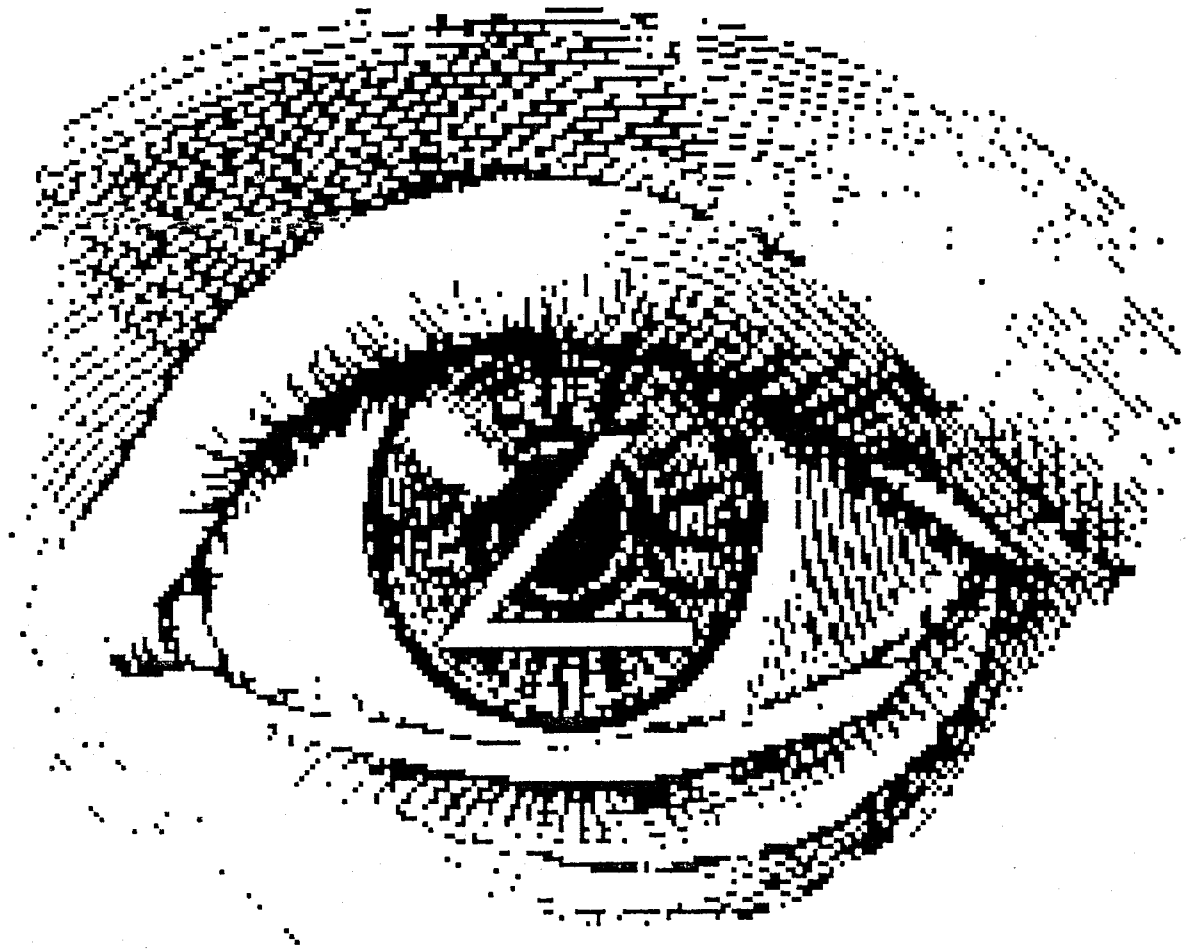
ORIC USER MONTHLY

with Oric Enthusiasts

Europe's longest running Oric Magazine

Number 55

March 1992



Edited and Distributed by Dave Dick, 65 Barnard Crescent, Aylesbury, Bucks HP21 9PW

* * E D I T O R I A L * *

HELLO,
and welcome to spring!
Do you remember that early seventies hit called 'Everything is Tuesday', by the 'Chairmen of the Board'. To our local postman (Ted the Truss), it certainly is as far as deliveries to OUM are concerned. Today (Tuesday Feb. 25th, he struggled with 3 pages from Peter, 5 pages from Allan and over 300 from Jon. Plus all the other mail. ..... Those of you who have tried to phone me recently will of got the dreaded answerphone. This will continue for another month,as I am still trying to clear the backlog.

And now to TWENTY FOUR PAGES of ORIC matters.

****

A NEW ORIC USER

On December 20th. the world welcomed a new Oric user born with the name Michael Benjamin Kimberley. Both mother and father (ROB) are doing well. Nice one Michael - I've enclosed a junior subscription form.
****

NECK & NECK

At January 1992,both the CEO and OUM had 98 readers.
The CEO are due to release 2 new games for the Telestrat and Sedoric Disc users. They are MAHJONG and ORICSWAP.
****

ERRATA

Owners of the book: "METEORIC PROGRAMMING" from Melbourne House may have had trouble getting some of the listings to run. If you didn't get the official correction sheets,then just send 2 x 24p stamps to OUM.
^^^^**^^**^^**^**^^**^^^^*^*^**^^*^^*^**^*^^*

Mini Book Review

'MADE - EASY - USING YOUR ORIC ATMOS' is from the pen of Garry Marshall. Originally selling for 3.95, this compact 137 page book has some interesting chapters. It looks at Databases,Spreadsheets and Wordprocessors in general, and then goes into some depth with help on ORICBASE,ORICCALC and AUTHOR. A must if you are trying to get to grips with these topics.
****

DEADLINES

ALL ARTICLES FOR INCLUSION IN THE APRIL ISSUE SHOULD BE WITH ME BY TUESDAY MARCH 24th and NOT THE 23rd AS PRINTED LAST TIME AROUND.
A BIG 'OUM' THANK YOU TO OUR WRITERS FOR MAKING IT ON TIME THIS MONTH - IT REALLY DOES MAKE LIFE THAT MUCH EASIER.
****

NEXT MONTH

MORE ARTICLES,NEWS,REVIEWS,LISTINGS,GAMES TIPS AND WHATEVER WE CAN AFFORD..... BYE - DAVE

THE FOURTH ORIC MEET

The fourth Oric meet will take place on Saturday July 18th. at the RIVETS SPORTS AND SOCIAL CLUB, MANDEVILLE ROAD, AYLESBURY, BUCKS.
Tickets are again 2 pound each. If you want tickets please send yer money. If you want to demo anything,please let me know in advance.

ARNTOGRAM

As we go to press; the wife is preparing to celebrate her 40th birthday with a party. I think an ideal present for her would have been one of those men from the CHIPPENDALES. (Sighs from the lady members). However at great expense and direct from his Oricland in Norway, I have arranged a special guest for her in the form of ARnt Erik Isaksen. HE promises to give her a good thrashing - at NHL ICE HOCKEY MANAGER. Arnt also promises to wear long trousers for the day!
Arnt is in London for the week - you have been warned!

THE JEAN GENIE

Jean (Mr. Atmos) Boileau will be in Southamoton for a WORLD HEALTH ORGANISATION meeting from April 6 - 11. He would like to know if there are any Oric owners in that area. Unfortunately, YES. Peter Wooley is in Portsmouth. Or perhaps others from not too distant would like to meet up with Jean. I can personally vouch for the fact that he is a very interesting guy. Please write to me or check out his address in the last contact list.

THREE INCHES

Jonathan Bristow informs me of a source for 3" discs at 1 pound each. You should contact: D.HAYWARD at 14 WILLIAM ST.,CEDWORTH CV12 9NS

SURVEY

Following a recent survey that I have done on OUM members,it appears that 66% are now disc based. With this in mind,added to the fact that more are switching from cassette; I have decided to reduce stocks of cassette software. Certain titles will be erased from the catalogue - basically those with many parts to them which I have to duplicate myself. e.g: GRAND PRIX. These will be phased out gradually so that you cassette users have the opportunity to get them.
SPECIAL OFFERS on cassettes will appear in this and future issues.

*** SPECIAL OFFERS ***

Special offers for the month of March are as follows:
48K ORIC 1 + PSU + Manual - 15.00 incl. post
48K ORIC1 + Manual + 10 Oric 1 games - 12.00 incl.post

Cassette software (incl post) -
HELLS TEMPLE - 1.70
BUCCANNEER - 1.50
CHOPPER - 2.10
CHUCKFORD - 1.70
DPTLQ - 3.25
GHOSTMAN - 50 pence
KRILLYS - 1.00
MANIC MINER - 2.80
SPOOKY MANSION - 1.50
TABLES TEST - 1.00
CAN YOU COUNT - 1.50
GOLDMINE - 2.10
All orders to Dave Dick at usual address.

SPECCY AD !!!

Nigel Alefounder sold his Spectrum to get a second Atmos (astute fellow) . He has one piece of as new Speccy software to dispose of., which is HISOFT's DEVPAC (assembler/debugger). It works with machines up to the 'PLUS 2'. All that is wanted for it is a modest donation to charity.
NIGEL is at:
17 DELPH ROAD, NORTH HYKEHAM, LINCOLN. LN6 9RT

U L A

Dave Utting is still in search of a circuit diagram for the ORIC ULA. He offers the grand sum of 20 pound.
I in fact spoke to Steve Hopps,who thought that perhaps the chief engineer at Oric may know. Steve said to give him a nudge and he will check it out.

STEVE HOPPS - YOU HAVE BEEN TANGO'ED !!

PETER's CAMERA

All those at last year's Oric meet will remember Peter Bragg's miniature camera. He informs me that the manufacturer ceased production when they could no longer obtain supplies of the light sensitive chip. They apparently sold about a thousand of the cameras to BBC micro users and so there must be quite a few gathering dust somewhere.

SOFTWARE CORNER

CLUB DISC'ORIC JEO DISC no.4 (Autumn '89).

Not to be confused with the CEO DISC no.4 of Spring '91.
The CDO disc (sorry,cassette users) is now available to OUM readers for the incredible LOW LOW price of 5 pound on 3" disc or 4 pound on 3.5".
One side contains the magazine,with page upon page of information. the other side has the software,and what software!
PSYCHIATRIC - a superb ladder game, SINGERIE - a very nice Honey Kong type of game, MUSIC - soothing, CONTACT LIST - an update of MULTIFILER stores all the CDO contacts and the 'creme de la creme' is EASYTEXT (as used for this page) - a superb word processor with many outstanding features. NEVER in the history of Oric will you get such a bargain again.
You had your order forms last month,so send your cash to Jon Haworth - pronto!

'BROTHER ' CAN YOU SPARE A DIME

Derek Gomer has for sale: 2 rolls of plain paper and 4 ribbons for a Brother HR5 printer. Price is 9 pound,or alternately he will swap for thermal paper.
Derek is at 42 GRAIG STREET,GRAIG,PONTYPRIDD,MID GLAMORGAN CF37 1NF

WANTED - DEAD or ALIVE

We want 30 Oric users to attend the next Oric meet so that we can break even on the cost of the hall,which will this year cost us 60 pound. See elsewhere for date and venue.

B A C K I S S U E S

We continue our look at early issues of O.U.M - see last issue for details of how to order.

ISSUE 21 (May 1989) - 9 pages - interview with author of DPTLQ, review of WINTER GAMES, report of Alternate micro show,news and tips ..... ISSUE 22 - 9 pages - reviews of TYRANN,COMPOSER,ROLAND GARROS and FOOTBALL, a look at Shiva's machine code book, Ramrom looks at loading errors, Print & and RND.,helpline,hints..... ISSUE 23 - 9 pages - interview with Alistair Way, Ramrom looks at more bugs plus the disc systems and DOS's avialble for the Oric, review of WILLY,tips and news..... ISSUE 24 - 9 pages - O.U.M is 2 years old, ZORGON still tops the charts, Robert Cook reflects, Ramrom reads your Microdisc eprom, The Byte Drive, review of SLALOM RACE, a listing to magnify characters in avariable,hints etc..... ISSUE 25 - 9 pages - questionaire results, RAmrom looks at all the ORIC books, mini-reviews and news..... ISSUE 26 - 9 pages - news,hints,mini-reviews, a look at The World of Oric, Ramrom takes a further look at the Byte Drive and EDIT command..... ISSUE 27 - 9 pages - news,Oric software with the initial 'A', Music Teach-in part 1, Ramrom, ARROW of DEATH solution,hints/tips..... ISSUE 28 - 9 pages - Jon views the Alternate micro show,reviews of CRICKET and CDOSOFT 2 (MLUCH & RISIKO), organisation of a SEDORIC disc,Music Teach-in part 2..... ISSUE 29 - 9 pages - reviews of GALACTOSMASH and the rest of CDOSOFT2 (YAHTZEE & OEIL DE SOLTEC), A to Z of ORIC people and products, NIBBLE - the descriptor and SEDORIC,news,MUSIC Teach-in part 3..... ISSUE 30 - 9 pages - review of GAMES COMPENDIUM,more on SEDORIC, Music teach-in part 4, news..... ISSUE 31 - 11 pages - listing of a drawing program,review of Jon Haworth's Oric book, Music teach-in continues, Oric Dos organisation, review of CHESS from IJK and preview of ESCAPE,news..... ISSUE 32 - 10 pages - Zorgon still tops charts,hints/tips,reviews of GRAND PRIX and TETRIS (the French version),Preview of KRYSTAL WORLDS, Jon looks at the current Public Domain for the Oric,Music teach-in pt.6,news..... ISSUE 33 - 12 pages - hints/tips,reviews of ARROW of DEATH II and CHESS MADNESS, RAMROM,more on SEDORIC,addition to drawing program,Music teach-in,news..... ISSUE 34 - 9 pages - Robert Cook's last issue, hints/tips,reviews of MIZAR and the O.N PACK, Jon looks at CLOAD problems,listing to convert Hex to Binary and viceversa,news..... ISSUE 35 - 12 pages - it is now July 1990 and Dave Dick becomes editor - reviews of GRENDEL & ROCKRUN, news from Europe,readers letters, Ramrom continues with a CLOAD,Fractals listing,news..... ISSUE 36 - 12 pages - OUM membership stands at 56, readers letters,free sectors on discs, Aylesbury Oric meet report, disassembly of page 2, a couple of short listings ,news etc..... ISSUE 37 - 12 pages - interview with Vincent Talvas (President of the CEO) , Stan Ellison looks at Oric machine code,Ramrom looks at all the Oric mags,more disassembly of page 2,Peter Bragg's software,mini-reviews,auto-stop routine,the story of KLUBB ORIC NORDEN,news etc..... ISSUE 38 - 14 pages - Stan continues on machine code,interview with Vincent continues, Jon looks at the latest P. Domain (Scrabble is top), menu listing, Dissassembly,review of ROBINSON CRUSOE,hints/tips/charts/news,help with French..... ISSUE 39 - 12 pages - 2 short listings,cassette relay control,readers letters,Disassembly,Interview with Geoff Philips,VORTEX, hints on LAST WARRIOR,news..... ISSUE 40 - 14 pages for the December 1990 issue - the editor looks at 10 pieces of his software, ROLAND GARROS instructions,volume control for your Oric, AMS 4 show report,more help on LAST WARRIOR,Disassembly,Interview with Geoff Philips continues,Oric Enthusiasts intro.,2 short listings,letters.news.....

WELL - that's 40 issues covered now. Next time we will bring you right up to date.

==========

G R A N D A D

COMING SOON - TO AN ORIC NEAR YOU - GRANDAD - THE LATEST TEXT ADVENTURE ESPECIALLY FOR YOU !!!

RAMBLING IN THE ROM - 36

'Magnetic Media'

For three years now I've been sending out hundreds of discs and cassettes without the slightest problem. In the last two months I've had more returns (although still only a small percentage) than ever before - and it was in correspondence with Peter Bragg that I think we realised what might be happening. It took three goes to get his CEO discs to him uncorrupted. Another clue was that the problems were primarily with 3½" and 5¼" discs, and with cassettes. It also happened particularly to post aimed at or via London. The good old thick plastic 3" discs got through unscathed. The answer must be that the Post Office have stepped up the level of their screening of mail with X-ray machines as a result of recent events. So if you get corrupted media through the post - send it back for a second attempt!

Rambling on...
This month a well-known command (note the ELSE in V1.1)...

'FOR' (COMMAND)

C841	LDA #80	C855	LDA #80	Indicate not array variable
C843	STA 2B	C857	STA 2B	
C845	JSR $CAD2	C859	JSR $CB1C	Do a LET for the index variable
C848	JSR $C3CA	C85C	JSR $C3C6	Does loop with this variable exist?
C84B	BNE C852	C85F	BNE C866	No, jump
C84D	TXA	C861	TXA	Yes, adjust stack pointer so as to
C84E	ADC #0F	C862	ADC #0F	jump this block
C850	TAX	C964	TAX	
C851	TXS	C865	TXS	
C852	PLA	C866	PLA	Take return address from the
C853	PLA	C867	PLA	interpreter
C854	LDA #09	C868	LDA #09	Allocate 18 bytes on the stack
C856	JSR $C43B	C86A	JSR $C437	
C859	JSR $CA1C	C86D	JSR $CA4E	Find end of instruction
C85C	CLC	C870	CLC	(result is in Y)
C85D	TYA	C871	TYA	
C85E	ADC E9	C872	ADC E9	and add to TXTPTR
C860	PHA	C874	PHA	save low byte on stack
C861	LDA EA	C875	LDA EA	and high byte, stacking
C863	ADC #00	C877	ADC #00	the effective start address
C865	PHA	C879	PHA	of the loop
C866	LDA A9	C87A	LDA A9	Stack the line number
C868	PHA	C87C	PHA	high byte
C869	LDA A8	C87D	LDA A8	and low byte
C86B	PHA	C87F	PHA	
C86C	LDA #&TO	C880	LDA #&TO	get 'TO'
C86E	JSR $CFDB	C882	JSR $D067	
C871	JSR $CE7A	C885	JSR $CF06	Verify numeric index (at last!)
C874	JSR $CE77	C888	JSR $CF03	get limit of value
C877	LDA D5	C88B	LDA D5	take the sign

```
C879  ORA #7F      C88D  ORA #7F      don't change b0-b6
C87B  AND D1       C88F  AND D1
C87D  STA D1       C891  STA D1       include the sign
C87F  LDA #8A      C893  LDA #9E      place return address
C881  LDY #C8      C895  LDY #C8      i.e. #C88A/#C89E
C883  STA 91       C897  STA 91       in #91-#92
C885  STY 92       C899 STY 92
C887  JMP $CF34    C89B  JMP $CFC0    Stack ACC1 and to next line

C88A  LDA #4B      C89E  LDA #81
C88C  LDY #DC      C8A0  LDY #DC      AY points to a value of 1, the
C88E  JSR $DE73    C8A2  JSR $DE7B    default step value; put in ACC1
C891  JSR $00E8    C8A5  JSR $00E8    take current character
C894  CMP #CB      C8A8  CMP #CB      is it STEP?
C896  BNE C89E     C8AA  BNE C8B2     no, jump
C898  JSR $00E2    C8AC  JSR $00E2    yes, jump it
C89B  JSR $CE77    C8AF  JSR $CF03    evaluate numeric expression
C89E  JSR $DF04    C8B2  JSR $DF13    take its sign in A
C8A1  JSR $CF25    C8B5  JSR $CFB1    stack A then ACC1
C8A4  LDA B9       C8B8  LDA B9       and save the address
C8A6  PHA          C8BA  PHA
C8A7  LDA B8       C8BB  LDA B8       of the index variable
C8A9  PHA          C8BD  PHA
C8AA  LDA #8D      C8BE  LDA #8D      and finally the code for FOR
C8AC  PHA          C8C0  PHA
```

EXECUTE A LINE

```
C8AD  JSR $C930    C8C1  JSR $C962    Test if Ctrl-C
C8B0  LDA E9       C8C4  LDA E9
C8B2  LDY EA       C8C6  LDY EA       take TXTPTR in AY
C8B4  BEQ C8BC     C8C8  BEQ C8D0     if direct mode, jump
C8B6  STA AC       C8CA  STA AC       if not, save for CONT, etc...
C8B8  STY AD       C8CC  STY AD       TXTPTR in #AC-D
C8BA  LDY #00      C8CE  LDY #00
C8BC  LDA (E9),Y   C8D0  LDA (E9),Y   take current character
C8BE  BNE C918     C8D2  BNE C92F     jump if not start of line
.....              C8D4  LSR 0252     indicate no IF for the moment
C8C0  LDY #02      C8D7  LDY #02

C8C2  LDA (E9),Y   C8D9  LDA (E9),Y   take link high byte
C8C4  CLC          C8DB  CLC          prepare addition & no error
C8C5  BNE C8CA     C8DC  BNE C8E1     if not end of program, continue
C8C7  JMP $C958    C8DE  JMP $C98A    if is, treat with END
C8CA  INY          C8E1  INY
C8CB  LDA (E9),Y   C8E2  LDA (E9),Y   take number of current line
C8CD  STA A8       C8E4  STA A8       high byte
C8CF  INY          C8E6  INY
C8D0  LDA (E9),Y   C8E7  LDA (E9),Y
C8D2  STA A9       C8E9  STA A9       and low byte
```

```
C8D4 TYA          C8EB TYA
C8D5 ADC E9       C8EC ADC E9      and adjust TXTPTR
C8D7 STA E9       C8EE STA E9      to effective start of line
C8D9 BCC C8DD     C8F0 BCC C8F4
C8DB INC EA       C8F2 INC EA      without forgetting the high byte
C8DD BIT 02F4     C8F4 BIT 02F4    TRON mode?
C8E0 BPL C8F5     C8F7 BPL C90C    no, jump
C8E2 PHA          C8F9 PHA         save A (useless!)
C8E3 LDA #'['     C8FA LDA #'['
C8E5 JSR $CC4D    C8FC JSR $CCFB   Display a bracket
C8E8 LDA A9       C8FF LDA A9
C8EA LDX A8       C901 LDX A8      then the line number
C8EC JSR $E0C1    C903 JSR $E0C5   (#E0BD/#E0C1 would be safer)
C8EF LDA #']'     C906 LDA #']'
C8F1 JSR $CC4D    C908 JSR $CCFB   and close brackets
C8F4 PLA          C90B PLA         recover A
C8F5 JSR $00E2    C90C JSR $00E2   take current character
C8F8 JSR $C8FE    C90F JSR $C915   execute command or assignment
C8FB JMP $C8AD    C912 JMP $C8C1   and start again
```

EXECUTE COMMAND IN A

```
C8FE BEQ C92D     C915 BEQ C960    if already end of line, exit
C900 SBC #80      C917 SBC #80     test for keyword
C902 BCC C915     C919 BCC C92C    if not keyword, assignment
C904 CMP #42      C91B CPM #42     #80+#42=#C2=last command
C906 BCS C91C     C91D BCS C94F    if not command, error
C908 ASL A        C91F ASL A       calculate displacement
C909 TAY          C920 TAY         as index
C90A LDA C007,Y   C921 LDA C007,Y  take address (high byte) of command
C90D PHA          C924 PHA         and put on stack
C90E LDA C006,Y   C925 LDA C006,Y  repeat for low byte
C911 PHA          C928 PHA         and put on stack
C912 JMP $00E2    C929 JMP $00E2   jump to command and execute
C915 JMP $CAD2    C92C JMP $CB1C   do a 'LET'

C918 CMP #':'     C92F CMP #':'    is it the separator
C91A BEQ C8DD     C931 BEQ C8F4    yes, execute next command
.....             C933 CMP #C8     is it ELSE?
.....             C935 BNE C945    no, jump
.....             C937 BIT 0252    yes, has there been an IF
.....             C93A BPL C94F    no, error
.....             C93C JSR $CAB1   yes, find next instruction
.....             C93F LSR 0252    and reset IF flag
.....             C942 JMP $ C8C1  go to next instruction
.....             C945 CMP #'"'    is it short REM?
.....             C947 BNE C94F    no, error
.....             C949 JSR $CA99   yes, find next line
.....             C94C JMP $C8C1   and continue execution there
C91C JMP $CFE4    C94F JMP $D070   'SYNTAX ERROR'
```

Machine Code for the Oric Atmos (Part 13)

Peter N. Bragg

The Story so far

We have looked at the basic requirements for machine code programming on the Oric and now have a small table (subset) of sixteen useful instructions. The last issue used a program of just two short routines to set the keyboard so that it could be made command three simple functions.

As the program was intended to concentrate on the keyboard operation, we "borrowed" three of the Oric's own routines (ZAP, SHOOT and EXPLODE) to aid the demonstration.

The listing shown on the next page is the same program, with just one change, it has been extended to include Oric's PING routine. If you compare last months listing with this one, you will also see that the description is in a slightly more compact form in this version. So now lets have a look at it.

The program consists of two routines. This makes life easier, because each routine can be dealt with seperately. The first routine gets an input from the keyboard, the second routine tests that input and calls up any action that may be requested.

"Fetch Item from Keyboard" Routine

This routine waits for a key to be pressed. Each key has it's own specific hex code. After a key is pressed, its code is tested, to see if it is one of the four command keys requiring action. Once that has been dealt with, the operation loops back again for another key press.

We have dealt with the procedure for reading the keyboard in Part 11 of the series. However, to recap briefly, the instructions 1010 and 1013 use another of the Oric's own routines (GTORKB at EB78) to make the program stop and wait until a key has been pressed. Once this has been done, a copy of the key (code) is saved in location 1001 and we can then go on to test that key code.

As we have just said, this routine is a loop and as it is the start of our program, we obviously need a means to escape from the loop, otherwise we would have no effective exit from the program, apart from the RESET button.

In this case the exit is provided by the two instructions 1018 and 101A, which provide the only access to the "Exit" at instruction 1022. It is usually much easier, if you provide the means to exit from the program, immediately after getting that particular command from the keyboard and before you get involved with anything else. Now we have dealt with the input from the keyboard, let us go on to see how it is tested and used.

"Control Select" Routine

There has been a slight change of name here. The routine in effect, selects the controls for our program. Let's ignore the first instruction at this stage. If you look at the main part of the routine, you should be able to see that it actually consists of four small subroutines, one for each of the four keys we are using. The four subroutines are nearly identical to each other, the only real differences are the actual key tested and the operation called up for it by the "JSR" instruction.

Lets look at the first key code test operation in instructions 1030 to 1037. The CMP instruction at 1030 tests the Accumulator for the code produced by pressing Key "Z". The result of that test sets or clears the "Zero Flag" which in turn controls the operation of the BNE (Branch) instruction at 1032.

```
Oric - Demo        Keyboard Control          5 Feb 92
-----------        ----------------          --------
[ CALL#1010 ]------------[ FETCH ITEM FROM KEYBOARD ]

1001:( )        .....    Storage for Key input.

        ---start---    ---Fetch a Key---
1010:20 78 EB : JSR EB78 : Read Keyboard
1013:10 FB    : BPL 1010 : until key pressed.

1015:8D 01 10 : STA 1001 : Save Key press input.

                       ---Continue or Finish ?---
1018:C9 1B    : CMP"ESC" : Was "ESC" key pressed ?
101A:F0 06    : BEQ 1022 : If "Yes"  Finish
                           If "No" then...
                       ---Continue---
101C:20 2D 10 : JSR 102D : Test and use input then
101F:4C 10 10 : JMP 1010 : back to read keyboard.

                       ---Finish---
1022:60       : RTS      : Exit back to Basic.
         ---end---


[ JSR 102D ]------------[ CONTROL SELECT ]----------

       ----start----
102D:AD 01 10 : LDA 1001 : Fetch key input for test.

                       ---Test for "Zap" key---
1030:C9 5A    : CMP "Z"  : Key "Z" pressed ?
1032:D0 04    : BNE 1038 : If "No" skip to next test.
                           If "Yes" then
1034:20 E1 FA : JSR FAE1 :  use Oric's "Zap" routine.
1037:60       : RTS      : Return to key input again.

                       ---Test for "Shoot" key---
1038:C9 53    : CMP "S"  : Key "S" pressed ?
103A:D0 04    : BNE 1040 : If "No" skip to next test.
                           If "Yes" then
103C:20 B5 FA : JSR FAB5 :  use Oric's "Shoot".
103F:60       : RTS      : Return to key input again.

                       ---Test for "Explode" key---
1040:C9 45    : CMP "E"  : Key "E" pressed ?
1042:D0 04    : BNE 1047 : If "No" skip to next test.
                           If "Yes" then
1044:20 CB FA : JSR FACB :  use Oric's "Explode".
1047:60       : RTS      : Return to key input again.

                       ---Test for "Ping" key---
1048:C9 50    : CMP "P"  : Key "P" pressed ?
104A:D0 03    : BNE 1047 : If "No" Return via "RTS".
                           If "Yes" then
104C:20 9F FA : JSR FA9F :  use Oric's "Ping".
104F:60       : RTS      : Return to key input again.
         ---end---
```

If the Accumulator does contain the correct code (5A for Key "Z"), the "Flag" is set to "1" and as a result the BNE instruction is disabled. Any other code tested, makes the BNE active instead.

If the BNE instruction is active, it will pass the operation on to the next test at 1038, which is for the key "S" (SHOOT).

However if the BNE is disabled (because key "Z" was pressed), the next instruction at 1034 will be used to produce the required "ZAP" effect and this will be followed by a return to the Keyboard Routine for another key input. The "RTS" at 1037 provides for that return to the keyboard input, via instruction 101F.

By the way, note that up to now, all the RTS instructions we have used, returned us to Basic. However, in this case, the "Control Select" routine was not called from Basic, but from instruction 101C in the "Keyboard" routine. As a result of this, we are now returned to the instruction 101F in that routine. That is the way the JSR and RTS instructions operate. The function of an RTS instruction is to return the program operation to the place that the current routine was called from.

Now if you look at the rest of the routine, you will see that it consists of three more sets of the same mini-routine we see in instructions 1030 to 1037. Each set of four instructions, tests for it's own specific key and calls up it's own particular operation if that key is found. Each CMP/BNE pair in effect passes the key code on from one test to the next, until it is either identified and used or it reaches the end of the routine, where it is discarded.

Note that the last BNE instruction in the routine must be shortened by setting it to D0 03 so that it goes to the last RTS. This ensures that if the key is not recognised, the routine will go back and get another key input. All the other BNE instructions are set to D0 04 to reach the next test in line.

Finally we can look at the very first instruction at 102D. This is used just to make sure that the Accumulator will always have a copy of the key code input at the start of the Control Select routine, even if the Accumulator has been overwritten.

So what's it all for ?
---------- This is a useful program, because it can be used to start up and control any software. Most software needs something to do that job Setting up for other software is just a matter of changing the key tested and the action which that key calls up.

Changing any key in the Control Select routine is a doddle. For example, you might prefer "B" for BANG instead of "S" for SHOOT. The code for "B" is 42, so change the CMP operand in location 1039 to 42 and key "B" will call the shots. In just the same way, you can change the action. As a simple example you could try out some of the other Oric routines or just make all four keys produce PING by changing all the "JSR" addresses to FA9F. Later, we can use the Control Select routine to control our own software.

Four keys for commands may not seem much. However it is easy to produce a longer version of the Control Select routine. This is how it is done.

I write a short a short version like the above and copy it again and again through memory to produce a routine for as many keys as I need. All the key tests are set to one code (one that is rarely if ever used) and each of these are made to call up a single harmless operation. In effect, this produces a ready made set of "blanks", for as many keys as you like.

Writing a routine like this takes all of five minutes. A key code of "00" and a call to PING, would be OK for all of the "blank" key tests. One important point The last BNE instruction should be set to D0 03 so it will access the last RTS instruction, all the rest are set to D0 04 for the next key test in line. Once this has been done, it can be saved and then used when writing new software.

As each operation is written it can be fitted into one of the Control Select routine "blanks", by putting it's address into the "JSR" instruction and then linking it together with a suitable key code. This is the way to build up to a large piece of useful software, by doing it one piece at a time in easy stages.

One for the Critics
---------- The experts will tell you how this routine could be more efficent, but is it really worth saving a few millionths of a second and a few bytes, if the program becomes more difficult to write and understand.
My own feeling is that machine code is very fast and takes little space when compared to other languages. If you can make your programming a lot easier by adding an extra instruction or a few bytes, here and there, it is well worth it

A good example of this is instruction 102D. Control Select can work without this instruction. However, when writing software, you may well add other routines between fetching a key and actually testing for it. Those later additions might cause problems by changing the Accumulator contents before "Control Select" is used. Instruction 102D will insure against this problem.

Likewise the gap (1023 - 102C) between the two routines will allow you some room to extend the "Fetch Item from Keyboard" routine if you need to later.

As we progress, you will find that there are a lot of things we can do to make programming easier. Even professional programmers rarely get software working first time (some never, ever !!). A very clever and compact piece software with bugs in it, is a sure way of boosting the sales of hair restorer and aspirin. It is best to get the software working first, then try the clever stuff later.

Thats how to start up.....next time, how to change the IBM's radio valves !

KRYSTAL WORLDS

KRYSTAL 5 - Inventory should read - 1) WAND ** , 2) HELMET (worn), 3) AMULET (worn), 4) LAMP **, 5) BRICK, 6) RUBY.
Items marked ** are not needed.
Tie rope to grapnel hook - insert sphere in socket. To get rope:- take parchment (read) - throw rope to ledge. Wake by barrier - insert brick.

Well,thanks to Peter Thornburn and that's Robert Cook's adventure solved. On the reverse of this page you will find Peter's maps tarted up by Paul Baker.

==========

GRANDAD

There now follows a party political broadcast on behalf of the BAKER Party.
Only kidding Paul. Here in all it's glory is Paul's introduction to his first adventure.

GRANDAD was written by me especially for O.U.M and it's members. This is no ordinary adventure;this is one of the new breed of adventure games for your ORIC. The game is 100% machine code,(written on the QUILL),so you don't have an infuriating wait after each command. Over 50 locatioms and 40 + objects,which you must find to achieve you goal,which is to help ERIC find the hidden treasure. If you need help on any section,you can write to the author's favourite magazine - O.U.M. for maps; but please don't phone Dave or myself as there aren't enough hours in the week for that. If you do,we will be forced to send the boys round or even worse, put the dreaded "CURSE OF THE MCP 40" on you,dreaded through the ages for being the main cause of pestilence.woh,doom and wobbly letters.

In this game,you play the part of Eric Dint,a 12 year old boy, visiting Grandad's manor in Henley. Grandad is the main character (as well as Einstein) - so a little more about him. Grandad is a crazy old character and is not happy unless tinkering with his latest invention. He is especially interested in Matter Transfer (at it's atomic level) and computer add-ons,which you will encounter in the adventure.
What will he have made this year? Mother says that he has been spending rather a lot of his rapidly diminishing wealth with a massive computer hardware company.

GRANDAD ALSO INCLUDES A SPECIALLY DESIGNED "CINEMA" STYLE FONT, WHICH IMPROVES THE WHOLE IMAGE OF THE GAME.

==========

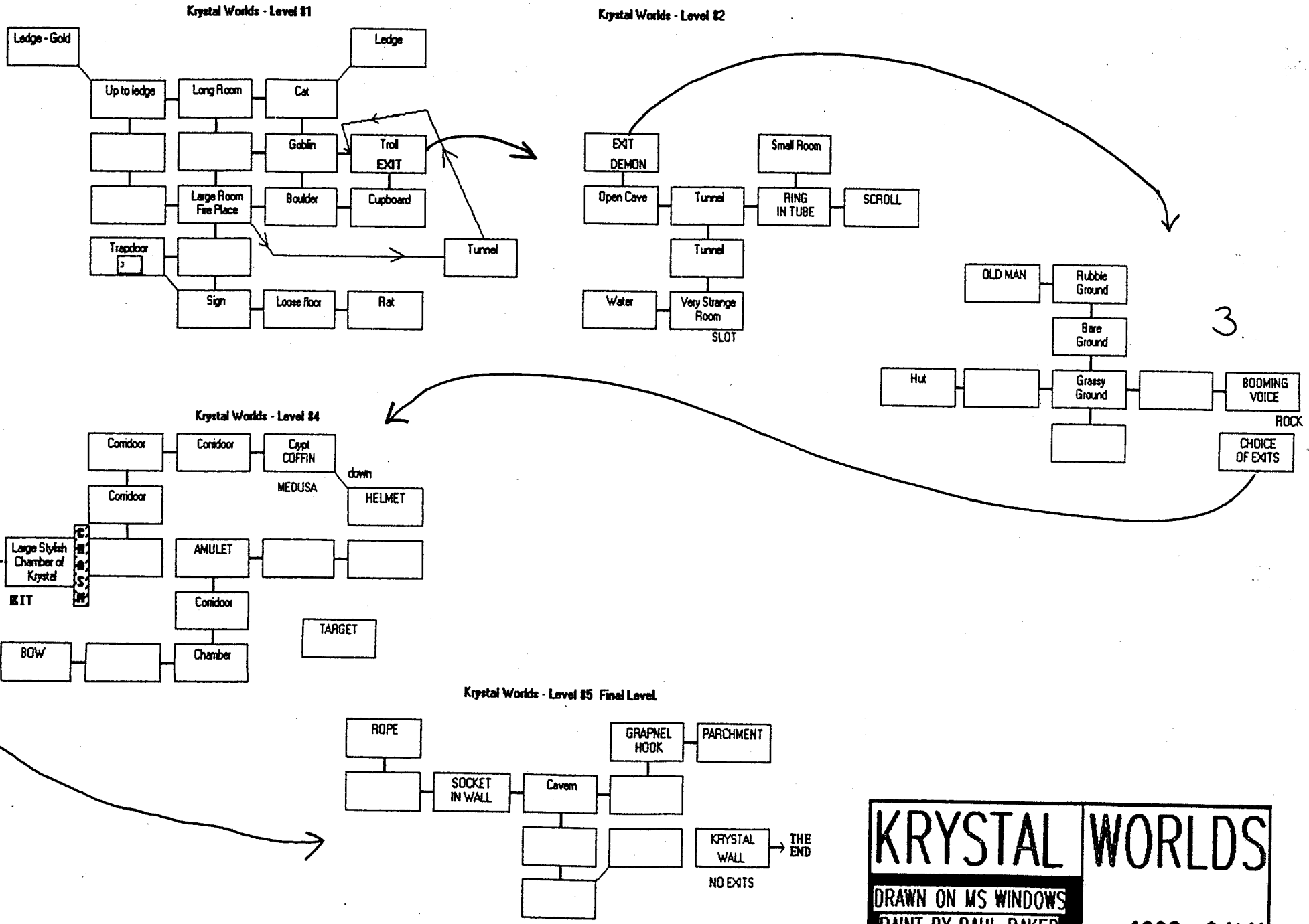
GRENDEL

Direct from Alistair,author of GRENDEL,and 'in his own write',so to speak, some POKES.
I've worked out the Pokes to sort out the 'impossible' screen on Grendel for Judy Simms and the rest.
The problem on the screen is the nasty spiked 'face' lying on the ground, which makes it a bit tricky to jump off the lamp post safely.
POKE#43F7,25 makes it a little easier,by shifting the face a touch to the right. Alternately, POKE#43F9,19 makes it a lot easier,by transforming the 'face' into a harmless 'bush'. I hope these help !

Thanks Ali. By the way folks, over the Xmas holidays He and his brother got rather hooked on TETRIS and FRENCH LUDO.

==========

(13.)



KRYSTAL WORLDS

DRAWN ON MS WINDOWS

PAINT BY PAUL BAKER

1992 O.U.M

D I S K C O N T R O L L E R

INTRODUCTION FROM THE EDITOR

Readers may remember that a few issues ago that I had a letter from one Harry Peters in Holland,who had made his V1.5 ROM and with the aid of a PC had designed a disc controller for the Atmos.
Harry tells me that the Rom is very compatible with Atmos programs,but that 300 bytes are different,and sometimes programs get the wrong information when testing for an Atmos or Oric 1. The solution is to patch the program so that it tests another byte in the Rom.
With regard to his disc controller for the Atmos; I now let Harry tell you in his own words.

THE DISC CONTROLLER

Firstly, I am not an expert in electronics - I am a chemist by profession. I cannot design even the smallest circuit with I.C's. But when there is a design,it is possible for me to draw a printed circuit board. My first project with the Oric was a board for a parallel to serial converter for my serial printer with an RS232 interface that I had at the time. It worked fine and that encouraged me ,via the construction of a ROM cartridge,to construct a disc controller.

Someone sent me the original circuit diagram of the Oric controller. On my PC with the program SMARTWORK, I designed a double-layered printed circuit board. It took many nights to master the program and to finish the drawings. The length of time was also due to bad readability of the Oric diagram. I followed the diagram,except in the cases whereby changing the multiple connections to IC ports made the layout easier.

With another program (LASERPLOTTER); I made the final drawings ; a copy of which are now with O.U.M. They are of good quality to produce films, which are used to expose photosensitive material on copper layered boards.

But this way was to difficult for me and beyond my possibilities. With a laserprinter I made copies on overhead transparencies,which were primed with 0.5% siliconoil in hexane and heated for 2-3 minutes at 100 degrees C. When the copper layer is well cleaned an you put the printed side of the overhead sheet on the copper layer; it is possible to transfer the pattern on the sheet by heating it with an iron. When you do this twice, the pattern is thick and tight enough to prevent dissolution of the underlying copper in an etch bath. I etched with 2M (5%) hydrochloric acid and hydrogenperoxyde. The peroxyde is added in small portions as a 5% solution until enough copper is etched away. After cleaning the board and drilling of 681 holes (Terrific),the board can be soldered. The black stripes in the component layout are decoupling IC's of 47 n.

Sometimes modifications to the computer are necessary to make the controller really work with it. When you have an Oric with two eproms 2764,you have to cut the connection of IC10 pin 27-28 and connect IC10 pin 27 to the ROMDIS signal IC9 pin 27.

The second modification is more serious and due to the weakness of the PH12 signal of the Oric.

Let me try and explain what I heard from some real hardware 'freaks'. In the ULA the 12 mhz clock signal is divided by 4 & 3 to achieve a clock signal of 1 mhz. Due to the division by 3,the signal is 0.67 usec low and 0.33 usec high. Therefore in the Oric , a 6502A processor is used,which can handle signals under 0.5 usec duration.
Another shortcoming of the Oric is the high load of the 6502 PH12 signal by VIA 6522, 1x74LS04 and when a diskcontroller is plugged in also a 74LS244. This is too much for a signal that is designed to drive only one TTL gate. The solution is to buffer the signal at the 6502. This is done by cutting the wire from 6502 pin 39 and feed this to pin 1 of a 74LS14. Solder pins 2 and 3 of the 7414 together and connect pin 4 to the cutted PH12 line of the computer. Connect pin 7 of the 7414 to the 0 volt (pin21/1 6502) and pin 14 of the 7414 to the + 5 volt (pin 38/2 6502).

(CONTINUED OVERLEAF)

I think with this short explanation it might be possible to build your own controller. You need also a power supply with +5 and +12 Volt and a drive. I use a switching power supply from the dump rated at +5V 4A and +12V 2A. The Oric is also fed with the +5V via a separate connection to the print. In Holland it is now and then possible to buy a drive for 30 guilders (approx. 10 pound sterling) at the Hobby Computer Exhibition.

There is still one thing. The possibility of buying the IC's 1793 and 9216. In particular the 9216 is sometimes difficult to get,but when you can they are by far the most cost effective.
I altered the drawing for use with the more sophisticated chip - WD1772, which is easily obtainable.

I have not built it yet and so do not know if it will really work. You can delete then the IC's 9216,74393,1793,74123 with resistors an condensators and half of 7411 and substitute this all with one 1772. I am not sure if it will work because the 1772 command set has a software controllable write precompensation command. That option is not used in the Oric controller. With the 1793, write precompensation is done in hardware via pin 17-18. With the 1772 write precompensation,is controlled in type 2-3 commands with the P-bit = bit 1. When the Dos sets this bit always to 0,then I think it will work O.K.

Also you need a 2764 Eprom programmed with the boot routine;really a mini Dos. Contact me or one of the hardware freaks in England to get one. I hope this story will at least give you more understanding of the Oric computer/controller hardware when you are not willing to build a controller.

- HARRY PETERS, Marathon 28, 2924 XD Krimpen a/d IJssel, The Netherlands.

FOOTNOTE FROM THE EDITOR

I now have in my possession the diagrams from Harry. Six pages in total and therefore too much to put into O.U.M.
Included are a circuit diagram of the controller,a parts list and excellent diagrams of the solder and component sides of the PCB. If anyone would like copies of this; just send 4 x 24 p stamps. Overseas readers can instruct me to reduce their subscription by one issue.

In summing up, I would like to thank Harry for all the trouble he has gone to in translating his documentation into English.

ON THIS PAGE AND OVERLEAF, WE CONCLUDE THE LISTING FOR 'VISIORIC', WHICH IS A VERY NICE LITTLE 'MATCHING PAIRS' TYPE OF GAME.

```
1010 DATA2,2,15,16,48,19,19,31
1020 DATA0,0,60,2,25,26,2,62
1030 DATA0,1,2,4,8,63,31,15
1040 DATA60,48,40,36,34,63,62,60
1050 DATA1,7,31,63,0,0,2,1
1060 DATA48,60,62,63,32,32,32,0
1070 DATA32,16,6,4,31,15,39,31
1080 DATA0,0,0,6,56,48,36,36
1090 DATA1,31,31,1,1,1,1,1
1100 DATA32,60,63,32,32,32,32,32
1110 DATA1,3,3,17,24,12,14,3
1120 DATA48,24,24,49,35,39,46,56
1130 DATA14,17,32,32,32,24,6,1
1140 DATA28,34,1,1,1,6,24,32
1150 DATA31,16,16,31,16,16,16,16
1160 DATA62,12,12,62,0,0,0,0
1170 DATA15,27,63,62,31,13,6,3
1180 DATA60,54,63,31,62,44,24,48
1190 DATA0,28,54,43,54,28,0,0
1200 DATA0,0,0,63,5,0,0,0
1210 DATA28,15,0,0,0,0,0,0
1220 DATA39,62,32,32,16,32,16,32
1230 DATA16,23,63,63,63,31,50,12
1240 DATA31,18,62,62,62,63,18,12
1250 DATA62,63,63,31,31,63,63,62
1260 DATA31,63,63,62,62,63,63,31
1270 DATA0,1,3,7,15,31,0,0
1280 DATA0,32,48,56,60,62,0,0
1290 RETURN
1300 REM
1310 PRINT@HC,V2;C1$(P(HC,V2)):HC(CO)=HC:V2(CO)=V2
1320 PRINT@HC,V2+1;C1$(P(HC,V2))
1330 IFCO=2THENNB=NB+1:PLOT27,24,STR$(NB)+"  "
1340 IFCO=2ANDP(HC(1),V2(1))=P(HC(2),V2(2))THENCP=CP+1:CV=12-CP
1350 IFCO=2ANDP(HC(1),V2(1))=P(HC(2),V2(2))THENPLOT14,20,STR$(CV)+"  "
1360 IFCP=12THENZAP:WAIT20:GOSUB1680:GOTO1540
1370 IFCO=2ANDP(HC(1),V2(1))=P(HC(2),V2(2))THENZAP:CO=0:RETURN
1380 IFCO<2THENRETURNELSEFORX=1TO100:IFPEEK(#208)<>56ORPEEK(#400)<>191THEN1400
1390 GOSUB1420:NEXT
1400 PRINT@HC(1),V2(1);C2$:PRINT@HC(1),V2(1)+1;C2$:PRINT@HC(2),V2(2);C2$
1410 PRINT@HC(2),V2(2)+1;C2$:CO=0:RETURN
1420 REM MESURE TEMPS
1430 TE=NI-INT((65535-DEEK(#276))/100):IFTE<0THENTE=0:GOTO1470
1440 IFCT=TE+1ANDU-J>300THENCALL#FB14
1450 CT=TE:PLOT21,22,STR$(TE)+" SECONDES  ":RETURN
1460 REM FIN DU JEU
1470 PRINT@7,0;CHR$(140)+CHR$(129)+"    TERMINE !       "
1480 PRINT@HC,V1;"          "
1490 MUSIC1,2,1,0:MUSIC2,2,5,0:MUSIC3,2,8,0
```


Hi-scores
from: Graeme Burton.
Styx-194600
Ultra-27940

× E.T. - PHONE HOME ×

+ JUDY SIMMS NOW HAS A NEW TELEPHONE NUMBER. IT IS 021 454 6906 +

==========

ON THE LEFT HAND SIDE OF THE PAGE,WE CONCLUDE 'VISIORIC', WHILST ON THE RIGHT WE HAVE JOHN HURLEY'S LATEST RENDITION.

ONE OF JOHN'S HOBBIES IS AN INTEREST IN MOTOR CYCLES OF THE OLDER TYPE,AND ENTERING THE VARIOUS CLASSIC RUNS AND LOCAL SHOWS OR EVENTS. HIS BIKE IS A 'TRIUMPH TIGER CUB' model T20, WHICH HE USES DAILY TO & FROM WORK. SO FOR OTHER BIKE USERS IN THE GROUP; JOHN'S PROGRAM IS A FORMULA FOR ESTABILISHING THE CORRECT TIMING AT THE POINTS USING THE PISTON TRAVEL AS THE METHOD.

1491 REM *** V I S I O R I C ***

```
1500 PLAY1,0,1,4000:WAIT100
1510 PLAY2,0,1,4000:WAIT100
1520 PLAY4,0,1,10000:WAIT200
1530 PLAY7,0,1,10000:WAIT500
1540 DE=0:GOSUB670:WAIT400:SHOOT
1550 FORX=48802TO49120STEP40:POKEX,20:NEXT
1560 PLOT5,20,"                    "
1570 PLOT5,22,"                    "
1580 PLOT5,24,"                    "
1590 IFNB=0THENSC=0:GOTO1610
1600 SC=CP*1E5/((NI-TE)*(2*NB)):SC=INT(SC)
1610 PRINT@8,20;CHR$(131)+"VOTRE SCORE:"+STR$(SC)+" POINTS"
1620 PRINT@10,23;CHR$(142)+CHR$(128)+"Voulez-vous rejouer ?"
1630 PRINT@10,24;CHR$(142)+CHR$(128)+"Voulez-vous rejouer ?"
1640 GETR$:IFR$<>"N"ANDR$<>"O"THEN1640
1650 FORX=48000TO48039:POKEX,0:NEXT
1660 IFR$="O"THENGOSUB1780:GOTO320
1670 CALL583
1680 REM MUSIC
1690 PRINT@15,0;CHR$(132)+"BRAVO !"
1700 MUSIC1,2,1,0:MUSIC2,2,6,0:MUSIC3,2,8,0
1710 FORT=1TO2:PLAY1,0,1,1000:WAIT20:NEXTT
1720 PLAY1,0,1,2500:WAIT30
1730 FORT=1TO2:PLAY3,0,1,2000:WAIT50:NEXT
1740 FORT=1TO2:PLAY4,0,1,2000:WAIT50:NEXTT
1750 MUSIC2,1,1,0:MUSIC1,3,1,0:PLAY3,0,1,4000:WAIT80:MUSIC1,2,10,0
1760 PLAY1,0,1,1000:WAIT20:MUSIC2,2,1,0:MUSIC1,2,6,0:PLAY3,0,1,6000:WAIT200
1770 RETURN
1780 REM INITIALISATION
1790 CLS:FORX=#BBA4TO#BBA7:POKEX,32:NEXT:PAPER0:INK1
1800 POKE618,10
1810 PRINT@8,10;CHR$(140)+"Quelques instants S.V.P"
1820 RETURN
1830 REM POIGNEE
1840 DATA#48,#AD,#01,#03,#48,#AD,#03,#03,#48,#A9,#C0,#8D,#03,#03,#A9,#80,#8D
1850 DATA#0F,#03,#AD,#0F,#03,#8D,#00,#04,#A9,#40,#8D,#0F,#03,#AD,#01,#03,#8D
1860 DATA#01,#04,#68,#8D,#03,#03,#68,#8D,#01,#03,#68,#4C,#22,#EE
1870 FORM=#402TO#431:READA:POKEM,A:NEXT:DOKE#0245,#402
1880 RETURN
```

PISTON

```
50 CLS:PAPER 0:INK 3
60 POKE #26A,10
70 GOSUB 200
80 CLS:PRINT:PRINT
100 INPUT "Stroke in mm..... ";A
110 PRINT
112 INPUT "Degrees B.T.D.C.. ";B
115 C=A/2
120 E=COS(B/57.2957796)
125 D=C-(C*E)
130 PRINT
135 PRINT"The distance before Top Dead Centre  will be :- "
137 PRINT:PRINT D;"mm"
145 F=D/25.4
147 PRINT:PRINT SPC(8)"OR"
150 PRINT:PRINT F;"in Thous of an inch..."
155 PRINT:PRINT:PRINT:PRINT CHR$(129)SPC(2)"Press any key to Re-Calculat
160 GET A$:GOTO 80
200 PRINT:PRINTCHR$(129)TAB(14)"POINTS SETTING"
205 PRINT:PRINT
210 PRINT:PRINT" A simple formulae for establishing"
215 PRINT"the Piston Travel before Top Dead"
220 PRINT"Centre. To enable the points setting"
225 PRINT"to be carried out without having to"
230 PRINT"remove any side covers on a motor "
235 PRINT"cycle to expose the Flywheel or the"
240 PRINT"magneto assy. This may be used when"
245 PRINT"only the Degrees are quoted in the"
250 PRINT"handbook or manual."
255 PRINT:PRINT" Piston Setting can then be carried"
260 PRINT"out via the Spark Plug hole using a"
265 PRINT"depth gauge type of instrument,or for"
270 PRINT"precision setting, the cylinder head"
275 PRINT"could be removed to expose the barel"
280 PRINT"and provide a flat surface to use a"
285 PRINT"Depth Micrometer."
290 PRINT:PRINTCHR$(130)"Press ANY key once ready to start"
295 GET A$:RETURN
```

O . U . M FOR FUN

* * F I N A N C E S * *

As previously promised , I now report on the financial situation with regard to 'ORIC USER MONTHLY'.
I will not waste space with columns and columns of figures. What you need to know is the real nitty gritty - the bottom line. That is not a healthy one.

SUMMARY OF ACCOUNTS

At FEB. 21st 1992 there was 228.23 pounds in the O.U.M account, compared to 182.12 exactly one year earlier.
Over the year monies were spent on advertising/mailshots,BACC and of course the magazine itself,stationary and postage. Income was obtained from a sweepstake,a raffle and subscriptions. The bulk of the raffle profit was used to fund 50% of the new O.U.M printer;with myself putting up the other half.
The last issue cost 89 pence per unit to despatch (photocopying + envelope + postage). We glean approx. 91 pence per issue from subscribers, as most pay year annually. In theory this leaves 2 pence per issue over. In fact this is generally not the case. The cost of producing O.U.M is normally above 89 pence. There have been a few freakish months recently - for full explanation,please drop me a line.
With 666 issues owed to subscribers;it means we have 34 pence to spend per issue. For that you get a cover (free from Jon), a whole page,an envelope and a stamp. It looks serious. perhaps it is ,perhaps it isn't. It's all part of life's rich tapestry.
It is only now that I have lloked in depth at the accounts. One tends to assume that if there is a couple of hundred pound in the bank;then everything is rosy. There is no panic; unless all readers ask for their subscription money back. I DO NOT INTEND TO CUT BACK ON PAGES PER ISSUES. In fact I want to make it bigger.

THE SOLUTIONS

A) ALL SUBSCRIPTION RATES TO INCREASE.
With effect from April 1st.1992, the rates for O.U.M are as follows:

British Isles:	5 issues	for 6.00
	12 issues	for 13.00
Eire/Europe:	6 issues	for 10.00
	12 issues	for 16.00

This represents nearly a 20% increase, BUT in the last 2 years there have been NO increases in price and yet a 100% increase in size of O.U.M. I do not think it is too much to ask.

B) RAFFLE - perhaps Rob Kimberley or even someone else will offer to run a raffle at the next Oric meet. The last one raised 80 pound,but due to the theft of my printer was used to help me buy a new one.

C) MAILSHOTS - instead of O.U.M funding our search for new members; you will be asked to do it. FULL DETAILS in this or the next issue. The last year has resulted in making new contact with users who have contributed significantly to O.U.M. We need new blood all the time.

D) SECOND-hand-software: There is a lot which is now the property of O. U.M. Please buy something. It is no good people donating it to the club if it is going to just rot in my garage.
E) YOUR IDEAS - are most welcome, perhaps a '100' club, OUM Bingo 1!!. Let me know.

==========

CHANGE OF ADDRESS

Our publicity officer (Brian Kidd) has moved . Luckily he has stayed in Wales.
Brian can now be found at 'THE DOG & DUCK' instead of 'THE KING'S HEAD'.

Alternately you can write to him at:
23 HOWARD CLOSE, NEWPORT, GWENT. NP9 9RF

LETTERS TO THE EDITOR

HI DAVE,
this is to update my information on your contact list. Please drop 'CESAR ANDRE' as anyone knowing me knows that 'JEAN' stands for a male Oric owner and not a pair of trousers.
My set- up has been upgraded:
2 TELESTRATS (2x3" + 1X3" + 3.5" Epson drive)
2 Atmos (2xMicrodiscs + 1 Cumana interface with 1 Amstrad DD1 - eek!)
The original Psu's have been binned and replaced by IBM. PC - PSU's, thus solving the booting problems.
MT-140, MCP 40 printers and a Canon BJ10 printer (fantastique) and a Voice synthesiser.
MY INTERESTS : - replacing Oric PSU's with IBM ones and adding 3.5" drives to Atmos's. Collecting software like POSTMAN SAM and A VIEW TO A KILL III ! and putting them onto disc.
I also enjoy doing anything that permits me to burn my fingers with my soldering iron; e.g making PASE compatible joystick interfaces.
BEST OF LUCK - JEAN BOILEAU (Bondy, FRANCE)

DEAR JEAN,
great to hear from you. It appears that you are the only member who has upgraded in the last year or else others couldn't find a pen and paper.
Do you realise that this is probably the first French letter to OUM - cheers Mate!
I bet you haven't solved View to a Kill III yet. From what I remember when in Paris, it was an impossible maze.

- DAVE

==========

DEAR DAVE,
I am pleased to inform you that the wife bought me a Citizen120D+ printer for Xmas.. I followed the instuctions from OUM for making up th lead and found it straigtforward. Carry on with the good work.
I would like through the pages of OUM to thank all those who helped with getting me out of trouble whilst learning to use my set,as I am only a novice at the game.
- TIM PHOENIX (Cleveland)

DEAR TIM,
I am glad that the articles and our members are being helpful. I know that David Wilkin helped out with advice on 'AUTHOR'.

==========

DEAR DAVE,
It is so nice to hear that some people have fun playing TETRIX and that it has not been in vain.
I now have a 3.5" twin drive connected to the Oric (80 track,double sided). It had once been connected to a Speccy,but I thought it deserved something better.
- Andre Widhani (Hamburg)

Dear Andre,
although yo won't become a rich man buy having written TETRIX; you will at least have enough money (almost) to give you free subscriptions to OUM and the CEO.
- DAVE

==========

DEAR DAVE,
a selection of my hi-scores for gamesters to beat:

PASTA BLASTA - 17360
OPERATION GREMLIN - 22617
KRILLYS - 28290
ZEBBIE - 211470
KRIKATILE WALTZ - 10025
TWO GUN TURTLE - 5890
INSECT INSANITY - 25050

- Graeme Burton

Dear Graeme,
SHOW OFF and wait untill I tell the wife what YOUR score on STYX was. She will be really peeved.
- DAVE

ORIC Enthusiasts (OUM 55)

INTRODUCTION

This month sees two more pages, each, on Geoff Phillips' book and how to access disc files from BASIC. Thanks must go to Bernard Grone for his information on controlling the step rate on ORIC and SEDORIC DOSs. I do not know if B & H Computers still have the 3" disc drives but if anyone is interested I will find out. The 3" disc drives would make a reasonable 2nd drive, but users may prefer the faster Hitachi, Panasonic drives. Bernard, if you are reading this, how about writing an article or two on the WD1793 chip and how to control it? I for one would welcome this information as it is one area where I haven't found too much information.

This brings me onto another point. After a series of articles in OUM and CEO and also after talking with Trevor Shaw and Ray McLaughlin it is apparent that a good development for the ORIC would be to provide the ability to transfer files from the ORIC to the IBM PC or compatible. To this extent, Trevor said that he would investigate the possiblity of accomplishing this through a serial interface, while Ray is to study the possibilities of a SEDORIC utility program to read and write PC discs. Since the IBM standard represents the link to other computer systems such as the ARCHIMEDES, AMIGA and ATARI, it would open up the routes to transfer ORIC files to all computers that either emulate or access PC standards. Do not hold your breath because I know everyone concerned is busy but enthusiasm may win the day.

Although I did receive any response to an earlier request for continued development of the ORIC, I still think that a few enthusiasts would like to see some changes. To that extent I intend to write a specification for what the developments should contain, in order to maintain a standard approach and one which could be advantageous to all. After brief discussions, it would most definitely involve providing better integration of the DOS with the machine operating system; provision for ROM/EPROM modules; new screen editor and a more powerful BASIC interpreter. I would like to circulate this specification to interested parties in order to gain an overall concensus of what the development should contain, but I am particular looking for experienced users of the ORIC who would be interested in sharing the development work. Write to me if you are interested.

BOOKS

Continuing with the series on books available from my price list. The next title is "Easy Programming for the ORIC-1" by Ian Stewart and Robin Jones.

Again, this is another book first published in 1983 but instead of providing an alternative to the ORIC-1 manual it was intended to complement it by provided a more thorough introduction to the machine and ORIC BASIC. Consequently it does not constitute a reference book and is actually written in a tutorial style with numerous sections, each introducing a topic, giving an introduction, a small number of small projects with answers at the end. The style the authors chose is friendly and humorous, with quite number of cartoons in the book, and includes some helpful advice on how not to aggravate the not-so-computer-literate of the family.

The book contains 162 pages with an Introduction, 31 sections, 2 appendices and 2 indices. The sections deal with UP and RUNNING; THE KEYBOARD; DIRECT COMMANDS; PROGRAMS; TEXT DISPLAY; VARIABLES; INPUTS; BRANCHING; PLOT POSITIONS; LOOPING, DEBUGGING I; SOUND and COLOUR; USING A CASSETTE RECORDER; DEBUGGING II; RANDOM NUMBERS; STRINGS; SUBSTRINGS; DEBUGGING III; THE ART OF COARSE GRAPHICS; MOVING GRAPHICS, ARRAYS, MUSIC and HARMONY; DEBUGGING IV; PEEK and POKE; HIGH-RESOLUTION GRAPHICS; COLOUR IN HIGH-RESOLUTION; DEBUGGING V; USER-DEFINED CHARACTERS; SUBROUTINES; DEBUGGING VI and WHAT NEXT? Appendix 1 is on Control Characters while Appendix 2 covers the Alternate Character Set. There is a Program Index and a Commands and Symbols Index.

If you are new to the ORIC and its BASIC then this book is intended for you. Again, it does not cover those enhancements in the V1.1 ROM but it would prove equally beneficial to the user who has just acquired an ATMOS and wants to complement the fine ATMOS manual. There are number of small listings but the book includes four programs called "Imprisoned ORIC", "Fruit Machine", "Cyril the Squirrel" and "Shockwaves". When using these programs on the ATMOS, remember to add 1 to the 'x' position in the PLOT command so that the PAPER attributes are not overwritten unintentionally.

In conclusion, the book achieves its objectives in that it helps to reinforce the principles of using an ORIC computer and mastering its in-built BASIC programming language. It is available at £1.00 , inclusive of postage and packaging, (original price was £5.95).

OE/0

ORIC ATMOS and ORIC-1 GRAPHICS & MACHINE CODE TECHNIQUES

Chapter 3 - BASIC (continued)

3.2 Storing machine code (continued)

4 Since the alternate character set is rarely used, the entire area between #B800 and #BB7F is available for a machine code program. This area of RAM is ideal for facilities like Renumber.

5 Another 'hidden' area lies between #BFE0 and #BFFF. This area will only be overwritten if HIMEM is incorrectly set, and survives the commands HIRES, TEXT and the reset button.

3.3 Type of machine code program When you write a program that is all in machine code you do not need to worry about interfacing with BASIC. If your program calls the BASIC ROM for certain functions you should keep clear of the same area of RAM that the particular sub-routine uses. For instance, if using the MUSIC command routines, keep away from the parameters area #2E0 to #2EF. Since a machine code program can be made to autorun at the start address of the load, it makes sense to use this feature and make your program start at the earliest address. (**N.B.** Disc users are not constrained in any way so the executable address may be anywhere within the program. Adapting a machine code program where the executable address is not at the start, for cassette autostart is easily done by adding a JMP statement, to the actual executable location, at the beginning of the program.) If you are using an Assembler program, such as ORICMON, you will also have to avoid the area of RAM used by that program. A common type of machine code program is used when a BASIC program needs an extra facility, or perhaps a machine code sub-routine is used to speed up part of the programming this case the BASIC program will often use DATA statements in order to set up the machine code. A more efficient way, for larger sections of code, is to load in a separate machine code file from tape or disc. Another method is to put the machine code after the BASIC code and modify the #9C pointer before saving to encompass the machine code. The first instruction in the program should reset the pointers #9C, #9E and #A0 back to the end of the BASIC program. For example:

BASIC program #501-#1F00
M/C program #2800-#2E00

Before saving, DOKE#9C,#2E00; then in the program use DOKE#9C,#1F02:CLEAR

An example of a BASIC program creating a machine code sub-routine can be found in section 3.4. The third type of machine code program occurs where a BASIC program is modified. Normally such a routine will be loaded separately from the BASIC, although you must remember to reset the #9C pointer on V1.0 machines - this can often be done by the machine code routine itself.

3.4 Creating a machine code program Nearly all the programs in this book have been listed in terms of the assembly mnemonics and the actual machine code. In order to set up the programs you are best advised to use a machine code monitor/assembler package. If such a facility is not available, you can quite easily use a short BASIC program to read in the machine code. Program 3.1 is an example of a program to read in a short section of code by using DATA statements. The program itself is very useful, as it totally disables the use of Ctrl-C. This works by testing for ASCII code 3 in a routine that is patched into the slow interrupt link.

```
5 REM   Ignore Ctrl-C
10 FOR I=#BFE0 TO #BFEE:READ D:POKE I,D:NEXT
20 DATA #8,#48,#AD,#DF,#2,#C9,#83,#D0,#3,#CE,#DF,#2,#68,#28,#40
30 IF PEEK(#D000)=166 THEN DOKE#231,#BFE0:POKE#230,76:    REM: For ORIC-1
40 IF PEEK(#D000)<>166 THEN DOKE#24B,#BFE0:POKE#24A,76:     REM for ATMOS
```

Program 3.1 Disable Ctrl-C

3.5 Calling a machine code routine A machine code program which is completely self-contained can be automatically run by using the AUTO command. Alternatively, a CALL can be used to start the program off. Where a BASIC program calls a machine code sub-routine, CALL is often used. If a CALL is to return to BASIC the sub-routine must end with the RTS (#60) instruction. Do not worry about saving registers when writing such a sub-routine. CALL is also often useful when entering add-on sub-routines, such as 'Renumber', when it is used as an immediate command. In addition to CALL NN, there are several alternatives:

1 USR and & functions
2 ! - the extension command

From the point of view of a machine code sub-routine, CALL NN is much the same as ! (**N.B.** parameters can be appended to the CALL statement, like CALL#9000,"DATA",4,3:REM provide parameters, and dealt with by the machine code routine), and USR(X) is identical to &(X). One difference is in the setting up. For the extension command '!' you DOKE the start address into #2F5, and for '&' you DOKE the address into #2FC. The USR facility uses DEF USR in order to set up the start address. (**N.B.** SEDORIC provides enhancements to these commands for the machine code programmer.) The difference between '&' and '!' (or USR and CALL) is that '&' is a function and returns a value; the ! command can only take in values. The rest of this chapter will only deal with & and !, although the same considerations apply for CALL and USR.

3.6 Passing information to machine code routines The most common method of passing small amounts of data to a machine code routine is with the DOKE and POKE commands. For small data areas, such as for addresses, use the area #0 to #B in page 0 (**N.B.** SEDORIC users will have to wary of this as the DOS makes use of this area in some of its commands) Chapter 5 will help you in determining other areas of memory available. The ! and & keywords can both take parameters, e.g. &(A1*3), and this will be explained in section 3.11. A machine code routine could read a BASIC variable, but this would involve quite a bit if searching and conversion.

3.7 Patching into BASIC Although BASIC is in unalterable ROM (**N.B.** unless you use the RAM versions of V1.0 or V1.1 ROMs supplied with SEDORIC DOS) there are several cases where it jumps out to an area of RAM. The reasons for doing this are :-

1 It lets programmers patch in extra facilities.
2 It allows for add-ons, such as disks.
3 It can be more efficient to write some instructions in page 0.

Each of the areas has been listed below, with the address for V1.0 ROMs given in brackets:

At **#1A** - a jump vector to the routine that prints 'Ready'. By changing this jump to your own routine it is possible to trap errors or prohibit Ctrl-C. See the ON-ERROR facility of Chapter 8. (**N.B.** For example to sound a PING instead of printing 'Ready' DOKE#1B,#FA9F)

At **#E2** lies a very important sub-routine. (**N.B.** Geoff didn't really explain what the routine did, so I will. The short routine at #E2 is used by the BASIC interpreter to read the next character, in a BASIC program, to be presented to it [if it is a space then it is skipped]. It is important to understand the workings of this routine when passing parameters with the ! command.) At #E2 the address at #E9,#EA is incremented. (**N.B.** These two locations represent the current executable address within a BASIC program.) Then at #E8, the contents of the address at #E9,#EA are loaded. (**N.B.** So entering the routine at #E8 allows you to read the current character, and entering at #E2 reads the next character.) This provides a very fast sub-routine for reading reading characters from the program. After getting the next character, the routine jumps back into ROM. It is a very simple matter to alter the routine at #E2 in order to jump to your own sub-routine. By doing this, you can look for special instructions (perhaps 'IMPLODE' and 'PONG'!). The important consideration is that you jump back into ROM as though nothing had happened - remember to save all the registers.

#244(#228) is the address of the 'fast' interrupt jump. By altering the jump address at #245,6(#229,A) you can provide your own interrupt handler.

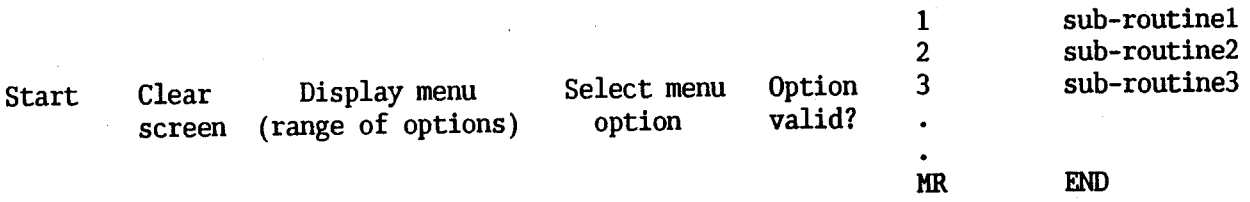
Next month -- Continuing Chapter 3 and the completion of 3.7

DISC FILE HANDLING TECHNIQUES USING BASIC

UNSORTED SEQUENTIAL FILES (continued)

This month I shall introduce the main routine for the development of the program PHONE and then start on each individual sub-routine one by one.

In data processing there is nearly always a number of operations to be performed on the data set in question, such as creation; amendment and deletion of records. In organising the user interface for such programs it is normal to adopt a menu offering a list of choices for the operations that may be performed by the user on the data. This program will be no exception to this so let me describe the menu techniques that I will adopt. The following hybrid flow/schematic diagram will serve to illustrate the design.



The individual operations, and, therefore, menu options can be configured to have a range from 1 to MR where MR will be a variable defining the range of menu options. A specific user selection can be stored in a variable named MS, which, to be valid, must lie between 1 and MS, inclusive. If the selection is invalid then the selection is requested again. If the selection is valid then you need to access the appropriate sub-routine to carry out the required operation. The actual construct for accessing the menu is provided, in modern computer languages, by the CASE statement. Now, the CASE statement may be considered to be a neater presentation of multiple IF..THEN..ELSE statements. The following example should help demonstrate the point.

```
IF MS=1 THEN GOSUB 1000 ELSE
        IF MS=2 THEN GOSUB 2000 ELSE
                IF MS=3 THEN GOSUB 3000 ELSE etc.
```

```
CASE equivalent:
CASE OF MS
  :1   sub-routine1
  :2   sub-routine2
  :3   sub-routine3
  .
  .
  :MR  stop
END CASE OF MS
```

In ORIC BASIC, the nearest (but still appropriate) command is the ON <variable> GOSUB, so we shall use that for our menu selection.

```
ON MS GOSUB 1000, 2000, 3000, 9999
```

so now the main routine will be able to access the appropriate the sub-routine and then return to the menu. For the sake of clarity each sub-routine begins with a round thousand number, but line 9999 will contain the END command.

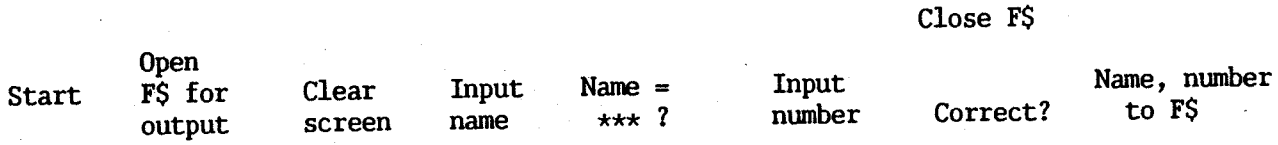
So moving onto to the BASIC code to produce the menu. The 6 sub-routines are each represented, for the moment, by a REM command and will be developed later. This allows the program to be produced in a modular fashion.

```
10 REM ***** PHONE *****
20 F$="PFILE.DAT"
30 CLS
40 PRINT TAB(10);"PHONE DIRECTORY"
50 PRINT:PRINT
60 PRINT TAB(10);"1 = Create directory"
70 PRINT TAB(10);"2 = Print directory"
80 PRINT TAB(10);"3 = Add new name and number"
90 PRINT TAB(10);"4 = Search for name"
```

```
100 PRINT TAB(10);"5 = Delete name and number"
110 PRINT TAB(10);"6 = Copy directory"
120 PRINT TAB(10);"7 = End"
130 PRINT:PRINT
140 PRINT TAB(10);"Select number, please"
150 INPUT MS
160 IF MS<1 OR MS>7 THEN 140
170 ON MS GOSUB 1000,2000,3000,4000,5000,6000,9999
180 GOTO 30
1000 REM ----- CREATE PFILE -----
1010 RETURN
2000 REM ----- READ AND PRINT FILE -----
2010 RETURN
3000 REM ----- ADD NEW NAMES AND NUMBERS -----
3010 RETURN
4000 REM ----- SEARCH FOR NAME -----
4010 RETURN
5000 REM ----- DELETE NAME AND NUMBER -----
5010 RETURN
6000 REM ----- COPY PFILE TO PFILE2 -----
6010 RETURN
9999 END
```

As you can see, the menu options are 'hard-coded' into the BASIC program and are therefore not all that flexible. For instance if we wanted to add another option we would have to add extra lines of BASIC and renumber the program (to keep things neat and tidy). I can develop the menu display later to accommodate this flexibility, but, for the time being I will continue with the first of the sub-routines.

Creating the file PFILE. The complete file will be created in one go since there are few names and numbers to enter. Later, new names can be added or old ones deleted. The following flow chart illustrates the procedure. If you have already created a file with this name, it should either be renamed or deleted, first, before selecting option 1 from the menu. A convention will be used in these routines to signify the end of data input. iT will be three asterisks '***' and this code will not be written to the file. As the phone number is stored as a sting, the number may be as long as you wish. If a mistake is made when entering data then the whole input will be deleted and you will start again. All these aspects would be frowned upon in a commercial program but would detract from the exercise at hand, so I will keep it simple for now.



The program lines are :-

```
1000 REM ----- CREATE PFILE -----
1010 GOSUB 1200:   REM --- Open PFILE for output
1020 CLS
1030 PRINT "TYPE NAME AND PHONE NUMBER (*** = STOP)":PRINT
1040 INPUT "NAME";N$
1050 IF N$="***" THEN 1120
1060 INPUT "PHONE NUMBER";P$
1070 INPUT "CORRECT (Y/N)";A$
1080 IF A$<>"Y" THEN PRINT "ERROR, PLEASE RE-ENTER":GOTO 1040
1090 PRINT#1,N$:PRINT#1,P$
```

MICROSOFT

Next month - Completion of CREATE PFILE and the next routine.