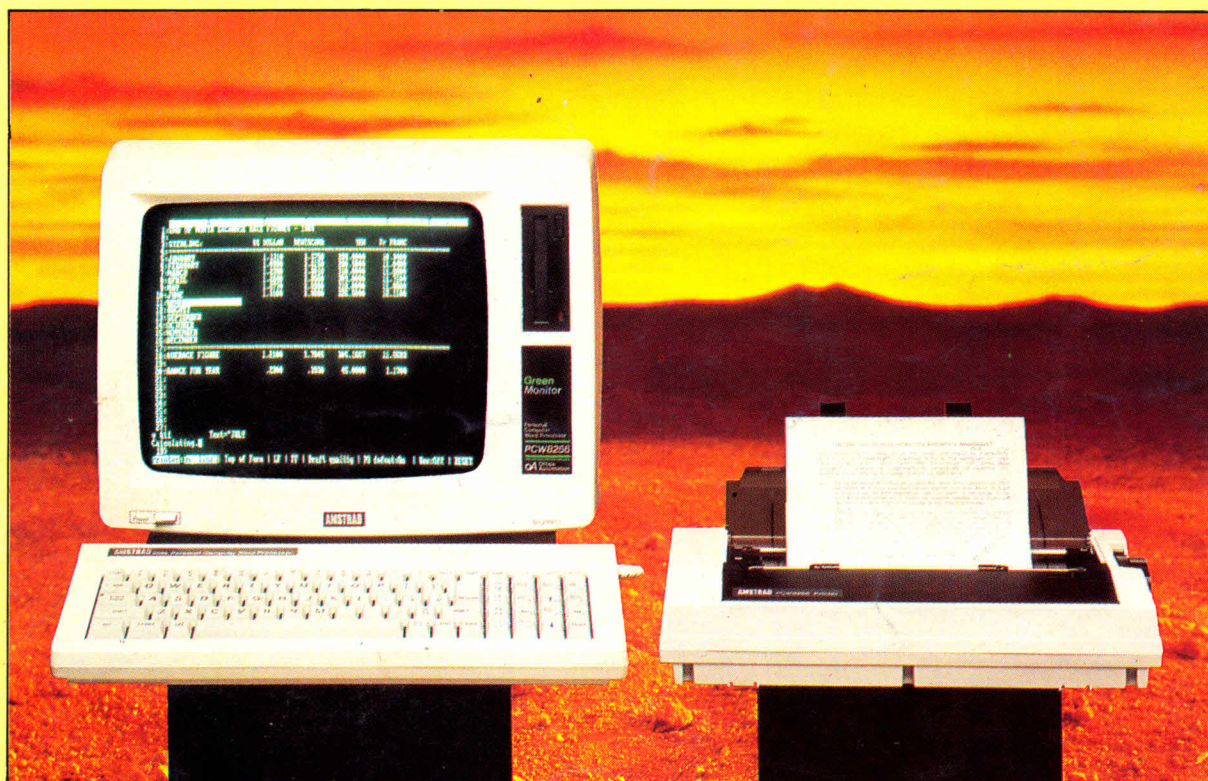


# THE AMSTRAD USER

Issue No. 11

\$3.50

December 1985



- 8 PAGE SOFTWARE REVIEW SUPPLEMENT
- A REVIEW OF THE NEW PCW-8256
- USER GROUP INFORMATION
- FIGHTING FORTS - A GAME

FOR THE NOVICE & EXPERIENCED USER

Registered by Australia Post - Publication No VBP7017



# BACK COPIES

They are disappearing *FAST* - and we wont be printing anymore! If you've missed any copies order NOW.

	MAG	TAPE		MAG	TAPE
Number 1 - Feb	.....	.....	Number 6 - July	.....	.....
Number 2 - Mar	.....	.....	Number 7 - Aug	.....	.....
Number 3 - Apr	.....	.....	Number 8 - Sep	.....	.....
Number 4 - May	.....	.....	Number 9 - Oct	.....	.....
Number 5 - Jun	.....	.....	Number 10 - Nov*	.....	.....

The cost of a single magazine is \$3.00 plus 50 cents towards postage and the cost of a single tape is \$4.50 plus 50 cents towards postage.

*\*Note: The cost of Issue Number 10 (November) and subsequent issues is \$3.50 plus 50 cents towards postage.*

We also have a limited number of back copies of the English Amstrad User - probably the only stocks left in Australia - so plug the gaps in your collection before it's too late!

Jan/Feb	...	\$4.00 incl. p & p	July	...	\$4.50 incl. p & p
March	...	\$4.00 incl. p & p	August	...	\$4.50 incl. p & p
April	...	\$4.00 incl. p & p	Sept	...	\$4.50 incl. p & p
May	...	\$4.00 incl. p & p	October	...	\$4.50 incl. p & p
June	...	\$4.00 incl. p & p	Nov	...	\$4.50 incl. p & p

*Please send your order and remittance to:*  
**Strategy Publications,**  
2/33 The Centreway, Blackburn Road  
Mt. Waverley, Victoria 3149      Tel: (03) 232 7055  
(Bankcard and MasterCard accepted)

# BACK COPIES



# THE AMSTRAD USER

Issue No. 11      December 1985

---

## CONTENTS

Letters.....	3
A Different Approach to Program Design - Part 2 .....	6
A Retailer's Point of View - interview with Billy Guyatts....	12
Hall of Fame.....	14
FIGHTING FORTS - a game from Lindsay Allen.....	15
<i>8 Page SOFTWARE REVIEW SUPPLEMENT</i>	
User Group Information.....	22
The Learning Centre - Introduction to Music Part 3.....	24
Tony Blakemore Goes Somewhere Else.....	26
A Special Christmas Message - for Tape Subscribers.....	28
Review of the new PCW 8256.....	29
The Amstrad User Software Review Index.....	31
Discounted Books for Subscribers.....	32

For Tape Subscribers, the programs/routines can be found at these approximate counter readings:

Side 1 - FFLOADER:4, FFORTS:13, AMSFILE1:117, SNIGHT1:132, SNIGHT3:146, CLOCKPT2:169

Side 2 - XMASLDR:4, XMASPIC:12

---

*All enquiries and contacts concerning this Publication should be made to The Amstrad User, Shop 2, 33 The Centreway, Blackburn Road, Mt. Waverley, Victoria 3149, Australia. [Telephone: (03) 232 7055].*

*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 \$3.50 is the recommended retail price only. The subscription rate (for Australia only) is \$35.00 for 12 issues of the magazine only, or \$75.00 for 12 issues of the magazine plus tape containing all programs*

*appearing in that issue. Postage is included in the above prices. Overseas prices available upon application.*

*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 welcome from readers or other interested parties. In most circumstances the following payments will apply to published material: Letters-\$5.00, Cartoons-\$5.00 and a rate of \$10.00 per page for programs, articles etc.*

*Contributions will not be returned unless specifically requested coupled with suitable stamped and addressed padded bag (for tapes) or envelope.*

---

# THE AMSTRAD USER

G'day,

As I write the last editorial for 1985, I have my fingers crossed that our first venture into colour is not our last. The production of The Amstrad User is such that the editorial is printed before the cover - so I haven't seen it yet.

I have seen the new PCW-8256 - the centrepiece of our colour extravaganza - and an impressive machine it seems to be. I had a quick fiddle with it before it was whisked away for review (on Page 29) and in that short time I had few problems in understanding the concepts and use of Locoscript (the Word Processing software). It looks like Amstrad have scored again, but this time going up-market.

You'll find another 8 Page Supplement with reviews on some seventeen pieces of software. Again, the supplement is so designed that you can remove it intact without the main magazine falling apart. And whilst on the subject of preserving your magazine, I have had an increasing number of people write to me on the subject of Binders to house their copies safely. So next month we will be announcing availability of The Amstrad User Binder which will accommodate 12 issues of the magazine. The price is not yet fixed, but will probably be \$12 - \$13 including postage.

I find it hard to get into a Festive mood sitting at my desk, but when I received this month's Learning Centre with the three voice Carol from Peter Campbell and put it through the SSA-1 (Speech Synthesiser), my brain's memory bank retrieved the Christmas file and, almost immediately, I could smell my Christmas dinner and sense the children's excitement, which is always something special this time of year.

Have a good Christmas,

Ed.



# Letters

I am writing to you to let you know of some of the troubles I am having in trying to start a user group in my area.

For some months now I have had my name in your magazine as a contact point for the formation of a user group, to date I have had no response from this direction. I have also made up some pamphlets and arranged with the local Amstrad dealer for them to be handed out to any customers he has for Amstrad wares, I have had only one phone call from that effort. To tell the truth I am a little disappointed with these results.

Because of this I wish to ask that you publish this letter in The Amstrad User in the hope that subscribers in the HAWKESBURY DISTRICT will read it and contact me with a view to forming a user group or at least to get together once in a while to help each other with any problems we might have.

If there is anyone interested out there then ring Terry on 045 765291 anytime.

Terry Webb, Glossodia, NSW.

Recently becoming the owner of an AMSTRAD CPC464 with its high resolution, colour and sound channels, I looked like being on a good thing until I realized all the programming I had done for my old faithful Dick Smith System 80 was not compatible.

Some of the programs I have are worth saving and transferring over. With this end in view I was prepared to type them in. It soon became apparent that this was not going to be a straight forward job.

All the print @ statements have to be changed to locate statements. The

System 80 has 64 characters per line. The AMSTRAD has a choice of 20, 40, or 80 per line. The System 80 shows 16 lines per screen, The AMSTRAD has 25.

Armed with these facts I contemplated the problem once more. I had to decide which mode to use. As modes 0 and 2 are non-linear as far as characters are concerned, either too wide or too thin, I opted for mode 1 for my conversion.

This gave me a base for my conversion, i.e. 40 characters per line and 25 lines per screen. Colour is not a problem as the System 80 was a two colour system, i.e. either on or off.

The next problem was to work out the new locations at approximately where the old print @'s would appear. Then the thought came to me.

Why not devise a proportional program, stick it in the AMSTRAD and let the computer do the work. With this in mind the following short program was devised.

```
2 'CONVERT PRINT @ TO LOCATE
4 'No.19857
10 CLS:MODE 1
20 CLS:INPUT" ENTER THE PRINT
   @ NUMBER YOU WISH TO
   CONVERT";P
30 A=INT(P/64)+1
40 Y=INT(A*1.6)
50 A=A-1:C=P-(A*64)
60 X=INT(C*40/64)+1
70 LOCATE 6,8:PRINT " X Y
   CO-ORDINATES";LOCATE
   6,10:PRINT X;";";Y
80 LOCATE 6,15:INPUT"PRESS
   ENTER TO CLEAR SCREEN";Z
90 GOTO 20
```

The lines 2 and 4 are REM statements that I head my programs with for printout and filing purposes. It saves

trying to work out what the H! is this program supposed to do. The file No. is a corruption of the date the program was written.

Henry G. Piner, Springvale, Vic.

At the moment I am campaigning with Amstrad to recall and replace the manual with Micropen, which I bought at no small cost a few months ago. It is a slipshod, inadequate excuse for ripping off unwary consumers! It is a careless bodge-up of an existing program to get money from Amstrad owners and I am telling Amsoft that they will ruin their image if they associate with that kind of enterprise. Have a look at it yourself!

John Jeffreys, Mt. Ku-ring-gai, NSW.

I was hoping you could publish this letter in your "LETTER" section in The Amstrad User.

You see, I have something to say..... H E L P !!

I've been trooping around the rooms in GEMS OF STRADUS for the past few weeks, almost non-stop, but have finally come to a halt.

I've been in every room (I think) but, for the likes of me, I just cannot get past the ALIEN.

Could you, or any other readers, help me?

What should I use to get past that "YELLA DUDE"?

And while I'm here if there's anything else to know about this game, tell me! 'Cause I could sure use the help. Thanks.

Tony Scott, Maffra, Vic.



I found the game "Maniac Mower" in the September issue to be good, apart from a minor problem. The scoring was based on the amount of fuel left at the end of mowing each lawn, but best scores were obtained by cheating and going out the gate as soon as possible without mowing much lawn!

I have modified the game to score on the proportion of lawn mown as well as the amount of fuel left. In addition, I modified it to start each run with a little less fuel than for the previous run.

```

280 SOUND 132,1:DIM lawn (20,25)
340 SC=0:e=3:f1=1050
350 f1=f1-50:f=f1:SC1%=0:ERASE
    lawn:DIM lawn (20,25)
510 PEN 6:PRINT pond$:lawn
    (r1,r2)=1
590 PEN 9:PRINT tree$:lawn
    (s1,s2)=1
690 PRINT tree$:lawn(a1,a2)=1
800 PRINT tree$:lawn(b1,b2)=1
920 lawn (t1,t2)=1:PEN7:PRINT
    CHR$(248);
1060 PRINT CHR$(248):lawn
    (c1,c2)=1
1230 PEN 0:PRINT mo$:IF lawn
    (x,y)=0 THEN lawn (x,y)=1
1350 .....SOUND
    132,1:GOSUB 1590:
    SC=SC+SC1:GOTO 350
1460 RESTORE 1580
1590 tot=0:FOR i%=1 TO 20
1600 FOR j%=1 TO 25
1610 IF lawn (i%,j%)=1 THEN
    tot=tot+1
1620 NEXT:NEXT
1630 tot=tot/500:
    sc1%=1000*tot*tot+f*tot*tot:
    RETURN
    R.E. Chapman, Glenbrook, NSW.

```

Most of us have grown used to the fact of life that a computer will be out-dated before it is taken out of its box, but it is unlikely that AWA-Thorn's record will be beaten for many years to come. Yes, even as people, such as myself, were handing over their \$1000 for a CPC 664 thousands of 6128's must have been rolling of the production line.

Ordinarily one would have shrugged this off as a one of those things that

happen in this fast moving world of high technology and made the most of what one had purchased, but, Dear Readers, there was nothing ordinary about these circumstances. AWA-Thorn, showing nothing but contempt for their customers, releases within two months of the CPC 664, a compact 128k machine together with a more advanced version of CP/M for **the very same price!** If this doesn't show a complete disregard for their customers then I'd hate to see what does. After all they must have known that the release of the 6128 was imminent.

So.....

A word of warning to buyers of the 6128. Beware! If it is AWA-Thorn's intention to release an up-graded model of the previous model every two months at the same price then you can expect to see a CPC 6512 colour computer in the shops by February 1986.

Thank you very much Amstrad and you too AWA-Thorn!

Brendan Grainger, Mt. Evelyn, Vic.

Not long ago I purchased the adventure game "Bastow Manor" and as yet I cannot enter the Manor. I would like some help so could someone please tell me how to enter the Manor.

A.F. Williams, Windsor, Vic.

I have a query concerning which printer I should buy and I think this will benefit other "printer seekers".

I am looking for a printer that is under \$500 and produces reasonable print-outs plus graphics dumps etc. I have heard (and read!) that the DMP-1 is not very good so after I read the review on the Seikosha SP-1000, I was sure it was the printer for me. But, once again, a local supplier assured me that he had a better model and that the Seikosha was unreliable. Could you please inform me of some good printers for the Amstrad?

Paul Mezzavia, Geelong, Vic.

*We have been using a Seikosha*

*SP-1000 pretty solidly for over three months now and have no complaints whatsoever. Perhaps your local dealer can't stock them, in which case that would explain his comment.*

I have just replaced my 464 with a 6128 and have noticed that the CAPS LOCK Trick, viz. POKE 46312,255, mentioned in your March issue does not work on the 6128. Since I use this feature in a number of my programs to ensure upper case input, I would appreciate it if anyone could tell me the appropriate address and/or code to use on the 6128.

Neale Yardley, McKellar, ACT.

I recently received my October issue of your magazine and, naturally enough, had to check out the "BUGHUNT" game just to make sure it was OK for the kids! I found it a good game but there was one extra unexpected "bug" in it. The program allows two BUGs to be placed at the one location but the scoring system does not score two hits when you zap them.

To correct this I modified Line 1013 to read:

```

1013 IF position(horiz(x),vert(x))=1
    GOTO 1011 ELSE
    position(horiz(x),vert(x))=1

```

This seems to solve the problem adequately.

G.C. McDowall, Stafford Hts., QLD.

Thank you to Brenton Ross for his Utility Trio .....Screen dumping on the Amstrad. (User magazine October 85).

I am enclosing a modified listing in Basic that should allow people to dump the Amstrad CPC 464 screen to DT80, CP80, U80, most of the low cost xx80's style printers, having the same Escape codes. At 640 graphic dots you can get some very nice printouts.

To use, cold start your machine



(switch on), enter this program and RUN it. If you get ERROR IN DATA STATEMENTS, recheck your work especially the DATA statements! When corrected and RUN you should get a screen message READY. Now fill your screen with something, a listing etc. When you wish to perform a screen dump, enter ORIGIN 0,0:|scremdmp (the bar before scremdmp is the shifted @ key the next key to the right of the "P").

See Brenton's article especially the section "A BASIC program to generate a screen dump RSX", for a fuller understanding. Remember once entered and RUN the listing can be DELETED or NEWed and a program LOADED to run some graphics etc. with a call to |scremdmp inserted in your program.

As long as HIMEM is not changed by a cold start or by your graphics program after RUNNING this utility it will be there till you switch off.

Happy Dumping!

Roy Eiberg, Manly, NSW.

### All I want for Christmas

A wish from Don Leith

Christmas time is getting nearer  
And I'm getting jumpy  
I wish I could control myself  
And stop from getting grumpy

Because there is a very special Gift  
I'd like from someone dear  
I've dropped all the hints I can  
Whenever she can hear

"Wish I had more 'ram'  
To run my 'DB 3"  
"41 k is not enough"  
I moan when she's near me

I'm always caught reading  
Billy Guyatt's Christmas blurb  
I leave reviews lying round  
Pages open to observe

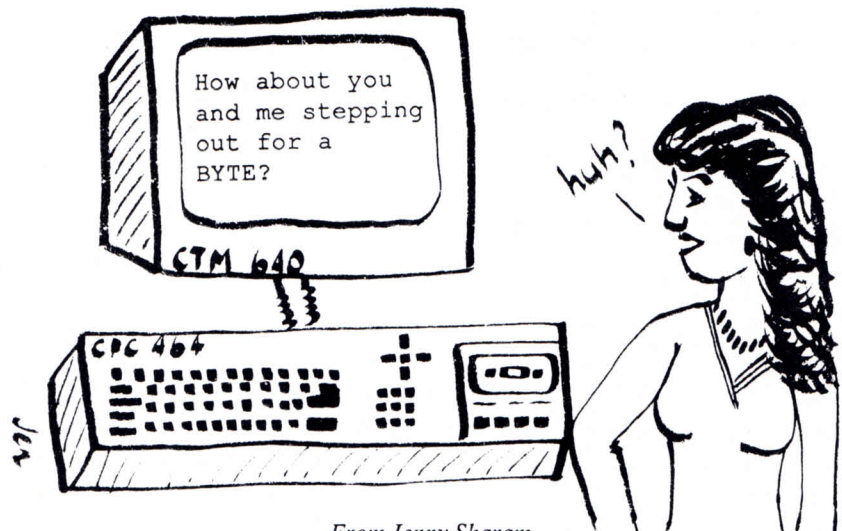
Then whenever we're out shopping  
There's one place that I head  
To the Computer section of the shop  
To put in practice what I've read

The wife is dragged protesting  
To watch me and my 'MATE'  
As I fondle my future Christmas present  
An Amstrad 128.

```

900 REM *****
910 REM ***** screen dump m/c *****
950 REM *****
1010 READ n:cs=n
1020 s=HIMEM-n:MEMORY s-1
1030 FOR i=1 TO n
1040 READ a:cs=cs+a:POKE s+i-1,a
1050 NEXT i
1060 READ n:cs=cs+n
1070 FOR i=1 TO n
1080 READ a,b:cs=cs+a+b
1090 a=a+s:ah=INT(a/256):a1=a-ah*256
1100 POKE s+b-1,a1:POKE s+b,ah
1110 NEXT i
1120 READ check:IF check=cs THEN CALL s: ELSE
PRINT "ERROR IN DATA STATEMENTS!"
1130 END
1140 DATA 191
1150 DATA 33,136,144,1,146,144,205,209,188,201
1160 DATA 151,144,195,158,144,83,67,82,68,77
1170 DATA 208,0,253,33,70,145,221,33,250,144
1180 DATA 205,235,144,1,144,1,121,214,4,79
1190 DATA 120,222,0,71,221,33,0,145,205,235
1200 DATA 144,17,0,0,205,15,145,19,62,2
1210 DATA 186,32,247,62,127,187,32,242,221,33
1220 DATA 6,145,205,235,144,197,62,66,205,30
1230 DATA 187,193,192,62,0,184,32,204,185,32
1240 DATA 201,221,33,9,145,205,235,144,201,221
1250 DATA 126,0,254,0,200,205,43,189,48,245
1260 DATA 221,35,24,241,27,65,3,0,0,0
1270 DATA 27,75,127,2,0,0,13,10,0,27
1280 DATA 64,0,0,0,0,197,213,96,105,229
1290 DATA 205,240,187,254,0,40,8,253,203,0
1300 DATA 198,253,203,0,206,225,209,213,35,35
1310 DATA 205,240,187,254,0,40,8,253,203,0
1320 DATA 214,253,203,0,222,253,126,0,205,43
1330 DATA 189,48,248,253,54,0,0,209,193,201
1340 DATA 0
1350 DATA 14
1360 DATA 0,2,10,5,15,11,22,14,190,25,114,29
1370 DATA 99,32,120,47,99,50,135,56,126,71,99,74
1380 DATA 129,94,99,97,24943

```



From Jenny Sharam



# A Different Approach to Program Design

## Part Two - from Alan Harris

---

For those of you who typed in the first part of this clock last month, congratulations. It's not easy to have to type all that source without knowing exactly what will happen. However, all will be explained, including part two of the program, which adds a feature for adjusting the clock without making a mess of the clock face.

**Lines 1030-1070:** These lines are responsible for drawing lines to fill in the outer and the centre of the clock face. They are drawn each time through the loop and that means for every degree between 0 and 359. This was originally done because of the lack of a 'paint' or 'fill' function in Locomotive BASIC. (I note that it has been included in 'ARNOLD MK II' the CPC 664 however.) I won't go into details here, I suggest you examine the code and you can learn yourself if you understand TRIG.

**Lines 1080-1130:** This block of code is only used when  $x$  is evenly divisible by 6. The reason is that its function is to draw the minute marks at the edge of the clock face which occur 60 times for one rotation. Here again we use the MOD function to test if the flag should be set. If so then we enter the inner loop to draw them otherwise we bypass it.

**Lines 1140-1160:** We need to lengthen the marks that correspond to the hour positions and we need to calculate the hour numbers that correspond to them. We use the variable max.hour here to determine whether to draw at 15 degree or 30 degree angles. If  $x \bmod (\text{hour angle})$  is zero then we need to draw a larger mark so we set the flag and enter this inner loop, otherwise we bypass it. The function 'TAG' is used so that ARNOLD uses the graphics cursor as a start position to display the hour numbers.

**Lines 1170-1220:** Lines 1170 and 1180 are used to draw the short line to indicate the hour positions using the centre of the clock and a distance from centre of 120 units. If you examine line 1190, you will see that an offset of 13 horizontally and 5 vertically is used for the origin. This is because text is printed to the right of the graphics cursor and it's used to centre the text. Line 1290 calculates the hour number again depending on the max.hour variable, and displays it beside the hour mark. The flag is now reset and

TAGOFF ensures that we don't mess up the display.

**Lines 1230-1260:** Line 1230 is the WEND of the hour mark section and  $x$  is incremented in line 1240 and tested in 1250. If we haven't reached 360 then we move back to the top and continue through again. If we've reached 360 degrees then the circle is complete and we can get out.

**Lines 1270-1290:** We now need to decide whether a disc drive is installed in the system. This is about the only time you will see me use the 'GOTO' statement. By using the 'ON ERROR GOTO' function we test to see if the drive is available. In line 1270 we can trap the error that would result if the command |DISC is used in a cassette system. If it were used in a non disc system, ARNOLD would not be aware of the meaning and would respond with 'Bad command', which would stop the program. By preceding the command with line 1270, we can instead make him GOTO line 1280 if an error occurs in line 1290. Here the RESUME 1360 will cause him to bypass the code in lines 1300 to 1360 which we'll look at now.

**Lines 1300-1360:** If a disc was attached then ARNOLD knows what |DISC means. It simply means initialise input/output to use discs instead of tapes. If this is the first time the program has run on a disc system, then we know that the user has loaded the program from a cassette and the variable 'discfile' will be reset. This will cause this section of code to be used. Its function is to save a copy of the clock face as a binary file on disc that can be loaded much faster than it can be drawn on the screen, once saved in line 1310. We now want the user to adjust the value of 'discfile' in the declaration lines at the beginning of the program. The prompt in line 1330 does this, and also tells the operator to save the program in its updated form on disc. For the first time user with discs this is where the program stops.

Once the program is re-run it will go through the same routine again except that this time 'discfile' is set which means that the DRAW routine is bypassed altogether because a file is available on disc. Program control passes instead to the line immediately after the DRAW loop. This time since discfile is set, we also bypass lines 1300-1360 and end up at line 1360.

This line is reached whether it's a disc system or not. The



ON ERROR GOTO is reset with the line 1360, and line 1370 loads the file from disc if discfile is set otherwise we've just drawn it so it's there anyway!!!

**Lines 2000-2090:** Well here we are with old friend the WINKEY routine. For those who have not seen my earlier remarks this is a general purpose WHILE/WEND INKEY\$ routine, that is the principal way I use to get information from an operator.

Line 2020 resets the variable that will hold a copy of the key pressed by the operator so that we enter the loop whose start is line 2030 and whose end is 3640 (Be patient it may seem a long way but we'll get there.)

We mentioned earlier the 'setup' variables and this is where they are used. Why? To set up the clock of course. (See what I mean about useful names?). If you examine line 2110 to 3640, you will see small blocks of code each of which handles a feature whose first letter is tested for by this routine. In the order they appear we have T(ick), C(hime), G(ong), A(larm), P(aper), I(nk), B(order), and V(ideo). Most of these functions are either ON or OFF, the exceptions being those concerned with the INK functions where we have 27 possible conditions.

In order to understand the setup routine we shall first examine the routines that are used not only by setup, but also by the operator. Beginning at 2110 we see that if the key pressed was a letter 'T' or 't' then the test in line 2110 passes and control transfers to the block of code in lines 2120 to 2160. If the variable 'tick.on' is set then the clock routine (5000 onwards) will make a tick sound similar to that of the old clockwork type clocks. If reset then the sound is not produced. This switch on/off action occurs in line 2120. Once the action is carried out, we inform the operator by displaying the current switch setting in lines 2130 to 2140. The command DI is used so that the cursor setting in line 2130 will not be in conflict with the interrupt service routine which also uses cursor movement. This situation could occur because the clock subroutine is called every second, and if we position the cursor here and then the clock subroutine needs to move the text cursor, we'd be printing all over the place. Once we've set the tick.on we re-enable the interrupts with EI.

Looking further down the listing you will see similar routines for chime, gong and alarm. These only differ in the character typed at the keyboard, except for the A(larm) function which when switched off also resets the 'chime' and 'gong' variables. Later these will be described in detail, but for now assume the condition where the alarm was set for 8:00 am and has just sounded. To turn it off we simply press the 'a' key on the keyboard. However, if gong.on is selected then it will be paused by the clock routine until the alarm has been silenced, at which time it will commence sounding to announce the fact that it's eight a.m. (which we already know since the alarm was set for eight a.m.).

The next three functions are a feature added to allow the

operator to select the ink colours. All that happens is the selected ink is incremented by 1 for each key press until the maximum of 26 is selected. Since we are counting the colours in modulo, that is 27, then when 26 is reached the ink will return to 0 on the next key press. The keys for Paper, Ink and Border therefore can be adjusted to suit the user.

Finally we have the 'V' for Video function. This was the result of a snag I encountered when calibrating the timing routine in 5000 onwards. You see I use the computer in my bedroom and in order to check the timing it was necessary to leave it running all night. The 'V' function simply sets all inks to 0 so I can turn the display off when it's not needed.

In the declaration section at the beginning of the program we defined most of the variables of this section as =reset, in other words, set to 0. In the usual course of events we would normally wish to know the condition of each of these sections and a well designed program will display such information on the screen. In line 2940 we see if a key is being pressed and transfer it to the 'key.pressed\$' string variable however in lines 2050 to 2100 we in fact ignore the key pressed if the numeric variable 'setup' is less than 10. We have already set it to 1 in the declaration lines so we can assume it is still that way and control enters the loop defined by 2060 to 2100.

Line 2070 will set 'key.pressed' to the same value as the FIRST letter of the 'setup\$', then 'setup' is incremented to point to the next letter and the flag reset so we can leave the loop.

The first letter in the setup\$ is 'T' and so is processed as though the operator had pressed this key. In other words the tick feature is reversed from the original state set by Line 230. The 'key.pressed\$' is 'reset\$' and we fall through the bottom of the loop with it reset to a null string. Finally we reach Line 3640 which directs control back one more to Line 2030. Here we assume no key is yet pressed by the operator and generate the next 'phantom key stroke' from 'setup\$' which is 'C' for chime and so on etc. etc. Finally the pointer 'setup' is pointing past the end of 'setup\$' and Lines 2050 to 2100 are not used again.

### ***Clock Routine***

Every 50 counts of the internal timer control passes to this subroutine, which according to AMSTRAD means every second. Since my computer appears to run faster than expected I have included a routine to increment a counter that, at regular intervals, causes an immediate return without adjusting the clock. This happens at intervals determined by the CONSTANT 'trim'. In my case it is set at 860. It is used as the modulus of a counter using a variable I've called 'do.trim'. As 'do.trim' is incremented every second it will eventually reach a value of 860. If you examine line 5010 you will see how it is reset to 0 because of the modulus 'trim'. Next time the subroutine is called by the interval timer, 'do.trim' will be 0 (or reset) and it is then set (i.e.



made equal to 1), so that the next time it is called (one second later) it will not cause an immediate return.

Assuming this to be so, the next section of code will un-draw the second hand, prior to updating its position. This is accomplished by drawing it in its current position using an ink value the same as the paper value. This will make the second-hand disappear. Now we can increment the seconds counter in mod 60, (since there are 60 seconds in one minute) in line 5040. If the switch for the tick is set then we produce the sound in line 5050. However to avoid confusion we make sure that both the 'chime' and 'gong' are not set. This means that during the chime or gong sounds we don't tick. In 5060 and 5070 we now draw the second hand in its new position and in line 5080, we update the 'clock\$' which contains a digital representation of the time.

In line 5110, if the value of 'second' is 0 (reset) we know that we must add 1 to the 'minute' so we set a flag to indicate that this function is required. If not then nothing further needs to be done to the clock itself and control passes from line 5110 all the way down to line 5810 and eventually to the RETURN in line 6240.

The remaining code between 5110 and 5810 is essentially the same as the block just described. We test to see if a change is needed for minute, hour, am/pm, day, month, or year, and if so we set a flag to enter a block of code to carry out the task. If not then the remainder of the code is bypassed and we reach the RETURN, which eventually takes us back to the WINKEY module.

A point worth noting is the difference between minutes and seconds in the first case and hours, days, months etc. in the other. For seconds and minutes we count from 0 and for the remainder we count from 1. To make use of the modulus feature we must *always* count starting from 0. This is why you will see the display uses '+1' in several lines and why we test for 11 in the hours module to see if am/pm needs changing rather than 12.

The section of code in Lines 5820 to 5930 is used to detect when the second hand is at the same location, or just past the hour/minute hands. If such is the case then part of the hour or minute hand may have been rubbed out when the second hand was erased prior to being updated. The result is to re-draw the hour and/or minute hand if they are close to each other.

When the digital display string is the same as the digital alarm string then we know we must trigger the alarm. This is accomplished by a test in line 5940 and a block of code is run that sounds the high pitched alarm sound and resets any chime or gong activity that might otherwise occur.

If the chime has been set then lines 6020 to 6080 will read the chime data and every second will sound a single note, reducing the chime count by 1 each time. Similarly the gong module in Lines 6090 to 6150 does the same thing except that unlike the chime each gong note is the same. Fine, but how do we know when to set chime and gong. First, line 6160 will only test for the chime time if 'chime.on' is set

otherwise it will bypass the block of code in lines 6170 to 6230. The function of this block of code is to detect when the quarter, half, three-quarter, and hour periods arrive. Each chime consists of 4 notes and a 1 second gap, in the following sequence:

15 minutes past the hour 4 notes  
30 minutes past the hour 5 notes  
45 minutes past the hour 12 notes  
Just before the hour 16 notes  
On the hour 1 gong for each hour  
(the number of gong's is set when the hour is incremented)

### *Update Routine*

The lines of code from 2790 to 3520 I refer to as the Interface area. Immediately the "U" or "u" key is pressed we leave the first long key scan module and enter a second key scan routine. In order that the operator is aware of this fact we alternately print an instruction line then a blank line. The interrupt function that calls the 'clock' subroutine in 5000 is disabled in line 2791, and because the interrupts that would normally occur the next time we use the EI are not needed we use a dummy routine in line 6280. (I found that the display was adversely affected if it were not used). The loop can only be exited by resetting 'update'. The three keys A, T, D, and the [ENTER] key are the only valid characters acknowledged. 'A' allows the Alarm time to be adjusted, 'T' allows the time to be adjusted and 'D' the date. To leave the routine the operator will press the [ENTER] key, which about 1 second later will cause the clock to start running.

The first sub-module in lines 2830 to 2930, sets first the parameters that are needed by both clock and alarm adjustment modules. Section 2940 to 3000 transfers the alarm\$ digits into a temporary 'work\$' if the 'A' key was pressed to select alarm adjustment. The Update routine is then called where the adjustments are made. On the return the 'work\$' is copied back to the 'alarm\$'. The key.pressed\$ is reset and control passes straight down to line to 3620 and then back to 2790 again. If the key pressed was 'T' then we clean up the clock face by un-drawing the hands in their current position in the same colour as the paper ink, transfer 'clock\$' into work\$ and call the Update routine. On return we re-draw the hands in their new position, transfer the new work\$ into clock\$ and reset key.pressed\$. Control is now transferred out of the 'T' routine and back to 2790.

When 'D' is selected we transfer the date\$ into work\$ and adjust the required modulus (What's the plural of modulus -- Moduli? - *Almost, it's moduli - Ed*) to suit Year, month and day from work\$, which is in the form DD/MM/YY, test that the date set is valid and if so expand it and display it at the top of the clock. We reset key.pressed\$ and control goes to the WEND at 3630 and then back up to 2790.

If the operator is satisfied that all is well then the [ENTER] key is pressed to signify the fact and since key.pressed\$ =finished\$ we exit the update routine by resetting 'update' and



in line 3610 the interrupt is set to call the timing routine at 5000 onward every second.

The routine from 4000 to 4690 is used to adjust the time, alarm setting and date. On occasion, small branches and tests are included since we need slightly different adjustments for each. The principal is that each of the pairs of digits in the centre of the clock are adjusted independently, so we need to move left/right and also we need to increase/decrease the value. In order to determine the maximum and minimum values of each pair of digits we use a suitable modulus. So that we can readily see the digits on which we work these are displayed in reverse colours in 4020 to 4050. The temporary variables 'temp()' are the values of pairs of digits in the work\$. A test is made in 4130 to see if we have NOT selected the date function. If we DID NOT then we must allow for an extra temporary variable to represent the AM/PM condition.

Prompt\$(2) indicates that the left/right arrow keys move the reversed display and the up/down keys increase/decrease the digits displayed. We now enter the third and final INKEY\$ loop defined by 4260 and 4640. This is where we actually carry out the adjustment. 4280 to 4370 test and carry out the increase/decrease function the value of temp(position) is adjusted as required using the value of modulus(position) to ensure it is within limits. A flag is then set to indicate that the value has been adjusted and key.pressed\$ is reset.

The code in lines 4380 to 4470 is used to modify the work\$ to reflect the updated value of temp() and work\$ is re-displayed at the correct position so the operator knows what is going on. Lines 4480 to 4620 reset the paper and ink values to normal and update the 'position' variable according to whether left or right arrow was pressed. The pair of digits in the new position is now reversed and they can be adjusted.

Once the operator has completed all the adjustments, key.pressed is set to finished\$ by pressing [ENTER] and control drops down to 4650, where the work\$ is displayed in normal colours and a return is made to the interface routine.

Well that just about wraps it up. For those of you ambitious enough to try it I suggest that consideration be given to the following possible alternatives:

1. For those with colour monitors adjust the program for MODE 1, and allow more colours for a more interesting display.

2. Make provision for an array of date\$(0) to date\$(n) each of which has the family Birthdays in and display a specific message for each member when that day arrives. Alter the Update routine to allow these dates to be entered without affecting the display.

3. If you're feeling really ambitious what about adding a calendar routine to allow the operator to type reminders similar to the birthdays for particular days.

Finally why have this program at all? I firmly believe that there are many of you out there who in spite of the fact that you understand the commands that are used in Locomotive Basic, may be unaware of the methods used to analyse a

problem with the view to providing a computer program as a solution. It may be that the experience I have gained over the years, which produces a program in the style of that just discussed may be of benefit to you. I know this program does not do very much, but the way in which it was DESIGNED rather than simply tried out on the keyboard, is indicative of the approach that professional programmers must adopt when they write important software.

Maybe I'll have other tips or programs for you in the future, so until then remember that in my opinion, the last and least of a programmers work is carried out at the keyboard. Careful analysis of the problem, followed by thoughtful implementation of small solutions to each tiny section means less work at the keyboard and more dollars in the pocket.

```
211 null$=STRING$(80,32)
280 REM
290 REM
300 REM
310 REM
320 REM
330 REM
340 REM
350 REM
360 REM
370 REM
380 REM
1380 REM
1390 REM
1400 REM
1410 REM
1420 REM
1430 REM
1440 REM
1450 REM
2780 WEND
2781 IF key.pressed$="U" THEN update=set
2790 WHILE update=set:REM (* Update Procedure *)
2791     EVERY 10000 GOSUB 6280
2792     LOCATE 1,24:PRINT null$
2800     DI:LOCATE 12,24
2810     PRINT prompt$(1):EI
2820     key.pressed$=UPPER$(INKEY$)
2830     IF key.pressed$="A" OR key.pressed$="T" THEN flag=set
2840     WHILE flag=set:REM (* Update Alarm or Clock *)
2850         x=34
2860         position=1
2870         p.max=4
2880         modulus(1)=max.hour
2890         modulus(2)=60
```



```

2900     modulus(3)=60
2910     modulus(4)=2
2920     flag=reset
2930     WEND
2940     WHILE key.pressed$="A":RE
M (* Update the Alarm *)
2950     work$=alarm$
2960     y=12
2970     GOSUB 4010
2980     alarm$=work$
2990     key.pressed$=reset$
3000     WEND
3010     WHILE key.pressed$="T":RE
M (* Update the Clock *)
3020     DI:EVERY 50 GOSUB 6280
3030     ORIGIN FN x(second),FN
y(second)
3040     DRAW FN p1(second,secon
d.hand),FN p2(second,second.h
and),black
3050     ORIGIN FN x(minute),FN
y(minute)
3060     DRAW FN p1(minute,minut
e.hand),FN p2(minute,minute.h
and),black
3070     ORIGIN FN x(tock),FN y(
tock)
3080     DRAW FN p1(tock,hour.ha
nd),FN p2(tock,hour.hand),bla
ck
3090     work$=clock$
3100     y=13
3110     position=1
3120     GOSUB 4010
3130     clock$=work$
3140     second=temp(3)
3150     ORIGIN FN x(second),FN
y(second)
3160     DRAW FN p1(second,secon
d.hand),FN p2(second,second.h
and),white
3170     minute=temp(2)
3180     ORIGIN FN x(minute),FN
y(minute)
3190     DRAW FN p1(minute,minut
e.hand),FN p2(minute,minute.h
and),white
3200     hour=temp(1)
3210     tock=(hour+1)*5+INT(min
ute/12)
3220     ORIGIN FN x(tock),FN y(
tock)
3230     DRAW FN p1(tock,hour.ha
nd),FN p2(tock,hour.hand),whi
te
3240     EI
3250     key.pressed$=reset$
3260     WEND
3270     WHILE key.pressed$="D":RE
M (* Update the date *)
3280     x=35
3290     y=11
3300     position=1
3310     modulus(1)=31
3320     modulus(2)=12
3330     modulus(3)=100
3340     work$=date$
3350     GOSUB 4010
3360     year=VAL(RIGHT$(work$,2
))
3370     month=VAL(MID$(work$,4,
2))-1
3380     day=VAL(LEFT$(work$,2))
-1
3390     date$=work$
3400     IF FN date.good (year+1
900,month+1,day+1) THEN flag=
set
3410     WHILE flag=set:REM (* G
ood date selected *)
3420     DI:LOCATE 27,1
3430     PRINT "
"
3440     LOCATE 27,1
3450     PRINT FN date$(year+1
900,month+1,day+1,d$,m$):EI
3460     flag=reset
3470     key.pressed$=reset$
3480     WEND
3490     WEND
3500     IF key.pressed$=finished$
THEN update=reset
3510     p.max=4
3520     WEND:REM (* En
d of procedure Update *)
3530     IF key.pressed$=finished$ T
HEN flag=set
3590     PRINT blank$:EI
4000     REM (* This is the ad
just procedure *)
4010     update$=key.pressed$
4020     PAPER black
4030     PEN white
4040     DI:LOCATE x,y
4050     PRINT work$:EI
4060     PAPER white
4070     PEN black
4080     DI:LOCATE x,y
4090     PRINT LEFT$(work$,2):EI

```



```

4100 temp(1)=VAL(LEFT$(work$,2))-1
4110 temp(2)=VAL(MID$(work$,4,2))-1
4120 temp(3)=VAL(MID$(work$,7,2))-1
4130 IF update$<>"D" THEN flag=set
:ELSE flag=reset:p.max=3:temp
(3)=temp(3)+1
4140 WHILE flag=set
4150   IF RIGHT$(work$,2)="AM" THE
N temp(4)=0
4160   IF RIGHT$(work$,2)="PM" THE
N temp(4)=1
4170   temp(2)=temp(2)+1
4180   temp(3)=temp(3)+1
4190   p.max=4
4200   flag=reset
4210 WEND
4220 DI:LOCATE 12,24
4230 CALL &BD19
4240 PRINT prompt$(2):EI
4250 key.pressed$=reset$
4260 WHILE key.pressed$=reset$
4270   key.pressed$=INKEY$
4280   WHILE key.pressed$=CHR$(24
0):REM (* UP Arrow - Increase
*)
4290     flag=set
4300     temp(position)=(temp(pos
ition)+1) MOD modulus(position)
4310     key.pressed$=reset$
4320     WEND
4330     WHILE key.pressed$=CHR$(24
1):REM (* Down Arrow - Decrease
*)
4340       flag=set
4350       IF temp(position)>0 THEN
temp(position)=temp(position
)-1:
ELSE temp(position)=modulus(
position)-1
4360       key.pressed$=reset$
4370       WEND
4380       WHILE flag=set:REM (* Up o
r Down Arrow *)
4390         IF position=4 THEN temp$
=temp$(temp(position)):
4400         IF update$<>"D" AND posi
tion<4 AND position>1 THEN

temp$=RIGHT$(STR$(temp(posit
ion)),LEN(STR$(temp(position)
))-1)
4410         IF (update$<>"D" AND pos

```

```

ition=1) OR update$="D" THEN

temp$=RIGHT$(STR$(temp(posit
ion)+1),LEN(STR$(temp(positio
n)+1))-1)
4420         IF LEN(temp$)=1 THEN tem
p$="0"+temp$
4430         MID$(work$, (position-1)*
3+1,2)=temp$
4440         DI:LOCATE x+(position-1)
*3,y
4450         PRINT temp$:EI
4460         flag=reset
4470         WEND
4480         IF key.pressed$=CHR$(242)
OR key.pressed$=CHR$(243) THE
N flag=set
4490         WHILE flag=set:REM (* Left
/Right Arrows *)
4500           PAPER black
4510           PEN white
4520           DI:LOCATE x,y
4530           PRINT work$:EI
4540           PAPER white
4550           PEN black
4560           IF key.pressed$=CHR$(24
2) AND position>1 THEN positi
on=position-1
4570           IF key.pressed$=CHR$(24
3) AND position<p.max THEN po
sition=position+1
4580           DI:LOCATE x+(position-1)
)*3,y
4590           PRINT MID$(work$, (posit
ion-1)*3+1,2):EI
4600           key.pressed$=reset$
4610           flag=reset
4620           WEND
4630           IF key.pressed$<>finished$
THEN key.pressed$=reset$
4640 WEND
4650 PAPER 0
4660 PEN 1
4670 DI:LOCATE x,y
4680 PRINT work$:EI
4690 RETURN
4700 END
4710 REM *****
Clock routine *****
*****

```

---

*Merge the above listing with last month's to get your complete Clock Program.*

---



# A Retailer's Point of View

## An Interview with Pieter van Wessem of Billy Guyatts

The Billy Guyatt's story goes back 26 years to a Mitcham newsagent Jack Taylor, who went to work for Hannam's Electrical Discount Stores. Jack subsequently bought out Hannam's and built it up to a chain of 13 stores before selling it in the early 1970's.

In 1973, Jack's family returned to the electrical retailing industry with the purchase of the Billy Guyatts chain. In 1985, Billy Guyatts acquired the Discount City chain of 8 stores - its major competitor.

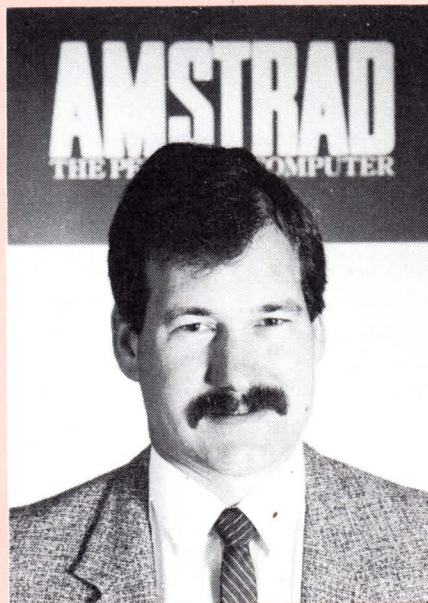
There are now 18 Billy Guyatts Discount City Stores in the metropolitan area, with plans to cover all suburbs and to move into major regional centres.

The Amstrad User was able to stop Pieter van Wessem for an hour recently and obtain the views and comments from the retailing side of the Amstrad industry. As Product Manager for Billy Guyatts, Pieter has the responsibility for all electronic products from the time they are purchased by Guyatts to the time they are purchased by the customer. These products, of course, include the Amstrad range of personal computers.

**TAU.** How did Billy Guyatts get into retailing personal computers?

**PvW.** In 1982 we were the first department store to retail personal computers. Unfortunately the product was unreliable, service was poor and this produced many irate customers and a large amount of unserviceable machinery in our stores. At this point we decided to leave the personal computer business to others. Then in August 1984, I was contacted by AWA concerning their Amstrad CPC464, and at first I was not at all interested in the machine. However, I did have the

opportunity to have a look at one and my first reaction was that it was either something amazing or there was something incredibly wrong with it, but knowing AWA I doubted the latter. I sought the advice of some senior sales people and a computer salesperson and they all confirmed my gut feeling - that the CPC464 was very good value for money.



*Pieter van Wessem  
Product Manager for Billy Guyatts*

**TAU.** How many Amstrads have you sold in Victoria?

**PvW.** We have sold 20% of Australian sales of Amstrad computers.

**TAU.** What sort of service do you get from AWA?

**PvW.** It's the best service on any product we sell. They give it a full 12 months warranty, which is a very good computer warranty. If someone has a problem, they take it back to our store,

we get the salesperson to ring AWA-Thorn who will pick it up, fix it and return it to the store within 48 to 72 hours. If they think they can't fix it quickly enough, they exchange the faulty item. It is an unbelievable service. In fact the customer can take their faulty equipment direct to AWA and receive the same service. I just don't know of any other service like that.

**TAU.** The customer would surely need his warranty card?

**PvW.** Yes, although the relationship with AWA is so good that I can ring and say to them that someone is coming without a card and they would fix it without further discussion. Only two complaints on service have reached me in the last twelve months and these have been rectified immediately by AWA.

**TAU.** I believe your policy towards stocking software is one of not allowing the customer to try it first?

**PvW.** To some degree a lot of programs have to be looked at to be sold, such as business programs, but we go back to the days of TV games where the machines were held up the whole time with people playing games, some spending an hour or two going from one to the other. Of course, the stores are set up to be selling machines and not to be entertainment complexes which would restrict our sales. Within our 18 stores some 6 or 8 are set up as key bases which stock larger quantities of fast moving lines and one each of the slow items on the chance that someone needs it. Basically, a smaller store would not be geared up to opening all the stock because he might only hold 100 items, half of which he has just one copy, and the next customer to come along would not want to buy an



opened one. But we train our sales team to preview the material so they can tell the customer about it.

**TAU.** One of the criticisms of Billy Guyatts is that your sales people are adept at selling televisions, fridges and so on, but there is a particular knowledge required in selling a computer in view of the more technical questions which arise, and that they are often not skilled enough to handle this.

**PvW.** I think that is a criticism which will always come up and the reason is that certain consumers are so much more educated in computers than the average salespeople can hope to be. The criticism very often comes from the highly skilled user as they often ring me here and even I am totally out of my depth in view of the intricate nature of their enquiries. But fortunately, with the growth of User Groups which we have seen in the last 12 months, the Groups are more than happy to get involved with these people and that is removing a lot of this type of pressure. I don't believe we can solve this type of problem entirely because as the training goes on, so the technology grows at such a great rate. However, we do have a person going around the stores non-stop just talking to the salespeople, offering advice on handling the machines and software. This generates a lot of questions which is good, but we can never achieve the level of expertise of someone who spends 20, 30 or 40 hours a week on their machine - we would be kidding ourselves if we thought we could.

**TAU.** Did you have any pangs of conscience when the 664 was so closely followed by the 6128 at the same price?

**PvW.** No. We didn't build the machine then follow it up with another, nor did we import the machine and follow it up with another. We simply bought the current model off the supplier as with any product. But of course there was some concern in my mind that it was being followed up too quickly, in other words one was being superseded by the other too quickly. The 664 had as much potential as a home computer as the average user

would ever require in a lifetime, and the 6128 had probably more than they would ever need anyway. But of course, in people's minds it is the later model, and they would say that even if I never used that extra 64k, I know I have got the latest one. Nevertheless, I was quite surprised to see that follow-up so quickly. It's like a car- if you buy the latest model, you know there is another one already in design.

**TAU.** That brings us to the 8256 - will Billy Guyatts be selling this more specialised model?

**PvW.** Yes we will, and it is very specialised. We will train a number of salespeople just to sell that machine. I



don't see us putting it into all the stores. We will pick certain stores and staff capable of marketing that product and all enquiries will be directed to them, because what I know of it already, it is more complex, more advanced and a totally different concept - yes, we will go into it. In fact, we would look seriously at anything Amstrad brings out, the standard has been so good and so far has been very well accepted. We really have put Amstrad on the map in Victoria. Most people didn't know the brand before, and we have put in a lot of work in that area and we will certainly keep that going.

**TAU.** Going back to software, have you noticed any effect of piracy on your sales?

**PvW.** No, I haven't, but the thing that strikes me as peculiar is that when you compare sales ratios of good titles,

say the six top sellers, to slower ones you would expect the ratio to be higher. But of course it (piracy) will always go on. It happened in the music industry where people were dubbing tapes and I don't think there is anything we can do to stop it. That is why the people who import the software don't bring in a lot. Some customers think that as soon as a new title comes out we should have 20 or 30 copies in the shops. The point is that the importers probably have only brought in 30 in total and will top-up later on if necessary. Nobody wants to get burnt like they did with the TV games a few years ago when too many people brought too much into the country.

**TAU.** There seems to be an unhealthy amount of parallel importing of software at the moment. How do you safeguard against this?

**PvW.** We are only buying, we are not importing ourselves but yes, there is a lot of parallel importing where different people come to me with the same software. I would rather deal with one or two wholesalers only, rather than take one title off one person, one off another and perhaps another from interstate and so on, because it makes a very messy way of restocking. There are some people who are supplying us very well, with good backup. They provide quick replacement of faulty pieces or where I suspect that a title is not very good, I can send them all back.

**TAU.** Apart from the Amstrad computers, what other peripherals do you sell and from where do you get them?

**PvW.** Printers, disc drives and so on. We get them from whoever is supplying at the time. For example, there was some criticism of the original printer from AWA, the DMP-1, and so we switched supplier, but then AWA came back with the Siekosha SP1000, so we switched back again only to discover that they didn't really have the stock. They appear not to have anticipated that the demand of a higher priced printer would be the same as the lower priced one, and consequently we were out of stock for three weeks. If



people have a product then we'll look at it, but I'd rather buy from AWA as they deliver around Melbourne every day, even if it's a \$7 item.

TAU. And you are also into books now.

PvW. Yes, that's interesting. Last November or December I introduced books to just four of our stores because I was a little nervous about them as it was something I hadn't touched before and I couldn't really see why someone would want a book. The first delivery was sold out within two days! We now put larger quantities into all the stores.

TAU. So where do you go from here? Can you see any major changes or expansion in the home computer market?

PvW. Well, obviously with the 8256 coming out we will have to wait to see how that is accepted - I believe that it will be a very worthwhile machine if we can market it the correct way. So first of all I want to see it, get my hands on it, get some people trained and get that one off the ground. I believe the \$1000 computer will just keep going, whether it's the 664, the 6128 or whatever it is called after that, and from what I can see, they (Amstrad) seem to be doing their homework because they are always ahead of whoever is out there. I can't think of any computer that's better value even if you double the price.

I think the future is only going to get better. Initially, computers for a lot of retailers were a flash in the pan, they were one hit wonders or so they thought, and didn't put them in the same class as the TV's, washing machines etc. which were solid sellers over the years. I think that Amstrad has now got a base that will go the same way. Of course it will be a different computer in ten years time, but it will be the latest Amstrad model. Everything that Amstrad has done so far they have done very effectively, and I am sure they will continue to gain ground in the market place as consumer awareness of the value of the product and the quality of service from AWA grows.

# THE AMSTRAD USER HALL OF FAME

Game	Score	Time	Achiever
------	-------	------	----------

<b>Roland in Time</b>	<b>72</b>	<b>18 mins</b>	<b>Paul Azzopardi</b>
-----------------------	-----------	----------------	-----------------------

*Hint: Practice well on cards 1, 6 and 4 to give a sound base for a good score. Don't try to do too much at the beginning.*

<b>Codename Mat</b>	<b>601</b>	<b>90 mins</b>	<b>Rowland Hayes</b>
---------------------	------------	----------------	----------------------

*Hint: Leave the cruisers alone until you have destroyed all the other ships.*

<b>Roland in the Caves</b>	<b>79884</b>	<b>4 mins</b>	<b>Emma Poynton</b>
----------------------------	--------------	---------------	---------------------

*Hint: Always jump right as your first move.*

<b>Moonbuggy</b>	<b>66500</b>	<b>23 mins</b>	<b>Bob Brown</b>
------------------	--------------	----------------	------------------

*Hint: Wear a sweat band!*

	<b>53740</b>	<b>10 mins</b>	<b>Ben Jarvis- 6 yrs</b>
--	--------------	----------------	--------------------------

*No Hints - but his Dad says he must be a natural born astronaut.*

## AMSTRAD ACHIEVERS

### Get your name in our "HALL OF FAME"

Register your name and score on the form below, or a copy, and if possible, send a photo of the screen.

Name .....

Address .....

Telephone Number .....

Game ..... Score .....

Achieved (date) ..... Game lasted (mins.secs) .....

Signed .....

#### THIS NEXT PART MUST BE COMPLETED

Witness' Name .....

Address .....

Telephone Number.....

Occupation .....

I confirm that the above claimed score is accurate and genuine

Signed .....

Post this form along with your tips for playing the game to:  
Amstrad Achievers, The Amstrad User, Shop 2, 33 The Centreway,  
Blackburn Road, Mt. Waverley, Victoria 3149.



# Fighting Forts

## A Game from Lindsay Allen

This game takes place in a country that is in the grip of a civil war. You are in command of a heavy Fortress which is under attack by another fort captured by rebels.

To destroy the fort you have to hit it 4 times with your main gun, but the wind will blow your shells off target. While you can adjust your firing force to allow for the wind, the wind speed or direction can alter while the shell is in the air (there is a reading of the current wind speed and direction at the bottom of the screen).

Unfortunately, the wind has carried several shells over the border into a powerful neutral country and they have sent their air force to bomb both forts. The planes have lasers that can explode your heavy shells in the air, but not your anti-aircraft shells.

If you shoot down a plane, it allows a supply truck to reach you with some repair materials.

2040 - 2060 Fire RH anti-aircraft gun.  
 2070 - 2130 END of game routine.  
 2150 - 2200 Set up M/L character reading routine.  
 2210 - 3160 Set up SYMBOLS and define shapes.

### CONTROLS

Right Hand	Key 3	Increases force
	Small Enter	Decreases force
	Key .	Fires Main Gun
	Key 0	Fires anti-aircraft gun
Left Hand	Key S	Increases force
	Key X	Decreases force
	Key Z	Fires Main Gun
	LH Shift	Fires anti-aircraft gun

### HOW IT WORKS

10 - 110 Set up screen, Symbols etc.  
 120 Error Trap for SYMBOL AFTER if program rerun.  
 130 - 380 Various routines for drawing the ground and objects.  
 390 - 550 Main ground drawing routine.  
 560 - 580 Draw display window.  
 590 - 610 Set up AFTERS and EVERY timers.  
 620 Main program Loop! (That's right, only one line - all the rest are interrupts).  
 630 - 750 Wind speed Interrupt routine.  
 760 - 880 Set up attacking aircraft  
 890 - 1490 Main interrupt where:  
 890 - 1180: LH Gun key scan, move shell, detect hits, etc.  
 1190 - 1490: RH Gun key scan, move shell, detect hits, etc.  
 1500 - 1570 Exploding shell routine  
 1580 - 1610 Exploding Fort routine  
 1620 - 1650 Sets values for RH fort to explode.  
 1660 - 1690 Sets values for LH fort to explode.  
 1700 - 1840 Move attacking aircraft (see line 890).  
 1850 - 1900 Detect hit on aircraft and alter danger level.  
 1910 - 1930 Aircraft falling and exploding.  
 1940 - 2000 Bomb drop and explode.  
 2010 - 2030 Fire LH anti-aircraft gun.

```

10 REM FIGHTING FORTS LOADER
20 MODE 0: BORDER 1: INK 0, 1: INK 1, 1: IN
  K 2, 1: DIM clval(30)
30 PLOT 0, 0, 2: MOVE 102, 304: TAG: PRINT"
  FIGHTING FORTS"; : TAGOFF: LOCATE 4, 7
  : PRINT CHR$(22); CHR$(1); : PRINT"FIG
  HTING FORTS": PRINT CHR$(22); CHR$(1
  );
40 pixa=96: pixu=304: WHILE pixa<560: FO
  R doub=0 TO 14 STEP 2: clval(doub)=
  TEST(pixa, pixu-doub): NEXT doub
50 PLOT pixa, pixu-2, clval(0): FOR doub
  =4 TO 28 STEP 4: PLOT pixa, pixu-dou
  b, clval(doub/2): PLOT pixa, pixu-dou
  b-2, clval(doub/2): NEXT doub
60 pixa=pixa+4: WEND
70 BORDER 18: INK 0, 18: INK 1, 6: INK 2, 0
  : INK 3, 10
80 LOCATE 9, 12: PEN 3: PRINT"FOR": LOCAT
  E 3, 16: PRINT"AMSTRAD CPC 464"
90 RUN"!"
  
```



```

10 REM FIGHTING FORTS.17 by L.J.A
    LLEN Box 591 MURRAY BRIDGE 5253
20 CLEAR:INK 0,7:INK 1,0
30 ON ERROR GOTO 120:S%=32
40 MODE 1 :LOCATE 13,6:PRINT"FIGHTING
    FORTS":LOCATE 2,12:PRINT"L.H. Co
    ntrols          R.H. Controls":P
    RINT
50 PRINT"    Key S    Increases Forc
    e    Key 3":PRINT"    KEY X    Dec
    reases Force    Sm. ENTER"
60 PRINT"    Key Z    Fires main gun
    Key .":PRINT"L.H.Shift Fires
    anti-aircraft    Key 0"
70 LOCATE 2,23:INPUT"Please enter Lev
    el ( 1 to 3 ) ",level:IF level<1 O
    R level>3 THEN 50 ELSE maxwind=lev
    el+2:minswind=maxwind-(maxwind*2)
80 INK 0,11:INK 1,0:INK 2,9:INK 3,21:
    PAPER 0:CLS
90 ENV 1,15,-1,3:ENV 2,15,-1,10:ENV 3
    ,15,-1,0.5
100 SYMBOL AFTER 130:GOSUB 2160
110 MODE 1:GOTO 390
120 IF ERR=5 AND ERL=100 THEN GOSUB 22
    10:RESUME NEXT ELSE PRINT" ERROR N
    o."ERR" in Line"ERL:STOP
130 PRINT CHR$(22)+CHR$(1);:WHILE y<22
140 LOCATE tr,y:PRINT CHR$(171);:y=y+1
    :r=INT(RND(17)*45)
150 IF tr=1 THEN 170
160 LOCATE tr,y:IF y>rts+4 AND r=5 THE
    N PRINT USING"&";onetree$; ELSE IF
    y>rts+4 AND r=9 THEN PRINT USING"
    &";twotree$; ELSE IF y>rts+4 AND r
    =15 THEN PRINT USING"&";smhse$; EL
    SE IF y>rts+4 AND r=18 THEN PRINT
    USING"&";jeep$;
170 WEND:PRINT CHR$(22)+CHR$(0);
180 RETURN
190 r=INT(RND(23)*12):ON r GOTO 200,21
    0,210,220,230,240,250,260,270,280,
    290,230,210
200 LOCATE tr,rts:PRINT CHR$(170);:y=r
    ts+1:GOTO 300
210 LOCATE tr,rts:PRINT CHR$(171);:y=r
    ts+1:GOSUB 320:GOTO 300
220 LOCATE tr,rts:PRINT CHR$(172);:y=r
    ts+1:GOTO 300
230 LOCATE tr,rts:PRINT CHR$(173);:y=r
    ts+1:PRINT USING"&";onetree$;:GOTO
    300
240 LOCATE tr,rts:PRINT CHR$(174);:y=r
    ts+1:PRINT USING"&";twotree$;:GOTO
    300
250 IF rts=12 THEN 190 ELSE LOCATE tr,
    rts-1:PRINT CHR$(175);:y=rts:GOTO
    300
260 rts=rts-1:IF rts<12 THEN rts=rts+1
    :GOTO 190 ELSE LOCATE tr,rts:PRINT
    CHR$(176);:y=rts+1:GOTO 300
270 IF rts>19 THEN 190 ELSE LOCATE tr,
    rts:PRINT CHR$(177);:y=rts+1:rts=r
    ts+1:GOTO 300
280 IF rts>19 THEN 190 ELSE LOCATE tr,
    rts:PRINT CHR$(178);:y=rts+1:rts=r
    ts+1:GOTO 300
290 rts=rts-1:IF rts<12 THEN rts=rts+1
    :GOTO 190 ELSE LOCATE tr,rts:PRINT
    CHR$(179);:y=rts+1:GOTO 300
300 GOSUB 130
310 RETURN
320 r=INT(RND(23)*6):ON r GOTO 330,340
    ,350,360,370,330
330 PRINT USING"&";smhse$;:GOTO 380
340 PRINT USING"&";bghse$;:GOTO 380
350 PRINT USING"&";jeep$;:GOTO 380
360 PRINT USING"&";lgnlh$;:GOTO 380
370 PRINT USING"&";lgnrh$;:GOTO 380
380 RETURN
390 CLS : REM DRAW GROUND
400 PEN 2:rts=INT(RND(13)*5)+13:rfo=IN
    T(RND(7)*7)+3:rft=INT(RND(9)*7)+29
410 y=rts+1:tr=1:LOCATE tr,rts:PRINT C
    HR$(170);:GOSUB 130:y=rts+1:tr=tr+
    1:LOCATE tr,rts:PRINT CHR$(174);:G
    OSUB 130
420 FOR tr=3 TO 12
430 IF tr=rfo THEN LOCATE tr,rts-1:PEN
    1:PRINT USING"&";gunlh$:PEN 2:flh
    ast=tr+3:flhust=rts-3
440 IF tr>rfo-1 AND tr<rfo+4 THEN LOCA
    TE tr,rts:PRINT CHR$(171);:y=rts+1
    :GOSUB 130:GOTO 460
450 GOSUB 190
460 NEXT tr
470 FOR tr=13 TO 28
480 GOSUB 190
490 NEXT tr
500 FOR tr=29 TO 38
510 IF tr=rft THEN LOCATE tr,rts-1:PEN
    1:PRINT USING"&"; gunrh$:PEN 2:fr
    hast=tr-1:frhust=rts-3
520 IF tr>rft-1 AND tr<rft+4 THEN LOCA
    TE tr,rts:PRINT CHR$(171);:y=rts+1
    :GOSUB 130:GOTO 540
530 GOSUB 190
540 NEXT tr
550 y=rts+1:tr=39:LOCATE tr,rts:PRINT
    CHR$(170);:GOSUB 130:y=rts+1:tr=tr
    +1:LOCATE tr,rts:PRINT CHR$(174);:
    GOSUB 130
560 WINDOW#1,1,40,22,25:WINDOW#0,1,40,
    1,21:PAPER#1,2:CLS#1:PEN#1,1:BORDE
    R 10
570 MOVE 0,60:DRAW 639,60,1:DRAW 639,0
    :DRAW 0,0:DRAW 0,60
580 LOCATE#1,2,2:PRINT#1,"FORCE DAMAGE
    WIND SPEED FORCE DAMAGE"

```



```

590 EVERY 25,2 GOSUB 890
600 AFTER 50,1 GOSUB 630
610 r=(INT(RND(23)*90)+15)*50:AFTER r,
    0 GOSUB 760
620 GOTO 620
630 DI:r=INT(RND(9)*7)+1:ON r GOTO 640
    ,650,660,670,680,690,690,690,690,6
    90
640 GOTO 750
650 wndspeed=wndspeed+1:GOTO 700
660 wndspeed=wndspeed+2:GOTO 700
670 wndspeed=wndspeed-1:GOTO 700
680 wndspeed=wndspeed-2:GOTO 700
690 GOTO 750
700 IF wndspeed>maxwind THEN wndspeed=
    maxwind
710 IF wndspeed<minswind THEN wndspeed
    =minswind
720 LOCATE#1,18,3:IF wndspeed>0 THEN P
    RINT#1," "wndspeed""+CHR$(197)
730 IF wndspeed<0 THEN PRINT#1," "wnds
    peed" " :LOCATE#1,18,3:PRINT#1,CH
    R$(199)+" ";
740 IF wndspeed=0 THEN PRINT#1," "wnds
    peed" "
750 r=r*25:AFTER r,1 GOSUB 630:EI:RETU
    RN
760 DI:r=INT(RND(5)*6)+1:ON r GOTO 770
    ,780,790,800,810,820,770
770 pln$=jtpllh$:pldr=-2:pla=38:pdown
    $=jtpldn$:plnote=2:plnoise=5:GOTO
    830
780 pln$=jtplr$:pldr=2:pla=2:pdown$=
    jtpldn$:plnote=2:plnoise=5:GOTO 83
    0
790 pln$=prplr$:pldr=1:pla=2:pdown$=
    prpldn$:plnote=1000:plnoise=10:GOT
    O 830
800 pln$=prpll$:pldr=-1:pla=38:pdown
    $=prpldn$:plnote=1000:plnoise=10:G
    OTO 830
810 pln$=hlcpr$:pldr=1:pla=2:pdown$=
    hlcpho$:plnote=1000:plnoise=15:GOT
    O 830
820 pln$=hlcpl$:pldr=-1:pla=38:pdown
    $=hlcpho$:plnote=1000:plnoise=15:G
    OTO 830
830 plu=INT(RND(6)*5)+2:LOCATE pla,plu
    :PRINT USING"&";pln$
840 SOUND 2,plnote,2500,7,0,0,plnoise
850 airatk=1:IF lhshair=1 THEN sndposa
    =flha:sndposu=flhu:lhshair=0:GOSUB
    1500
860 IF rhshair=1 THEN sndposa=frha:snd
    posu=frhu:rhshair=0:GOSUB 1500
870 IF lhshair=2 THEN SOUND 130,plnote
    ,75,4,0,0,plnoise:GOTO 2080 ELSE I
    F rhshair=2 THEN SOUND 130,plnote,
    75,4,0,0,plnoise:GOTO 2090
880 lhaashu=0:rhaashu=0:EI:RETURN
890 DI:IF airatk>0 THEN 1700 : REM L.
    H. Control
900 IF rhhit=2 THEN 1190
910 IF lhshair=1 OR lhshair=2 THEN 990
920 IF INKEY(71)>-1 THEN flha=flhast:f
    lhu=flhust:GOTO 960
930 IF INKEY(60)>-1 AND lhforce<20 THE
    N lhforce=lhforce+1:LOCATE#1,3,3:P
    RINT#1,lhforce
940 IF INKEY(63)>-1 AND lhforce>1 THEN
    lhforce=lhforce-1:LOCATE#1,3,3:PR
    INT#1,lhforce" "
950 GOTO 1190
960 lhshair=1:PEN 3:LOCATE flha,flhu:P
    RINT CHR$(227);:SOUND 132,0,45,0,1
    ,0,15:FOR dl=1 TO 20:NEXT:PEN 1:LO
    CATE flha,flhu:PRINT CHR$(227);:FO
    R dl=1 TO 30:NEXT:LOCATE flha,flhu
    :PRINT CHR$(32);:flha=flha+1:flhu=
    flhu-1
970 lhupgo=1:lhup=2+lhforce:lhvertgo=0
    :lhvert=2+lhforce:lhdn=0
980 GOTO 1170
990 IF lhupgo=lhup THEN 1020 ELSE lhup
    go=lhupgo+1:GOSUB 1160:flha=flha+1
    :flhu=flhu-1:IF flha>40 THEN flha=
    1:GOTO 1170 ELSE 1170
1000 IF lhupgo=lhup THEN lhvert=lhvert+
    wndspeed:IF lhvert<2 THEN lhvert=2
1010 RETURN
1020 IF lhvertgo=lhvert THEN 1050
1030 IF lhvertgo=0 THEN lhvertgo=lhvert
    go+1:GOSUB 1160:flha=flha+2:flhu=f
    lhu-1:GOSUB 1000:IF flha>40 THEN f
    lha=flha-40:GOTO 1170 ELSE 1170
1040 lhvertgo=lhvertgo+1:GOSUB 1160:flh
    a=flha+1:IF flha>40 THEN flha=1:GO
    TO 1170 ELSE 1170
1050 IF lhdn=0 THEN lhdn=lhdn+1:GOSUB 1
    160:flha=flha+2:flhu=flhu+1:IF flh
    a>40 THEN flha=flha-40:GOTO 1170 E
    LSE 1170
1060 GOSUB 1160:flha=flha+1:flhu=flhu+1
    :IF flha>40 THEN flha=1:GOTO 1170
    ELSE 1170
1070 lhshair=0
1080 IF S%>245 AND S%<252 AND rhdanger<
    3 THEN rhdanger=rhdanger+1:LOCATE#
    1,36,3:PRINT#1,rhdanger:sndposa=fl
    ha-1:sndposu=flhu-1:GOTO 1500
1090 IF S%>245 AND S%<252 AND rhdanger>
    2 THEN rhdanger=rhdanger+1:LOCATE#
    1,36,3:PRINT#1,rhdanger:GOTO 1620
1100 IF S%>169 AND S%<180 THEN sndposa=
    flha-1:sndposu=flhu-1:GOTO 1500
1110 IF S%>179 AND S%<191 THEN sndposa=
    flha:sndposu=flhu:GOTO 1500
1120 IF S%=237 THEN sndposa=flha:sndpos
    u=flhu:rhshair=0:GOTO 1500
1130 IF S%>239 AND S%<246 AND lhdanger<

```



```

3 THEN lhdanger=lhdanger+1:LOCATE#
1,10,3:PRINT#1,lhdanger:sndposa=fl
ha-1:sndposu=flhu-1:GOTO 1500
1140 IF S%>239 AND S%<246 AND lhdanger>
2 THEN lhdanger=lhdanger+1:LOCATE#
1,10,3:PRINT#1,lhdanger:GOTO 1660
1150 PRINT S%:STOP
1160 IF flhu<1 THEN RETURN ELSE LOCATE
flha,flhu:PRINT CHR$(32);:RETURN
1170 IF flhu<1 THEN 1190
1180 !SCT ,flha,flhu,@S%:IF S%<>32 THEN
1070 ELSE LOCATE flha,flhu:PRINT
CHR$(237);
1190 REM R.H. Control
1200 IF lhhit=2 THEN 1490
1210 IF rhshair=1 OR rhshair=2 THEN 129
0
1220 IF INKEY(7)>>-1 THEN frha=frhast:fr
hu=frhust:GOTO 1260
1230 IF INKEY(5)>>-1 AND rhforce<20 THEN
rhforce=rhforce+1:LOCATE#1,29,3:P
RINT#1,rhforce
1240 IF INKEY(6)>>-1 AND rhforce>1 THEN
rhforce=rhforce-1:LOCATE#1,29,3:PR
INT#1,rhforce" "
1250 GOTO 1490
1260 rhshair=1:PEN 3:LOCATE frha,frhu:P
RINT CHR$(227);:SOUND 129,0,45,0,1
,0,15:FOR dl=1 TO 20:NEXT:PEN 1:LO
CATE frha,frhu:PRINT CHR$(227);:FO
R dl=1 TO 30:NEXT:LOCATE frha,frhu
:PRINT CHR$(32);:frha=frha-1:frhu=
frhu-1
1270 rhupgo=1:rhup=2+rhforce:rhvertgo=0
:rhvert=2+rhforce:rhdn=0
1280 GOTO 1470
1290 IF rhupgo=rhup THEN 1320 ELSE rhup
go=rhupgo+1:GOSUB 1460:frha=frha-1
:frhu=frhu-1:IF frha<1 THEN frha=4
0:GOTO 1470 ELSE 1470
1300 IF rhupgo=rhup THEN rhvert=rhvert-
wndspeed:IF rhvert<2 THEN rhvert=2
1310 RETURN
1320 IF rhvertgo=rhvert THEN 1350
1330 IF rhvertgo=0 THEN rhvertgo=rhvert
go+1:GOSUB 1460:frha=frha-2:frhu=f
rhu-1:GOSUB 1300:IF frha<1 THEN fr
ha=frha+40:GOTO 1470 ELSE 1470
1340 rhvertgo=rhvertgo+1:GOSUB 1460:frh
a=frha-1:IF frha<1 THEN frha=40:GO
TO 1470 ELSE 1470
1350 IF rhdn=0 THEN rhdn=rhdn+1:GOSUB 1
460:frha=frha-2:frhu=frhu+1:IF frh
a<1 THEN frha=frha+40:GOTO 1470 EL
SE 1470
1360 GOSUB 1460:frha=frha-1:frhu=frhu+1
:IF frha<1 THEN frha=40:GOTO 1470
ELSE 1470
1370 rhshair=0: REM R.H. Hit
1380 IF S%>239 AND S%<246 AND lhdanger<
3 THEN lhdanger=lhdanger+1:LOCATE#
1,10,3:PRINT#1,lhdanger:sndposa=fr
ha+1:sndposu=frhu-1:GOTO 1500
1390 IF S%>239 AND S%<246 AND lhdanger>
2 THEN lhdanger=lhdanger+1:LOCATE#
1,10,3:PRINT#1,lhdanger:GOTO 1660
1400 IF S%>169 AND S%<180 THEN sndposa=
frha+1:sndposu=frhu-1:GOTO 1500
1410 IF S%>179 AND S%<191 THEN sndposa=
frha:sndposu=frhu:GOTO 1500
1420 IF S%=237 THEN sndposa=frha:sndpos
u=frhu:lhshair=0:GOTO 1500
1430 IF S%>245 AND S%<252 AND rhdanger<
3 THEN rhdanger=rhdanger+1:LOCATE#
1,36,3:PRINT#1,rhdanger:sndposa=fr
ha+1:sndposu=frhu-1:GOTO 1500
1440 IF S%>245 AND S%<252 AND rhdanger>
2 THEN rhdanger=rhdanger+1:LOCATE#
1,36,3:PRINT#1,rhdanger:GOTO 1620
1450 PRINT S%:STOP
1460 IF frhu<1 THEN RETURN ELSE LOCATE
frha,frhu:PRINT CHR$(32);:RETURN
1470 IF frhu<1 THEN 1490
1480 !SCT ,frha,frhu,@S%:IF S%<>32 THEN
1370 ELSE LOCATE frha,frhu:PRINT
CHR$(237);
1490 EI:RETURN
1500 IF sndposa>20 THEN SOUND 129,0,150
,0,2,0,15 ELSE SOUND 132,0,150,0,2
,0,15
1510 IF sndposu<1 THEN 1570
1520 IF sndposa>40 THEN sndposa=1 ELSE
IF sndposa<1 THEN sndposa=40
1530 LOCATE sndposa,sndposu:PEN 3:PRINT
CHR$(238);:BORDER 26:FOR dl=1 TO
20:NEXT:LOCATE sndposa,sndposu:PEN
1:PRINT CHR$(238);:FOR dl=1 TO 25
:NEXT
1540 LOCATE sndposa,sndposu:PRINT USING
"&";CHR$(32)+CHR$(11)+CHR$(8)+CHR
$(32);:BORDER 10
1550 IF lhhit=2 THEN 2080 ELSE IF rhhit
=2 THEN 2090
1560 IF airatk=1 THEN RETURN
1570 EI:RETURN
1580 LOCATE ftexpa,ftexpu:PEN 3:PRINT C
HR$(238);:BORDER 26:FOR dl =1 TO 2
0:NEXT:LOCATE ftexpa-1,ftexpu:PRIN
T CHR$(238)+CHR$(32)+CHR$(238);:FO
R dl =1 TO 30:NEXT:SOUND 2,300,175
,0,2,0,15:BORDER 10:LOCATE ftexpa-
1,ftexpu-1:PRINT CHR$(238)+CHR$(23
8)+CHR$(238)
1590 PEN 1:LOCATE ftexpa-1,ftexpu:PRINT
CHR$(238)+CHR$(32)+CHR$(238);:FOR
dl =1 TO 35:BORDER 26:NEXT:LOCATE
ftexpa-1,ftexpu-1:PRINT CHR$(238)
+CHR$(238)+CHR$(238);
1595 LOCATE ftexpa,ftexpu-2:PRINT CHR$(
238):FOR dl =1 TO 35:NEXT

```



```

1600 LOCATE ftexpa-1,ftexpu:PRINT" "; :AFTER r,0 GOSUB 760
:LOCATE ftexpa-1,ftexpu-1:PRINT" 1840 EI:RETURN
";:BORDER 10:FOR dl =1 TO 35:NEXT 1850 IF lhaashu>0 THEN LOCATE lhaasha,1
:LOCATE ftexpa,ftexpu-2:PRINT" "; haashu:PRINT" ";
1610 RETURN 1860 IF rhdanger>0 THEN rhdanger=rhdang
1620 ftexpa=frhast+2:ftexpu=frhust+2:SO er-1:LOCATE#1,36,3:PRINT#1,rhdange
UND 129,0,155,0,2,0,15:GOSUB 1580 r:GOTO 1890
1630 IF rhshair=1 THEN rhshair=2:rhhit= 1870 IF rhaashu>0 THEN LOCATE rhaasha,r
2:EI:RETURN haashu:PRINT" ";
1640 IF lhhit=2 THEN 2070 1880 IF lhdanger>0 THEN lhdanger=lhdang
1650 GOTO 2090 er-1:LOCATE#1,10,3:PRINT#1,lhdange
1660 ftexpa=flhast-2:ftexpu=flhust+2:SO 1890 SOUND 130,0,150,0,2,0,15:BORDER 26
UND 132,0,155,0,2,0,15:GOSUB 1580 :FOR dl=1 TO 20:NEXT:BORDER 10:LOC
1670 IF lhshair=1 THEN lhshair=2:lhhit= ATE pla,plu:PRINT USING"&";plwipe$
2:EI:RETURN ;:LOCATE pla,plu:PRINT USING"&";pl
1680 IF rhhit=2 THEN 2070 down$;;airatk=2
1690 GOTO 2080 1900 EI:RETURN
1700 IF airatk=2 THEN 1910 ELSE IF aira 1910 PEN 0:LOCATE pla,plu:PRINT USING"&
tk=3 THEN 1940 :REM plane attack ";pldown$;;plu=plu+1:IF plu>7 THEN
1710 IF lhaashu<1 THEN GOSUB 2010 ELSE sndposa=pla:sndposu=plu:airatk=3:
LOCATE lhaasha,lhaashu:PRINT" "; SOUND 130,0,150,0,2,0,15:GOTO 1500
1720 IF rhaashu<1 THEN GOSUB 2040 ELSE 1920 PEN 1:LOCATE pla,plu:PRINT USING"&
LOCATE rhaasha,rhaashu:PRINT" "; ";pldown$;;IF bomb>0 THEN 1940
1730 LOCATE pla,plu:PRINT USING"&";plwi 1930 EI:RETURN
pe$; 1940 IF bomb=0 THEN 1830
1740 r=INT(RND(7)*3)-1:plu=plu+r:IF plu 1950 LOCATE bmba,bmbu:PRINT" ";:bmbu=bm
<1 THEN plu=1 ELSE IF plu>6 THEN p bu+1:IF bmbu<8 AND bmba>2 AND bmba
lu=6 <38 THEN bmba=bmba+pldr
1750 pla=pla+pldr:IF pldr>0 AND pla>37 1960 !SCT,bmba,bmbu,@S%:IF S%=32 OR S%=
THEN airatk=3:SOUND 130,plnote,75, 237 OR S%<170 THEN LOCATE bmba,bmb
6,0,0,plnoise:SOUND 2,plnote,75,5, u:PRINT CHR$(252);:EI:RETURN
0,0,plnoise:SOUND 2,plnote,75,3,0, 1970 bomb=0:IF S%>239 AND S%<246 THEN 1
0,plnoise:GOTO 1940 hdanger=lhdanger+1:LOCATE#1,10,3:P
1760 IF pldr<0 AND pla<2 THEN airatk=3: RINT#1,lhdanger:IF lhdanger<4 THEN
SOUND 130,plnote,75,4,0,0,plnoise: sndposa=bmba:sndposu=bmbu-1:GOTO
SOUND 2,plnote,75,5,0,0,plnoise:SO 1500 ELSE 1660
UND 2,plnote,75,3,0,0,plnoise:GOTO 1980 IF S%>245 AND S%<252 THEN rhdanger
1940 =rhdanger+1:LOCATE#1,36,3:PRINT#1,
1770 LOCATE pla,plu:PRINT USING"&";pln$ rhdanger:IF rhdanger<4 THEN sndpos
; a=bmba:sndposu=bmbu-1:GOTO 1500 EL
1780 lhaasha=lhaasha+2:lhaashu=lhaashu- 1990 IF S%<180 THEN bmbu=bmbu-1
2:IF lhaashu<1 THEN 1790 ELSE !SCT 2000 sndposa=bmba:sndposu=bmbu:GOTO 150
,lhaasha,lhaashu,@S%:IF S%=32 OR 0
S%=252 OR S%=237 THEN LOCATE lhaas 2010 IF INKEY(21)=-1 THEN RETURN
ha,lhaashu:PRINT CHR$(237); ELSE 1 2020 lhaasha=flhast:lhaashu=flhust:SOUN
870 D 132,0,45,0,1,0,15:LOCATE lhaasha
1790 rhaasha=rhaasha-2:rhaashu=rhaashu- ,lhaashu:PEN 3:PRINT CHR$(227);:FO
2:IF rhaashu<1 THEN 1800 ELSE !SCT R dl=1 TO 20:NEXT:PEN 1:LOCATE lha
,rhaasha,rhaashu,@S%:IF S%=32 OR asha,lhaashu:PRINT CHR$(227);:FOR
S%=237 OR S%=252 THEN LOCATE rhaas dl=1 TO 10:NEXT:LOCATE lhaasha,lha
ha,rhaashu:PRINT CHR$(237); ELSE 1 ashu:PRINT CHR$(32);
850 2030 RETURN
1800 IF bomb=0 THEN r=INT(RND(13)*7):IF 2040 IF INKEY(15)=-1 THEN RETURN
r=4 THEN bomb=1:bmba=pla:bmbu=plu 2050 rhaasha=frhast:rhaashu=frhust:SOUN
+2:LOCATE bmba,bmbu:PRINT CHR$(252 D 129,0,45,0,1,0,15:LOCATE rhaasha
);:GOTO 1820 ,rhaashu:PEN 3:PRINT CHR$(227);:FO
1810 IF bomb>0 THEN 1940 R dl=1 TO 20:NEXT:PEN 1:LOCATE rha
1820 EI:RETURN asha,rhaashu:PRINT CHR$(227);:FOR
1830 airatk=0:r=(INT(RND(23)*90)+15)*50

```



```

dl=1 TO 10:NEXT:LOCATE rhaasha,rha
ashu:PRINT CHR$(32);
2060 RETURN
2070 FOR dl =1 TO 1800:NEXT:MODE 1:INK
0,7:PEN 1:LOCATE 4,12:PRINT"BAD LU
CK! YOU BOTH LOOSE!":GOTO 2100
2080 FOR dl =1 TO 1800:NEXT:MODE 1:INK
0,7:PEN 1:LOCATE 10,12:PRINT"CONGR
ATULATIONS!":LOCATE 6,14:PRINT" RI
GHT HAND PLAYER WINS!":GOTO 2100
2090 FOR dl =1 TO 1800:NEXT:MODE 1:INK
0,7:PEN 1:LOCATE 10,12:PRINT" G
OOD WORK!":LOCATE 6,14:PRINT" LEF
T HAND PLAYER WINS!":GOTO 2100
2100 IF INKEY$<>" THEN 2100
2110 LOCATE 2,23:PRINT"Do you want anot
her game?"
2120 a$=INKEY$: IF a$="" THEN 2120
2130 IF a$="y"OR a$="Y" THEN 10 ELSE EN
D
2140 STOP
2150 REM SCREEN TEST (SCT) ROUTINE --us
e" \SCT ,x,y,@S% "
2160 MEMORY 39999:FOR q=40000 TO 40065:
READ w:POKE q,w:NEXT
2170 CALL 40000
2180 DATA &01,&4A,&9C,&21,&7F,&9C,&CD,&
D1,&BC,&C9,&4F,&9C,&C3,&53,&9C,&53
,&43,&D4,&00,&CD,&78,&BB,&7C,&32,&
7E,&9C,&7D,&32
2190 DATA &7D,&9C,&DD,&66,&04,&DD,&6E,&
02,&CD,&75,&BB,&CD,&60,&BB,&DD,&6E
,&00,&DD,&66,&01,&77,&3A,&7E,&9C,&
67,&3A,&7D,&9C
2200 DATA &6F,&CD,&75,&BB,&C9,00,00,00,
00,00,00
2210 SYMBOL 237,0,0,0,24,24,0,0,0
2220 SYMBOL 240,0,3,7,15,31,63,127,255
2230 SYMBOL 241,60,255,255,255,255,255
,255,255
2240 SYMBOL 242,0,192,224,240,248,252,
254,255
2250 SYMBOL 243,0,1,3,7,15,15,15,15
2260 SYMBOL 244,126,255,255,255,255,25
5,255,255
2270 SYMBOL 245,1,130,196,200,208,224,
224,224
2280 SYMBOL 246,126,254,255,255,255,25
5,255,255
2290 SYMBOL 247,0,128,192,224,240,240,
240,240
2300 SYMBOL 248,128,65,35,19,11,7,7,7
2310 SYMBOL 249,0,192,224,240,248,252,
254,254
2320 SYMBOL 250,0,2,7,15,31,63,127,255
2330 SYMBOL 251,60,255,255,255,255,255
,255,255
2340 gunlh$=CHR$(240)+CHR$(241)+CHR
$(242)+CHR$(11)+CHR$(8)+CHR$(
8)+CHR$(8)+CHR$(243)+CHR$(2
44)+CHR$(245)
2350 gunrh$=CHR$(250)+CHR$(251)+CHR
$(249)+CHR$(11)+CHR$(8)+CHR$(
8)+CHR$(8)+CHR$(248)+CHR$(2
46)+CHR$(247)
2360 SYMBOL 170,143,223,255,255,255,25
5,255,255
2370 SYMBOL 171,255,255,255,255,255,25
5,255,255
2380 SYMBOL 172,129,195,227,255,255,25
5,255,255
2390 SYMBOL 173,153,255,255,255,255,25
5,255,255
2400 SYMBOL 174,231,255,255,255,255,25
5,255,255
2410 SYMBOL 175,0,0,0,0,8,28,126,255
2420 SYMBOL 176,1,3,7,7,15,63,127,255
2430 SYMBOL 177,128,192,224,248,248,25
2,254,255
2440 SYMBOL 178,128,192,224,224,240,25
2,254,255
2450 SYMBOL 179,1,3,7,31,31,63,127,255
2460 SYMBOL 180,82,82,36,24,8,8,8,8
2470 SYMBOL 181,153,86,36,36,36,36,36,
36
2480 SYMBOL 182,52,74,246,94,251,110,2
15,126
2490 SYMBOL 183,0,0,1,10,5,11,2,0
2500 SYMBOL 184,0,0,32,80,168,208,64,1
60
2510 SYMBOL 185,126,126,126,90,90,126,
126,126
2520 SYMBOL 186,0,0,0,0,24,60,126,255
2530 SYMBOL 187,255,255,153,153,255,15
3,153,255
2540 SYMBOL 188,0,16,32,225,255,255,10
2,102
2550 SYMBOL 189,128,64,36,24,28,62,126
,231
2560 SYMBOL 190,1,2,36,24,56,124,126,2
31
2570 onetree$=CHR$(15)+CHR$(1)+CHR$(11)
+CHR$(8)+CHR$(180)+CHR$(11)+CHR$(8
)+CHR$(8)+CHR$(15)+CHR$(3)+CHR$(18
3)+CHR$(182)+CHR$(15)+CHR$(2)+CHR$(
10)+CHR$(10)
2580 twotree$=CHR$(15)+CHR$(1)+CHR$(
11)+CHR$(8)+CHR$(181)+CHR$(11)
+CHR$(8)+CHR$(8)+CHR$(15
)+CHR$(3)+CHR$(183)+CHR$(182
)+CHR$(184)+CHR$(15)+CHR$(2
)+CHR$(10)+CHR$(10)
2590 smhse$=CHR$(15)+CHR$(1)+CHR$(11)
+CHR$(8)+CHR$(185)+CHR$(11)
+CHR$(8)+CHR$(186)+CHR$(15
)+CHR$(2)+CHR$(10)+CHR$(10)
2600 bghse$=CHR$(15)+CHR$(1)+CHR$(11)
+CHR$(8)+CHR$(187)+CHR$(11)
+CHR$(8)+CHR$(187)+CHR$(15
)+CHR$(2)+CHR$(10)+CHR$(10)

```



```

2610 jeep$=CHR$( 15 )+CHR$( 1 )+CHR$( 1
1 )+CHR$( 8 )+CHR$( 188 )+CHR$( 15
)+CHR$( 2 )+CHR$( 10 )
2620 lgnlh$=CHR$( 15 )+CHR$( 1 )+CHR$(
11 )+CHR$( 8 )+CHR$( 189 )+CHR$( 1
5 )+CHR$( 2 )+CHR$( 10 )
2630 lgnrh$=CHR$( 15 )+CHR$( 1 )+CHR$(
11 )+CHR$( 8 )+CHR$( 190 )+CHR$( 1
5 )+CHR$( 2 )+CHR$( 10 )
2640 SYMBOL 130 , 0 , 3 , 7 , 63 , 25
5 , 7 , 3 , 1
2650 SYMBOL 131 , 0 , 129 , 3 , 198 ,
254 , 4 , 128 , 192
2660 SYMBOL 132 , 0 , 129 , 192 , 99
, 127 , 32 , 1 , 3
2670 SYMBOL 133 , 0 , 192 , 224 , 252
, 255 , 224 , 192 , 128
2680 SYMBOL 134 , 96 , 56 , 28 , 8 ,
8 , 8 , 152 , 218
2690 SYMBOL 135 , 254 , 126 , 60 , 24
, 24 , 24 , 8 , 8
2700 SYMBOL 136 , 192 , 224 , 224 , 2
25 , 225 , 252 , 255 , 252
2710 SYMBOL 137 , 225 , 225 , 224 , 2
24 , 224 , 224 , 192 , 0
2720 SYMBOL 138 , 0 , 0 , 0 , 128 , 1
92 , 227 , 255 , 127
2730 SYMBOL 139 , 3 , 7 , 7 , 135 , 1
35 , 63 , 255 , 63
2740 SYMBOL 140 , 0 , 0 , 0 , 1 , 3 ,
199 , 255 , 254
2750 SYMBOL 141 , 135 , 135 , 7 , 7 ,
7 , 7 , 3 , 0
2760 SYMBOL 142 , 248 , 240 , 224 , 1
92 , 192 , 192 , 224 , 224
2770 SYMBOL 143 , 255 , 255 , 254 , 2
24 , 224 , 224 , 64 , 88
2780 SYMBOL 144 , 127 , 127 , 63 , 0
, 0 , 0 , 0 , 3
2790 SYMBOL 145 , 0 , 195 , 199 , 47
, 31 , 15 , 2 , 31
2800 SYMBOL 146 , 128 , 240 , 136 , 1
32 , 228 , 252 , 16 , 254
2810 SYMBOL 147 , 0 , 0 , 0 , 0 , 0 ,
219 , 128 , 128
2820 SYMBOL 148 , 0 , 0 , 0 , 0 , 0 ,
109 , 0 , 0
2830 SYMBOL 149 , 1 , 15 , 17 , 33 ,
39 , 63 , 8 , 127
2840 SYMBOL 150 , 0 , 195 , 227 , 244
, 248 , 240 , 64 , 248
2850 SYMBOL 151 , 0 , 0 , 0 , 0 , 0 ,
219 , 1 , 1
2860 SYMBOL 152 , 0 , 0 , 0 , 0 , 0 ,
182 , 0 , 0
2870 SYMBOL 153 , 60 , 66 , 255 , 129
, 90 , 60 , 24 , 36
2880 SYMBOL 154 , 4 , 4 , 14 , 24 , 4
8 , 32 , 0 , 0
2890 SYMBOL 155 , 0 , 36 , 24 , 24 ,
24 , 24 , 24 , 0
2900 SYMBOL 156 , 32 , 32 , 112 , 24
, 12 , 4 , 0 , 0
2910 SYMBOL 157 , 0 , 0 , 0 , 0 , 125
, 0 , 0 , 0
2920 SYMBOL 158 , 0 , 0 , 0 , 0 , 190
, 0 , 0 , 0
2930 SYMBOL 159 , 60 , 66 , 153 , 153
, 153 , 129 , 66 , 60
2940 SYMBOL 160 , 0 , 0 , 0 , 127 , 2
, 2 , 6 , 4
2950 SYMBOL 161 , 0 , 0 , 0 , 254 , 6
4 , 64 , 96 , 32
2960 SYMBOL 162 , 189 , 66 , 165 , 15
3 , 153 , 129 , 66 , 60
2970 jtpllh$=CHR$( 130 )+CHR$( 131 )+CH
R$( 0 )+CHR$( 0 )+CHR$( 0 )+CHR$(
0 )+CHR$( 0 )+CHR$( 0 )
2980 jtplrh$=CHR$( 132 )+CHR$( 133 )+CH
R$( 0 )+CHR$( 0 )+CHR$( 0 )+CHR$(
0 )+CHR$( 0 )+CHR$( 0 )
2990 jtpldn$=CHR$(134)+CHR$(10)+CHR$(8)
+CHR$(135)
3000 prplrh$=CHR$( 138 )+CHR$( 136 )+CH
R$( 10 )+CHR$( 8 )+CHR$( 137 )+CHR
$( 0 )+CHR$( 0 )+CHR$( 0 )
3010 prpllh$=CHR$( 139 )+CHR$( 140 )+CH
R$( 10 )+CHR$( 8 )+CHR$( 8 )+CHR$(
141 )+CHR$( 0 )+CHR$( 0 )
3020 prpldn$=CHR$( 142 )+CHR$( 10 )+CHR
$( 8 )+CHR$( 143 )+CHR$( 8 )+CHR$(
8 )+CHR$( 144 )+CHR$( 0 )
3030 hlcprh$=CHR$(148)+CHR$(147)+CHR$(1
0)+CHR$(8)+CHR$(8)+CHR$(145)+CHR$(
146)
3040 hlcplh$=CHR$(151)+CHR$(152)+CHR$(1
0)+CHR$(8)+CHR$(8)+CHR$(149)+CHR$(
150)
3050 plwipe$=CHR$(32)+CHR$(32)+CHR$(10)
+CHR$(8)+CHR$(8)+CHR$(32)+CHR$(32)
3060 bomblh$=CHR$( 154 )+CHR$( 0 )+CHR$(
0 )+CHR$( 0 )+CHR$( 0 )+CHR$( 0 )
+CHR$( 0 )+CHR$( 0 )
3070 bombdw$=CHR$( 155 )+CHR$( 0 )+CHR$(
0 )+CHR$( 0 )+CHR$( 0 )+CHR$( 0 )
+CHR$( 0 )+CHR$( 0 )
3080 bombrh$=CHR$( 156 )+CHR$( 0 )+CHR$(
0 )+CHR$( 0 )+CHR$( 0 )+CHR$( 0 )
+CHR$( 0 )+CHR$( 0 )
3090 SYMBOL 163 , 128 , 192 , 32 , 32
, 32 , 192 , 0 , 32
3100 SYMBOL 164 , 0 , 1 , 2 , 2 , 2 ,
1 , 0 , 2
3110 SYMBOL 165 , 0 , 1 , 3 , 3 , 3 ,
1 , 0 , 2
3120 SYMBOL 166 , 128 , 192 , 224 , 2
24 , 224 , 192 , 0 , 32
3130 SYMBOL 167 , 0 , 1 , 1 , 3 , 63
, 3 , 0 , 0
3140 hlcpho$=CHR$(148)+CHR$(147)+CHR$(1
0)+CHR$(8)+CHR$(8)+CHR$(164)+CHR$(
163)
3150 hlcpto$=CHR$( 165 )+CHR$( 166 )+CH
R$( 11 )+CHR$( 8 )+CHR$( 8 )+CHR$(
148 )+CHR$( 147 )+CHR$( 0 )
3160 RETURN

```



## User Group Contact List

Please note that the following names are listed as contact points for new user groups and should NOT be viewed as a problem solving service. See other list for established groups.

### NSW

Chris Craven	Canowindra	(063) 44 1150
Trevor Farrell	Coolah/Mudgee area	(063) 77 1374
T.J. Webb	Glossodia	(045) 76 5291
David Higgins	Inverell	(067) 22 1867
John Patterson	Lismore	(066) 21 3345
Paul Wilson	Moruya	(044) 74 3160
Frank Humphreys	Mummulgum	(066) 64 7290
Martin Clift	Narrabri	(067) 92 3077
Bob Hall	Newcastle	(049) 52 6915
Reuben Carlsen	North Sydney	(02) 957 2505
Stephen Gribben	Singleton	(065) 72 2732
Ken Needs	St. Ives	(02) 449 5416
Chas Fletcher	Toongabbie	(02) 631 5037
Nick Bruin Snr.	Tweed Valley	(066) 79 3280
Jim Owen	Uranga	(066) 55 6190
John Harwood	Windale	(049) 48 5337

### Vic

Stuart McLean	4/304 Albert St. Sebastopol, 3356	
David Carbone	Burwood	(03) 29 4135
Rod Anderson	Camperdown	(055) 93 2262
Paul Walker	Heathmont	(03) 729 8657
Terry Dovey	Horsham	(053) 82 3353
Andrew Portbury	Leongatha	(056) 62 3694
Ron Butterfield	Leopold	(052) 50 2251

Sue Kelly	Manangatang	(050) 35 1402
Keith McFadden	Numurkah	(058) 62 2069
Mrs. G. Chapman	South Clayton	(03) 551 4897
Lindsay Parker	Wandin North	(059) 64 4837

### QLD

Debbie Topp	Bribie Island	(075) 48 1688
Steven Doyle	Caloundra	(071) 91 3147
Mick O'Regan	Gladstone	(079) 79 2548
Kylie Telford	Goondiwindi	Calingunee246 (weekendsonly)
D.F. Read	Ingham	(077) 77 8576
Tim Takken	Ipswich	(07) 202 4039
Michael Toussaint	Loganlea	(07) 200 5414
Alan Laird	Maryborough	(071) 22 1982
R.C. Watterton	Toowoomba	(076) 35 4305

### SA

Lindsay Allen	Murray Bridge	(085) 32 2340
---------------	---------------	---------------

### WA

Dave Andersen	6 Kitchener Rd Merredin, 6415	
Graeme Worth	Scarborough	(09) 341 5211
P.M. Nuyens	Waroona	(095) 33 1179

### TAS

Conal McClure	Scottsdale	(003) 52 2514
---------------	------------	---------------

### NT

G.P. Heron	Tiwi	(089) 27 8814
------------	------	---------------

# NATIONWIDE USER GROUPS

## ROCKINGHAM/KWINANA USER GROUP

Contact Bob Harwood on 095 27 1777 for further details on meeting times.

## WESTERN AUSTRALIA

### AMSWEST, Perth

President:	Tony Clitheroe	(09 275 1257)
Secretary:	Mrs. P.T. Ardron	(09 361 8975)
Treasurer:	Eric Stallard	(09 339 6361)

Regular meetings take place at a venue in Shenton Park on the first and third Tuesdays of each month starting at 7.30p.m.

### SOUTHSIDE AMSTRAD USER CLUB

President:	John Marshall	(09 390 7335)
Secretary:	Linda Marshall	(09 390 7335)
Treasurer:	Eric Tytherleigh	(09 390 8865)
Librarian:	Roy Depurouzel	(09 457 9026)

SAUC meets from 7.00 p.m. every 2nd and 4th Tuesday of each month at Thornlie Technical College. All meetings are socially orientated with a minimum of business matters and can include software and hardware demonstrations. Discounts have been obtained from most local dealers and are available to financial members.

## SOUTH AUSTRALIA

### AMSTRAD COMPUTER CLUB INC. (SA)

President:	Chris Sowden	(08 295 5923)
Vice Pres:	Frank Matzka	(08 382 2101)
Treasurer:	Les Jamieson	(08 356 9612)

The group now meets each Tuesday at the Unley High School between 6.30 p.m. and 9.00 p.m. Any of the above officers can be contacted for further details and correspondence can be addressed to PO Box 210, Parkholme, 5043.

### PORT PIRIE AMSTRAD USER GROUP

President:	Rick Cable	(086 32 5967)
Treasurer:	Dave Green	(086 32 6834)

The group meets at 7.30 p.m. on the first Monday of each month at the Princess Park Scout Hall, Three Chain Road, Solomontown. Meetings are well attended with members from Pt. Broughton, Warnertown and even Burra. For further details contact Rick Cable who will advise on the benefits of belonging to this group.



## VICTORIA

### WESTERN AMSTRAD USER GROUP

President: Mike McQueen (03 312 5594)  
Secretary: Peter Pilbeam (03 336 0705)  
Treasurer: Frank Melino (03 337 2495)

The meetings are held on each alternate Tuesday and Sunday (to allow for shift workers) at the Tottenham North Primary School, South Road, Braybrook.

### CENTRAL AMSTRAD USER SOCIETY

President: Rimon Russo (03 428 4281)  
Vice-Pres: Dennis Whelan (03 367 6614)  
Treasurer: Fred Gillan (03 598 5780)  
PR Officer: John Holmes (03 434 1607)

Meetings are held twice a month in the Hall at the corner of Church and Somerset Streets, Richmond on the first Sunday of each month commencing at 4.00 p.m. and generally twelve days later on a Friday evening starting at 7.00 p.m. All meetings are conducted in a friendly atmosphere - families are welcome.

### EASTERN AMSTRAD USER GROUP

President: Tony Blakemore (03 878 6212)  
Secretary: Andrew Martin (03 729 8471)  
Treasurer: Ron Dunn (03 277 7868)

Regular meetings are held on the first Sunday of every month at the Box Hill Scout Hall, Tyne St. (The Hall is located in Halligan Park between Watts and Mersey Streets). Proceedings commence at 2.00 p.m.

### SOUTHERN AMSTRAD USER GROUP

President: Mike Prezents (03 781 2158)  
Secretary: Martin Scragg (059 78 6949)  
Treasurer: Steve Issell (03 786 9340)

Meetings are held on the third Tuesday of every month (except December) from 7.30 p.m. to 10.30 p.m. The venue is the Senoir Campus at John Paul College, Frankston.

### NORTHERN AMSTRAD USER GROUP

Contact: Brian Ellis (03 469 4425)

This group caters for users in the Preston/Coburg areas. Meetings are devoted to learning more about computers and consist of lectures, demonstrations and practical workshops of projects such as modems, expansion busses etc. Arcade games are banned from meetings. The Group is privately funded and there are no membership fees.

### SALE AMSTRAD GROUP

Organiser: Alan Harris (051 44 1454)

The Group meets informally every Thursday night from 7.00p.m. at the Sale Neighbourhood House in Leslie Street. In addition, small group tutorials are held twice a month. Contact Alan Harris for further details.

## ACT

### ACT AMSTRAD USER GROUP

Convenor: Arthur McGuffin (062 31 9437)  
Secretary: Kevin Loughrey (062 31 2991)  
Treasurer: Kevin Cryer (062 91 9881)

The group meets at 7.30 p.m. on the first Wednesday of each month in the Seminar Room of the Oliphant Building at the Research School of Physical Science, Australian National University.

## NEW SOUTH WALES

### JUBOL AMSTRAD USER GROUP of COFFS HARBOUR and DISTRICT

Contacts: Bruce Jones (066 52 8334)  
Jim Owen (066 55 6190)

The "JUBOL" User Group is currently a small group covering the Coffs Harbour area. They have already met a few times in an informal manner and are very keen for other users in the area to contact them.

### SYDNEY AMSTRAD COMPUTER CLUB

President: Raja Vijayenthiran (02 519 4106)  
Secretary: Reed Walters (02 560 9487)  
Treasurer: Jim Chryss (02 327 7872)  
Junior Rep: Daniel Story

This club, which services the inner City area, meets on the first Monday of every month at 7.00 p.m. A permanent meeting place has not yet been established so prospective members are advised to contact one of the above officers.

## QUEENSLAND

### BRISBANE AMSTRAD COMPUTER CLUB

President: Paul Witsen (07 371 9259)  
Secretary: Mal Harper (07 288 3578)  
Treasurer: Ian Cartwright (07 369 9364)

Meetings are held on the first Tuesday of each month at Junction Park State School, Annerley starting at 7.30 p.m. in Room 15a.

### SOUTHSIDE AMSTRAD USER GROUP (QLD)

President: Michael Toussaint (07 200 5414)  
Secretary: Sylvia Wilson (07 209 1947)  
Treasurer: Col Liebke (07 200 5555)

Meetings take place every third Saturday of the month at 10 Carramar Street, Loganlea starting at 2.00 p.m. The group was formed to service the southern outskirts of Brisbane.

## TASMANIA

### SOUTH TASMANIA AMSTRAD USER CLUB

Meetings will take place at the Elizabeth Matriculation College (off Elizabeth Street) on the first Wednesday of each month, commencing at 7.30 p.m. Enquiries should be made to Graham West - (002) 34 5817.









# The Learning Centre

## An Introduction to Music - Part Three

by Peter Campbell

Before we commence the third part of our Music tutorial, you may have noticed a gap in last month's article and a reference to Figure 4. Well, Figure 4 should have been where the gap was! (Ten demerit points to the printer). To correct the omission, below is the missing picture, although if you have entered the listings so far, you would have been presented with a diagram containing similar information.

Semibreve		Quaver	
Minim		Semi-quaver	
Crotchet		Demisemi-quaver	

You may have gathered by now that the listings which have accompanied the articles in this series to date will be linked together (possibly edited a little on the way), to eventually produce ..... well, wait and see. However, before we go any further and merge more routines it is necessary to look at the SOUND command.

"The sound features of a computer determine whether or not it sounds like a bluebottle in an empty cocoa tin - or if it can produce an acceptable representation of an electronic musical instrument." So says the User Instructions for the CPC464.

The Amstrad computers use an AY-3-8912 sound chip (a programmable sound generator of the same type used in the MSX series of computers). This gives three channels over an eight octave range, with, as it says in the aforementioned User Instructions, 'full control of the amplitude and tone envelopes'. Such versatility comes at the price of some fairly lengthy command structures.

That last paragraph gives more than a clue to what is contained in the SOUND command and the two other commands with which it can be associated, the Envelopes for Volume and Tone (ENV and ENT). We need to be able to specify which of the three channels, what note in the eight octaves and which envelopes.

### Channels

Not unexpectedly, therefore, the first number after the SOUND command specifies which channel. What is unexpected is the range from 1 to 225! The manual unhelpfully says 'the input integer range is given in decimal form, but when converted to an 8 bit value (thus yielding a

bit significant result), the active bits specify the ... commands.' Some sense can be made of that by looking at the binary number equivalents of our decimal number inputs.

Decimal	Binary	Bit	Command
1	00000001	0	send SOUND to channel A
2	00000010	1	send SOUND to channel B
4	00000100	2	send SOUND to channel C
8	00001000	3	rendezvous with channel A
16	00010000	4	rendezvous with channel B
32	00100000	5	rendezvous with channel C
64	01000000	6	hold
128	10000000	7	flush

Binary numbers are based on the number 2 and consist entirely of 1's and 0's. What counts is the position of the digit '1' as this determines which 'bit', or 'bits', of the 'byte' is affected. Don't worry if you don't really follow all that. As long as you can see the connection between the decimal number and the command, you will be able to manage quite successfully. The important thing to remember is to add the numbers together whenever you want to combine commands.

Consider these examples:

The desired instruction reads as follows:

*Flush the sound from channels A,B and C.*

Flush is 128, sound to channel A is 1, sound to channel B is 2 and sound to channel C is 4. Thus the number we require is  $128+1+2+4 (=135)$ .

If we wish to *rendezvous channels A and C*, then the number game is played like this:

Sending sound to channel A is 1, sending sound to channel C is 4, rendezvous with channel C is 32 and with channel A is 8. Therefore our first sound command will start SOUND  $1+32 (=33)$ , whilst the second command will start SOUND  $4+8 (=12)$ .

If we have programmed a sound which we don't want immediately then we hold it using 64. *To send a sound to channel B and hold* becomes SOUND  $2+64 (=66)$ . *To release the sound later*, we use the associated BASIC command RELEASE (surprise! surprise!), which takes the form RELEASE 2. The range of values for the RELEASE command is from 1 to 7, as they are the only numbers we can make from 1, 2 and 4 - the three numbers which send sound to a channel or combination of channels.



## Tone or Pitch

The next number specifies the property of the sound which Amstrad calls 'sound period'. The more common way of describing a sound is in terms of its frequency. Once measured in kilocycles/second, frequency is now measured in hertz and kiloHertz. Guess what? A Hertz is one kilocycle/second!

Frequency (expressed in Hertz) is related to period by the simple equation:

Frequency = 1/Period (expressed in seconds)

The relationship between frequency and Amstrad's <tone period> is:

Frequency (expressed in Hertz)=125000/<tone period>

This is one of the two formulae that we need to program music, if we wish to avoid the tedium of looking each note up in the table in the Appendices of the User Instructions.

The frequency of any note can be calculated from International A as follows:

Frequency =  $440 * (2^{(Octave + ((n-10)/12))})$

(where n is 1 for C, 2 for C#, 3 for D etc.)

(It works, but I don't pretend to understand it!!!)

That monstrosity gives us the second formula that we need. Tone periods, however determined, must be integers (whole numbers) in the range 1 to 4095.

## Duration

The third number in our SOUND command determines the duration (length) of the note. This can have a range from -32768 to 32767. How can a sound have a negative length? Actually it can't! Negative numbers and zero are used in conjunction with volume envelopes, which we have yet to discuss.

## Volume

How loud will our sound be? That is determined by the fourth number in our SOUND command. This can have a range of 0 to 7, if we are not using an envelope, but 0 to 15 when we do.

## The Result

We now have enough of a SOUND command to program a simple tune. As our beloved editor suggested to me when I went to see him a while back, (something I don't do often because it's a 600km journey!), this episode is scheduled to appear in December and what is December without Christmas carols? Let's try one.

I have given two listings. The first is a straightforward arrangement of just the melody line. This will play quite satisfactorily on Arnold's rather inadequate speaker, while the second sounds better played through the stereo amplifier of Amstrad's SSA-1 speech synthesiser (or through some other stereo amplifier). This is because the second 'arrangement' has a bass part and you will get better reproduction of the bass notes from the amplifier and its (even slightly) better speakers.

I have added one more variable to the SOUND command, that of 'tempo'. By increasing the value of 'tempo' the music will play more slowly. Decrease it to speed things up. In both listings 'tempo' is at the start of the program. I have also introduced a volume envelope (affecting only the last two notes) to smooth the ending of the second arrangement. Note, too, assigning a value of 1 to 'octave' sets the melody one octave above that containing Middle C. I think it sounds better, but try values of 2 and 0 to see how the Carol sounds when up a further octave or down to the octave in which the music was written.

Next month we shall look at the volume and tone envelopes both of which have a number of uses, not only in music, but also in sound effects.

---

```
10 .....
20 '
30 ' S I L E N T N I G H T '
40 ' for one voice '
50 '
60 .....
70 octave=1:tempo=1.2 'Vary pitch by chan
   ging octave, speed by changing tempo
80 FOR i=1 TO 46
90 READ status,notenum,er,duration,volu
   me
100 frequency=440*(2^(octave+((notenum
   ber-10)/12)))
110 period=ROUND(125000/frequency)
120 SOUND status,period,duration*tempo,vol
   ume
130 NEXT
140 DATA 2,8,60,7, 2,10,20,6, 2,8,40,6, 2,5
   ,120,7, 2,8,60,7, 2,10,20,6, 2,8,40,6,
   2,5,120,7, 2,15,80,7, 2,15,40,6, 2,12,1
   20,7, 2,13,80,7, 2,13,40,6, 2,8,120,7,
   2,10,80,7, 2,10,40,6, 2,13,60,7, 2,12,2
   0,6, 2,10,40,6
150 DATA 2,8,80,7, 2,10,20,6, 2,8,40,6, 2,5
   ,120,7, 2,10,80,7, 2,10,40,6, 2,13,60,7
   , 2,12,20,6, 2,10,40,6, 2,8,60,7, 2,10,
   20,6, 2,8,40,6, 2,5,120,7, 2,15,80,7, 2
   ,15,40,6, 2,18,60,7, 2,15,20,6, 2,12,40
   ,6, 2,13,120,7, 2,17,120,7, 2,13,40,7,
   2,8,40,6, 2,5,40,
```

---

```
10 .....
20 '
30 ' S I L E N T N I G H T '
40 '
50 ' for three voices '
60 '
70 '
80 .....
90 CLS:octave=1:tempo=2.4 'Vary octave to
   vary pitch, tempo to vary speed
100 FOR i=1 TO 100
110 READ status,notenum,er,duration,volum
   e
120 frequency=440*(2^(octave+((notenum
   ber-10)/12)))
130 period=ROUND(125000/frequency)
```



```

140 SOUND status,period,duration*tempo,volum
    e
150 NEXT
160 FOR i=1 TO 2 ' Use COPY key to repeat l
    ines 90-120
170 READ status,notenumber,duration,volum
    e
180 frequency=440*(2^(octave+((notenumber
    -10)/12)))
190 period=ROUND(125000/frequency)
200 SOUND status,period,duration*tempo,volum
    e,1 '(Volume ENvelope no. 1)
210 ENV 1,1,-1,250,4,-1,1 'Avoids abrupt
    ending
220 NEXT
230 DATA 33,8,30,7, 12,5,30,5, 33,10,10,6,
    12,6,10,4, 33,8,20,6, 12,5,20,4, 33,5,6
    0,6, 12,1,60,4, 33,8,30,7, 12,5,30,5, 3
    3,10,10,6, 12,6,10,4, 33,8,20,6, 12,5,2
    0,4, 33,5,60,6, 12,1,60,4, 49,15,40,7,
    28,0,40,5, 42,8,40,5
240 DATA 49,15,20,6, 28,0,20,4, 42,8,20,4,
    49,12,60,6, 28,0,60,4, 42,3,60,4, 49,13
    ,40,7, 28,1,40,5, 42,5,40,5, 49,13,20,6
    , 28,1,20,4, 42,5,20,4, 49,8,60,6, 28,1
    ,60,4, 42,5,60,4, 33,10,40,7, 12,6,40,5
    , 33,10,10,6, 12,6,10,4, 33,13,30,6
250 DATA 12,10,30,4, 33,12,10,6, 12,8,10,4,
    33,10,20,6, 12,6,20,4, 33,8,30,7, 12,5
    ,30,5, 33,10,10,6, 12,6,10,4, 33,8,20,6
    , 12,5,20,4, 33,5,60,6, 12,1,60,4, 33,1
    0,40,7, 12,6,40,4, 33,10,20,6, 12,6,20,
    4, 33,13,30,6, 12,10,30,5, 33,12,10,6,
    12,8,10,4
260 DATA 33,10,20,6, 12,6,20,4, 33,8,30,7,
    12,5,30,5, 33,10,10,6, 12,6,10,4, 33,8,
    20,6, 12,5,20,4, 33,5,60,6, 12,1,60,4,
    49,15,40,7, 28,-4,40,5, 42,0,40,5
270 DATA 49,15,20,6, 28,-4,20,4, 42,0,20,4,
    49,18,30,6, 28,0,40,4, 42,-4,40,4, 1,1
    5,10,6, 49,12,20,6, 28,-4,20,4, 42,3,20
    ,4, 49,13,60,7, 28,1,60,5, 42,5,60,5, 4
    9,17,60,6, 28,8,60,4, 42,1,60,4, 33,13,
    30,7, 12,5,40,5, 1,8,10,6, 33,5,20,6, 1
    2,1,20,4
280 DATA 33,8,30,7, 12,5,30,4, 33,6,10,6, 1
    2,3,10,4, 33,3,20,6, 12,0,20,4, 33,1,0,
    12, 12,-4,0,5
290 '
300 REM Last two notes have a duration of 0
    so that volume ENvelope takes over

```



# Tony Blakemore Goes Somewhere Else

## Sequel to *The Trials of Tony Blakemore*

It seems only yesterday that I sat down at my first computer and took the first of many stumbling steps along the road to learning Basic. At the time, there was very little available in book form for my type of computer. What a different situation with the Amstrad. Nearly forty different titles have become available in a period of less than a year, with more to follow.

With the shortage of specific books I was forced to use material for all kinds of computers. Every book that I read had, with few exceptions, a mailing list of some kind and I set out to write a mailing list of my own. It took over three months to write and used most of the Basic commands that were available to me at the time. Because the computer that I had was very simple, the program uses very simple routines and I feel it is the ideal way to become familiar with the day to day routines that are used in writing most Basic programs.

Converting the program to the Amstrad was not very difficult. The hardest part was resisting the temptation to embellish it with the wealth of new commands that became available on the Amstrad. Apart from using a few graphics commands, the program is in essence the same one that I originally wrote. I hope you will have as much fun with the program as I did. It contains the basic building blocks for many different types of data bases and you should not have any problems in adapting it for any specific applications that you may have.

The program contains six main areas:

1. Load old file from tape or disc
2. Enter a new record
3. View or alter a record
4. View all or part of the file
5. Printing of all records or a mailing list
6. Save new file to tape or disc with an option to sort A-Z

These are the main subroutines. They in turn call many minor subroutines that assist in producing the required results.

One of the problems I had found with many data bases is that once you had entered a menu option, you had to complete it. Amsfile allows a return to the menu from any option that you may select. Where possible, single key entry has been used to enable speedy use of the program.

This month we will set up the introduction screen, initialise and dimension the strings that we will use, and print the menu. Though not strictly necessary, I have initialised all the strings that we will use for storage of the data that will be entered for each record. This will enable a better understanding of the program.



To enable the program to run at its correct speed, remark lines have been kept to an absolute minimum. Any remarks that are in the program are to indicate the function of the subroutine. These are indicated with the shortened version of REM, the shifted 7.

### VARIABLES USED

FILENO - keeps a check of the number of records that are created and is used mainly for saving and loading the file to tape or disc.

### STRINGS USED

NAME\$ - stores the name. Names should be entered surname first as the sort uses this.

ADDR\$ - stores address, street number and name.

SUBR\$ - stores the suburb.

PHNE\$ - stores the telephone number.

STATE\$ - stores the state.

PCDE\$ - stores the postcode.

FILE\$ - combines NAME\$ + ADDR\$ + SUBR\$ + PHNE\$ + STATE\$ + PCDE\$ into one long string and is used to hold the record information on tape or disc.

PACKING\$ - contains sixteen blanks. Amsfile uses fixed fields for storing information. This means that all the separate bits that are combined in FILE\$ all start and finish at the same place. PACKING\$ is added to the information that is entered from the keyboard and then shortened to the required length. It also ensures that if no information is entered for, say, the street, that the required spacing is maintained when stored in FILE\$.

GUIDE\$ - contains sixteen dashes (-). This is printed, shortened to the required length underneath the record entries. This gives a guide to the required length of entry.

MENU1\$ to MENU6\$ - the menu headings are stored for use in printing the menu and also for labeling the various menu options.

SNAME\$ to SPCDE\$ - used to label the entries on the screen display.

### DEFINT A-Z

Locomotive Basic allocates five bytes for variables. If not told otherwise, it allows for the possibility that the variables will contain a large number or have a decimal content, eg. 123.89765. If we DEFINT any variable we declare that variable as an integer and it is allocated only two bytes. This not only saves memory but also makes the program run a lot faster.

### DEFINE WINDOWS

To enable rapid erasure of record information, a series of windows are set up under the area that we wish to erase.

### HOW IT WORKS

10 Sends the program to a subroutine starting at 10000 that initialises all variables, strings etc.

50 Sets screen mode and all inks to background colour. This makes the screen set invisible until all the inks are set new colours.

60 Positions graphic cursor at the required location for printing the background to the title. Tags on the text cursor to enable movement one pixel at a time. The background is offset one pixel to give a shadow effect.

70 Prints the backgrounds. Note that a semi-colon is required after the tagged lines to stop control characters from being printed. MOVE repositions the cursor.

110 Removes the tag and the cursor resumes normal spacing. Sets up the transparent mode. Any text now printed will be transparent in the blank areas of the character cell.

120 Prints all the foregrounds.

170 Allocates new values to the inks. This makes the printing visible and prints it all in one go rather than one character at a time.

180 Removes transparent mode.

190 Short delay, clear screen

200 Prints menu headings. Uses same routines as screen setup.

350 Ink 3 set to flash slightly

360 Sets up keyboard to receive input. If no input, wait.

370 If input is out of range (above 6 or below 1) then wait.

380 If there are no files in memory, print NO FILES and wait.

```

10 GOSUB 10000' INITIALISE
50 MODE 1: INK 1, 1: INK 2, 1: INK 3,
  1
60 BORDER 1: PLOT 261, 365: TAG
70 PEN 2: PRINT "AMSFIL."; CHR$(1
  64);
80 MOVE 229, 316: PRINT "MAILING L
  IST";
90 MOVE 212, 172: PRINT "Tony Blak
  emore";
100 MOVE 228, 140: PRINT "October 1
  985";
110 TAGOFF: PRINT CHR$(22); CHR$(1)
120 LOCATE 17, 3: PRINT "AMSFIL." "CH
  R$(164)
130 LOCATE 15, 6: PRINT "MAILING LI
  ST"
140 LOCATE 14, 15: PRINT "Tony Blak
  emore"

```



```

150 LOCATE 15,17:PRINT "October 1
985"
160 PEN 3:LOCATE 17,10:PRINT"-aut
hor-"
170 INK 1,0:INK 2,26:INK 3,11
180 PRINT CHR$(22);CHR$(0)
190 FOR a=1 TO 4000:NEXT:CLS
200 INK 1,1:INK 2,1:INK 3,1
210 MOVE 254,352:TAG
220 PEN 2:PRINT "-MENU-";:TAGOFF
230 PRINT CHR$(22);CHR$(1);
240 LOCATE 17,4:PRINT"-MENU-"
250 LOCATE 11,7:PRINT MENU1$
260 LOCATE 11,9:PRINT MENU2$
270 LOCATE 11,11:PRINT MENU3$
280 LOCATE 11,13:PRINT MENU4$
290 LOCATE 11,15:PRINT MENU5$
300 LOCATE 11,17:PRINT MENU6$
310 PEN 2:PLOT 190,96:TAG
320 PRINT"- SELECT 1-6 -";
330 TAGOFF:LOCATE 13,20
340 PEN 3:PRINT"- SELECT 1-6 -"
350 INK 1,0:INK 2,26:INK 3,13,26
355 PRINT CHR$(22);CHR$(0);CHR$(7
)
360 I$=INKEY$:IF I$="" THEN 360
370 IF I$<"1" OR I$>"6" THEN 360
380 IF I$> "2" AND FILENO=0 THEN
LOCATE 15,20:PEN 2:PRINT"NO R
ECORDS":PRINT CHR$(7):GOTO 23
0
10000 DIM FILE$(500):DEFINT A-Z
10040 MENU1$="1.LOAD -OLD- FILE."
10050 MENU2$="2.ENTER NEW RECORD.
10060 MENU3$="3.VIEW/ALTER RECORD."
10070 MENU4$="4.VIEW ALL/PART FILE.

10080 MENU5$="5.PRINTING FUNCTIONS.
"
10090 MENU6$="6.SAVE -NEW- FILE."
10130 PACKING$=SPACE$(16)
10170 FILENO=0:NO=0
10210 GUIDE$=STRING$(16,45)
10250 NAME$="":SNAME$="NAME:"
10260 ADDR$="":SADDR$="ADDR:"
10270 SUBR$="":SSUBR$="SUBR:"
10280 PHNE$="":SPHNE$="PHNE:"
10290 STATE$="":SSTATE$="STATE:"
10300 PCDE$="":SPCDE$="P/CDE:"
10340 WINDOW #1,11,26,10,12
10350 WINDOW #2,11,26,10,12
10360 WINDOW #3,31,34,10,12
10370 WINDOW #4,15,40,17,19
10380 RETURN

```

FOR TAPE SUBSCRIBERS ONLY

## A SPECIAL CHRISTMAS MESSAGE

from Andrew Reynes and Nicholas Bolvary

Most people like to draw simple, pretty designs but are put off by the sheer seeming complexity of the programs or the volume of data lines needed. We prefer to use and play with all the graphic utilities on the market. We have three, about which we have heard varied comments ranging from "wow" to "absolute garbage"!

To this end we have constructed a simple Christmas picture using a well known program called Screen Designer and arranged to have it saved to this month's tape.

If you haven't got a disc drive and want to run the program, just type RUN"XMASLDR" and watch the picture slowly appear.

If you have got a disc drive, then to get the picture into your computer and save it to disc, you must type in the following (don't worry about the colours at this point):

```

10 |TAPE
20 MODE 0
30 LOAD "!XMASPIC"
40 |DISC
50 SAVE "!XMASPIC",B,49152,16384

```

OK, now that you have saved it to disc, you might want to re-load the picture and put your own program underneath it such as a nice Christmas jingle, or a Carol (see this month's Learning Centre). This is what you have to do:

```

10 MODE 0
20 BORDER 3
30 INK 1,13:INK 2,26:INK 3,12
40 INK 4,24:INK 5,9:INK 6,18
50 INK 7,10:INK 8,6,20:INK 9,1
60 INK 10,2:INK 11,11:INK 12,4
70 INK 13,8,0:INK 14,3:INK 15,6
80 LOAD "!XMASPIC"
65535 GOTO 65535

```

The last line in this routine prevents the screen from being overwritten with the "READY" prompt and allows you to insert your program from line 90 to 65535.



# Review of the new PCW-8256

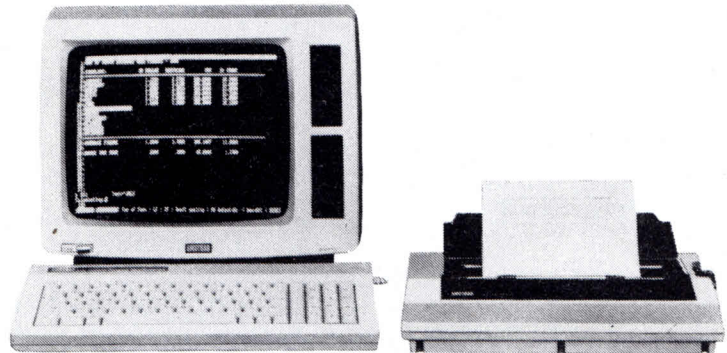
by Simon Anthony

With the release of the PCW 8256 comes Amstrad's first serious attempt to move into the business market, or at least the office level. This new machine is effectively two computers in one, operating as a word processor or as a personal computer. On the surface there is nothing unusual about this except, in this case, the 8256 has been designed in terms of word processing from scratch and developed around Locomotive Software's Locoscript. The concept of building a machine around software, rather than making the software fit the machine, is an approach which should make the overall operation easier and more effective.

Like a child with a new toy, I eagerly opened the box containing the yet to be released PCW 8256. It was clear that the deadline for the review to hit the November issue was too close, and that I would be able to spend more time putting it through its paces. Inside the box I found the green screen monitor with fitted disc drive, a keyboard, a printer, two 3" floppy discs, two manuals, various pieces of cabling, another small box with the printer ribbon inside, a packet with the tractor feed mechanism for the printer and finally two strange looking flat pieces of plastic - I wonder what they are for?

It took me a few minutes to rush around and find a power plug and attach it to the mains lead. (Don't take any notice of the manual which talks about an English fused plug). I also discovered where those two pieces of plastic belong - they are paper tray extensions and fit, naturally enough, on the top of the paper tray.

By now, you should know what the PCW8256 looks like, (if you don't, you haven't seen the front cover of this



months' magazine), so it is not necessary to provide a detailed description. Suffice to say that the whole unit is well designed in complementary grey colours. At first I thought that Amstrad had really 'gone ergonomic' with a swiveling/tilting screen - but alas it only looks like that - there are no adjustments.

The keyboard requires a relatively soft touch on the dished keys, and is presented in a standard QWERTY layout with the special word processing keys (doubling up as a numeric keypad) and cursor keys to the right. Two special keys of interest are ALT and EXTRA. Used independently, or with Shift (ALT only) they produce a range of special characters.

The green screen provides 90 columns by 32 lines of text as opposed to the standard 80 by 24, and houses the disc drive for the 3" floppy.

The printer looks, at least to me, like a "sawn-off" Sieksha SP-1000. Because it is driven by the software it is not necessary to fill it with boards. Certainly the paper loading mechanism is the same. It has two modes of operation - Letter quality printing at 20 cps or Draft quality printing at 90 cps, both in Elite pitch.

On loading Locoscript, the screen presents three sections of information. The first section (the top two lines) provide details of the type of operation that can be carried out along with their associated keys. For example, E= Edit existing document or C= Create new document. The second section provides information on the disc(s) and files that are resident in the system. The third gives details of the documents that are available. This screen is called the Disc Manager and is the starting point for all word processing operations.

I found using the 8256 as a word processor running with Locoscript to be fairly simple. Of course, with all new pieces of software and machinery it takes a while to get the hang of things and this configuration is no different except that users should be able to start producing meaningful work in a shorter space of time than is usual with others.

The ease of using Locoscript is best illustrated by an example - in this case the production of a letter. The steps are quite straightforward:

1. On the Disc Management screen, press C to create a new document. This action generates the display of a Menu box superimposed on the current screen, which provides the opportunity to



either accept a suggested name for the new document or enter your own. The 'pull down' menu is a main feature of this word processing system and is what makes it succeed in being user friendly.

2. Press Enter a couple of times (to default to a Template) and a new screen appears with the prepared format, ready for entering.

3. Type the letter and, when finished, hit the EXIT key. This will generate another menu of options which allows you to save the document, print it or finish editing.

4. Select SAVE and PRINT option, load the paper, hit the Enter key and .....Voila - the letter.

Of course, the above example hasn't included any frills, but it certainly served as an indication of the ease of use.

There is a danger of this review becoming a tutorial in word processing, so I'll mention just a few of the keys that can be used for editing a document.

CUT - pressing this key followed by a second CUT deletes the text between the places the two CUTs were made.

COPY - followed by a second COPY and a digit or a letter copies the text between the two into another block identified by the digit or letter.

PASTE - followed by a digit or a letter pastes the contents of a block or phrase into the document at that point.

FIND - searches for a piece of nominated text.

EXCH - searches for a piece of nominated text to be replaced by another.

Locoscript uses current information as the starting point for any change. This means that it makes a note of, say the last layout that was used, and offers it again at the start of something new - undoubtedly a help when making minor modifications. It also holds the current document until told to change. Again, this saves the bother of repeating the document details if you have been working on some other function (not another document) in the meantime.

The Disc Manager screen operates in a

similar fashion in that whatever is displayed can be taken as the current 'hold' situation, and by merely re-directing the cursor to the name of a document is sufficient to instruct Locoscript to change files.

As mentioned in the simple example above, menus form an important part of Locoscript because they are the medium through which most of the codes or commands are inserted. I uncovered eight.

SHOW - which allows options such as to show any rulers that have been set, show blanks as dots, show spaces (not blanks) as "bullets" and so on.

LAYOUT - allows modification to the existing layout or set up a new layout. Selection from this menu generates further sub menus to step you through.

EMPHASIS - provides the facility to underline, embolden characters, double-strike when printing etc.

STYLE - this gets you into italics, superscript, subscript and pitch settings.

LINES - apart from allowing seven different line spacings (0 to 3 in half increments), it also provides for centring or justification.

PAGES - this is the menu which helps to get over the problem of page breaks in the wrong place. It also allows for the insertion of a page number.

MODES - this is especially useful if the disc has become full when you are in the middle of a document. It allows you to enter the disc housekeeping utility whilst still in your document or pick out a document you want to insert in the current text. It also has an Edit Header facility which provides the means of entering certain parameters with the aid of function keys, for example, to prevent "widows and orphans", set pagination or set zero characters with or without a slash.

BLOCKS - enables the listing of all blocks or phrases that have text stored in them, save a block to disc or save the current set of phrases to a new PHRASE file (this is where you keep your commonly used pieces of text - up to a maximum of 26).

I make no apology for saying that whilst this article is supposed to be a review of the PCW8256, it has so far become more a review of Locoscript. But that is natural for, in terms of this word processing, one can't work without the other.

I was impressed by the speed of the system, undoubtedly due to the "RAM-disc" which stores the document currently being worked on before dumping to the floppy. Each floppy can hold nearly 180000 characters of information - this adds up to a storage capacity of between 120 and 180 A4 pages.

Now let's take a look at the 8256 as a personal computer.

On the two discs supplied with the machine are CP/M Plus operating system with GSX graphics enhancements, Locomotive Software's Mallard extended Basic interpreter and the educational and training language DR LOGO which is compatible with the 6128. With CP/M+ this should open access to a wide range of standard software such as Supercalc, Multiplan etc. using the VT52 terminal emulation. The GSX graphics system (a DR product) is supplied to provide a standard software interface for graphical programs, although I am not sure how much software is available to make use of this system.

The microprocessor is a Z80A with 256k bytes of RAM. 112k of this is allocated as the "RAM-disc" to assist in overlaying when using CP/M. There are also separate microprocessors which control the printer and keyboard.

Options include an RS232 serial and parallel interface and a second disc drive (FD2) which has a formatted capacity of 720k. Apparently the drive may have to be fitted by a qualified engineer. I haven't seen any of these options so cannot offer any comments.

So there it is. A neat, carefully designed and presented computer that Amstrad reckon exceeds the speed and functionality of a basic IBM PC. I am inclined to agree with them, and with a price of around \$1500 they must win hands down.



# TAU INDEX RATING

Based on The Amstrad User Index (TAU Index) which was introduced a few months ago to give an overall rating of software being reviewed, the list below is a summary of the games to date. Whether or not you agree with our Review panel, it should give you some indication of the quality of the software.

TITLE	TAU INDEX
Splat	84%
Moon Buggy	80%
Dungeon Adventure	80%
Mordon's Quest	80%
A View to Kill	79%
Blogger	78%
Grand Prix Rally II	78%
Detective	78%
Manic Miner	75%
The Scout Steps Out	75%
Hunchback	73%
The Survivor	71%
World Pool (with Speech)	71%
Quack-a-Jack	71%
Jigsaw Magic	71%
Stockmarket	70%
Dark Star	70%
Graphic Magic	66%
Mr Wong's Loopy Laundry	66%
Mutant Monty	62%
Astro Attack	62%
Fantastic Voyage	61%
Pyjamarama	61%
Space Hawks	61%
Alien	59%
Centre Court	55%
Galactic Plague	43%

Judging by the reviews so far received, it would seem that anything over 65% is worth a look, but don't take our word for it - everyone has got their own likes and dislikes!

If you have a piece of software that hasn't already been reviewed and would like to share your opinions with other readers, send your summary to the Editor. Try to keep it to 200 words and don't forget the ratings at the end. Reviews of new software is especially welcome.

DO YOUR  
PROGRAMS SUFFER  
FROM FATIGUE?

ARE THEY  
LISTLESS AND  
TIRED?

Now you can PEP them  
up with ISSCOM!

*ISSCOM is an integer  
only Basic Compiler  
written to take full  
advantage of the  
Amstrad's sound and  
graphics with  
enhancements.*

For further details contact:

**STRATEGY SOFTWARE**

2/33 The Centreway

Mt. Waverley, Vic. 3149

Telephone: (03) 232 7055

Watch out for details  
of our Binder offer in  
the January 1986  
issue of  
**THE AMSTRAD USER**



## **DISCOUNTED BOOKS FOR SUBSCRIBERS ONLY**

Title	Subscriber Price	Normal Price
Advanced User Guide	\$ 19.75	\$ 21.95
Adventure Games for the Amstrad	\$ 25.15	\$ 27.95
Amstrad Compendium	\$ 20.65	\$ 22.95
Amstrad Computing	\$ 21.55	\$ 23.95
Amstrad Games Book (Pitman/Ramshaw)	\$ 14.35	\$ 15.95
Amstrad Games Book (Melbourne House)	<b>NEW</b> \$ 20.65	\$ 22.95
Amstrad Pentacle Adventure Creator	<b>NEW</b> \$ 8.05	\$ 8.95
Amstrad Users Omnibus - 464/664/6128	<b>NEW</b> \$ 17.95	\$ 19.95
Basic BASIC	\$ 11.45	\$ 12.75
Basic Programming on the Amstrad	\$ 22.45	\$ 24.95
Brainteasers for the Amstrad	\$ 19.75	\$ 21.95
Childs' Guide to the Amstrad Micro	\$ 9.85	\$ 10.95
Disc System, The Amstrad CPC 464	<b>NEW</b> \$ 25.15	\$ 27.95
Dynamic Games for the Amstrad	\$ 17.95	\$ 19.95
Exploring Adventures on the Amstrad	\$ 21.55	\$ 23.95
Filing Systems And Databases for the Amstrad	\$ 26.95	\$ 29.95
Games and Graphics Programming - 464/664/6128	<b>NEW</b> \$ 26.95	\$ 29.95
Graphics Programming Techniques	<b>NEW</b> \$ 22.45	\$ 24.95
Ins and Outs of the Amstrad	\$ 20.65	\$ 22.95
Machine Code for Beginners on the Amstrad	\$ 17.95	\$ 19.95
Machine Language for the Absolute Beginner	<b>NEW</b> \$ 20.65	\$ 22.95
Making Music on the 464/664	<b>NEW</b> \$ 19.75	\$ 21.95
Master Machine Code on your 464/664	<b>NEW</b> \$ 19.75	\$ 21.95
On the road to Artificial Intelligence	\$ 17.95	\$ 19.95
Pitman's First Book of Amstrad Games	\$ 11.65	\$ 12.95
Practical Programs for the 464	\$ 21.55	\$ 23.95
Programming the Amstrad CPC 464	<b>NEW</b> \$ 17.95	\$ 19.95
Ready made Machine Language routines - 464/664	<b>NEW</b> \$ 20.65	\$ 22.95
Structured Programming on the 464/664/6128	<b>NEW</b> \$ 26.95	\$ 29.95
Whole Memory Guide	<b>NEW</b> \$ 26.95	\$ 29.95
Working Amstrad	\$ 17.95	\$ 19.95
Writing Adventure Games on the 464/664	<b>NEW</b> \$ 20.65	\$ 22.95
Your first Amstrad Program	\$ 20.65	\$ 22.95
40 Educational Games for the Amstrad	\$ 19.75	\$ 21.95
60 Programs for your Amstrad	\$ 19.75	\$ 21.95

### **How to Order**

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**

*Shop 2, 33 The Centreway, Blackburn Road, Mount Waverley, Victoria, 3149*

*Bankcard or Mastercard orders accepted by phone on (03) 232 7055*



# The 40cent Programs

by Ivor Joystick

Once upon a time, a sensible man bought an Amstrad computer.

Thirsting for information, he took out a subscription to The Amstrad User, the largest selling magazine for the Amstrad range in Australia.

He didn't need his machine to work out that, for less than \$3\* per month for 32 pages (and quite often more) of solid Amstrad information, he was getting programs for less than 40 cents each!

This made him live very happily ever after.

*... but that's not the end of the story*

*Every month The Amstrad User is packed with a range of articles, programs, hardware and software reviews, utilities, User Group Information, Letters, Tutorials, Hints and Tips - all devoted entirely to the Amstrad computers.*

*And to save you the trouble of keying-in and correcting your typing errors, a cassette containing the programs appearing in the magazine each month is also available.*

**Make your Amstrad a success story by subscribing to The Amstrad User now.**

\* Based on yearly subscription of \$35.

Please send me THE AMSTRAD USER for 12 months

Magazine only: \$35  Magazine and cassette: \$75  (PNG and NZ add \$12 airmail)

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, Shop 2, 33-45 The Centreway,  
Blackburn Road, Mt. Waverley, Vic 3149 Tel 03-232 7055

(OVERSEAS PRICES ON APPLICATION TO ABOVE ADDRESS)



# THE LATEST VIDEO GAMES FROM AMSTRAD

## DARE YOU TAKE THEM ON?

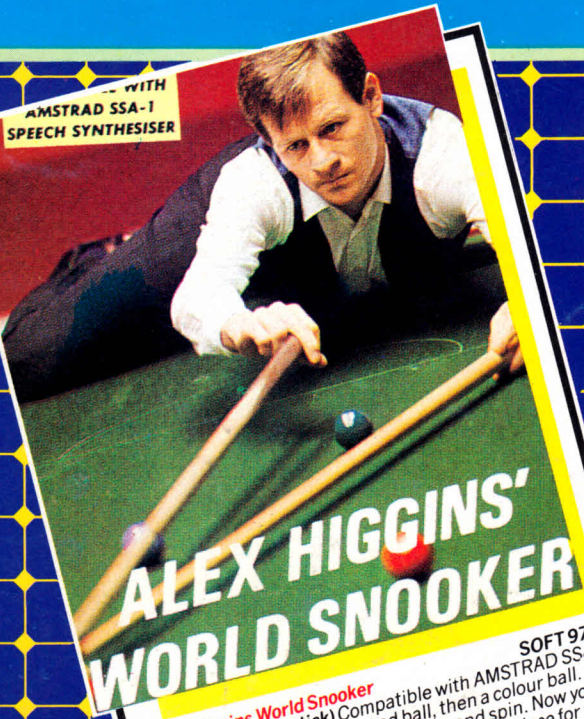
So you're pretty hot at video games are you? We dare you to take on these. They're the latest video game challenges from Amstrad and they're out to win.

The substantial RAM in the Amstrad CP464 has provided our software authors with greater scope to bring you more challenging, more cunning, more exciting fun than ever. If you dare!

Ask your nearest Amstrad dealer for full details on these and all Amstrad software for home or business.

**When ordering please quote catalogue number.**

WITH  
AMSTRAD SSA-1  
SPEECH SYNTHESISER

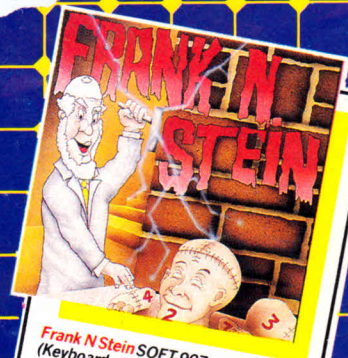


### ALEX HIGGINS' WORLD SNOOKER

**Alex Higgins World Snooker** SOFT 978  
(Keyboard or Joystick) Compatible with AMSTRAD SSA1 Speech Synthesiser. Pot a red ball, then a colour ball. Got your angles right? Control force and spin. Now you can play snooker with this software masterpiece for the CPC464/664. Challenge your friends, or go for a world record break of 147. Happy potting.



**Fu-Kung in Las Vegas** SOFT 979  
(Keyboard or Joystick) Fu Kung is a US Marshall working undercover in Las Vegas. His task — to uncover crooked casinos without get through each casino without being electrocuted, crushed or otherwise liquidated? Test your skill against some very odd odds.



**Frank N Stein** SOFT 907  
(Keyboard or Joystick) Guide Professor Stein around the dungeon picking up the bones. Be sure you get the right order. Then throw the activate switch. The quicker you he'll be on subsequent levels!

### Dragons



**Dragons** SOFT 977  
(Keyboard or Joystick) Kill the dragons and gather the gems before the dragons get you. You'll be climbing creepers, walking ledges and jumping for your life. Uh oh, the dragons change and get fiercer, you'd better watch out!



**Subterranean Stryker** SOFT 06010  
(Keyboard or Joystick) Your mission: rescue the humans captured in the alien cave system. Your Stryker is an amazing craft, but so too are the hazards! Your fuel is limited so you'll have to steal more. Watch out for the alien command saucer. Good luck, you'll need it.



**Super Pipeline** SOFT 987  
(Keyboard or Joystick) Protect your pipeline and workmen against attacks by manic power drills, gnawing caterpillars, ferocious hammers, the six-legged Venusian Pipe Spider and other weird dangers. Your water supply is limited so watch those leaks!



**The Scout Steps Out** SOFT 988  
(Keyboard or Joystick) Your mission: be good a moral example to your scout troop. Go boldly, dive for examples of sea life. Clean windows for pensioner flats; gather mushrooms for camp tea (not poisonous ones); but take care — the enemy is everywhere.



**Sorcery+** SOFT 1983 (Disc only)  
(Keyboard and Joystick) The evil forces of the necromancer are creeping in. Can you free your ancient land? A breath-taking disc-based enhanced version of best selling game "Sorcery". 75 atmospheric locations with great graphics and sound effects!

# AMSTRAD

Distributed and guaranteed throughout Australia by

# AWA-THORN