR$(32) THEN GOSUB360:GOTO160
250 PRINT A$;
260 IF A$=CHR$(20) THEN K=LEN(WB$): WB$=LEFT$(WB$,K-1):GOSUB360:GOTO160
270 IFA$=" " THEN320
280 WB$=WB$+A$
290 IF A$=CHR$(32) AND QQ=1 THEN QQ=0: GOTO 320
300 IF LEN (WB$)=70THEN QQ=1: GOSUB 360
310 GOTO160
320 PRINT#123,WB$
330 KK=KK+1
340 SV$(KK)=WB$
350 A$="":WB$="":PRINT:GOTO160
360 POKE 36878,15
370 POKE 36875,200
380 FORG=1TO100:NEXT:POKE36878,0
390 POKE36875,0
400 RETURN
500 REM
520 A=37136
530 B=37148
540 C=37149
550 X=223
560 Y=255
570 POKE(A+2),Y
580 POKEB,X
590 DIM A$(255)
600 FORAR=0TO63
610 READD$
620 A$(AR)=D$
630 NEXT
640 DATA 4642426263
650 DATA 32446263,14446263,31446263,30446263,44446263,59296263,3026446263,324430
266263
660 DATA 47446263,2632446263,8932446263,59246263,5912516263,5913136263,53456263
670 DATA 37373737446263,8944456263,7243436263,5931316263,42446263,60456263,15446
263
680 DATA 3051142460456263,5925316263,4521446263,1859306263
690 DATA 3037351362312545464,62144347255942186263,3719451330316263
700 DATA 252438401862312545464362144347255942186263,4254406256416237472345586219
156263
710 DATA 6943384562245929426263,31316263,1469206263,254538426263,284339306263
720 DATA 305324586263,375862315913426263,691313306263,6937383142433829446263
730 DATA 3837351362144347255942186263,252438401862144347255942186263,42213912186
263
740 DATA 372451316263,253812366263,1221391351316263,37381013426263
750 DATA 3039152139305930621421106263,184443536263,4550136263,4254406263,5743446
263
760 DATA 29525252436263,292139156263,313925316263,31592958136263,3233426263
770 DATA 132160136263,25382452136263,315912396225382452136263,245931625736136263
780 DATA 4489451924186263,2843324258625736136263,25454443396263
790 FORAR=64TO127
800 A$(AR)="2843472939256263"
810 NEXT
820 REM IMPROVING SOME WORDS
830 A$(33)="6525312524501233173513123643256263"
840 A$(94)="373736416263"
850 A$(38)="466413306263"
860 RETURN
870 FORSC=4096TO40601
880 GETE$
890 IFE$=" " THENRETURN
900 S=PEEK(SC)
910 IFS=>128THENS=S-128
920 IF PEEK (SC)=32 AND PEEK (SC+1)=32 THEN 1020
930 CO=1
940 B$=A$(S)
950 VX$=MID$(B$,CO,2)
960 POKEA,VAL(VX$)
970 POKEB,Y:POKEB,X

```

TALKING TYPEWRITER



We have been waiting for a program like this. It comes from David Calderwood, and David is blind: but he has found a way to make the Vic work for him, to make it fit a specific need and to do things that would otherwise be difficult or impossible. That has applied in a few areas already, notably games and some business functions — both of which fill the criterion of making the most of the computer. So let's have more uses for Vic like that!

It was the arrival of my braille mathematical tables that made me realise my need on a cold night in bed: you can read it under the blankets. But to have to 'do sums' using a book that makes the telephone directory look like a paperback is just not on.

"Why don't you get yourself a 'Vic'?" said a friend; and off I went to the shop to try one. The salesman reverted quickly from his "good colour and graphics" patter and started to point out the superb keyboard and sound.

He was right, of course, and I paid over my deposit. Three weeks later the shop did a moonlight flit and I came face to face with Commodore. They were most helpful and sent a Vic direct to my home, allowing me the £25 I had previously lost.

With the help of the Royal National Institute for the Blind and others with their generous students' grants I have been able to fully expand my Vic using the Stack system. I also run the printer and a Vox Box speech synthesiser. The latter device speaks the answers to my Open University problems to me!

The general purpose 'talking' subroutine which I have developed can be run on a smaller Vic by changing the top left-hand and bottom right-hand corner screen numbers in the main FOR-NEXT loop.

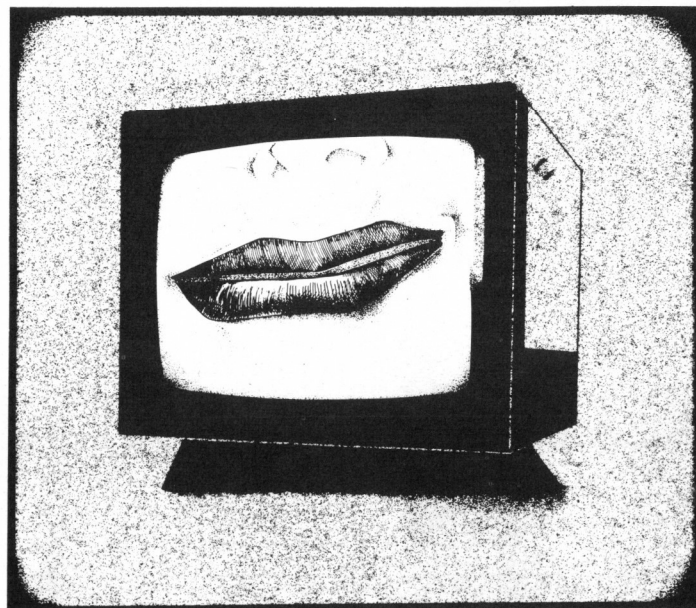
The idea is a simple one; it's just a case of peeking into each of the locations on the screen memory map to see what's there. The numbers of the array are the same as the screen codes of the Vic characters; and they are then poked out to the Vox Box and spoken. The result is that whatever is on the screen is spelt out. It's possible to make up



```

980 IFPEEK(C)<>16THEN980
990 CO=CO+2
1000 IFCO-1=LEN(B$)THEN1020
1010 GOTO950
1020 NEXT
1030 RETURN
1040 FOR RT=0 TO KK
1050 PRINT# 123,SV$(RT)
1060 NEXT:GOTO160
1070 PRINT "J"
1080 PRINT "PLEASE ENTER FILE NUMBER FOR SAVING": INPUT PT
1090 AL$="4": FI$="#"
1100 OPEN PT,1,1
1110 FOR RT=0 TO KK
1120 PRINT#PT,SV$(RT)
1130 NEXT
1140 PRINT# PT,FI$
1150 CLOSEPT
1160 PRINT "FILE " PT " SAVED."
1170 GOTO160
1180 PRINT"ENTER FILE NO. TO BE READ":INPUT PT
1190 OPEN PT,1,0
1200 GET#PT,A$
1210 IF A$="#" THENPRINT"FILE ENDS":GOSUB360:CLOSE PT:GOTO160
1220 PRINTA$:PRINT#123,A$:
1230 WB$=WB$+A$
1240 LC=1
1250 GOTO1200

```



```

60000 PRINT"REDO MK2"
60020 QA=37136
60030 QB=37148
60040 QC=37149
60050 QX=223
60060 QY=253
60070 POKE(QA+2),QY
60080 POKEQB,QX
60090 DIMQA$(253)
60100 FORQR=0TO63
60110 READQD$
60120 QA$(QR)=QD$
60130 NEXT
60140 DATA 464246263
60150 DATA 32446263,14446263,31446263,30446263,44446263,59296263,3026446263,3244
30266263
60160 DATA 47446263,2632446263,8932446263,59246263,5912516263,5913136263,5345626
3
60170 DATA 37373737446263,8944456263,7243436263,5931316263,42446263,60456263,154
46263
60180 DATA 3051142460456263,5925316263,4521446263,1859306263
60190 DATA 3837351362312545464362144347255942186263,3719451330316263
60200 DATA 252438401862312545464362144347255942186263,42544062564162374723455862
19156263
60210 DATA 6943384562245929426263,31316263,1469206263,254538426263,284339306263
60220 DATA 305324586263,375862315913426263,691313306263,6937383142433829446263
60230 DATA 3837351362144347255942186263,252438401862144347255942186263,422139121
86263
60240 DATA 372451316263,253812366263,1221391351316263,37381013426263
60250 DATA 3839152139305930621421106263,184443536263,4550136263,4254406263,57434
46263
60260 DATA 29525252436263,292139156263,313925316263,31592958136263,3233426263
60270 DATA 132160136263,25382452136263,315912396225382452136263,2459316257361362
63
60280 DATA 4489451924186263,2843324258625736136263,25454443396263
60290 FORQR=64TO127
60300 QA$(QR)="2843472939256263"
60310 NEXT
60320 REM IMPROVING SOME WORDS
60330 QA$(33)="6525312524501233173513123643256263"
60340 QA$(94)="373736416263"
60350 QA$(38)="466413306263"
60360 FORQD=4096TO4601
60370 QS=PEEK(QD)
60380 IFQS>128THENSQS=QS-128
60390 IF PEEK (QD)=32 AND PEEK (QD+1)=32 THEN 60490
60400 QO=1
60410 QB$=QA$(QS)
60420 QV$=MID$(QB$,QO,2)
60430 POKEQA,VAL(QV$)
60440 POKEQB,QV:POKEQB,QX
60450 IFPEEK(QC)<>16THEN60450
60460 QO=QO+2
60470 IFQO-1=LEN(QB$)THEN60490
60480 GOTO60420
60490 NEXT

```

TALKING SUBROUTINE

commonly-used words like 'sine' and 'tangent': but in the main, spelling is quite good enough for my purposes.

I also use the printer as a typewriter (via another of my programs also listed below). It really does work — I'm writing this with it now!

The Vic enables me to play games as well (yes, apart from Simon). I have been able to modify Bug Byte's excellent Vic Gammon so that the writing on the screen is read out. I copy the computer's moves onto a tactile back-gammon board and then enter my own move in response.

I am also working on some other games, including adventure programs — an ideal game for blind people to play.

While this complete system may not be the most sophisticated, it can be purchased by any blind person for a few hundred pounds rather than the many thousands that custom-built machinery costs. Of course it has meant that my wife has had to me for many many hours (including **Vic Computing** from cover to cover) and she has also written countless letters of enquiry on my behalf. Rita also typed in many of the first experimental programs before we reached the point where we finally devised the talking subroutine which enabled me to take over.

But without this effort on our part this technology would not be available to me and I would still be wrestling with my braille four-figure tables.

Obviously I am still very much a learner, so please overlook the inelegance of my listings: the main thing is that they work and fulfil a particular need.

