

THE AMSTRAD USER

Issue No. 4 \$3.00

May 1985



- USERS' CONTACT LIST
- SOFTWARE REVIEWS
- \$2500 COMPETITION
- UTILITIES FOR THE CPC-464

FOR THE NOVICE & EXPERIENCED USER

NEW

The Quill

Adventure Writing System

VOTED

UTILITY OF THE YEAR
BY CRASH MICRO &
POPULAR COMPUTING WEEKLY

NOW AVAILABLE FOR

THE AMSTRAD

CPC 464

Enquiries to:

*Strategy Software, Suite 4a, 33-45 The Centreway, Blackburn Road,
Mt. Waverley, Victoria 3149 Tel: (03) 233 9227*

THE AMSTRAD USER

Issue No. 4 May 1985

CONTENTS

Editorial	2
Letters – and some answers	3
User Group Information	5
23 Matches – A.M. Urankar makes the CPC464 learn	6
Writing Adventure Programs – by Philip Riley	8
Trials of Tony Blakemore	9
Owl-phabet – a game for the Kindergarten	10
Amstrad Basic Find Utility from S. Barnett	12
Bytes and Pieces – a trio of listings	18
Memory Dump from Steve Miles	19
MATHS TABLES – four types from Jim Oehlman	20
More Software Reviews – Roland in Time, 'Er-Bert and Flight Path 737	22
RUNNER – another game to build on, with a prize	24
Binary to Hexadecimal Tutor from Steve Miles	28
Artificial Intelligence – Arthur Harris makes you think	30
Competition – extended for three months	32

All enquiries and contacts should be made to The Amstrad User,
Suite 4a, 33-45 The Centreway, Blackburn Road, Mt. Waverley
3149, Australia. Tel: (03) 233 9227.

The Amstrad User is published each month by Strategy
Publications. Reprinting of articles in The Amstrad User is
strictly forbidden without written permission. All rights
reserved. Copyright 1985 by Strategy Publications. The single
copy price of \$3.00 is the recommended retail price only. Also

available on subscription for 12 months.

Please note that whilst every effort is made to ensure the
accuracy of all features and listings herein, we cannot accept any
liability whatsoever for any mistakes or misprints.

Contributions are welcomed from readers or other interested
parties. If you want them returned, then please send a large SAE
with all submissions.

THE AMSTRAD USER

G'day,

This month The Amstrad User is generally available in most newsagents, so regular readers will forgive me while I introduce this publication to new readers who, hopefully, will already own or are just about to purchase an Amstrad CPC-464.

Like the CPC-464, this magazine is new. In fact the copy you are holding is only the fourth issue, but already the circulation has risen to 3000 per month. It is compiled, printed and published totally in Australia for Australian users. Browsing through the magazine, you should understand that apart from assisting newcomers to programming and the CPC-464, it is also the medium for helping local user groups to get established. We don't forget the more experienced users either, and there is plenty of opportunity to use the magazine as a forum for ideas, programs and observations.

In view of this expansion in distribution, we have decided to extend the Competition by a further three months to give all newcomers a chance to win, after all, we couldn't expect them to complete their entries in a matter of days, as the original closing date was 15th April 1985. Previous entrants will be given the option to revise their entry if they so wish.

The network of user groups being established across Australia is slowly growing - but they appear to be reluctant to let us know how they are proceeding. Surely some of the original buyers of the CPC-464 must now be quite experienced users. So stop staring at your screens for a moment and fill us in on what's happening, and perhaps save other people 're-inventing the wheel'. You won't get any bigger if you don't spread the news.

See you next month.

Ed.

Letters

Another look into our bulging postbag. If you want a personal reply, send a stamped and self-addressed envelope, and remember, all letters published will earn \$5 for their authors.

I enclose a photocopy of a program from the book 'Sensational Games for the Amstrad' along with the input on a tape. I have checked it together with a second person, but each time I run it, I get 'DATA EXHAUSTED IN 22090'. The tape also contains another program that will not answer correctly. Could someone check this out for me?

JMB, Belconnen

We are publishing this letter, not to embarrass the correspondent but to illustrate to other new programmers the importance of correctly keying-in programs. The first problem 'DATA EXHAUSTED IN 22090' would indicate that a READ command has attempted to read beyond the end of the last DATA. So in the first instance, check that you have sufficient data for all your READ's. When we looked at the data fields we discovered that a comma was missing between two of them - easily done and easily fixed - but frustrating.

The second problem concerned a program which was supposed to check additions and subtractions but resulted in a Syntax Error. On checking the offending line, we discovered an error common to many program failures. It concerned an instruction which wrapped-round to a second line. On the surface the instruction appeared valid, but taking a closer look, ELSE was the last word on the first line (the last character of ELSE occupying the last position on the line) and IF was the first word on the next line. Thus 'Arnold' read the instruction as ELSEIF !

So beware, make sure that when a line wraps-around as described above, you don't forget the space. You

should also note that when entering an instruction which occupies all character positions on a line, the cursor will move to the first position of the next line awaiting an ENTER to allow you input the next instruction. If you fail to ENTER here, the next instruction will become part of the previous one.

Do you supply computer paper and address labels? The Strategy Software catalogue mentions a Tape Copier for \$24.95. I would like to know if it is possible to copy all purchased tapes with this, so as to have working (or playing) copies and be able to safely store the original away. I have recently purchased an Amstrad 164 tape, which is marked side 1 and 2, however, when I put side 2 on it goes for a while then "error-rewind tape" comes up on the screen. I rewind but get no further. Can you suggest what might be the problem?

I wish to purchase the book "Amstrad Computing with the CPC464" by Ian Sinclair, so can you tell me where I might be able to obtain it. Our local newswagents did not have it.

My children have had a couple of games given to them, and we have had problems with them. One is "Return to Eden" - can't seem to get on with that game, and the other is "Flight Path 737" - can't get the plane to take off in spite of pressing F1 to take off as stated in the instructions.

JAB, Yarraman

No, we do not sell computer paper or labels and if you are having trouble, we would suggest Dick Smith or Tandy.

Unfortunately Tape Copier does not copy all programs - some are too

large and some use the same area of memory as Tape Copier itself. Version 2 (about a month away) may well solve most of these problems.

The Amstrad 164 tape is most probably recorded at slow speed on side A and fast speed on Side B. The problems encountered with programs saved at fast speed are due to the tape recorders' head alignment being very slightly out. This is not a major problem and most Software houses provide a copy of the program at each speed on each cassette. Should you be able to load the fast version, that is your good luck but this cannot be guaranteed by anyone.

The book 'Amstrad Computing' is available from your local dealer or from Strategy Publications at \$15.95. (Subscribers to The Amstrad User are eligible for a discount on this and other books mentioned later in this issue).

The instructions you received with 'Flight Path' would appear to be those which relate to the Commodore version of the game. The following should help: by using the numeric keys at the top of the keyboard, 7 increases the speed by 20 kns and 4 increases by 10 kns, and 0 decreases by 20 kns.

I am hoping you can help me with two problems. The first is in a program called 'Pontoon' from a book called 'Sensational Games for the Amstrad' by Jim Gregory. It is a fabulous book of programs but for this one. The game keeps referring to line 11100, but it doesn't exist.

The other program is from a book by John Braga called 'Amstrad Explored'. The whole book has been produced on a daisy

wheel printer and is a mess. The only program I have tried is *Hungry Hiffalump*, and crashes every time I run it despite many alterations.

JE, Brisbane

We have received a number of complaints concerning the quality of the programs in Sensational Games, and we understand that a revised edition is being printed. It appears to be one of those books that have been hastily converted from another machine to run on the CPC464.

We have not had time to key the program in, so cannot accurately determine whether references to line 11100 should in fact refer to another line (11000?).

If anyone else has solved this problem, and any others in the book, please let us know.

I receive your magazine regularly and I think it would be a good idea to start a 'Pen Pal' corner or something along the same lines. If you do decide to do this, could you please print the following "Wanted Pen-Pal, about 11 or 12 years old, to swap ideas and programs". Would you like me to do Junior Reviews of your games? I am aged 11 years and think the Amstrad computer is terrific.

DM, Merredin

This letter certainly raises a couple of good points. First, the idea of providing space in The Amstrad User for our junior subscribers. We have considered this and, if the demand warrants it, we would most certainly include a "Junior Jotters" section. Secondly, the point on reviewing games. Anyone is welcome to submit reviews of games (based upon 'How

to Review' - Issue 1), and we could divert those from younger readers to "Junior Jotters". So the ball is in your court - just make sure that any correspondence is clearly marked JUNIOR JOTTERS if relevant

It will be of interest to know that we are currently planning to provide software reviews as often as possible. To achieve this, we intend to supply an item of software to a reviewer for him/her to keep by way of payment for the review when published. If you consider yourself eligible to become a reviewer, please drop us a line and mention why you should be considered to become a member of our review panel.

THE HELP-DESK

A special service for novice programmers.

If you have a problem concerning your Amstrad CPC-464 and related software, you may ring any of the volunteers below, who have kindly offered to set aside some time to help you.

TONY BLAKEMORE — (03) 878 6212
Saturday between 4 p.m. - 9 p.m. and **Sunday**
between 10 a.m. and 4 p.m.

MARTIN SCRAGG — (059) 786 949
Weekdays between 4 p.m. and 10 p.m.
(except Tuesday or Wednesday) and between
6 p.m. and 10 p.m. at Weekends.

HANS HILL — (02) 671 2929
Weekends only between 9 p.m. and 5 p.m.

User Group Information

We are happy to report that our Contact List is still growing, but reports from the Groups already established are not. Still, it's early days and we do appreciate that there may be some administrative problems which take precedence over letters to The Amstrad User.

One of the major advantages of belonging to a User Group is the fact that it allows you the freedom to talk to other people who have a micro-computer which is (in most cases) identical to your's and who have experienced, or still are experiencing a new world of programming techniques or problems.

The Groups that have so far established themselves appear to be evenly split between experienced users and newcomers. So for all those people new to computing who are afraid of getting swallowed up by the jargon, or feeling inadequate amongst more experienced mortals – fear not!

You will find that most of the skilled members are only too happy to impart their knowledge to others – let's face it, they too were novices once.

As this issue goes to press, we can report that the

Frankston (Victoria) group is thriving and expects at least 24 members to the next meeting. We have been promised a report for publication next month, detailing their meeting place and times, and some useful tips for other groups. Don Buchanan of the Grange Group (SA) has reported a steady attendance to his meetings at Grange Primary School. The Richmond Users' Group (Victoria) has just formed, but we don't have specific details at the moment, however, by ringing (03) 428 4281 Mr. Russo should provide the information you require.

But as mentioned in the introduction above, we rely upon someone within the groups to keep us updated on what is happening at the meetings and supply any 'discoveries' that can be passed on through these pages to other groups. So come on, stop writing programs for a moment and start writing some letters!

User Group Contact List

<p>NSW</p> <p>John Patterson Lismore (066) 21 3345 Chris Craven Canowindra (063) 44 1150 Paul Wilson Moruya (044) 74 3160 Frank Humphreys Mummulgum (066) 64 7920 R. Vijayenthiran Newtown (02) 519 4106 Hans Hill Blacktown (02) 671 2929</p>	<p>QLD</p> <p>Paul Witsen Bulimba (07) 371 9259 Mick O'Regan Gladstone (079) 79 2545 R.C. Watterton Toowoomba (076) 35 4305 D.F. Read Ingham (077) 77 8576 Kylie Telford Goondiwindi Calingunee 246 (weekends only)</p>
<p>ACT</p> <p>John Payne Phillip (062) 82 2277 Arthur McGuffin Kambah (062) 31 9437</p>	<p>SA</p> <p>Don Buchanan Grange (08) 356 2664 Lindsay Allen Murray Bridge (085) 32 2340</p>
<p>Vic</p> <p>R.A. Russo Richmond (03) 428 4281 Tony Blakemore Nunawading (03) 878 6212 Martin Scragg Pearcedale (059) 78 6949 Don Leith Brunswick (03) 383 1498 Michael Prezens Frankston (03) 781 2158 Mrs. G. Chapman South Clayton (03) 551 4897 Mike McQueen Braybrook (03) 312 5594 Paul Walker Heathmont (03) 729 8657 David Carbone Burwood (03) 29 4135 Sue Kelly Manangatang (050) 35 1402 Alan Harris Sale (051) 44 1454</p>	<p>WA</p> <p>Bob Harwood Cooloongup (095) 27 1777 Dave Andersen 6 Kitchener Rd Merredin, 6415 Graeme Worth Scarborough (09) 341 5211 Warren Pynt Dianella (09) 276 9883</p>
	<p>TAS</p> <p>Conal McClure Scottsdale (003) 52 2514</p>

23 Matches

Here is an old game brought up to date by A.M. Urankar of Eltham. It has an interesting 'twist' as the narrative describes, and although simple in concept is quite frustrating the more it is played.

Back in the "old" days, which my daughters are convinced must have been in the Stone Age, we played a game called 23 Matches. I have re-worked the game to suit Ami (our system has been christened "Ami").

By way of explanation of the program, all user input is through the INKEY function to provide a measure of 'user friendliness'. There are no levels of difficulty as such: but don't be fooled. A very basic form of learning has been incorporated in the program, which means that there is an effective increase of difficulty with each game. The graphics are very simply created by DATA statements to allow for easy manipulation.

```
10 ' Program Name = 23MATCHES
20 ' Written for AMI by A.M.URANKAR
30 GOTO 480
40 '
50 '** subroutines area
60 WHILE remove>0
70 IF remaining<=0 THEN RETURN
80 LOCATE x(position),y(position):PRINT blank$
90 remove=remove-1:position=position-1
100 SOUND 1,80,20
110 FOR rest=1 TO 300:NEXT
120 WEND:RETURN
130 FOR matches=1 TO 23:READ x(matches),y(matches)
140 SOUND 1,matches*10
150 PEN 1:LOCATE x(matches),y(matches):PRINT body$
160 PEN 2:LOCATE x(matches),y(matches):PRINT CHR$(211)
170 NEXT:RESTORE:RETURN
180 '
190 '** main program area
200 CLS #1
210 PRINT#1,"How many do you want to remove?";
220 remove$=INKEY$:IF remove$="" THEN 220
230 remove=VAL(remove$):PRINT#1,remove
240 IF remove<1 OR remove>3 THEN PRINT #1,"Please s
tick to the rules!!! Only 1, 2, or 3 can be removed at any
time":PRINT #1:GOTO 210
250 remaining=remaining - remove:GOSUB 60
260 IF remaining<=0 THEN 390
270 game(remaining)=-1
280 IF remaining=1 THEN 410
290 IF remaining=2 THEN 360
300 IF remaining=3 THEN 340
310 IF exper(remaining-3)<exper(remaining-2) THEN 340
320 IF exper(remaining-3)<exper(remaining-1) THEN 360
330 ami=3:GOTO 370
340 IF exper(remaining-2)<exper(remaining-1) THEN 360
```



```

350 ami=2:GOTO 370
360 ami=1
370 PRINT #1,"I have decided to take";ami:remove=ami:remaini
ng=remaining-remove:GOSUB 60
380 game(remaining)=1:GOTO 210
390 CLS #1:PRINT #1,"Sorry about that, but I won!!"
400 FOR x=1 TO 23:exper(x)=exper(x)+game(x):NEXT:GOTO 430
410 CLS #1:PRINT #1,"You won - but you won't be so lucky nex
t time"
420 FOR x=1 TO 23:exper(x)=exper(x)-game(x):NEXT
430 PRINT#1,"do you want another game (y/n)"
440 answer$=INKEY$:IF answer$="" THEN 440
450 IF answer$="y" THEN GOSUB 130:position=23:remaining=23:G
OTO 200 ELSE PRINT #1:PRINT #1,"thanks for the fun - see you
soon.":END
460 '
470 '** initialization parameters
480 CLS:MODE 1
490 INK 2,6
500 WINDOW #1,1,40,18,25
510 DIM x(23),y(23)
520 DIM exper(23),game(23)
530 position=23:remaining=23
540 SYMBOL AFTER 210
550 SYMBOL 210,&18,&18,&18,&18,&18,&18,&18,&18,&18
560 SYMBOL 211,&0,&18,&3C,&7E,&7E,&7E,&7E,&7E,&7E,&3C
570 body$=CHR$(210)+CHR$(10)+CHR$(8)+CHR$(210)+CHR$(10)+CHR$
(8)+CHR$(210)+CHR$(128)+CHR$(10)+CHR$(8)+CHR$(128)+CHR
$(8)+CHR$(128)+CHR$(10)+CHR$(8)+CHR$(128)+CHR$(10)+CHR
$(8)+CHR$(128)+CHR$(10)+CHR$(8)+CHR$(128)
590 DATA 8,1,11,1,14,1,17,1,20,1,23,1,26,1,29,1
600 DATA 8,6,11,6,14,6,17,6,20,6,23,6,26,6,29,6
610 DATA 8,12,11,12,14,12,17,12,20,12,23,12,26,12
620 GOSUB 130
630 FOR x=1 TO 23:exper(x)=0:game(x)=0:NEXT:exper(1)=100
640 PRINT #1,"The object of this game is to NOT BE THEONE to
take the last match.":PRINT #1
650 PRINT#1,"You can only remove 1, 2, or 3 matches at each
turn. Being generous I will let you go first. When ready, p
ress any key."
660 IF INKEY$="" THEN 660 ELSE 200

```

Writing Adventure Programs

Philip Riley, author of our first adventure program 'Down the Mine', provides the inside information on how he develops this particular type of program.

I was amazed to discover the other day that some adventure games can take three months to develop. I am very keen on these types of games, and using a method which I will now describe, I can generally write a good game in under two weeks.

The first thing to do when writing such games is to think up a storyline. This is just an explanation as to what you are doing and where you are, how you came to be there and what you must do to get out. The next problem I find the hardest; that is devising new and devious problems and traps to overcome.

Now we come to the mapping out of the game. I normally do this on a large piece of cartridge paper which will only set you back about twenty cents. Once you have mapped out the rooms, caves or whatever, you then mark out the problems and traps onto the map. This is where you must be careful and make sure that everything can be reached. It is no use putting a lamp after a dark room or a key behind a locked door.

Now we have our game mapped out, we must work out the vocabulary. These are the words that you will use to provide commands/actions during the game. Most adventure games use the two word commands to input into the computer, for example, get sword, drop sword etc., in a verb and noun format. I write the first and second words in two columns on the cartridge paper. By looking at the map you will find it fairly simple to work out all the words that you will need.

When you are satisfied with the list of words for the vocabulary, you can start programming. The way that you

tackle this phase is most important in order to make the game as simple as possible to program. First, I type in the vocabulary and the routines that verify the input. At this point I would like to stress that it is no use typing in the whole game and then trying to fix all the bugs. It is far better to get one section of the program running and then move onto the next section.

When you have typed in the vocabulary routines you then type in the routines that will allow you to move. When both of these routines are running, you type in the descriptions of the rooms and the routines for sending the program to the particular room when you move. At this point do not put in any of the problems or traps. You should now have all the rooms programmed into the computer and you should be able to move around all of the rooms freely without having to do anything.

Now all that is left is to put in the traps and problems. Start with the last problem that you will have to encounter. You put in the routine at the correct place in the program that will stop you from going past that point unless you have done what ever is needed to get past. For example, you cannot get past a locked door until you have used the key to unlock it. When done you move onto the second last problem and so on until you get to the first problem to be encountered. The reason for starting at the last problem and working your way back to the first is that it is a lot easier to check out each routine if you can move directly to the particular part of the game. If you started at the first problem that you encounter and worked your way to the last, you would

have to play through the whole game up to the point that you are programming to check out if the particular routine works. This can be very time consuming and boring. The final thing to be done is to type in any single word commands such as HELP, LOOK or INVENTORY. I have found the above system is about the best way of writing adventure games, but it is not the only way and you may know of one better.

To summarise, the procedure is as follows:

1. Invent a storyline.
2. Invent some new problems and traps.
3. Map out the game.
4. Mark out the problems and traps on the map.
5. Work out the vocabulary.
6. Program the vocabulary and routines for inputting words to the computer and checking those inputs with the vocabulary.
7. Program the routines for moving around the game.
8. Type in the descriptions of the rooms.
9. Type in the problems and traps.
10. Type in the routines for the single word commands if any.

Another thing that I should point out is that I write down all of the variables (numeric and string) and what they are used for, on the cartridge paper. This makes it very easy to check back on things later. And putting it down on the cartridge

paper rather than a separate sheet means that everything is laid out in front of you and you have no worries about losing vital information on little bits of paper, (after all it is rather hard to lose a large piece of cartridge paper)!

Finally, I should mention that this article is only a brief explanation of

how I plan and program an adventure game. Anyone who wants to write adventures but is unsure about programming certain routines please write in and ask, I will be only too happy to help you as best I can.

Note from the Editor.

Over the next few months we will feature

an Adventurer's column. This will provide an opportunity for all you enthusiasts to swap notes, clues and perhaps, if you are really stuck, the answers. You may also send your opinions on certain adventure games, and as Philip has mentioned above, get some help on developing your own programs.

The Trials of Tony Blakemore

A Column dedicated to the absolute beginner.

The errors that we have discussed so far have been, in the main, syntax errors and are usually easy to find. What we will discuss this month are the other types of errors that you will encounter and which are harder to define and correct.

'Improper argument' would have to be one of the more common errors that you will come across and though the line number is stated when the error occurs it can cover a multitude of mistakes. You may, for example, generate a number for the locate command that is impossible to execute. This is very obvious if you are using numbers but not so obvious if using variables, ie. LOCATE XPOS, YPOS. When the program once again grinds to a halt and gives you the line number where the error occurs, print out the value of the variables. Type PRINT xpos, ypos. If either one of the values is out of range BASIC cannot of course execute it. Go back to the line where the variable is initialised (given a value) and correct the mistake!!!! It's getting a bit complicated now isn't it? That's the problem with the more obscure bugs, the program line that the error occurs at is not always the problem line. It could have been generated at any point in the program.

'Data exhausted' is met when a read loop is larger than the amount of data in the data lines. The most

common error here is when a comma is missed or replaced with a full stop. BASIC can only read the data statements that are separated by a comma and it is very easy to miss one when entering large amounts of data. A handy hint on data lines is to make all your entries the same length, ie. put, say, eight or ten data statements on one line. This makes it very easy to count and pick up any errors that may occur.

'Unexpected next' or 'Next missing'. When first using loops get into the habit of declaring the NEXT statement ie. NEXT a or NEXT b or whatever variable you may be using. This makes tracing an errant next, which may be many lines further on than the FOR statement, much easier. If you have only terminated the loop with a NEXT and you come up with either of the above errors it can be very hard to trace.

'Type mismatch'. When you start to write your own programs you will generate this error time and time again, ie. a= "time" where you have tried to allocate a string statement to a variable. Or a\$ = time where you have tried to allocate a variable to a string. Usually the mistake is obvious and easy to correct.

The most baffling of all errors is when the program runs without producing any error messages but will not run correctly. The screen may

suddenly blank out for no reason or produce results that are totally unexpected. This is known as logic error and is the hardest of all errors to correct. A very useful tool is provided by LOCOMOTIVE BASIC called TRON and TROFF, (Chapter 8, page 49.) TRON means trace on and prints on the screen all the line numbers as the program is running. When you first use it, it can be confusing as it pushes all screen displays aside. It is very helpful in tracing the logical moves of any program. If a program is doing what it is not supposed to, TRON will enable you to follow the logic of the execution and pinpoint the area where the program may divert from the course that you have set. You may have a GOTO instead of a GOSUB. You may have put GOSUB 1000 instead of 2000. By following the line numbers you should be able to find where the program goes wrong. When the error has been corrected type TROFF which removes the trace. Save the program and then run it. Hopefully the error should have been corrected.

The one area that has created many problems is typographical errors. Errors that occur in books and magazines. These are the hardest to correct and all I can suggest is that you get involved with a Users Club and bring the programs that you are

Owl-phabet

Here is another simple program by Tony Blakemore which illustrates the use of PRINT SPACES\$, SYMBOL and SOUND. The completed program should suit pre-school children in their efforts to learn the alphabet.

```
10 ' OWL-PHABET Pre school Alphabet game.
20 ' The alphabet is shown on screen one
30 ' letter at a time. If the right match
40 ' is made from the keyboard the child
50 ' is rewarded by ahappy face and sound
60 ' Incorrect selection, by a sad face
70 ' and sound. If the child gets all the
80 ' letters correct The Alphabet song is
90 ' played.
100 ' ***** CONTROL *****
110 '
120 GOSUB 180(Initialise)
130 GOSUB 320(Screen)
140 GOSUB 470(Main loop)
150 '
160 ' ***** INITIALISE *****
170 '
180 MODE 0
190 WINDOW #2,1,20,11,24
200 RANDOMIZE TIME
210 DEFINIT a-z
220 right=0:wrong=0
230 SYMBOL 240,128,192,160,145,138,68,64,92
240 SYMBOL 241,2,6,10,18,162,68,4,116
250 SYMBOL 242,128,136,148,162,170,170,162,148
260 SYMBOL 243,2,34,82,138,170,170,138,82
270 SYMBOL 244,72,227,241,121,121,120,120,122
280 SYMBOL 245,36,142,30,60,60,60,60,188
290 SYMBOL 246,123,121,121,116,106,201,136,20
300 SYMBOL 247,188,60,60,92,172,38,34,80
310 RETURN
320 PRINT CHR$(240);CHR$(241);
330 PRINT SPACES$(16);CHR$(240);CHR$(241);
340 PRINT CHR$(242);CHR$(243);
350 PRINT SPACES$(16);CHR$(242);CHR$(243);
360 PRINT CHR$(244);CHR$(245);
370 PRINT SPACES$(3);"OWL-PHABET"; SPACES$(3);
380 PRINT CHR$(244);CHR$(245);
390 PRINT CHR$(246);CHR$(247);
400 PRINT SPACES$(16);CHR$(246);CHR$(247);
410 LOCATE 5,8:PRINT CHR$(224);
420 PRINT SPACES$(10);CHR$(225);
430 RETURN
440 '
450 ' ***** MAIN LOOP *****
460 '
470 FOR x=65 TO 90
480 CLS #2
490 FOR i=0 TO 15
```



```

500 LOCATE #2,INT(RND*18)+1,1,INT(RND*14)+1
510 PEN #2,i
520 PRINT #2,CHR$(x)
530 SOUND 1,i+20,5,4
540 SOUND 1,500+i,4,1
550 FOR a=1 TO 100:NEXT
560 NEXT
570 FOR A=1 TO 1400:NEXT
580 CLS #2:LOCATE #2,10,7:PEN #2,1:PRINT #2,"??"
590 ,
600 , **** CHECK CORRECT ****
610 ,
620 K$=INKEY$
630 a$=INKEY$:a$=UPPER$(a$)
640 IF a$="" THEN 630
650 IF a$<>CHR$(x) THEN 750
660 CLS #2:LOCATE #2,8,5
670 PRINT #2,CHR$(224);" "CHR$(x);" "CHR$(1);CHR$(6)
680 FOR a=1 TO 150 STEP 5
690 SOUND 1,a,5,4:SOUND 1,500+a,4,1:NEXT
700 right=right+1:LOCATE 6,8:PRINT right
710 GOTO 830
720 ,
730 , **** INCORRECT ****
740 ,
750 CLS #2:LOCATE #2,8,5
760 PRINT #2,CHR$(225)" "a$" "CHR$(88)
770 FOR a=1000 TO 3000 STEP 15
780 SOUND 1,a,2,5:NEXT
790 LOCATE #2,8,7
800 PRINT #2,CHR$(224)" "CHR$(x)" "CHR$(1)+CHR$(6)
810 wrong=wrong+1
820 LOCATE 12,8:PRINT wrong
830 FOR a=1 TO 3000:NEXT:NEXT X
840 IF right=26 THEN 930
850 MODE 0:LOCATE 8,3:PRINT CHR$(249);" "CHR$(248)
860 LOCATE 8,5:PRINT "R S"
870 A$=INKEY$:A$=UPPER$(A$):IF A$="" THEN 870
880 IF A$="R" THEN RUN
890 IF A$="S" THEN STOP ELSE 870
900 ,
910 , **** ALL CORRECT TUNE ****
920 ,
930 BORDER 16,11
940 RESTORE
950 FOR A=1 TO 14
960 READ NOTE,DURATION
970 SOUND 1,NOTE,DURATION,5
980 SOUND 2,NOTE-5,DURATION,5
990 IF NOTE = 320 THEN FOR Z=1 TO 900:NEXT
1000 FOR Z=1 TO 450:NEXT
1010 NEXT
1020 BORDER 1
1030 GOTO 850
1040 DATA 478,25,478,25,319,25,284,25,284,25,320,75
1050 DATA 358,25,358,25,379,25,426,25,426,25,478,100

```

Amstrad Basic Find Utility

By S. Barnett

When developing a program it is sometimes useful to find keywords or variables (eg: PRINT, CLS, A\$, A2 . . .) or literal text (eg: HELLO THERE JANE). Scanning the listing line by line can be tedious, time consuming and inefficient and, to this end, the "FIND UTILITY IN BASIC" was written. The program has also been written to inconvenience the user as little as possible and so utilizes the programmable function keys namely, the numeric keypad. (see Amstrad User March 1985).

When you wish to use the utility turn your Amstrad on and load this program from cassette (where it had previously been saved after you correctly typed it in) and run it. Upon completion of this step the "Ready" prompt is presented and, if you try to list the program, it will not be there. After recovering from this shock discovery you may load from cassette or type in your program. If you wish to find some literal text enter

1 "text" - followed by ENTER (There is one space between 1 and "text")

where 'text' is the text that you are searching for. This enters (as you might guess) line number one into your program. Now type in <CTRL><SMALL-ENTER-KEY> (this means holding the control and small enter keys down simultaneously). A line of BASIC appears on the screen and then, hopefully, the "Ready" prompt. This means that a search pattern has been set. Lastly, strike the small enter key and be amazed as the following appears . . .

a line of BASIC which has very little meaning, the words "Found in:" and a series of line numbers which should contain the search string.

```
10 KEY 139, "FOR a=1 TO 189:POKE 105+a, PEEK(43710+a):NEXT:"+CHR$(13)
20 KEY 140, "a$="+CHR$(34)+CHR$(34)+" :c=368:for x=2 to peek(c)-5:a$=a$+chr$(peek(371+x)):next"+CHR$(13)
30 MEMORY 43709:FOR a=1 TO 189:POKE 43710+a, PEEK(588+a):NEXT:NEW
40 PRINT "Found in:", l=99:WHILE l>0:c=c+PEEK(c):l=PEEK(c+2)+PEEK(c+3)*256:b$="":FOR x=c+4 TO c+PEEK(c)-2:b$=b$+CHR$(PEEK(x)):NEXT:FOR x=1 TO SGN(INSTR(b$,a$)):PRINT l,:NEXT:WEND
50 REM BY S.Barnett, 9 Feb 1985
```

NOTE: when finding literal text the CASE of the text is very important.

When finding keywords or variables the procedure is the same except that you enter line one as

1 search-target - followed by ENTER

There is a double space between the line number and the search target and, with variables, the case of any letters used is of the utmost importance. The remainder of the find procedure is the same as with straight text.

For the technically minded:

So that both tokenized and unaltered search patterns can be located, the BASIC interpreter is used to change any given pattern to its tokenized format. Upon running, the routine deletes the program in memory (usually only itself) and so this should occur before loading any development software.

It was originally intended to program a function-key (part of the numeric keypad) with the one-liner. Unfortunately, these keys allow a maximum of 105 characters for expansion in total and reducing the utility to this size was an impossible task. In lieu of this, the routine was sub-divided into two miniature programs, with the majority being stored in very high memory in tokenized format. This program is then down loaded into immediate command memory (locations 64 and onwards) and executed. This is a useful method of overcoming buffer limitations.

**HOW TO GO FROM
PROGRAMMER
TO
SUPER-PROGRAMMER
IN FOUR EASY
PAGES**

P. N. O.

SPECIAL OFFER TO SUBSCRIBERS



60 Programs for your Amstrad CPC464

A massive software library for the price of a single cassette. Explosive games, dynamic graphics and invaluable utilities, this specially commissioned collection takes BASIC to the limits and beyond.

Subscriber's Price
\$17.10



The Amstrad CPC464 Advanced User Guide

This book assumes you have got your CPC464 working and have already done some simple programming. It provides explanations on how the CPC464 communicates with external devices, strings and characters, data structures, Graphics and Sound.

Subscriber's Price
\$15.65



The CP/M Handbook with MP/M

If your future in computing goes beyond playing games – you'd better have a copy of this book on your shelf. It will satisfy the requirements of first time users as well as veteran CP/M experts.

Subscriber's Price
\$26.95

SPECIAL OFFER TO SUBSCRIBERS



CP/M User Guide.

This updated best-seller covers everything you need to know about the CP/M operating system from Digital Research. Organised for both beginner and advanced users, the information you'll need most often is presented first, followed by more in-depth technical data.

Subscriber's Price
\$24.45



An Amstrad CPC464 Compendium

A fascinating selection of thirty games, each presented with a complete listing in LOCOMOTIVE BASIC, a helpful introductory description and a screenshot. There is something to appeal to the traditional and ultramodern computer gamers alike.

Subscriber's Price
\$20.65

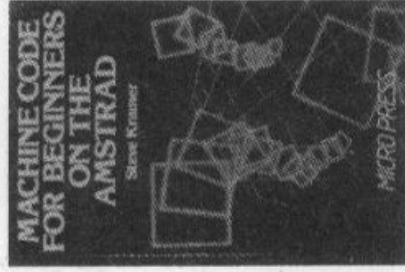


Logo Programming

This entertaining book will give you the knowledge and experience required to make the most of LOGO. Learn to use this ingenious medium and soon you will be constructing your own micro-worlds, simulations and demonstrations.

Subscriber's Price
\$11.65

SPECIAL OFFER TO SUBSCRIBERS



Machine Code for Beginners on The Amstrad

As the title suggests, this book is intended for the beginner wishing to learn how to use Machine code on the CPC464. Extensive use is made of the machine operating system allowing results from programs to be seen immediately.

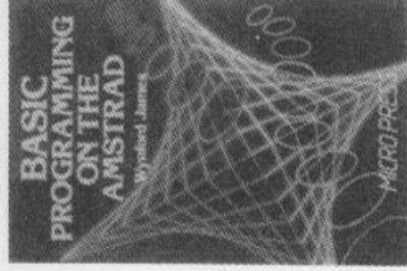
Subscriber's Price
\$16.15



40 Educational Games for the Amstrad CPC464.

The programs in this book have been designed to help the younger members of a family to handle the Amstrad and to increase their general knowledge – whilst enjoying themselves. Subjects include languages, mathematics and science.

Subscriber's Price
\$14.35



Basic Programming on the Amstrad

The first two chapters are for the complete beginner, the rest introduces the majority of the BASIC commands available, and the last few deal with topics of more specialised interest such as the use of sound, and how to create files and read data from them.

Subscriber's Price
\$17.95

SPECIAL OFFER TO SUBSCRIBERS



Amstrad Computing

Takes you step by step, in detail, through the commands on your CPC464. You will find how business calculations are worked out, how vivid displays are obtained and how attention-catching sounds can be arranged.

Subscriber's Price

\$14.35



The Amstrad Games Book

A collection of exciting games for your Amstrad, including arcade-style space and combat games, maze games, impossible and dangerous missions for superheroes and many more.

Subscriber's Price

\$13.45



The Working Amstrad

This book is a collection of solid application programs. The areas covered include home finance and tax, information storage and retrieval, household and diary management, creative graphics, and effective display techniques.

Subscriber's Price

\$17.95

How To Order

All subscriber prices shown above have been discounted by 10%. Send a list of the titles and quantities you require along with a cheque for the total **plus \$5.00** postage and packing (regardless of the quantity you order) to Strategy Publications, Suite 4a, 33-45 The Centreway, Blackburn Road, Mt. Waverley, Victoria 3149. We will also accept Bankcard or Mastercard orders by phone on (03) 233 9227.

Bytes & Pieces

This month we present a trio of small programs which you may wish to experiment with. Two of them are graphics oriented and the other is a reasonable example of a sort.

Sort Demonstration

```
10 REM ** SORT DEMO **
20 entries=-1
30 DIM store$(20)
40 PRINT "Please enter strings"
50 entry$=""
60 WHILE entry$<>""
70 INPUT entry$
80 IF entry$="" THEN 120
90 LET entries=entries+1
100 store$(entries)=entry$
110 IF entries=20 THEN PRINT "Array now full"
120 WEND
130 PRINT "Sorting...."
140 FOR STRINGS=0 TO entries-1
150 compare$=store$(strings)
160 position=strings
170 FOR rest=strings+1 TO entries
180 IF compare$>store$(rest) THEN compare$=store$(rest):position=rest
190 NEXT rest
200 IF position=strings THEN 250
210 FOR moves=position-1 TO strings STEP-1
220 store$(moves+1)=store$(moves)
230 NEXT moves
240 store$(strings)=compare$
250 NEXT strings
260 PRINT "...finished"
270 FOR strings=0 TO entries
280 PRINT store$(strings)
290 NEXT strings
```

3D Ball

```
10 size=200
20 sizey=200
30 MODE 2
40 MOVE size*SIN(100)+320, sizey*COS(100)*SIN(100*0.95)+200
50 FOR a=100 TO 225.8 STEP 0.2
60 DRAW size*SIN(a)+200, sizey*COS(a)*SIN(a*0.95)+200
70 NEXT a
```

Cross-Hatch and Circle

```
10 REM ** CROSS HATCH GENERATOR **
20 REM **
30 CLS
40 PRINT:PRINT:PRINT:PRINT
50 PRINT " CROSS HATCH and CIRCLE GENERATOR" :PRINT:PRINT
60 FOR y=1 TO 2000:NEXT y:CLS
70 MODE 1
80 LET m=0
90 FOR i=1 TO 15
100 MOVE m,0
110 DRAW m,1027
120 LET m=m+45.6
130 NEXT i
140 LET n=0
150 FOR i=1 TO 13
160 MOVE 0,n
170 DRAW 1280,n
180 LET n=n+33.2
190 NEXT i
200 ORIGIN 320,200
210 FOR angle=1 TO 360 STEP 0.25
220 xpoint=190*COS(angle)
230 ypoint=190*SIN(angle)
240 PLOT xpoint,ypoint
250 NEXT
260 FOR z=1 TO 1000000: NEXT z
270 END
```


Memory Dump

Steve Miles of Sale has submitted this program to dump the contents of memory in Hexadecimal. If you are unfamiliar with binary and conversions to hexadecimal, Steve has also supplied a tutor which appears on Page 28 of this issue.

```
10 MODE 1
15 INK 0,1:INK 1,26:BORDER 1
20 LOCATE 1,1:PRINT"1. This program is to read RAM memory
only."
22 LOCATE 1,4:PRINT"2. Input start address in DECIMAL or
HEXADECIMAL."
25 LOCATE 1,7:PRINT"3. Memory contents will be displayed in
HEXADECIMAL."
40 LOCATE 5,15:PRINT"Press <ENTER> to continue."
50 a$=INKEY$:IF a$="" THEN 50
55 IF a$<>CHR$(13) THEN 50
60 MODE 2
65 INK 0,10:INK 1,0:BORDER 10
70 INPUT"start addr";n:CLS
80 z=0:CLS:PRINT"
8 9 A B C D E
F":PRINT
90 n$=HEX$(n,4)
100 a$=HEX$(PEEK(n),2):n=n+1
110 b$=HEX$(PEEK(n),2):n=n+1
120 c$=HEX$(PEEK(n),2):n=n+1
130 d$=HEX$(PEEK(n),2):n=n+1
140 e$=HEX$(PEEK(n),2):n=n+1
150 f$=HEX$(PEEK(n),2):n=n+1
160 g$=HEX$(PEEK(n),2):n=n+1
170 h$=HEX$(PEEK(n),2):n=n+1
180 i$=HEX$(PEEK(n),2):n=n+1
190 j$=HEX$(PEEK(n),2):n=n+1
200 k$=HEX$(PEEK(n),2):n=n+1
210 l$=HEX$(PEEK(n),2):n=n+1
220 m$=HEX$(PEEK(n),2):n=n+1
230 q$=HEX$(PEEK(n),2):n=n+1
240 o$=HEX$(PEEK(n),2):n=n+1
250 p$=HEX$(PEEK(n),2):n=n+1
260 z=z+1:PRINT" ";n$;" ";a$;" ";b$;" ";c$;" ";o$;"
";e$;" ";f$;" ";g$;" ";h$;" ";i$;" ";j$;" ";k$;" ";
l$;" ";m$;" ";q$;" ";o$;" ";p$
270 IF z<>22 THEN 90
280 LOCATE 10,25:PRINT"Press <SPACE BAR> for next page. Pres
s <E> to End or <R> to Run."
290 z$=INKEY$:IF z$="" THEN 250
300 IF z$="" THEN 80
310 IF z$="r" THEN RUN
320 IF z$="e" THEN END ELSE 300
```

Maths Table

This simple program from Jim Oehlman may provide assistance to youngsters in practising, not only the traditional 'times tables', but also division, subtraction and addition.

On loading the program, you will be presented with a Menu from which to select the type of calculation you want to run. You are then asked 'WHICH TABLE'. Choose virtually any number you wish to be the basis of the calculation, providing it is not less than 1. The 'maximum limit' is the number of calculations you wish to see. For example, if you selected the multiplication

module (1) and the 10 table, by entering 12 as the maximum limit you will produce the standard '10 times table'. You also get the option to print the table. In the event of entering a large maximum limit which would scroll automatically, you can pause the display by pressing ESCAPE once, then ENTER to continue.

```
10 CLS
20 PRINT "MENU"
30 PRINT:PRINT
40 PRINT "1=MULTIPLICATION"
50 PRINT "2=DIVISION"
60 PRINT "3=ADDITION"
70 PRINT "4=SUBTRACTION"
80 PRINT:PRINT
90 PRINT "PLEASE SELECT (1-4)"
100 a$=INKEY$
110 IF a$="1" THEN GOSUB 160
120 IF a$="2" THEN GOSUB 380
130 IF a$="3" THEN GOSUB 600
140 IF a$="4" THEN GOSUB 820
150 GOTO 100
160 REM MULTIPLICATION TABLES
170 CLS: PRINT "MULTIPLICATION TABLES"
180 INPUT "WHICH TABLE";t
190 IF t<1 THEN PRINT "DON'T BE SILLY": GOTO 180
200 INPUT "SET MAXIMUM LIMIT";l
210 INPUT "DO YOU WANT A PRINTOUT (Y/N)";b$
220 IF b$="Y" THEN GOSUB 320
230 FOR w=1 TO 1000: NEXT w
240 REM NOW PRINT THE TABLE
250 CLS
260 LOCATE 8,4:PRINT CHR$(24); " ";t;"MULTIPLICATION TABLE";C
HR$(24)
270 PRINT:PRINT:PRINT
280 FOR i=1 TO l
290 PRINT TAB(5);t;"x";i;TAB(18);"=";t*i
300 NEXT
310 RETURN
320 LOCATE 8,4:PRINT#8," ";t;"MULTIPLICATION TABLE"
330 PRINT:PRINT:PRINT
340 FOR i=1 TO l
350 PRINT#8, TAB(5);t;"x";i;TAB(18);"=";t*i
360 NEXT
370 RETURN
380 REM DIVISION TABLES
390 CLS:PRINT "DIVISION TABLES"
400 INPUT "WHICH TABLE";l
410 IF t<1 THEN PRINT "DON'T BE SILLY":GOTO 400
420 INPUT"SET MAXIMUM LIMIT";t
```



```

430 INPUT "DO YOU WANT A PRINTOUT (Y/N)";b$
440 IF b$="Y" THEN GOSUB 540
450 FOR w=1 TO 1000:NEXT w
460 REM NOW PRINT THE TABLE
470 CLS
480 LOCATE 8,4:PRINT CHR$(24); " ";t;"DIVISION TABLE";CHR$(24
)
490 PRINT:PRINT:PRINT
500 FOR i=1 TO 1
510 PRINT TAB(5);t;CHR$(172);i;TAB(18);"=";t/i
520 NEXT
530 RETURN
540 LOCATE 8,4:PRINT#8," ";t;"DIVISION TABLE"
550 PRINT:PRINT:PRINT
560 FOR i=1 TO 1
570 PRINT#8, TAB(5);t;"/";i;TAB(18);"=";t/i
580 NEXT
590 RETURN
600 REM ADDITION TABLES
610 CLS: PRINT "ADDITION TABLES"
620 INPUT "WHICH TABLE";t
630 IF t<1 THEN PRINT"DON'T BE SILLY": GOTO 620
640 INPUT "SET MAXIMUM LIMIT";l
650 INPUT "DO YOU WANT A PRINTOUT (Y/N)";b$
660 IF b$="Y" THEN GOSUB 760
670 FOR w=1 TO 1000:NEXT w
680 REM NOW PRINT THE TABLE
690 CLS
700 LOCATE 8,4:PRINT CHR$(24); " ";t;"ADDITION TABLE";CHR$(24
)
710 PRINT:PRINT:PRINT
720 FOR i=1 TO 1
730 PRINT TAB(5);t;"+";i;TAB(18);"=";t+i
740 NEXT
750 RETURN
760 LOCATE 8,4:PRINT#8," ";t;"ADDITION TABLE"
770 PRINT:PRINT:PRINT
780 FOR i=1 TO 1
790 PRINT#8, TAB(5);t;"+";i;TAB(18);"=";t+i
800 NEXT
810 RETURN
820 REM SUBTRACTION TABLE
830 CLS:PRINT"SUBTRACTION TABLE"
840 INPUT "WHICH TABLE";t
850 IF t<1 THEN PRINT "DON'T BE SILLY":GOTO 840
860 INPUT "SET MAXIMUM LIMIT";l
870 INPUT "DO YOU WANT A PRINTOUT (Y/N)";b$
880 IF b$="Y" THEN GOSUB 980
890 FOR w=1 TO 1000:NEXT w
900 REM NOW PRINT THE TABLE
910 CLS
920 LOCATE 8,4:PRINT CHR$(24); " ";t;"SUBTRACTION TABLE";CHR$(
24)
930 PRINT:PRINT:PRINT
940 FOR i=1 TO 1
950 PRINT TAB(5);t;"-";i;TAB(18);"=";t-i
960 NEXT
970 RETURN
980 LOCATE 8,4:PRINT#8," ";t;"SUBTRACTION TABLE"
990 PRINT:PRINT:PRINT
1000 FOR i=1 TO 1
1010 PRINT#8,TAB(5);t;"-";i;TAB(18);"=";t-i
1020 NEXT
1030 RETURN

```

More Software Reviews

Joe and Tony take a good look at three very popular games for the CPC464 – Roland in Time, 'Er-Bert and Flight Path 737. It does seem that a reviewer's job is a lot of fun!

Roland in Time

It was with a little trepidation that I was talked into buying one more in the series, called ROLAND IN TIME, for my elder daughter. Being a series programmer of course I never play games (!!!!) and would not be caught ever playing one or, heaven forbid, actually enjoying one.

A silence had descended on the house, believe me a rarity in itself. All I could hear was a rendition of "I love to go awandering" being played over and over again. Poking my head round the door I saw both my daughters' totally engrossed eyes glued to the monitor and not saying a word. Had I discovered the secret formula that all parents had been searching for for years? No way! *Roland* had achieved in a short time what I and countless other parents had been trying to achieve for years. Total attention.

This was certainly something that warranted further investigation. Waiting until everyone was in bed I loaded *Roland* into the 464 and prepared to study, purely in the interest of science of course, this phenomenon called *Roland in Time*. What a pleasant surprise. I could not believe the sheer variety and complexity of sprites that unfolded before my eyes. Screen by screen, more than fifty I was later to discover, as I wandered through time – Each with a pseudo historical name. Names like 'Down at Dollis Brook', 'Njorls Dragon Boat', 'The Roman Fort', all unfolded as I made my way, not I must say without vast loss of lives.

The object of the game is to gather

crystals and, judging by the counter, there must be hundreds of them. But that did not seem to be to important. I was just enjoying going from location to location just to see what was there. The variety of characters and objects that you meet is truly amazing. Everything from trains to helicopters. Lifts of all kinds, some obvious and some not so obvious. Every location has to be explored to find the hidden paths and obstacles. Some of the crystals are easy to reach and some seem almost impossible to obtain. As you travel from location to location, by the way you can start at any one of ten, various types of monsters are waiting to greet you!!!! Be prepared for an attack as soon as you enter another zone. Only in the early screens are they a little more friendly. The sound effects are adequate and if "I Love to go awandering" gets on your nerves it can be switched off.

If you love adventures this is the game for you. If you prefer arcade type of games this is also for you. You make of it what you will. It is the first of many I hope, that at last is starting to show the true capabilities of the 464. What more can I say. Closet players come in from the cold. The CPC 464 can now truly hold its head high alongside all other games machines. This has got to be one of the most engrossing, exciting games that I have ever played and on a scale of one to ten rates eleven. I can only hope that other software houses follow the example of Gemsoft in producing what will become a classic in the games area. Now has anybody got any tips for getting past that darned Pterodactyl on the way to The End of Time

Flight Path 737

FLIGHT PATH 737 has been around on other computers for some time now, and though the graphics and sound effects are not what you would call exciting they are quite effective. You can use Joystick/keyboard or the keyboard alone.

The cockpit layout is a reduced version of the real thing and the dials are easy to read. The view from the cockpit, as you take off and fly, continually changes. The impression of flight is very good.

There are six levels, each level developing your flying skills. The first level is quite simple. You have to take off, fly over a mountain and land on the other side. As your skills develop you can advance to the other levels which contain flying hazards such as crosswinds, engine fires etc. At the completion of the flight your score is shown and advice offered as to whether you are ready to advance to other levels.

The placement of the command keys is a little difficult but practise soon makes perfect Fuller instructions would have been helpful and close reading is required to find all the key instructions. Key response is slow and can lead to frustration when you think that, for example, the undercarriage or flaps are up when they are not.

Overall FLIGHT PATH 737 should hold your interest and if you like technical, more than knock 'em down type of games, will keep you engrossed for hours. Now if you will excuse me I have a Plane to fly.

Er-Bert

Sproooooiiiiinnnggggggggg Well well what a surprise, Coily just got me again, I say to myself not feeling at all surprised. After all it is only the umpteenth time that he has killed me. Not to mention Boris who has thrown in his share of killings. Oh, by the way Coily is a snake and Boris is a gorilla who both seem to have homicidal tendencies. The problem is that they normally vent these on poor little Er-Bert who you are trying to guide around a pyramid. As if this wasn't enough, Er-Bert also has to avoid the large purple balls that are constantly cascading down the pyramid, and watch out for black holes while being careful not to jump over the edge.

Er-Bert is a very good copy of an arcade game with a similar name. The idea of this game is very simple - change all the yellow blocks on the pyramid to blue by guiding little Er-Bert onto them. Each time he jumps onto a yellow block it magically turns blue. Simple I here you say. Don't you believe it!

Anyway, back to the description of the game. The graphics are excellent and there is nothing jerky about the movement of the nasties or Er-Bert.

The game consists of four screens, the first, as already mentioned, is a pyramid. the next two screens have steps and paths leading everywhere, at which you have to make sure you don't get caught down a dead-end with Coily or Boris on your heels. The final screen is a solid square of blocks. After this screen you return to the first screen but the game speeds up a little.

As I said earlier you must change the yellow blocks to blue without letting little Er-Bert get killed by Coily and friend. But don't worry they don't have it all their own way. On each screen you have two coloured discs, jump onto these when you are being pursued by Coily and Boris and they will jump after you. Unfortunately for them they miss the disc and disappear into the bottomless depths of your monitor. But don't think for a minute that you have got rid of your troubles, they are soon replaced by identical twins. The colour discs are not the only means of help for Er-Bert. Collect the rotor hats and Er-Bert can fly vertically up or down two blocks.

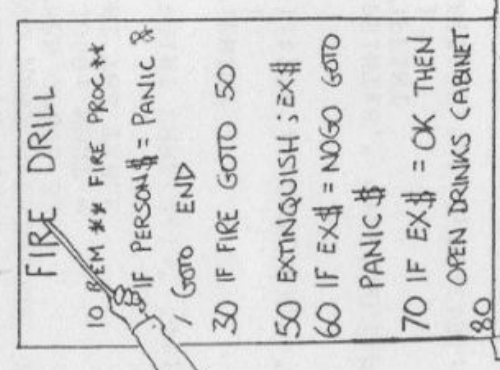
You can also collect double points if you pick up Boris' banana. But be warned, he doesn't like people

holding his banana and he may chase you even more than normal. If this happens you can drop his banana and run for your life.

Now you may think you are getting pretty good after you have played this game for a while so you may try a higher level. There are in fact 9 different levels for you and Er-Bert to test your skills. You also have two speeds to try, fast or slow. Then add the little problem of changing the colour of the blocks twice. First you change the blocks to blue and then to white, Wonderful eh?

And for people with joysticks, you can use them on this game. But don't worry if you haven't I personally found it easier to play from the keyboard. One useful tip if you do play this way, I found it a great help to put stickers onto the keys you can use with arrows drawn on them.

What more can I say except that Er-Bert is habit forming. And we take no responsibility if you miss meals because you can't drag yourself away from your computer. It is certainly one of the best computer games that I have played in a long time. I would recommend it to anyone who likes arcade games and is great value at \$19.95.



FIRE DRILL

10 REM ## FIRE PROC ##

IF PERSON ## = PANIC &
/ GOTO END

30 IF FIRE GOTO 50

50 EXTINGUISH ; EX #

60 IF EX # = NOGO GOTO
PANIC #

70 IF EX # = OK THEN
OPEN DRINKS CABINET

80



PROGRAMMING
ROOM

W H A M

Runner

This is a game which provides you with the opportunity to develop it further – for a start you can redefine certain characters giving it more animation. We will give a choice of software to the value of \$25 to the best enhanced version we receive by the end of June 1985. The instructions for this original version are contained within the program.

```
10 CLS
20 PRINT:PRINT:PRINT"
30 PRINT:PRINT"
40 PRINT:PRINT"You must guide the man down through the level
s to the bottom right hand corner of the screen. But you mu
st avoid the balls that roll backwards and forwards at ea
ch level. You only have one hole in the floor to drop down to
the next level
50 PRINT"When over the hole you will fall through automatical
ly. But you are able to jump back up through the holes by pr
essing the space bar. You also have one safe spot on eac
h level at the far right of the screen. Use your cursor key
s to move";
60 PRINT"left and right and the space bar to jump up a level.
"
70 PRINT"press any key to start."
80 a$=INKEY$:IF a$=""THEN 80
90 INK 0,0:INK 1,6:INK 2,2
100 PAPER 0:PEN 1
110 CLS
120 BORDER 9
130 h=1:v=2:e=2:x=1:z=1:ee=3
140 FOR i=1 TO 12
150 FOR y=1 TO 40
160 LOCATE h,v:PRINT CHR$(208)
170 h=h+1
180 NEXT
190 v=v+2:h=1
200 NEXT
210 PEN 2
220 DIM w(24),r(24),z(12),qq(12),tt(12)
230 FOR u=1 TO 24 STEP 2
240 w(u)=e
250 e=e+2
260 NEXT
270 FOR y=1 TO 24 STEP 2
280 r(y)=INT(39*RND(1))+1)
290 LOCATE r(y),w(y):PRINT" "
300 NEXT
310 FOR aa=1 TO 12
320 zz(aa)=ee:qq(aa)=1:LOCATE qq(aa),zz(aa):PRINT CHR$(202)
330 ee=ee+2
340 NEXT
350 LOCATE x,z:PRINT CHR$(249)
360 a$=INKEY$:IF a$=""THEN 390
```

RUNNER"

written by Philip Riley"


```

370 a=ASC(a$)
380 IF a=32 THEN 1410
390 IF a$=CHR$(242) THEN LOCATE x,z:PRINT " ":x=x-1:IF x<1 TH
EN x=x+1:LOCATE x,z:PRINT CHR$(249) ELSE LOCATE x,z:PRINT CH
R$(249)
400 IF a$=CHR$(243) THEN LOCATE x,z:PRINT " ":x=x+1:IF x>39 T
HEN x=x-1:LOCATE x,z:PRINT CHR$(249) ELSE LOCATE x,z:PRINT C
HR$(249)
410 ON z GOTO 1030,1,1040,1,1060,1,1080,1,1100,1,1120,1,1140
,1,1160,1,1180,1,1200,1,1220,1,1240,1,1260
420 ON z GOTO 430,1,430,1,430,1,430,1,480,1,530,1,580,1,630,1,680,
1,730,1,780,1,830,1,880,1,930
430 LOCATE qq(1),zz(1):PRINT " ":IF tt(1)=1 THEN qq(1)=qq(1)+
1 ELSE qq(1)=qq(1)-1
440 IF qq(1)<1 THEN qq(1)=qq(1)+1:tt(1)=1
450 IF qq(1)>38 THEN qq(1)=38:tt(1)=2
460 LOCATE qq(1),zz(1):PRINT CHR$(202)
470 IF z=1 THEN 1020
480 LOCATE qq(2),zz(2):PRINT " ":IF tt(2)=1 THEN qq(2)=qq(2)+
1 ELSE qq(2)=qq(2)-1
490 IF qq(2)<1 THEN qq(2)=1:tt(2)=1
500 IF qq(2)>38 THEN qq(2)=38:tt(2)=2
510 LOCATE qq(2),zz(2):PRINT CHR$(202)
520 IF z=3 THEN 1020
530 LOCATE qq(3),zz(3):PRINT " ":IF tt(3)=1 THEN qq(3)=qq(3)+
1 ELSE qq(3)=qq(3)-1
540 IF qq(3)<1 THEN qq(3)=1:tt(3)=1
550 IF qq(3)>38 THEN qq(3)=38:tt(3)=2
560 LOCATE qq(3),zz(3):PRINT CHR$(202)
570 IF z=5 THEN 1020
580 LOCATE qq(4),zz(4):PRINT " ":IF tt(4)=1 THEN qq(4)=qq(4)+
1 ELSE qq(4)=qq(4)-1
590 IF qq(4)<1 THEN qq(4)=1:tt(4)=1
600 IF qq(4)>38 THEN qq(4)=38:tt(4)=2
610 LOCATE qq(4),zz(4):PRINT CHR$(202)
620 IF z=7 THEN 1020
630 LOCATE qq(5),zz(5):PRINT " ":IF tt(5)=1 THEN qq(5)=qq(5)+
1 ELSE qq(5)=qq(5)-1
640 IF qq(5)<1 THEN qq(5)=1:tt(5)=1
650 IF qq(5)>38 THEN qq(5)=38:tt(5)=2
660 LOCATE qq(5),zz(5):PRINT CHR$(202)
670 IF z=9 THEN 1020
680 LOCATE qq(6),zz(6):PRINT " ":IF tt(6)=1 THEN qq(6)=qq(6)+
1 ELSE qq(6)=qq(6)-1
690 IF qq(6)<1 THEN qq(6)=1:tt(6)=1
700 IF qq(6)>38 THEN qq(6)=38:tt(6)=2
710 LOCATE qq(6),zz(6):PRINT CHR$(202)
720 IF z=11 THEN 1020
730 LOCATE qq(7),zz(7):PRINT " ":IF tt(7)=1 THEN qq(7)=qq(7)+
1 ELSE qq(7)=qq(7)-1
740 IF qq(7)<1 THEN qq(7)=1:tt(7)=1
750 IF qq(7)>38 THEN qq(7)=38:tt(7)=2

```

```

760 LOCATE qq(7),zz(7):PRINT CHR$(202)
770 IF z=13 THEN 1020
780 LOCATE qq(8),zz(8):PRINT " ":IF tt(8)=1 THEN qq(8)=qq(8)+
1 ELSE qq(8)=qq(8)-1
790 IF qq(8)<1 THEN qq(8)=1:tt(8)=1
800 IF qq(8)>38 THEN qq(8)=38:tt(8)=2
810 LOCATE qq(8),zz(8):PRINT CHR$(202)
820 IF z=15 THEN 1020
830 LOCATE qq(9),zz(9):PRINT " ":IF tt(9)=1 THEN qq(9)=qq(9)+
1 ELSE qq(9)=qq(9)-1
840 IF qq(9)<1 THEN qq(9)=1:tt(9)=1
850 IF qq(9)>38 THEN qq(9)=38:tt(9)=2
860 LOCATE qq(9),zz(9):PRINT CHR$(202)
870 IF z=17 THEN 1020
880 LOCATE qq(10),zz(10):PRINT " ":IF tt(10)=1 THEN qq(10)=qq
(10)+1 ELSE qq(10)=qq(10)-1
890 IF qq(10)<1 THEN qq(10)=1:tt(10)=1
900 IF qq(10)>38 THEN qq(10)=38:tt(10)=2
910 LOCATE qq(10),zz(10):PRINT CHR$(202)
920 IF z=19 THEN 1020
930 LOCATE qq(11),zz(11):PRINT " ":IF tt(11)=1 THEN qq(11)=qq
(11)+1 ELSE qq(11)=qq(11)-1
940 IF qq(11)<1 THEN qq(11)=1:tt(11)=1
950 IF qq(11)>38 THEN qq(11)=38:tt(11)=2
960 LOCATE qq(11),zz(11):PRINT CHR$(202)
970 IF z=21 THEN 1020
980 LOCATE qq(12),zz(12):PRINT " ":IF tt(12)=1 THEN qq(12)=qq
(12)+1 ELSE qq(12)=qq(12)-1
990 IF qq(12)<1 THEN qq(12)=1:tt(12)=1
1000 IF qq(12)>30 THEN qq(12)=30:tt(12)=2
1010 LOCATE qq(12),zz(12):PRINT CHR$(202)
1020 GOTO 360
1030 IF x=r(1) THEN 1280 ELSE 420
1040 IF x=r(3) THEN 1280
1050 IF x=qq(1) THEN 1600 ELSE 420
1060 IF x=r(5) THEN 1280
1070 IF x=qq(2) THEN 1600 ELSE 420
1080 IF x=r(7) THEN 1280
1090 IF x=qq(3) THEN 1600 ELSE 420
1100 IF x=r(9) THEN 1280
1110 IF x=qq(4) THEN 1600 ELSE 420
1120 IF x=r(11) THEN 1280
1130 IF x=qq(5) THEN 1600 ELSE 420
1140 IF x=r(13) THEN 1280
1150 IF x=qq(6) THEN 1600 ELSE 420
1160 IF x=r(15) THEN 1280
1170 IF x=qq(7) THEN 1600 ELSE 420
1180 IF x=r(17) THEN 1280
1190 IF x=qq(8) THEN 1600 ELSE 420
1200 IF x=r(19) THEN 1280
1210 IF x=qq(9) THEN 1600 ELSE 420
1220 IF x=r(21) THEN 1280

```



```

1230 IF x=qq(10) THEN 1600 ELSE 420
1240 IF x=r(23) THEN 1280
1250 IF x=qq(11) THEN 1600 ELSE 420
1260 IF x=39 THEN 1330
1270 IF x=qq(12) THEN 1600 ELSE 420
1280 FOR q=1 TO 2
1290 LOCATE x,z:PRINT " "
1300 z=z+1:LOCATE x,z:PRINT CHR$(249)
1310 NEXT
1320 GOTO 420
1330 FOR q=1 TO 200:NEXT
1340 PEN 1
1350 CLS:PRINT"well done you have won"
1360 PRINT"another game <y or n>"
1370 a$=INKEY$
1380 IF a$="y"THEN ERASE r,w,qq,zz,tt:GOTO 110
1390 IF a$="n"THEN CLS:END
1400 GOTO 1370
1410 ON z GOTO 1020,1,1420,1,1430,1,1440,1,1450,1,1460,1,147
0,1,1480,1,1490,1,1500,1,1510,1,1520,1,1530
1420 IF x=r(1) THEN 1540 ELSE 1020
1430 IF x=r(3) THEN 1540 ELSE 1020
1440 IF x=r(5) THEN 1540 ELSE 1020
1450 IF x=r(7) THEN 1540 ELSE 1020
1460 IF x=r(9) THEN 1540 ELSE 1020
1470 IF x=r(11) THEN 1540 ELSE 1020
1480 IF x=r(13) THEN 1540 ELSE 1020
1490 IF x=r(15) THEN 1540 ELSE 1020
1500 IF x=r(17) THEN 1540 ELSE 1020
1510 IF x=r(19) THEN 1540 ELSE 1020
1520 IF x=r(21) THEN 1540 ELSE 1020
1530 IF x=r(23) THEN 1540 ELSE 1020
1540 FOR ff=1 TO 2
1550 LOCATE x,z:PRINT " "
1560 z=z-1:LOCATE x,z:PRINT CHR$(249)
1570 NEXT
1580 LOCATE x,z:PRINT " :x=x+1:LOCATE x,z:PRINT CHR$(249)
1590 GOTO 1020
1600 FOR hh=1 TO 500:NEXT
1610 PEN 1
1620 CLS:PRINT"you have been killed. Bad luck"
1630 GOTO 1360

```

Binary to Hexadecimal Tutor

Newcomers will find this Tutor from Steve Miles of great assistance in converting a number into Binary and then to Hexadecimal. However, practise makes perfect, so test yourself as often as possible.

```
10 REM *****
20 REM BINARY/HEXADECIMAL TUTOR
30 REM *****
40 REM BY STEVE MILES
50 REM *****
60 CLS
70 POKE 46312,255
80 INK 0,1:INK 1,24:INK 2,26:INK 3,21
90 WINDOW #1,1,40,1,11
100 PAPER #1,0:PEN #1,2:CLS #1
110 WINDOW #2,14,30,13,14
120 PAPER #2,0:PEN #2,3:CLS #2
130 WINDOW #3,14,30,13,14
140 PAPER #3,0:PEN #3,1:CLS #3
150 WINDOW #4,14,36,16,18
160 PAPER #4,0:PEN #4,3:CLS #4
170 WINDOW #5,14,36,21,21
180 PAPER #5,0:PEN #5,1:CLS #5
190 GOSUB 640
200 PAPER 0:PEN 2
210 LOCATE 14,15:PRINT"128 64 32 16 8 4 2 1"
220 LOCATE 14,19:PRINT" 8 4 2 1 8 4 2 1"
230 PAPER 0:PEN 1
240 RANDOMIZE TIME
250 n=INT(RND*255)+1
260 LOCATE 1,14:PRINT n
270 PAPER 0:PEN 3
280 LOCATE 2,17:PRINT"BINARY:"
290 FOR zz=2 TO 23 STEP 3
300 b$(zy)=INKEY$:IF b$(zy)=" THEN 300
310 LOCATE #4,zz,2:PRINT #4,b$(zy):zy=zy+1
320 IF zy=8 THEN 340
330 NEXT zz
340 zy=0:b$=b$(0)+b$(1)+b$(2)+b$(3)+b$(4)+b$(5)+b$(6)+b$(7)
350 LOCATE 2,19:PRINT b$
360 IF b$<>BIN$(n,8) THEN 370 ELSE 410
370 tb$="Wrong! try again."
380 LOCATE #2,14,13:PRINT #2,tb$
390 FOR t=1 TO 1000:NEXT t:CLS #4:CLS #2
400 GOTO 280
410 GOSUB 690
420 PAPER 0:PEN 1
430 LOCATE 15,23:PRINT"A(10) B(11) C(12)"
440 LOCATE 15,25:PRINT"D(13) E(14) F(15)"
450 PAPER 0:PEN 1
```



```

460 LOCATE 2,21:PRINT"HEXADECIMAL:"
470 FOR aa=2 TO 14 STEP 12
480 h$(az)=INKEY$:IF h$(az)=" THEN 480
490 LOCATE #5,aa,1:PRINT #5,h$(az):az=az+1
500 IF az=2 THEN 520
510 NEXT aa
520 az=0:h$h=h$(0)+h$(1)
530 LOCATE 2,23:PRINT "&"
540 LOCATE 3,23:PRINT h$h
550 IF h$h<>HEX$(n,2) THEN 560 ELSE 600
560 th$="Wrong! try again."
570 LOCATE #3,14,13:PRINT #3,th$h
580 FOR t=1 TO 1000:NEXT t:CLS #5:CLS #3
590 GOTO 460
600 CLS #3:PAPER 0:PEN 3
610 LOCATE 2,12:PRINT"Press <SPACE BAR> for another number."
620 z$=INKEY$:IF z$=" THEN 620
630 IF z$=" THEN RUN
640 LOCATE #1,1,1:PRINT #1," To convert the random number to
    BINARY, look at the line of numbers 128 to 1.
650 LOCATE #1,1,3:PRINT #1," Any number that adds up TO th
    e exact total of the random number, input a <1> under the co
    rrect number."
660 LOCATE #1,1,6:PRINT #1," Where a number does not add up
    to the random total, input a <0>."
670 LOCATE #1,1,8:PRINT #1,"For example 199 would be 1 1 0 0
    0 1 1 1 that is 128+64+4+2+1 as numbers 32,16,8 in any comb
    ination would not add up to 199."
680 RETURN
690 CLS #2:CLS #1
700 PAPER #1,0:PEN #1,2
710 LOCATE #1,1,1:PRINT #1," To convert to HEXADECIMAL, wher
    e a <1> matches with a number in the LOWER row, add the 1st
    set of four numbers (8 to 1) and input."
720 LOCATE #1,1,5:PRINT #1," Repeat the same for the 2nd set
    of four numbers. Where the total of any set exceeds <9>, in
    put the LETTER code equivalent to the numbers in the bracket
    s."
730 LOCATE #1,1,9:PRINT #1," For example binary no. 1 1 0 0
    0 1 1 1 would be C7 in hexadecimal."
740 RETURN

```

Don't pass the luck

— earn it

an observation or whatever
 — merely put pen to paper
 and direct your comments
 to the Editor.

LETTERS published in The
 Amstrad User attract a pay-
 ment of \$5. We don't mind
 whether you want to pass
 on a tip, have a moan, make

CARTOONS can also earn
 you \$5. Just make sure
 they are drawn in black ink
 on white paper and not
 folded when you submit

them to the Editor, who may
 well be folded when he
 sees them.

Please send your letters or
 cartoons to: The Editor, The
 Amstrad User Suite 4a,
 33-45 The Centreway,
 Blackburn Road,
 Mt. Waverley 3149.

Artificial Intelligence

Following on from the article last month covering 'The Fifth Generation', Arthur Harris now deals with the complementary subject of AI.

This is the second report based on a series of articles from New Scientist magazine. The issue of 29 November, 1984 still hasn't turned up. I will report anything significant from that issue when it arrives.

Firstly, what is artificial intelligence (AI)? There were a series of articles in issues of New Scientist prior to about 1980 which discussed the development work going on in this field. The general gist of these articles was that AI was the ability to have a computer reason out the answer to a problem, based on the information contained in its data banks. The first problem being tackled was that of interpretation of natural language. This involves deciding which shade of meaning should be applied to a word, according to the context in which it is being used.

A famous joke in this area involves work at one of the English Universities in the mid-1950's. Their computer had been programmed to translate from Russian to English and vice versa. After "bugs" had been ironed out, using everyday phrases, it was decided to start some more rigorous testing. The phrase chosen for this was "The spirit was willing but the flesh was weak". This was translated into Russian and back into English. The computer solemnly printed out the answer - "The vodka was good but the meat was rotten". Whoops - back to the drawing board!

In the circa 1980 articles, it appeared that work had progressed to the stage where attempts were being made to sort out the semantics of the saying, "Time flies like an arrow". There are three meanings that can, strictly, be applied to this statement,

depending on which of the first three words is considered to be the verb. How to make a computer decide which is the generally accepted (and probably correct) meaning? Further work on the drawing board!

This purely academic view of AI is the one generally accepted by the public. A different, but equally accepted, view is that of the perfect game playing machine which can beat all human comers at the "serious" games of chess, go, etc. Another use of AI, accepted by the public, is of the perfect diagnostic computer used by your local GP when you go to him with some rare disease. Other examples of "legitimate" AI include the advent of the housekeeping robot which is all-seeing, all-hearing, all-doing - "Your wish is my command" type of machines.

The welter of publicity given to the above examples has influenced public thinking in the huge burst of publicity given by the media to the race for the fifth generation computer and its ability to achieve AI.

As indicated in the last article, some of the projects being fostered around the world are proceeding along these lines. However, the bodies that have the funds to sink into AI research in larger amounts than anyone else are the Defence Departments of the various countries.

There is general agreement, even among the faithful, that AI is being oversold. One emerging issue is that AI will place computer scientists in much the same dilemma as atomic scientists are already. Because funds come from military establishments, the work on AI will become increasingly important in the deployment of

nuclear and other weapons.

The belief that AI is being overplayed comes from the ranks of computer scientists, academics, industry analysts, company executives and AI researchers themselves. There is genuine concern that because public expectations have been raised so high, there will be a backlash from investors and funding agencies when researchers cannot deliver the goods. This has previously occurred in other fields following extravagant claims by researchers.

Some examples of AI that are constantly being paraded, to exemplify the benefits to be obtained, are simply expert systems working with a restricted set of data. The two mentioned in the articles are in geology and medicine. These expert systems are also restricted by the fact that they use facts in association with a set of rules. Further, the existing systems have all been hand-tailored for a single purpose. There is no one program that could, on its own, have achieved the results of each of these systems.

The revolutionary advances needed to achieve AI require the programming of the human traits of perception, criticism and multiple levels of reasoning.

The annual meeting of the American Association for Artificial Intelligence, in August, 1984, was attended by 3000 people, with another 1000 being turned away, but presented only 70 research papers. It was estimated that a similar meeting in the area of materials science would have attracted at least 10 to 20 times that number of research papers. Obviously, the experts in AI are still very thin on the ground. The head of MCC (discussed

last month) was not impressed by the standard of the papers at the August meeting.

A former IBM scientist, Herbert Grosch, has predicted that voice input, robotic vision and natural language are not possible within the next 40 - 50 years. It has been predicted that the programs available in 10 years time, although extremely sophisticated by today's standards, will not be thought to be truly smart or truly intelligent.

One of the reasons for the problems surrounding perceptions of AI is given as the name itself. Had it been called symbolic programming, public interest probably would not be as high.

There is a huge gap between the research laboratories and those marketing AI. One company president of a marketing company claimed that 99% of the executives at the August meeting would not have understood the prize-winning paper - "Semantic Process for Syntactic Disambiguation". The whole situation was likened to the discussions on quantum mechanics that preceded the introduction of the semiconductor industry by more than 40 years.

Market analysts are impressed by "do-it-yourself" AI software. This allows non-experts to develop an expert system using a "shell" or "framework". A variation on this theme is a "dynamic" or "real-time" expert system which responds to changing facts. LISP Machines Inc. has released a program which can monitor manufacturing processes and suggest answers to problems.

While all the confusion reigns in the academic and business worlds, The Defence forces are quietly pushing ahead with their own version of AI. As mentioned last month, in the US, DARPA is working on three projects. These are: an autonomous "robot" land vehicle, a computerised co-pilot for fighter planes and a naval battle management system. DARPA reasons that each of these machines requires the development and combination of the elements of AI

machines - expert systems, vision, speech recognition and understanding of natural language.

The speed at which a computer must process information to accomplish these tasks is mind-boggling. For example, using current von Neumann processors, a computer would have to process up to 1E11 instructions per second just to "see" and interpret the terrain. The best machines today handle about 4E8 steps per second. After this, the machine must navigate using an expert system consisting of maybe 6500 rules, processed at a rate of about 7000 rules per second. Current expert systems contain about 2000 rules, processed at a rate of about 100 per second.

It is clear that the weapons of the future require symbolic "inference" programs that process concepts rather than the current digital on-off signals. This need applies to all systems that hope to use AI for any purpose.

At Columbia University, in New York, scientists are developing a rule processor. This machine will use 1023 individual processors that handle rules at 10 times the current rate. Another project at this University involves 1 million processors linked in a tree-hierarchy, each with its own memory.

In the UK, defence planners are already experimenting with AI systems that analyse signals from radar, sonar, infra-red detectors and passive detection systems. The idea is to use a computer to direct the fighting forces.

Well, there it is - the bare bones of AI. There are obviously high expectations of what research will achieve. How long it will be before we see the results of this research is a matter of pure conjecture. Vast sums of money are being expended in the hope of achieving a breakthrough. The only consolation I can find in the military involvement is that, traditionally, developments in this area eventually flow on for the good of the masses in peace time. Here's hoping we are still around to reap those benefits.

The Trials of Tony Blakemore

having trouble with the meetings and see if one of the more experienced programmers can help solve the problem. If you want to contact a club in your area please ring me on 878 6212 and I will put you in contact with your nearest club. All clubs have members from beginners to experienced and love to see new members joining, regardless of their expertise.

If you have studied the error messages listing (Appendix VIII) you may have noticed that they are all numbered. There is a reason for this. LOCOMOTIVE BASIC provides, as well as error messages, facilities to trap errors and execute your own remedies. Four commands are available ON ERROR, ERROR, ERR and ERL (chapter 8 pages 16-17).

To comprehensively cover all known bugs would take all of the next twenty issues and unfortunately I don't have the space. The areas covered so far should enable you to enter and run most programs that you want to at this stage. I cannot stress enough that experience is the best teacher. Only by entering and studying other peoples programs will you gain the experience to enable you to understand LOCOMOTIVE BASIC and all of its' commands. As you gain more experience you will find that the errors diminish, and you will correct typographical errors as you enter the programs.

Next month we will look at two areas that are not very well explained in the manual. Defining key functions and defining your own characters. I would like some letters on the areas that you would like explained in future issues. This is your column and the more feedback I get on the problems that you are encountering the more I can try and help you to make programming the rewarding experience that it has become for many thousands of people.

4th Chance

•COMPETITION•

\$2500 worth of prizes to be won over four classes

Class 1

Best overall program
Wins an AWA Video Recorder

Class 2

Best amusement/adventure
Wins a new DD1 disk drive

Class 3

Best educational software
Wins a new DD1 disk drive

Class 4

Best business software
Wins a new DD1 disk drive

How to enter

Think about your program and map it out in a series of events or features. Write the program onto cassette based around these events and check that the program runs as intended. Once you are satisfied, send a copy of the cassette in a suitable envelope along with the following:

- 1 A brief summary of the program in 500 words or less.
- 2 A clear program listing if available.
- 3 A stamped, self addressed envelope of adequate dimensions if you would like your entry returned.
- 4 Your name and address.

You may make as many submissions as you want, but no entrant may win more than one prize.

Conditions of Entry

- 1 All entries must run on a CPC464, and must include a cassette copy of the program (plus loading instructions where necessary), a brief summary of the program and its purpose and, if possible, a full listing.
- 2 All entries must arrive by 15th August 1985, and winners will be printed in the October edition of The Amstrad User.
- 3 The decision of the judges is final.
- 4 It is a condition of entry that all entrants have exclusive ownership of the copyright of the material submitted, and the winners agree to assign all copyright in the winning submissions to The Amstrad User. Where the entrant is more than one individual, then one person must be nominated and empowered to act on behalf

of the entire group. All entrants must undertake not to submit the same or a similar program to any other magazine, publisher or organisation until after the announcement of the winning entries.

We, The Amstrad User, may offer to publish programs other than the winners in the magazine or as commercial software, in which case we will agree terms on an individual basis with the author(s) concerned. We reserve the right to amend, alter or revise any program we publish.

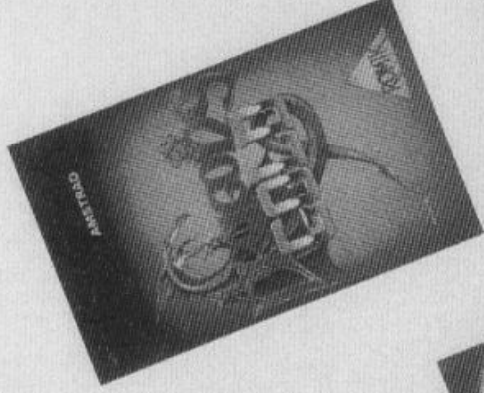
No employees of The Amstrad User or Strategy Publications, or their relatives may enter this competition.

The Amstrad User cannot be held responsible for any loss or damage to any submission.

No entrant may win more than one prize.

AMSTRAD ARCADE ALLEY

**Five Exciting Quality Games from Strategy
at only \$19.95**



SPECIAL OFFER FOR MAY 1985 ONLY
Any one - \$19.95 Any three - \$39.95
All five - \$59.95
(Prices include postage and packing)

Name

Address

..... Postcode

Creditcard

or cheque for \$

(Tick which required)

- TER-BERT
- DRAGONS GOLD
- ALIEN BREAK-IN
- ATOM SMASHER
- 3D MONSTER CHASE

SAVE \$450

Q. A.

How many computer magazines would I have to buy to get at least 30 pages of information and program listings for the new Amstrad CPC464?

On average, most popular magazines will devote 2½ pages to the AMSTRAD CPC464. This means you will need to buy 12 magazines at a cost of around \$40 per month, or **\$480 per year!**

Q. A.

Surely there must be a more sensible and cheaper way of getting the information I need?

There is.

THE AMSTRAD USER is a brand new monthly publication packed with articles, reviews, listings, hints etc. for Amstrad users only, and costs just **\$30 a year.**

Q. A.

How can I get a copy delivered to my home each month with an optional cassette containing all the program listings?

By completing and returning this order.

Please send me THE AMSTRAD USER for 12 months

Magazine only: \$30 Magazine and cassette: \$70

Payment by: Cheque Bankcard or Mastercard

Card number _____ Expiry date

Name Phone

Address

..... Postcode

Signed **Please start with Issue No**

Return to THE AMSTRAD USER, Suite 4a, 33-45 The Centreway
Blackburn Road, Mt. Waverley, Vic 3149 Tel 03-233 9227

(OVERSEAS PRICES ON APPLICATION TO ABOVE
ADDRESS)