

THE AMSTRAD USER

Issue No. 7 \$3.00

August 1985



- RECIPE DATABASE PROGRAM FOR DISK DRIVES
- USER GROUP INFORMATION AND SAMPLE RULES
- NEW JUNIOR JOTTERS SECTION
- PLAY THE POKIES

FOR THE NOVICE & EXPERIENCED USER

THE AMSTRAD USER

Issue No. 7

August 1985

CONTENTS

Editorial	2
Letters	3
Amstrad Cordon Bleu - a Recipe database on disc	5
User Group Information	11
The Learning Centre - Machine Code: Part Two	14
Junior Jotters - for younger Amstrad users	15
Auld Lang Syne - a Poker Machine game	16
Bytes and Pieces - more Machine Code routines	25
Discounted Books for Subscribers	28
Sound and Fury - sound and graphics demonstration	29
An insight into Basic, Pascal and Fortran	30
Competition - the last chance	32

For Tape Subscribers, the programs/routines can be found at these approximate counter readings:
Side 1 - Auld Lang Syne: 0, More MC Routines: 100, Kaboom (Sound and Fury): 115
Side 2 - Recipe: 0

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 \$3.00 is the recommended retail price only. The subscription rate (for Australia only) is \$30.00 for 12 issues of the magazine only, or \$70.00 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

It's been a bit of a rush to compile this month's issue of The Amstrad User, primarily due to the PC85 show in Melbourne being staged at a time when the magazine had to be ready for the printer. Except for this Editorial which I shall have to keep brief, we just made it before the show commenced.

The PC85 show is one of a number of regular exhibitions which are held each year in various states. You will remember that we attended the one in Sydney a few months ago. But unlike that show, we had our own stand in the Computer Club Corner. Representatives of User Groups around Melbourne were on hand to answer questions. During the few quiet moments we 'evaluated' some of the games, and succeeded in wrecking a joystick with Decathlon. (A bill will be sent shortly to Mr. Russo!)

Of course AWA-Thorn had a stand, and much interest was shown in the new CPC664. However, we found that many people who had visited that stand then came to ours for 'inside information' from the end-user.

There is no question that membership of the local user groups should expand as a result of this show, and I would urge other groups to keep in contact with their nearest AWA state office and thereby be informed of any exhibitions or shows at which User Groups could be represented.

Finally, the last reminder for competition entries. The closing date is 15th August and the winners will be announced in the October issue.

See you next month.

Ed.

Letters

Some Months ago I purchased an Amstrad computer for home use and I was so delighted that I bought a second one for work. I wish to make four comments which will be of interest to other users.

First, so far I have found only one bug in the BASIC which sometimes occurs when the commands to OPENIN or ENOUT files are used. For example, ENOUT FILE1.DAT may create a file with the name XYFILE1.d where the first two letters from some other program in memory appear at the beginning of the file name and two characters from the desired name are tacked off the end. After OPENIN, a similarly scrambled file name is searched for in vain. Both my machines behave in this way. However, it is possible to rectify the problem by using a garbage collection before each of these statements is issued by preceding them with the statement:

```
DUMMY=FRE("")
```

Second, I would be interested if other users have had similar problems to me with three commercial software packages. The first is Master Chess which had a bug on level 9 causing it to make its own queen and prevented me from making valid moves. Eventually the program became hung up. After informing it to my dealer, I obtained a replacement copy which again takes its time on level 9 but this time replaces with a third rook for me. This program also becomes hung up, and the computer cannot be reset but must be switched off and the program reloaded. Level 0 the program seems to work satisfactorily but on the remaining levels 1-8, does not evaluate repetitive moves as indicated by the documentation. From my experience

with the two copies of this program, I suspect that it has some serious software bugs.

Third, the next program I wish to comment on is Codename Mat which has grossly incomplete documentation. This is a superb program which works well once one has discovered the purpose of the game, the details of general strategy to be followed, and what the various keys on the keyboard control. An extra half hour in the preparation of the document would have avoided all this. Below is a tabulation of what I have discovered to be the functions of the various keys (those omitted from the original documentation are in italics)

- 1- *Slows the spacecraft down*
- 2- *Speeds the spacecraft up*
- 3- *Stops the spacecraft*
- 4- Sets the speed to 30
- 5- *Sets the speed to 100*
- 6-9 *Cursor controls for keyboard operation.*
- 0- *fire button for keyboard operation.*
- W- Warp
- R- Rear view
- T- Transmit
- A- Switch tracker between attackers.
- S- Sector scan
- D- Shield toggle
- F- Front view
- H- *Halts the game for later restarting.*
- K- Tracker toggle
- L- long range scanner
- C- Solar system scan

Perhaps there are other options of which I am still unaware.

Finally, I have purchased a cassette

version of Tasword (as opposed to Amsword which is supposed to be identical) It is an excellent word processor but will not print page numbers exactly in the centre of the page. They are printed about one third of the way across the top of the page. Any suggestions?

Bernard M. Chapman
Valley Heights, N.S.W.

I have just received my back issues of The Amstrad User and have read them all from cover to cover. I am very glad to see such a very informative and practical publication, and I wish to congratulate you on a beautiful magazine.

Combined with the Home computer course and the Advanced Home computer course by Orbis, I am grasping most of the essentials of computing. I heartily agree with you that User Groups are the answer to a better understanding of the CPC464 and to solve any problems. If anyone else in the Maryborough area wishes to form a User Group, I would be only too happy to assist in any way. (*Alan's contact number appears in the User Group Contact List*).

I agree with JE of Brisbane's letter (Issue 4) about the programs in the book 'Amstrad Explored'. After many corrections to the music program on Page 67, and finding that the symbol in line 3010 " ^ " is actually " ↑ ", my neighbours and I were astounded by the results. I'm still trying to convince them that I haven't bought an electronic organ, but merely wired up a 4mm stereo plug. This feeds the output of the CPC464 into the aux. jack of my 65W stereo.

Another problem an experienced user could help me with is a music program on Page 35 of the March edition of the English magazine. Line 440 reads:

```
period=base/(2XXXXXoctave) and  
prevents the program working. The  
only way I can get even a sick tune is  
to substitute 10* or 20* for 2XXXX  
and the pitch gets higher the larger the  
number used.
```

Another problem in the same edition concerns Electric Eddie. At midnight in frustration I rang England and after much apologising I was told to add the following line to the end of the program: 1890 DATA thyme,reason. Without this line you get Data exhausted in 230. I have since read why in your May letters. There are only 238 items of DATA instead of 240 to match its READ line.

Alan Laird, Maryborough, QLD.

Line 440 contains a printing error. The XXXX should be replaced with an upwards arrow (the character on the £ key).

705 PRINT #3,AS;
825 PRINT #3,CHR\$(X);

Robert Wright, Bradbury, N.S.W.

The Review Panel

My request to readers to assist in formation of a panel of review provided a deluge of prospective participants. Most replies contain good reasons why they should be selected, and many went as far as including a review as a sample. I still didn't make the task of selecting the final panel any easier!

However, after a great deal of deliberation, I have chosen the following people as an initial panel and you will see the results of their labour in next month's issue. My thanks to those who responded but didn't get selected this time; you will be kept in reserve.

James Gallagher
Sydney Jackson
Stephen Kerr
John Parker
Darren Robinson
Justin Scotchbrook
Andre Urankar
Graham Watt

Kenthurst, NSW
St. Kilda, Vic
Carindale, Qld
Tullamarine, Vic
Reservoir, Vic
Maddington, WA
Eltham, Vic
Padstow, NSW

NEW PRINTER FROM AWA

Whilst the DMP-1 has been of use to many Amstrad owners, the quality for the more serious user has often been criticised. Clearly, the comments have not fallen upon deaf ears with the impending announcement of a jointly marketed AWA SEIKOSHA dot matrix printer. This device will offer near letter quality, auto and tractor feeding, 100 characters per second printing speed and a typeface which (unlike the DMP-1) incorporates correct positioning of the lower case decenders.

The AWA SEIKOSHA printer is due to be available in the middle of August at a cost of less than \$500. In the next issue of The Amstrad User, we will provide a full review following our extensive testing.

• 195 WINDOW #3,4,17,5,6
• 196 PAPER #3,6;PEN #3,8

MICRO-MART The Market Place for The Amstrad User Readers

THREE INCH DISCS - now available on mail order in packs of five discs for \$42.50 plus \$2.50 airmail postage, from The Amstrad User. Bankcard or Mastercard accepted including phone orders on 03-232 7055.

MICRO-MART is available for all readers to advertise for a nominal cost of \$10 per insertion. For further details ring Strategy Publications on 03-232 7055 or send your ad with cheque to Shop 2, 33 The Centreway, Mt. Waverley, Vic 3149.

Amstrad Cordon Bleu

A recipe storing program from Andre Urankar

ture the following scene: A most successful dinner by - your boss' wife remarks "you simply *must* let me have a copy of that recipe". Whereupon your wife responds "I'll give you one in equal aplomb "Ami will be able to print out a copy in 10 minutes" (Ami being the CPC464, not the maid!). Alternatively, picture a kitchen drawer crammed with recipe books and the frustrated cries of "the recipe I want is *nowhere* in here".

was the second scene that prompted me to use the power of Ami to lend a hand and act as a RECIPE DATA BANK. This program provides the following features:

Add a recipe

Search for a recipe - by either name, or part of a name, or ingredient

Print-out/Display - a specific recipe in detail, an index for a specific group of recipes or a total catalogue of all recipes stored

Corrections to a recipe - in case errors have been made or later changes are required, or to remove a recipe (when you discover that Aunt Myrtle's recipe for yoghurt pie should have remained with Aunt Myrtle) or to be effective, data banks require rapid access to mass storage devices. This program uses the micro-disc and is not recommended for cassette operation.

The concept of 'top-down' programming has been applied, which means that those people with an adventurous bent should be able to easily add expanded procedures. The program modules are as follows:

Lines 1 to 10 - Preface

Lines 100 to 999 - Subroutines area

)) Handling of valid/invalid responses to "input" questions.

)) Message display while files are getting updated.

)) Display routine for the various "index groups".

)) Definitions of windows for MODE 1.

)) Definitions of windows for MODE 2.

)) Reading of the first four blocks of data from an individual file.

)) Main part of the correction routine.

)) Horizontal dividing line.

)) File handling routine. Using a "TEMP" file which is Data renamed into the original file.

Lines 1000 to 8999 - Main Program area

)) Main Command Choice.

)) Command 1: "Add".

2100 Command 2: "Search".

3100 Command 3: "Display".

4100 Command 4: "Correct".

Lines 9000 to 9999 - Definitions of variables
Lines 10000 and on - Initialisation of all

indexes. This is rather a dangerous section in that it will clear all indexes. I would recommend that after initialisation, you 'REM-out' all except line 10036

STARTING UP

(1) Transfer the program from cassette to a newly formatted (both sides) disk. I would suggest that a whole disk be made available since this allows up to 400 recipes to be saved.
(2) RUN the program. A title page will be screened.
(3) Press the <break> key twice.

(4) Type in GOTO 10000 (and <ENTER>). After initialization the program returns to the title page.

(5) Repeat steps (3) and (4) for the second side of the disk.

(6) Optional (but strongly recommended): <break> the program and "REM-out" lines 10000 to 10035.

(7) RUN the program normally and respond as required to the prompts.

SOME PROGRAM FEATURES

(1) The "INKEY\$" subroutine (line 100) will RETURN only if a character within the "valid\$" is encountered. All other responses will result in a beep-tone, an error message, and the input question is repeated. The string defined as CONTROL\$ does all the cursor control for this procedure. Note that the character "+" (plus sign) is used as an escape key to restart the program at any time that a single character entry is required.

(2) Separate windows are defined for MODE 1 and MODE 2 operation, since the shifting from one mode to the other clears window specifications.

(3) In the Correction procedures, a recipe may be totally removed if its name is set as "REMOVE" (in upper case as a precaution). Each part of the recipe can be individually corrected; and additional data can be entered.

(4) The "LINE INPUT" instruction has been used to allow comma's as separators in the text. However, " (double apostrophies) must not be used.

(5) The Search routine uses the powerful instruction INSTR to locate a match to a name or part-of a name.

(6) HELPFUL HINT: since this program is written in BASIC, the maximum string length is 255. I would therefore suggest that longer paragraphs in the "preparation"

be broken down to smaller sections for typing in. You will find that this also helps when corrections need to be made.

This program has been in use in our household for some time now, and as with all "living" software changes/improvements are inevitable. (Additions can be evidenced by the strange line numbering within the main - and sub program areas).

The overflowing drawer is being steadily reduced, and looking forward to the "dinner party" scene to cap off effectiveness of the program. I hope that you find it interesting and useful as we have.

(So do I, but for those people who really want to make good impression, replace "Horses Doovers" in line 22 with "hors d'oeuvres" - Ed).

```
10 ' Program Name:  RECIPE
20 ' Developed By:  A.H.URANKAR
30 ' Date:         April 1985
40 ' Version:     2.1
50 '
60 GOTO 2040
70 '
80 '*** Subroutines Area
90 CLS #4:PRINT #4,questions:
100 as=INKEY$:IF as="" THEN 100 ELSE answers$=LOWER$(a$):PRINT #
    4,answers
110 operation=INSTR(valld$,answers)
120 IF as="+-" THEN CLOSEIN:CLOSEOUT:RUN
130 IF operation=0 THEN PRINT #5,"Incorrect Input - please try
    again":PRINT CHR$(7):GOTO 90 ELSE RETURN
140 '
150 CLS:LOCATE 20,10:PRINT "PLEASE WAIT WHILE THE FILES ARE BEI
    NG UPDATED":RETURN
160 '
170 FOR x=1 TO 11:PRINT#3,TAB(offset) types(x):NEXT:RETURN
180 '
190 WINDOW#1,1,40,1,3: WINDOW#2,21,40,1,6: WINDOW#3,1,40,3,24
200 WINDOW#4,1,40,24,24:WINDOW#5,1,40,25,25:WINDOW#6,1,40,10,25
210 RETURN
220 '
230 WINDOW#1,1,80,1,22:WINDOW#4,1,80,24,24:WINDOW#5,1,80,25,25
240 RETURN
250 '
260 FOR x=1 TO 4:IF EOF THEN CLOSEIN:RETURN ELSE INPUT#9,x$(x):
    NEXT:RETURN
270 '
280 CLS#4:CLS#5:questions="Correction/addition to (n)ane, <q>ua
    ntity, <l>ngredient, <p>reparation method?":valld$="nqip":G
    OSUB 90
290 CLS:PRINT"EDITING PROCEDURES":PRINT
300 ON operation GOSUB 310,340,350,410:RETURN
310 PRINT"Present name is:...":names$:PRINT
320 PRINT"To remove a recipe completely, type REMOVE as the nam
    e":PRINT
330 INPUT"New name to be:...":names$:RETURN
340 PRINT"Present quantity is for:...":qtys:INPUT"New quantity w
    ill be for:...":igtys:RETURN
350 FOR x=1 TO ingred:PRINT STR$(x)+" "+ingred$(x):NEXT:PRINT
360 INPUT#4,"Line number & <ENTER> = change,(a) & <ENTER> = add
    ,<e> = exit without change ",lin.lng$:IF VAL(lin.lng$)>Ingr
    ed THEN PRINT#4,control$:GOTO 360
370 IF lin.lng$="e" THEN RETURN
380 IF lin.lng$="a" THEN Ingre$=ingred+1:LINE INPUT"Additional
    ingredient is:...":ingred$(Ingre$):RETURN
390 op=VAL(lin.lng$):IF op=0 THEN PRINT#4,control$:GOTO 360
400 PRINT"Present line says:...":ingred$(op):LINE INPUT"To be re
    placed by:...":ingred$(op):RETURN
410 FOR x=1 TO prep:PRINT x TAB(5) prep$(x):NEXT:PRINT
420 INPUT#4,"Line number & <ENTER> = change,(a) & <ENTER> = add
    ,<e> = exit without change ",lin.lnss$:IF VAL(lin.lnss$)>prep
    THEN PRINT#4,control$:GOTO 420
430 IF lin.lnss$="e" THEN RETURN
440 IF lin.lnss$="a" THEN prep=prep+1:LINE INPUT"Additional inst
    ruction is:...":preps$(prep):RETURN
450 op=VAL(lin.lnss$):IF op=0 THEN PRINT#4,control$:GOTO 420
460 PRINT"Present line says:...":preps$(op):LINE INPUT"To be repl
    aced by:...":preps$(op):RETURN
```



```

1020 MODE 1:GOSUB 190:GOTO 620
1030
1040 '*** Sub-Program: Search for a recipe
1050 PEN yellow:PRINT-SEARCH FOR A RECIPE":PEN green
1060 PRINT#3," Through which group do you want to search?":
ffset=10:GOSUB 170
1070 questions="Please enter the required group":valids="esmgdx
bht":GOSUB 90:types=as:type=operation:CLS#3:CLS#5:filename
s=index,"+types
1080 PRINT#3,"The following search modes are available":PRINT#3
1090 PRINT#3,TAB(5)*1 = by name (or part of a name)"
1100 PRINT#3,TAB(5)*2 = by an ingredient":PRINT#3
1110 questions="Which search mode ":valids="12":GOSUB 90:search
=operation:PRINT#3,"Search mode "search:PRINT#3:CLS#4:CLS#5
1120 ON search GOTO 1130,1150
1130 INPUT#3,"What recipe name (or part name) are you looking fo
r":in$=names=LOWER$(n$)
1140 CLS:PEN yellow:PRINT-Searching for:.. CHR$(34) names CHR$(
34):PEN green:PRINT" Please stand by.":LOCATE#3,1,2:PRINT
#3,"Recipe Name":line.at=330:GOSUB 480:PRINT#3:GOTO 1170
1150 PRINT#3,"One ingredient may be specified. The result
ing list will name the recipes that use this ingredient.":PR
INT#3:INPUT#3,"ingredient:..":ingred$:ing$=LOWER$(ingred$)
1160 CLS:PEN yellow:PRINT-Searching for a recipe containing:..
:PRINT TAB(12) CHR$(34) ingred$ CHR$(34):PEN green:PRINT-p1
ease stand by.....":LOCATE#3,1,4:PRINT#3,"Recipe Name":line.
at=300:GOSUB 480:PRINT#3
1170 filenames=filenames
1180 OPEN IN filenames:z=1
1190 FOR x=1 TO 50:recipex(x)="":NEXT
1200 ON search GOTO 1210,1260
1210 WHILE NOT EOF:found=0:GOSUB 260
1220 found=INSTR(LOWER$(xs(1)),names):IF found>0 THEN PRINT#3,x$
(1):recipex(z)=xs(1):z=z+1
1230 FOR fast.forward=1 TO VAL(xs(3))+VAL(xs(4)):LINE INPUT#9,x$
:NEXT
1240 WEND:CLOSE IN:GOTO 1320
1250
1260 WHILE NOT EOF:found=0:GOSUB 260
1270 FOR x=1 TO VAL(xs(3)):LINE INPUT#9,x$
1280 found=INSTR(LOWER$(xs),ing$):IF found>0 THEN PRINT#3,x$(1):
recipex(z)=xs(1):z=z+1
1290 NEXT
1300 FOR fast.forward=1 TO VAL(xs(4)):LINE INPUT#9,x$:NEXT
1310 WEND:CLOSE IN
1320 IF z=1 THEN INPUT#3,"Sorry, no recipes found. Press <ENTER>
to return to MAIN PROGRAM.",z$:CLS:GOTO 620
1330 CLS#5:questions="Is a printout needed of this list (y/n)":v
alids="yn":GOSUB 90:IF operation=2 THEN CLS:GOTO 620
1340 headings=MID$(types,type),4)
1350 PRINT#8,headings:PRINT#8,STRINGS$(LEN(headings)+1,"*"):FOR x
=1 TO z:PRINT#8,recipex(x):NEXT:CLS:GOTO 620
1360
1370 '*** Sub-Program: Display of recipe
1380 PEN yellow:PRINT-PRINTOUT OR DISPLAY OF RECIPES":PEN green:
CLS#3
1390 PRINT#3,"Do you require a copy on paper (c) or display
to the screen (s)? ":questions="":valids="cs":GOSUB 90:o
utps=LOWER$(as):PRINT#3,as:LOCATE#3,1,8
1400 PRINT#3,"The following printout possibilities are available:
...":PRINT#3
1410 PRINT#3,"1 = Full printout of a specific recipe.":PRINT#3
1420 PRINT#3,"2 = Index (or catalogue) of recipes":PRINT#3,TAB(8
) = within a group.":PRINT#3
1430 PRINT#3,"3 = Complete catalogue of all recipes."
1440 questions="Which printout operation is required?":valids="1
23":GOSUB 90:all=0
1450 IF outps="s" THEN device=1:MODE 2:GOSUB 230 ELSE device=8
1460 CLS#3:ON operation GOTO 1470,1660,1770
1470 PRINT#3,"Please enter as accurately as possible the name o
f the recipe. If in doubt, use the 'SEARCH' procedure to
find out the correct name of the recipe.":PRINT#3
1480 INPUT#3,"The name:..":n$:names=LOWER$(n$)
1490 CLS:PEN yellow:PRINT-Printout of:.. CHR$(34) names CHR$(34
):PEN green:LOCATE#3,1,3

```



```

1500 PRINT#3,"I also need to know to which group it belongs. Th
e valid groups are...":offset=10:GOSUB 170
1510 questions$="To which group?":valid$="esmgd*cbhtr":GOSUB 90:t
ype$=a$
1520 CLS:IF device=8 THEN LOCATE 15,12:PRINT"PLEASE STAND BY:...
1530 filename$="index."+types
1540 OPENIN filenames$
1550 WHILE NOT EOF:found=0:GOSUB 260
1560 found=INSTR(LOWER$(x$(1)),names):IF found=0 THEN 1630
1570 PRINT#device,x$(1):PRINT#device,STRING$(LEN(x$(1)),"*")
1580 IF VAL(x$(2))>0 THEN PRINT#device,"This recipe is for "x$(2
) " persons." ELSE PRINT#device
1590 PRINT#device,"INGREDIENTS:":
1600 FOR x=1 TO VAL(x$(3)):LINE INPUT#9,x$:PRINT#device,TAB(18)
x$:NEXT:PRINT#device:IF device=1 THEN INPUT#4,"There is mor
e one the next 'page'. Press <ENTER> key to continue.",e$
1610 CLS#1:PRINT#device,"PREPARATION:":
1620 FOR x=1 TO VAL(x$(4)):LINE INPUT#9,x$:PRINT#device,USING"##
-:x:":PRINT#device," " x$:PRINT#device:NEXT:IF device=1 TH
EN INPUT#4,"End of recipe!! Press <ENTER> key to continue
operations. ",e$:CLOSEIN:GOTO 1640 ELSE CLOSEIN:GOTO 1640
1630 FOR x=1 TO VAL(x$(3))+VAL(x$(4)):LINE INPUT#9,x$:NEXT
1640 WEND:CLOSEIN:MODE 1:GOSUB 190:GOTO 620
1650 ,
1660 PRINT#3,"Index listing for all recipes from one of the foll
owing groups":offset=10:GOSUB 170
1670 question$="which group?":valid$="esmgd*cbhtr":GOSUB 90:file
name$="index."+a$:CLS#1
1680 PRINT#device,MID$(types$(operation),4):PRINT#device
1690 OPENIN filenames$
1700 WHILE NOT EOF:GOSUB 260
1710 PRINT#device,x$(1)
1720 FOR fast.forward=1 TO VAL(x$(3))+VAL(x$(4)):IF EOF THEN CLO
SEIN ELSE LINE INPUT#9,x$:NEXT
1730 WEND:IF all=1 THEN 1810
1740 CLOSEIN:IF device=1 THEN INPUT#4,"press <ENTER> to continue
.",q$
1750 MODE 1:GOSUB 190:GOTO 620
1760 ,
1770 MODE 1:GOSUB 190: IF device=1 THEN device=3:CLS:PEN yellow:
PRINT"TOTAL INDEX":PEN green
1780 FOR zx=1 TO 11:IF zx=6 THEN 1820 ELSE filename$="index."+LE
FT$(TYPE$(zx),1)
1790 PRINT#device,UPPER$(MID$(types$(zx),3)):PRINT#device,STRING$
$(LEN(types$(zx)),"*"):PRINT#device
1800 all=1:GOTO 1690
1810 IF device=8 THEN FOR zz=1 TO 6:PRINT#8:NEXT ELSE PRINT#devi
ce
1820 CLOSEIN:NEXT:IF device=3 THEN INPUT#5,"press <ENTER> to con
tinue.",q$
1830 CLS:GOTO 620
1840 ,
1850 '*** Sub-Program: Modify a recipe
1860 PEN yellow:PRINT"CORRECTION/MODIFICATION TO A RECIPE":PEN 9
reen:CLS#3
1870 INPUT#3,"Recipe Name:..":n$:name$=LOWER$(n$):PRINT#3
1880 PRINT#3,"From which group:..":offset=17:GOSUB 170
1890 question$="Which group?":valid$="esmgd*cbhtr":GOSUB 90:file
name$="index."+a$:CLS:PEN yellow:LOCATE 5,12:PRINT"PLEASE S
TAND BY:..."
1900 OPENIN filenames$:OPENOUT temp$:z=0
1910 WHILE NOT EOF:found=0:GOSUB 260
1920 found=INSTR(LOWER$(x$(1)),names):IF found=0 THEN 1940
1930 name$=x$(1):qty$=x$(2):ingred=VAL(x$(3)):prep=VAL(x$(4)):FO
R x=1 TO ingred:LINE INPUT#9,ingreds(x):NEXT:FOR x=1 TO pre
p:LINE INPUT#9,preps(x):NEXT:z=1:GOTO 1960
1940 FOR x=1 TO 4:PRINT#9,x$(x):NEXT
1950 FOR fast.forward=1 TO VAL(x$(3))+VAL(x$(4)):LINE INPUT#9,x$
:PRINT#9,x$:NEXT
1960 WEND:edit.exist=1:CLOSEIN
1970 IF z=0 THEN PRINT#1,"Sorry, recipe not found. Press <ENTER
> to return to the MAIN PROGRAM.- ELSE GOTO 900

```



```

1980 IF INKEYS="" THEN 1980 ELSE CLOSEIN:CLOSEOUT:CLS:GOTO 620
1990 GOSUB 560:MODE 1:GOSUB 190:GOTO 620
2000
2010 STOP
2020
2030 '*** Variables Definition Area
2040 MODE 1
2050 black=0:INK black,0
2060 green=1:INK green,22
2070 blue=2:INK blue,11
2080 yellow=3:INK yellow,24
2090 BORDER black:PAPER black:GOSUB 190:CLS
2100 WIDTH 75
2110 DIM types$(11),ingrds$(20),preps(20),recipes$(50)
2120 type$(1)="e = Entree"
2130 type$(2)="s = Soup"
2140 type$(3)="m = Main Course"
2150 type$(4)="g = Greens/Vegetables"
2160 type$(5)="d = Dessert"
2170 type$(6)=""
2180 type$(7)="c = Cakes"
2190 type$(8)="b = Biscuits"
2200 type$(9)="h = Horses Doovers"
2210 type$(10)="t = Tit-bits & Lollies"
2220 type$(11)="r = Refreshing Drinks"
2230 coms(1)="1 a j = Add a new recipe"
2240 coms(2)="f s l = Search for a recipe"
2250 coms(3)="l p l = Printout/Display of a recipe"
2260 coms(4)="f c l = Correct a recipe"
2270 temps="temp"
2280 controls=CHR$(11)+CHR$(18)+CHR$(11)+CHR$(7)
2290 PEN yellow:LOCATE 8,1
2300 PRINT-THE RECIPE FILE BUILDER"
2310 PEN green:PRINT:PRINT
2320 PRINT" This program will allow you to store all those fav
write recipes of yours, andrecall them as required.":PRINT:
PRINT
2330 PRINT" Follow the instructions as they are displayed on
the screen. Note that all single character entries do not
need the<ENTER> key to be pressed."
2340 PRINT:PEN yellow:PRINT"***** Special Note: pressing the <
+ > key will return you to this starting point. This sh
ould only be used as an emergency exit."
2350 PRINT#5,"press any key when ready to start.":
2360 IF INKEYS="" THEN 2360 ELSE CLS:GOTO 620
2370
2380 'Initializing routine !!!!! DANGER !!!!!
2390 CLS:LOCATE 5,12:PRINT-PLEASE STAND BY"
2400 FOR x=1 TO 11:filename$="index."+LEFT$(types(x),1)
2410 IF x=6 THEN 2430
2420 OPENOUT filename$:PRINT#9,"":CLOSEOUT
2430 NEXT
2440 RUN

```



(continued from Page 14)

MANUAL starting at chapter 14. You will find that most routines have ENTRY CONDITIONS consisting of values passed in registers. This would suggest that we need some way of passing values to the registers from BASIC as it is impossible to do this directly from BASIC. (ARNOLD has a method, but it is complicated and requires a greater understanding of all components of the Z80 than is necessary to use the firmware routines).

User Group Information

A tremendous amount of interest was shown in The Amstrad User Group stand at the PC85 exhibition in Melbourne last month. The stand was manned by various representatives from User Groups around Melbourne whose groups will no doubt benefit from increased membership. In addition, a number of enquiries were made concerning the establishment of new groups. All in all, it was a most successful first attempt to publicise the groups and we shall certainly push for a stand in the 1986 exhibition.

As the number of groups continues to increase, so too do the headaches and administrative worries with producing a constitution. In this issue, we are grateful to the Eastern Amstrad Users Group (Victoria) for allowing us to print a sample of their constitution and rules for emerging groups. With just a few local amendments, most groups in Victoria are using this format.

But first, news from groups around the country. Remember that these pages are the forum for any and all groups to advise readers of their existence and activities. If your group does not appear, it is either because no one has written or at the deadline (the end of the first week of the month prior to publication) has been missed. Remember too, that User Group Information is printed FREE OF CHARGE.

ICTORIA

Western Amstrad Users Group

The first meeting of the Western Amstrad Users Group was held towards the end of June, at Tottenham North Primary School, South Road, Tottenham in the multi-purpose room, which 15 members were present.

Don Leith (of the Central Users Group) opened the meeting and provided advice on Constitutional matters and answered a number of points concerning the establishment of the group. The Constitution was expected to be offered for adoption at the next meeting.

The meeting was also attended by Brian Jones (Editor of the Amstrad User).

The following were elected as members of the first committee: Mike McQueen (President:312 5594), Malcolm Auslan (Vice-President:749 5400), Peter Pilbeam (Secretary:336 0705) and Frank Melino (Treasurer:337 2495).

Regular meetings will be held each alternate Tuesday and Friday (this allows for shift workers). The next three meetings will be held on 6th and 18th August, and 3rd September.

A cordial invitation is extended to prospective new members and other User Groups. Further details can be obtained from the above officers.

Eastern Users Group

The first official meeting was held at the beginning of July at the Box Hill Scout Hall, Tyne Street. (The Hall is located in Halligan Park between Watts and Mersey Streets). The constitution and committee were ratified. Discussions were held on the formation of a training program, debugging clinic, a library, swap mart and additional seminars.

Further information can be obtained from Tony Blakemore on 878 6212.

QUEENSLAND

Brisbane Amstrad Computer Club

We welcome another new group, the Brisbane Amstrad Computer Club, which has recently formed in Annerley. With the help of Alliance Computers, prospective members were informed of the impending formation, which resulted in an attendance of approximately sixty people.

Meetings take place on the first tuesday of each month at the Junction Park State School (Room 15a) in Annerley and start at 7.30p.m.

Further details can be obtained from the following officers: Paul Witsen (President: 07 371 9259), Mal Harper (Secretary 07 288 3578) or Ian Cartwright (Treasurer: 07 369 9354).

Rockhampton User Group

We understand that a User group now exists in the Rockhampton area, but at the time of going to press we have not received details of contacts or meetings. We hope to be able to provide this information in next month's issue.

SOUTH AUSTRALIA

Amstrad Computer Club Inc.

As mentioned briefly last month, the first club to be established in South Australia appears to be the first club in the country to incorporate. They would be interested to know if any other groups have or are intending to take the same course.

Chris Sowden (President) reports that the current membership ranges from accountants to salesmen to engineers, professional programmers, a commercial pilot and firemen. By the time this magazine has been produced, the

club is expected to have been given a demo of the five and a quarter inch disc drive from a local supplier as well as a mod to Amstrad that exercises the second drive for text storage, and makes full use of graphic character sets for letterhead and logo construction.

The club is also beginning a course in Basic for new members, an introductory course in digital electronics and hopes soon to include tutorials on CP/M and machine code. Chris also hopes to eventually establish an electronic bulletin board with other groups in other states.

User Group Contact List

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

NSW

Mark Kelloway	Barrack Point	(042) 95 1581
Hans Hill	Blacktown	(02) 671 2929
Chris Craven	Canowindra	(063) 44 1150
Bruce Jones	Coffs Harbour	(066) 52 8334
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
R. Vijayenthiran	Newtown	(02) 519 4106
Ken Needs	St. Ives	(02) 449 5416
Chas Fletcher	Toongabbie	(02) 631 5037
Jim Owen	Uranga	(066) 55 6190
John Harwood	Windale	(049) 48 5337

ACT

Chris Rogers	Fraser	(062) 58 5749
Arthur McGuffin	Kambah	(062) 31 9437

Vic

Mike McQueen	Braybrook	(03) 312 5594
Don Leith	Brunswick	(03) 383 1498
David Carbone	Burwood	(03) 29 4135
Rod Anderson	Camperdown	(055) 93 2262
Frank Merlino	East Keilor	(03) 337 2495
Michael Prezens	Frankston	(03) 781 2158
Paul Walker	Heathmont	(03) 729 8657
Ron Butterfield	Leopold	(052) 50 2251
Sue Kelly	Manangatang	(050) 35 1402
Tony Blakemore	Nunawading	(03) 878 6212
Martin Scragg	Pearcedale	(059) 78 6949
R.A. Russo	Richmond	(03) 428 4281
Alan Harris	Sale	(051) 44 1454
Mrs. G. Chapman	South Clayton	(03) 551 4897
Peter Pilbeam	St. Albans	(03) 366 0705

QLD

Mial Harper	Bellbird Park	(07) 288 3
Paul Witsen	Bulimba	(07) 371 9
Steven Doyle	Caloundra	(071) 91 3
Mick O'Regan	Gladstone	(079) 79 2
Kylie Telford	Goondiwindi	Calinginee
D.F. Read	Ingham	(weekends)
Tim Takken	Ipswich	(077) 77 8
Michael Toussaint	Loganlea	(07) 202 4
Alan Laird	Maryborough	(07) 200 5
Ian Cartwright	Paddington	(071) 22 19
R.C. Watterton	Toowoomba	(07) 369 93
		(076) 35 4

SA

Chris Sowden	Morphetville	(08) 295 59
Lindsay Allen	Murray Bridge	(085) 32 23
Rick Cable	Pt. Pirie	(086) 59

WA

Mrs. P. Ardron	Carlisle	(09) 361 897
Bob Harwood	Cooloongup	(095) 27 177
Dave Andersen	6 Kitchener Rd	
	Merredin, 6415	
Tony Clitheroe	Morley	(09) 275 125
Graeme Worth	Scarborough	(09) 341 521
P.M. Nuyens	Waroona	(095) 33 117

TAS

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

NT

G.P. Heron	Tiwi	(089) 27 5792
------------	------	---------------

SAMPLE CONSTITUTION AND RULES

NAME

1. The name of the club shall be

hereinafter known as "The Club".

OBJECTS

2. The objects of the club will be to foster, stimulate and help provide facilities for the development of interest in the Amstrad range of computers, peripherals and software and to encourage and assist all incidental activities.

To encourage the development of software, no illegal copying of copyright material will be tolerated. To this end, all tape recorders will be banned from club premises. This does not of course subjugate the right of any member to make a legitimate back up security copy of any software program for their own purposes. Members may also swap copies of their own and public domain software.

The club will provide assistance to all levels of users in the form of regular seminars and tutorials to help develop programming skills.

The Learning Centre

An Introduction to Machine Code - Part Two

by Shane Kelly

This month we are going to build on the foundations laid down in the last article. Before we can proceed we must have a little bit of knowledge about the heart of our ARNOLD. This is the Z80 microprocessor, arguably the most popular of the microprocessors for home use. The Z80 is what is known as an 8 bit microprocessor i.e. it can only handle data that is one byte wide at one time. Bytes and bits are fundamental concepts that you should be aware of and if you are at all shaky on what they are consult your AMSTRAD CPC464 USER GUIDE appendix 2. This contains a short but informative look at these and other basic necessities.

The Z80 has built into it some special areas of memory that facilitate the transfer and manipulation of data. These areas are called REGISTERS and are referred to by capital letters e.g. A, B, C, D, E, F, H, L, IX and IY. Each of the registers named by a single letter are 8 bits wide, but the IX and IY registers are 16 bits wide. Some 8 bit registers can be paired with others to form registers 16 bits wide. The 16 bit register pairs are AF, BC, DE and HL. As you will learn there are great advantages in being able to pair registers in this manner. Some of these registers are more useful than others and some are special purpose registers such as the F register. This register is NOT available to the programmer directly, but by using certain opcodes we are able to change the contents of this, the FLAG REGISTER. It is called the FLAG register because it flags or reports on the results of certain operations carried out by the microprocessor. You have already used flags in your BASIC programs (probably without knowing it!) when

you use the IF.....THEN.....ELSE statement. Flags work like this:-

If an operation results in zero (for instance) then the ZERO FLAG is set i.e. the ZERO FLAG takes on a value of one. Our program tests for this and performs an operation that is different to the operation to be performed if the result is not zero. There are a number of flags in the flag register but only two are of special significance in our use of the FIRMWARE routines already built in to ARNOLD. These are the CARRY FLAG and the ZERO FLAG. In general the ARNOLD sets to one the value of the CARRY FLAG and the ZERO FLAG is reset to zero i.e. given the value zero if the routine we are using is carried out successfully. Turn to page 14-138 in the FIRMWARE GUIDE (or page 63 in the INS and OUTS).

Look at the entry named CAS OUT OPEN. Run your eye down the page to EXIT CONDITIONS. Read through them carefully. Each possible result is reported in the combination of each of these two flags. Those firmware routines that have several possible outcomes generally use this method of reporting those outcomes. Of the other registers, the A register is the only one that holds the result of an eight bit ADD or SUBTRACT operation. The other registers may hold data, but the A register is the one that holds the result. Of the 16 bit REGISTER PAIRS, HL is the one that most 16 bit operations use as a depository for their results. The IX and the IY registers are 16 bit registers and are used in a specialised fashion and need not concern us here, except to say that they are generally available to us as temporary storage areas for some data that we are not using at the moment, but may need

soon. One other 16 bit register is called the SP register or the STACK POINTER. A simple explanation of this register is to say that it references an area of memory that is set aside for the use of the system, but may, on occasion be used by us. Generally it is not wise to interfere with the contents of this register without a detailed knowledge of the workings of Z80.

Now, I realise that was a huge mouthful to swallow, but an understanding of REGISTERS and REGISTER PAIRS is vital for effectively using the FIRMWARE routines because most of the data required by those routines is PASSED to and from those routines IN REGISTERS! If you are having difficulty understanding the concept of registers and why they make programming easier, I will give you a method that I use that works quite well in gaining an understanding of new concepts. It is called parallel reading and basically it involves reading about the SAME concepts from different sources. Refer to the reference book that you should have on the Z80. Read the section in it explaining the registers and their usage. Keep referring back to this article and then again to your reference book. The cross-pollination should eventually result in a glimmer of light shining through (if you have any other reference sources use them also).

Up until now, any firmware routine that we have used have required no parameters (a fancy name for values or data) to be passed to it, but generally this is not the case as most routines need data to operate on to do anything useful. Take a break here and have a quick perusal through the FIRMWARE

(continued on Page 10)

JUNIOR JOTTERS

A Column for Young

Amstrad Users

As promised, I have now found the space to include a special section for our younger readers.

Of course, to continue this page will mean that you will have to keep me well supplied with information about your Amstrad. This could cover small programs you have developed to help you in a particular way (like the ones below) or a routine which you have found useful in your programs, and you think would help other people.

Don't forget that I would prefer to receive a tape containing your program as my typing is not very good, and a listing if possible, together with an explanation of what the program is supposed to do and any other information you think may be useful.

Alternatively, you could tell me for what you use your Amstrad - is it for playing games, for learning about computers or helping you with your homework? Do you use Amstrad at school?

Whatever you send in, it would be helpful if you could mention your age so that I can give a fair coverage to as many age groups as possible. I would be glad to look forward to hearing from you.

The Editor, Junior Jotters.

LETTERS

I have received a Junior Jotter and have a short program for other Jotters, to help with their multiplication.

```
LET A=0
LET B=0
LET C=INT(RND(1)*20+1)
LET D=INT(RND(1)*20+1)
PRINT "MULTIPLIERS"
PRINT "ARE";C,D;"EQUALS"
INPUT X
LET A=A+1
IF X=C*D THEN PRINT "YOU'VE GOT IT RIGHT!"
IF X=C*D THEN LET B=B+1
IF X<>C*D THEN PRINT "WRONG!"
IF A<6 THEN GOTO 30
PRINT "YOU'VE GOT";B;"OUT OF SIX"
END
```

R. Herbert, Warrnambool, Vic.

The enclosed routine may help other users. It can replace 'INPUT' and 'LINE INPUT' so that if an event timer is in place it will not be stopped. The same applies to an 'EVERY' command, making the keyboard entry possible without stopping the clock.

```
10 E$="":A$="":WHILE A$<>CHR$(13)
20 PRINT "*";CHR$(8);:A$="":WHILE
A$="":A$=INKEY$;WEND
30 IF A$=CHR$(127) AND E$<>" THEN
PRINT " ";CHR$(8);CHR$(8);:E$=LEFT$(E$,LEN(E$)-1):GOTO 50
40 IF A$<>CHR$(13) THEN E$=E$+A$:
PRINT A$;
50 WEND:PRINT " "
```

Tom Robertson, Manly, N.S.W.

SOFTWARE REVIEWS

To start the ball rolling, I asked Paul Dennis (age 10) to take a look at two games currently available for the Amstrad. This is what he thought about them.

Bridge It: The game is about men who come from a house and have to walk a zig-zag path along which there are four bridges. To allow the men to get to the end of each path, the bridges must be lowered to enable them to get across the water. Skill is required to lower the bridges so that the men do not fall into the water. Bridge It takes a long time to master - up to two weeks before you master level one, and up to five weeks to master level nine.

I like the game because it gives you a choice of lives, has nine levels and good instructions. I like the graphics with the contrasting colours and the challenge of the game as it quickens a persons hand movements.

I think there should be more points than one if a person gets home. I would prefer there to be more variety than a man coming out all the time, eg. could be a dog. I don't like that sometimes some of the platform is missing. I would prefer a less sudden start. I also think the music is great.

Overall, I enjoyed the idea of the game, and I find it amusing and I like the way it is planned out. It is one of the hardest games to play and is a lot of fun.

Write it Right: This tape is to help with spelling, and the difference between words which sound the same. The things I like about this game are the bright screen, the clear letters and the way it sets out the words. I have to think hard about the meaning of the words. The program is broken up into sets of ten words. If you spell a word wrongly, the program tells you that you have mis-spelt the word.

My brother (9) and a friend (15) have played the game which they have both found good for them.

Auld Lang Syne

A Poker Machine program from Tim Baldock

THE GAME

You are on holiday in Tasmania when you decide you need some excitement. So you go on a gambling spree at the Wrest Point Casino and spot an old-time Poker Machine and decide to try your luck. After many minutes (and many dollars) you begin to realise that the Poker Machine you are playing seems very much like a 'one-armed bandit'.

The game is based upon a realtime Poker Machine, so your odds of winning are slim! But don't be discouraged - there is a familiar song waiting for you when you hit the Jackpot. It costs you \$1 a game and you are given \$20 at the start.

HOW IT WORKS

10-40	REMS
50-60	Initialise
70-270	Draw game
300-540	Choose fruit
550-660	Determine win or not
670-770	Game play
840-970	Pull handle
980-1050	Draw lemon
1060-1230	Draw strawberry
1240-1370	Draw Grapes
1380-1430	Draw Bars
1460-1520	Spinning effects
1530-1990	Instructions
2000-2590	Combination table
2600-2710	In-game combination table
2720-2830	End Game
2840-2920	Animation for handle
2930-3070	Theme song 'Auld Lang Syne'

VARIABLES USED

MONEY

WIN

JACKPOT

X

Y

I

DE

A

B

C

PO

AB

A\$

AI\$

Your initial and remaining money

Size of Jackpot

Determine whether a Jackpot or Spit

X co-ordinate) Decide where to move

Y co-ordinate) the graphics and text

Set up delays and move text

Delays

Random no. to select 1st window fruit

Random no. to select 2nd window fruit

Random no. to select 3rd window fruit

Conditional GOSUBs

Conditional GOSUBs

Text to move

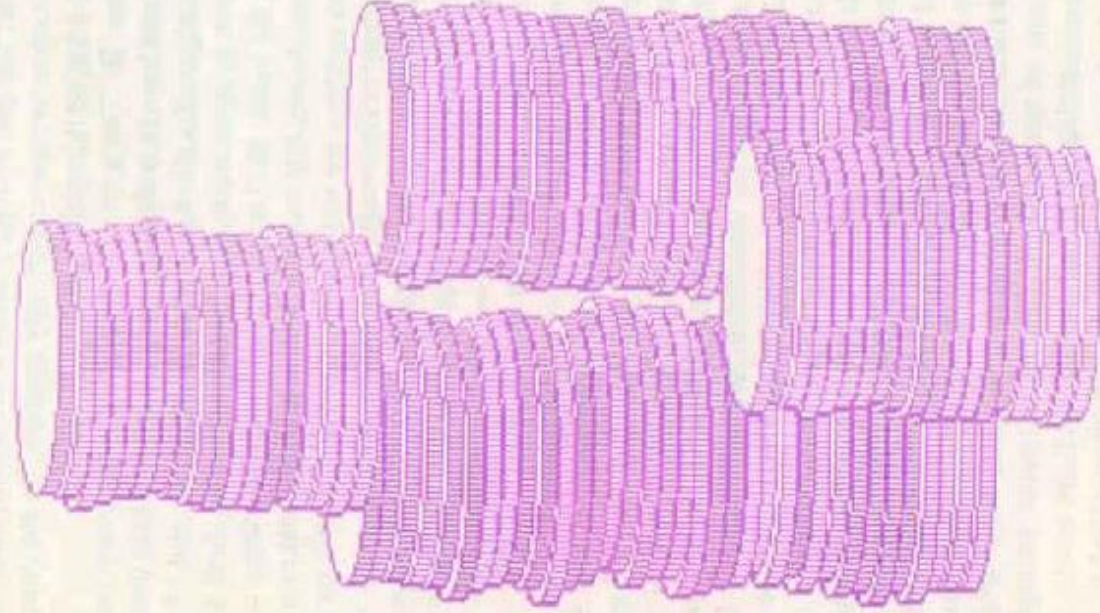
Inkey strings

HINTS

Take your time when typing in this game - it's rather long - but at least you won't have nagging bugs preventing the program from working.

ACKNOWLEDGEMENTS

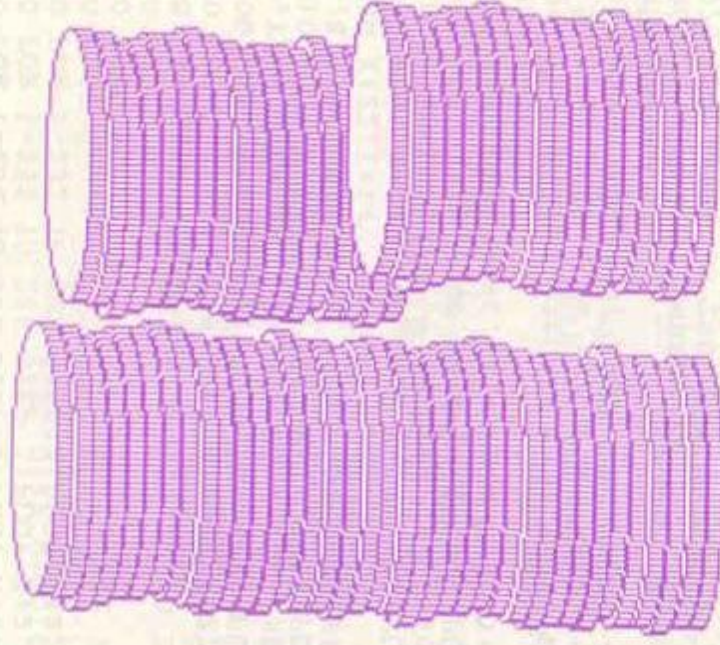
To Anne Baldock for information on Poker Machines, to Tony Blakemore for last minute adjustments and Tibor Gyore for general help.




```

5 MODE 1
10 '** AULD LANG SYNE **
20 '** A PROGRAM FOR THE AMSTRAD CPC 464 *
60 '** BY -TIM BALDOCK (C)COMPUTECH *
40 '** FOR THE AMSTRAD USER CLUB *
60 WIN=0:MONEY=20:JACKPOT=0
60 BORDER 0:PAPER 0:GOSUB 1530:GOSUB 2000
70 MODE 1:CLS:PLOT 0,0,1:DRAW 0,395:DRAW 635,395:DRAW 635,0:DRAW 0,0
80 PLOT 5,5,3:DRAW 5,390:DRAW 630,390:DRAW 630,5:DRAW 5,5
90 '** DRAW MACHINE **
00 PLOT 60,20,1:DRAW 60,320:DRAW 100,360:DRAW 300,360:DRAW 340,320:DR
W 340,20:DRAW 60,20
10 PLOT 60,40:DRAW 340,40
20 PLOT 80,310:DRAW 320,310:DRAW 320,210:DRAW 80,210:DRAW 80,310:PLOT
100,290:DRAW 300,290:DRAW 300,230:DRAW 100,230:DRAW 100,290
30 PLOT 100,290:DRAW 80,310:PLOT 300,290:DRAW 320,310:PLOT 300,230:DR
W 320,210:PLOT 100,230:DRAW 80,210
40 PLOT 110,60:DRAW 290,60:DRAW 340,40:PLOT 110,60:DRAW 60,40:PLOT 11
,60:DRAW 80,210:PLOT 290,60:DRAW 320,210
50 PLOT 120,80:DRAW 280,80:DRAW 300,190:DRAW 100,190:DRAW 120,80
60 LOCATE 9,15:PRINT"AULD LANG":PEN 3:LOCATE 11,16:PRINT"SYNE":PEN 1
70 '** MONEY TRAY **
80 PLOT 140,100:DRAW 240,100:DRAW 240,140:DRAW 140,140:DRAW 140,100:P
LOT 140,120:DRAW 240,120
90 PLOT 140,100:DRAW 160,110:DRAW 220,110:DRAW 240,100
00 PLOT 160,110:DRAW 160,120:PLOT 220,110:DRAW 220,120
10 '** HANDLE **
20 PLOT 340,200:DRAW 370,200:DRAW 370,300:DRAW 376,300:DRAW 376,190:D
RAW 376,190:DRAW 340,190
30 '** BALL **
40 PLOT 368,300:DRAW 378,300:PLOT 366,302:PLOT 364,304:DRAW 364,308:P
LOT 366,310:PLOT 368,312:DRAW 378,312:PLOT 380,310:PLOT 382,308:DRAW 3
82,304:PLOT 380,302
50 IF AS=1 THEN RETURN: '*FOR MOVEMENT*
60 FOR I=100 TO 300 STEP 66.6:PLOT I,290:DRAW I,230
70 NEXT
80 GOSUB 2600:PO=1:GOSUB 740:PO=0
90 IF BS=" " THEN 330
00 AS=INKEYS:IF AS="" THEN 300
10 IF AS=" " OR BS=" " THEN 330
20 GOTO 300
30 GOSUB 2840
40 AS=0:A=RND(1)
50 IF A>0.14 THEN 340
60 X=100:Y=230:GOSUB 1450
70 X=X+66:GOSUB 1450
80 X=X+66:GOSUB 1450
90 IF A>0 AND A<0.03 THEN X=100:GOSUB 980
00 IF A>0.03 AND A<0.1 THEN X=100:GOSUB 1060
10 IF A>0.1 AND A<0.12 THEN X=100:GOSUB 1240
20 IF A>0.12 THEN X=100:GOSUB 1380
30 B=RND(1)
40 IF B>0.156 THEN 430
50 IF B>0 AND B<0.07 THEN X=X+66:GOSUB 980
60 IF B>0.07 AND B<0.1 THEN X=X+66:GOSUB 1060
70 IF B>0.1 AND B<0.15 THEN X=X+66:GOSUB 1240
80 IF B>0.152 THEN X=X+66:GOSUB 1380
90 C=RND(1)
00 IF C>0.27 THEN 490
10 IF C>0 AND C<0.05 THEN X=X+66:GOSUB 980
20 IF C>0.05 AND C<0.1 THEN X=X+66:GOSUB 1060
30 IF C>0.1 AND C<0.2 THEN X=X+66:GOSUB 1240
40 IF C>0.2 THEN X=X+66:GOSUB 1380
50 WIN=0
60 IF A>0.12 AND B>0.15 AND C>0.2 THEN WIN=200:X1=1
70 IF A>0 AND A<0.03 AND B>0 AND B<0.07 AND C>0 AND C<0.05 THEN WIN=3

```




```

0:X1=1
580 IF A>0.1 AND A<0.12 AND B>0.1 AND B<0.15 AND C>0.1 AND C<0.2 THEN
WIN=10:X1=1
590 IF A>0.03 AND A<0.1 AND B>0.07 AND B<0.1 AND C>0.05 AND C<0.1 THEN
WIN=5:X1=1
600 IF A>0 AND A<0.03 AND B>0 AND B<0.07 AND C>0.2 THEN WIN=15:X1=1
610 IF A>0.1 AND A<0.12 AND B>0.1 AND B<0.15 AND C>0.2 THEN WIN=7:X1=1
620 IF A>0.03 AND A<0.1 AND B>0.07 AND B<0.1 AND C>0.2 THEN WIN=3:X1=1
630 IF A>0.1 AND A<0.12 AND B>0.1 AND B<0.15 AND C<0.1 THEN WIN=2:X1=2
640 IF A>0.1 AND A<0.12 AND B>0.1 AND B<0.15 AND C>0.1 THEN WIN=2:X1=2
650 IF A>0.1 AND A<0.12 AND B<0.1 THEN WIN=1:X1=2
660 IF A>0.1 AND A<0.12 AND B>0.15 THEN WIN=1:X1=2
670 MONEY=MONEY-1
680 MONEY=MONEY+WIN
590 IF MONEY<=0.5 THEN FOR I=1 TO 1000:NEXT I:GOTO 2720
700 IF WIN=0 THEN 740
710 IF X1=1 THEN LOCATE 27,15:PEN 3:PRINT"JACK*POT"
720 IF X1=2 THEN LOCATE 27,15:PEN 3:PRINT"SPIT"
730 IF X1=1 OR X1=2 THEN PEN 1:LOCATE 27,16:PRINT"WINS";WIN;".00"
740 LOCATE 27,18:PRINT"YOU NOW HAVE"
750 LOCATE 29,20:PRINT"$";MONEY;".00"
760 LOCATE 27,24:PRINT"PRESS <SPACE>"
770 IF PO=1 THEN RETURN
780 '** DECIDE HOW MUCH MUSIC TO PLAY **
790 IF WIN=5 THEN AB=1:GOSUB 2930
800 IF WIN=7 THEN AB=2:GOSUB 2930
810 IF WIN=10 THEN AB=3:GOSUB 2930
820 IF WIN=15 THEN AB=4:GOSUB 2930
830 IF WIN=30 OR WIN=200 THEN AB=5:GOSUB 2930
840 '** PULL HANDLE **
850 BS=INKEY$:IF BS="" THEN 850
860 IF JO=2 THEN 880
870 IF B$<>" THEN 850
880 FOR DELAY=1 TO 1000:NEXT
890 LOCATE 27,15:PRINT"
900 LOCATE 27,16:PRINT"
910 X=100
920 FOR I=232 TO 288 STEP 2:PLOT X+2,I,0:DRAW X+64,I:NEXT I:X=X+66
930 FOR I=232 TO 288 STEP 2:PLOT X+2,I:DRAW X+64,I:NEXT I:X=X+66
940 FOR I=232 TO 288 STEP 2:PLOT X+2,I:DRAW X+64,I:NEXT I
950 PLOT 0,0,1
960 GOTO 90
970 END
980 '** DRAW LEMON **
990 FOR I=232 TO 288 STEP 2:PLOT X+2,I,3:DRAW X+64,I:NEXT
1000 PLOT X+6,Y+28,1:PLOT X+6,Y+30:PLOT X+8,Y+32:DRAW X+8,Y+26:PLOT X+
10,Y+36:DRAW X+10,Y+22:PLOT X+12,Y+38:DRAW X+12,Y+20
1010 PLOT X+14,Y+40:DRAW X+14,Y+18:PLOT X+16,Y+42:DRAW X+16,Y+16:PLOT
X+18,Y+42:DRAW X+18,Y+16:PLOT X+20,Y+44:DRAW X+20,Y+14:PLOT X+22,Y+44:
DRAW X+22,Y+14
1020 PLOT X+24,Y+46:DRAW X+24,Y+12:PLOT X+26,Y+46:DRAW X+26,Y+12:FOR I
=28 TO 40:PLOT X+1,Y+48:DRAW X+1,Y+10:NEXT I
1030 PLOT X+42,Y+46:DRAW X+42,Y+12:PLOT X+44,Y+46:DRAW X+44,Y+12:PLOT
X+46,Y+44:DRAW X+46,Y+14:PLOT X+48,Y+44:DRAW X+48,Y+14:PLOT X+50,Y+42:
DRAW X+50,Y+16
1040 PLOT X+52,Y+42:DRAW X+52,Y+16:PLOT X+54,Y+40:DRAW X+54,Y+18:PLOT
X+56,Y+38:DRAW X+56,Y+20:PLOT X+58,Y+36:DRAW X+58,Y+22:PLOT X+60,Y+32:
DRAW X+60,Y+26:PLOT X+62,Y+30:PLOT X+62,Y+28
1050 RETURN
1060 '** DRAW STRAWBERRY **
1070 FOR I=232 TO 288 STEP 2:PLOT X+2,I,0:DRAW X+64,I:NEXT
1080 PLOT X+20,Y+50,3:DRAW X+46,Y+50:PLOT X+16,Y+48:PLOT X+18,Y+48:PLO
T X+14,Y+46:PLOT X+14,Y+44:PLOT X+12,Y+42:PLOT X+12,Y+40:PLOT X+11,Y+3
8:DRAW X+11,Y+30
1090 PLOT X+12,Y+28:DRAW X+12,Y+24:PLOT X+14,Y+22:PLOT X+14,Y+20:PLOT
X+16,Y+18:PLOT X+16,Y+16:PLOT X+18,Y+14:PLOT X+20,Y+12:PLOT X+22,Y+10:

```




```

PLOT X+24,Y+8:PLOT X+26,Y+6:PLOT X+28,Y+6:PLOT X+30,Y+4:DRAW X+36,Y+4
1100 PLOT X+38,Y+6:PLOT X+40,Y+6:PLOT X+42,Y+8:PLOT X+44,Y+10:PLOT X+4
6,Y+12:PLOT X+48,Y+14:PLOT X+50,Y+16:PLOT X+52,Y+18:PLOT X+54,Y+20:PLO
T X+56,Y+22:PLOT X+58,Y+24:DRAW X+54,Y+28:PLOT X+56,Y+30:DRAW X+56,Y+3
8
1110 PLOT X+54,Y+40:PLOT X+54,Y+42:PLOT X+52,Y+44:PLOT X+52,Y+46:PLOT
X+50,Y+48:PLOT X+48,Y+48
1120 REM ** DRAW LEAF **
1130 PLOT X+32,Y+48:DRAW X+32,Y+44:PLOT X+30,Y+42:PLOT X+30,Y+40:PLOT
X+28,Y+38:PLOT X+28,Y+36:PLOT X+26,Y+34:DRAW X+16,Y+34:PLOT X+14,Y+36:
PLOT X+12,Y+38
1140 PLOT X+34,Y+42:PLOT X+34,Y+40:PLOT X+36,Y+38:PLOT X+36,Y+36:PLOT
X+38,Y+34:DRAW X+50,Y+34:PLOT X+52,Y+36:PLOT X+54,Y+38
1150 ' ** DRAW STARK **
1160 PLOT X+34,Y+50,9:DRAW X+38,Y+54:PLOT X+38,Y+56:DRAW X+36,Y+56:DRA
W X+30,Y+50
1170 PLOT X+20,Y+30:PLOT X+28,Y+30:PLOT X+36,Y+30:PLOT X+44,Y+30:PLOT
X+18,Y+28:PLOT X+22,Y+28:PLOT X+26,Y+28:PLOT X+30,Y+28:PLOT X+34,Y+28:
PLOT X+38,Y+28:PLOT X+42,Y+28:PLOT X+46,Y+28
1180 PLOT X+20,Y+26:PLOT X+28,Y+26:PLOT X+36,Y+26:PLOT X+44,Y+26
1190 PLOT X+24,Y+22:PLOT X+32,Y+22:PLOT X+40,Y+22:PLOT X+22,Y+20:PLOT
X+26,Y+20:PLOT X+30,Y+20:PLOT X+34,Y+20:PLOT X+38,Y+20:PLOT X+42,Y+20
1200 PLOT X+24,Y+18:PLOT X+32,Y+18:PLOT X+40,Y+18
1210 PLOT X+28,Y+14:PLOT X+36,Y+14:PLOT X+26,Y+12:PLOT X+30,Y+12:PLOT
X+34,Y+12:PLOT X+38,Y+12:PLOT X+28,Y+10:PLOT X+36,Y+10
1220 PLOT 0,0,1
1230 RETURN
1240 ' ** DRAW GRAPES **
1250 FOR I=232 TO 288 STEP 2:PLOT X+2,I,0:DRAW X+64,I:NEXT I
1260 ' ** DRAW LEAF **
1270 PLOT X+30,Y+56,3:DRAW X+30,Y+44:PLOT X+28,Y+50:PLOT X+26,Y+52:PLO
T X+24,Y+54:PLOT X+22,Y+56:DRAW X+18,Y+56:PLOT X+16,Y+54:PLOT X+14,Y+5
4:PLOT X+12,Y+52:PLOT X+14,Y+50
1280 PLOT X+16,Y+48:PLOT X+18,Y+48:PLOT X+20,Y+46:DRAW X+26,Y+46:PLOT
X+28,Y+48
1290 PLOT X+32,Y+50:PLOT X+34,Y+52:PLOT X+36,Y+54:PLOT X+38,Y+56:DRAW
X+42,Y+56:PLOT X+44,Y+54:PLOT X+46,Y+54:PLOT X+48,Y+52:PLOT X+46,Y+50:
PLOT X+44,Y+48:PLOT X+42,Y+48:PLOT X+40,Y+46:DRAW X+34,Y+46:PLOT X+32,
Y+48
1300 PLOT X+8,Y+40,7:DRAW X+8,Y+36:PLOT X+10,Y+42:DRAW X+10,Y+34:PLOT
X+12,Y+44:DRAW X+12,Y+32:PLOT X+14,Y+44:DRAW X+14,Y+32:PLOT X+16,Y+44:
DRAW X+16,Y+32:PLOT X+18,Y+42:DRAW X+18,Y+34:PLOT X+20,Y+40:DRAW X+20,
Y+36
1310 PLOT X+24,Y+40:DRAW X+24,Y+36:PLOT X+26,Y+42:DRAW X+26,Y+34:PLOT
X+28,Y+44:DRAW X+28,Y+32:PLOT X+30,Y+44:DRAW X+30,Y+32:PLOT X+32,Y+44:
DRAW X+32,Y+32:PLOT X+34,Y+42:DRAW X+34,Y+34:PLOT X+36,Y+40:DRAW X+36,
Y+36
1320 PLOT X+40,Y+40:DRAW X+40,Y+36:PLOT X+42,Y+42:DRAW X+42,Y+34:PLOT
X+44,Y+44:DRAW X+44,Y+32:PLOT X+46,Y+44:DRAW X+46,Y+32:PLOT X+48,Y+44:
DRAW X+48,Y+32:PLOT X+50,Y+42:DRAW X+50,Y+34:PLOT X+52,Y+40:DRAW X+52,
Y+36
1330 PLOT X+16,Y+26:DRAW X+16,Y+22:PLOT X+18,Y+28:DRAW X+18,Y+20:PLOT
X+20,Y+30:DRAW X+20,Y+18:PLOT X+22,Y+30:DRAW X+22,Y+18:PLOT X+24,Y+30:
DRAW X+24,Y+18:PLOT X+26,Y+28:DRAW X+26,Y+20:PLOT X+28,Y+26:DRAW X+28,
Y+22
1340 PLOT X+32,Y+26:DRAW X+32,Y+22:PLOT X+34,Y+28:DRAW X+34,Y+20:PLOT
X+36,Y+30:DRAW X+36,Y+18:PLOT X+38,Y+30:DRAW X+38,Y+18:PLOT X+40,Y+30:
DRAW X+40,Y+18:PLOT X+42,Y+28:DRAW X+42,Y+20:DRAW X+44,Y+26:DRAW X+44,
Y+22
1350 PLOT X+24,Y+12:DRAW X+24,Y+8:PLOT X+26,Y+14:DRAW X+26,Y+6:PLOT X+
28,Y+16:DRAW X+28,Y+4:PLOT X+30,Y+16:DRAW X+30,Y+4:PLOT X+32,Y+16:DRAW
X+32,Y+4:PLOT X+34,Y+14:DRAW X+34,Y+6:PLOT X+36,Y+12:DRAW X+36,Y+8
1360 PLOT 0,0,1
1370 RETURN
1380 ' ** DRAWS BARS **
1390 FOR I=232 TO 288:PLOT X+2,I,0:DRAW X+64,I:NEXT I

```



```

1400 FOR I=54 TO 46 STEP -2:PLOT X+6,Y+I,1:DRAW X+60,Y+I:NEXT I
1410 FOR I=34 TO 26 STEP -2:PLOT X+6,Y+I:DRAW X+60,Y+I:NEXT I
1420 FOR I=14 TO 6 STEP -2:PLOT X+6,Y+I:DRAW X+60,Y+I:NEXT I
1430 RETURN
1440 END
1450 ENV 1,5,3,1,2,-3,1,1,0,10,3,-3,1:SOUND 1,767,-6,11,1
1460 PLOT X+6,Y+56:DRAW X+6,Y+40:PLOT X+6,Y+34:DRAW X+6,Y+28:PLOT X+6,
Y+20:DRAW X+6,Y+4:PLOT X+12,Y+56:DRAW X+12,Y+4:PLOT X+16,Y+56:PLOT X+1
6,Y+54:PLOT X+16,Y+48:DRAW X+16,Y+42:PLOT X+16,Y+38:PLOT X+16,Y+36:PLO
T X+16,Y+32:PLOT X+16,Y+26:PLOT X+16,Y+20
1470 PLOT X+16,Y+18:PLOT X+16,Y+12:PLOT X+16,Y+6:PLOT X+16,Y+4:PLOT X+
22,Y+46:DRAW X+22,Y+50:PLOT X+22,Y+38:DRAW X+22,Y+10:PLOT X+22,Y+8:DRA
W X+22,Y+4:PLOT X+28,Y+56:DRAW X+28,Y+4
1480 PLOT X+34,Y+52:PLOT X+34,Y+44:DRAW X+34,Y+42:PLOT X+34,Y+24:PLOT
X+34,Y+22:PLOT X+34,Y+18:DRAW X+34,Y+4:PLOT X+38,Y+56:DRAW X+38,Y+50:P
LOT X+38,Y+44:PLOT X+38,Y+38:DRAW X+38,Y+32:PLOT X+38,Y+24:PLOT X+38,Y
+22:PLOT X+38,Y+16:DRAW X+38,Y+4
1490 PLOT X+44,Y+40:DRAW X+44,Y+32:PLOT X+44,Y+18:DRAW X+44,Y+6:PLOT X
+52,Y+54:PLOT X+52,Y+54:PLOT X+54,Y+28:DRAW X+54,Y+22:PLOT X+54,Y+12:D
RAW X+54,Y+4
1500 PLOT X+48,Y+56:DRAW X+48,Y+40:DRAW X+58,Y+40:DRAW X+58,Y+10
1510 FOR I=1 TO 250:NEXT I:RETURN
1520 END
1530 CLS:PLOT 0,0:DRAW 0,395:DRAW 635,395:DRAW 635,0:DRAW 0,0:PLOT 5,5
:DRAW 5,390:DRAW 630,390:DRAW 630,5:DRAW 5,5
1540 AS="AULD LANG SYNE"
1550 I=0:X=38
1560 I=I+1:X=X-1
1570 FOR DE=1 TO 100:NEXT
1580 PEN 2:LOCATE X,3:PRINT LEFT$(AS,I); " "
1590 IF X<15 THEN GOTO 1610
1600 GOTO 1560
1610 AS="PROGRAM BY - TIM BALDOCK"
1620 I=0:X=38
1630 I=I+1:X=X-1
1640 FOR DE=1 TO 100:NEXT
1650 PEN 3:LOCATE X,5:PRINT LEFT$(AS,I); " "
1660 IF X<10 THEN GOTO 1690
1670 GOTO 1630
1680 INK 0,0:PEN 1
1690 LOCATE 3,7:PRINT"You are on a holiday in TASMANIA"
1700 LOCATE 3,8:PRINT"when you decide to go on a gambling"
1710 LOCATE 3,9:PRINT"spree at the WESTPOINT CASINO.There"
1720 LOCATE 3,10:PRINT"you spot an inconspicuous looking"
1730 LOCATE 3,11:PRINT"Poker Machine, but little did you"
1740 LOCATE 3,12:PRINT"know that it was a 'one armed bandit"
1750 LOCATE 3,13:PRINT"set on stealing your money. So be"
1760 LOCATE 3,14:PRINT" wary, or you will end up broke."
1770 INK 1,24:INK 0,0:INK 2,26:INK 3,6
1780 AS="YOU HAVE $20.00 TO RISK"
1790 X=38:I=0
1800 X=X-1:I=I+1
1810 PEN 3:LOCATE X,15:PRINT LEFT$(AS,I); " "
1820 IF X<10 THEN 1840
1830 GOTO 1800
1840 AS="GOOD LUCK"
1850 X=38:I=0
1860 X=X-1:I=I+1
1870 PEN 13:LOCATE X,17:PRINT LEFT$(AS,I); " "
1880 IF X<17 THEN 1900
1890 GOTO 1860
1900 X=230:Y=70
1910 GOSUB 1000:X=X+64:GOSUB 1000:X=X+64:GOSUB 1000
1920 AS="<PRESS ANY KEY TO CONTINUE)"
1930 X=38:I=0
1940 X=X-1:I=I+1

```



```

950 PEN 2:LOCATE X,23:PRINT LEFT$(A$,I);" "
960 IF X<10 THEN 1980
970 GOTO 1940
980 C$=INKEY$:IF C$="" THEN 1980
990 X=0:Y=0:A$="":RETURN
0000 CLS:PLOT 0,0:DRAW 0,395:DRAW 635,395:DRAW 635,0:DRAW 0,0:PLOT 5,5
0010 :DRAW 5,390:DRAW 630,390:DRAW 630,5:DRAW 5,5
0020 PLOT 0,0,1
020 A$="COMBINATION TABLE"
030 X=38:I=0
040 X=X-1:I=I+1
050 PEN 3:LOCATE X,3:PRINT LEFT$(A$,I);" "
060 IF X<14 THEN 2080
070 GOTO 2040
080 X=200:Y=230:GOSUB 1380:X=X+64:GOSUB 1380:X=X+64:GOSUB 1380
090 PEN 1:LOCATE 29,9:PRINT"=$200"
000 PEN 2:LOCATE 12,12:PRINT"BAR-BAR-BAR WINS $200"
0010 PEN 13:LOCATE 3,20:PRINT"<SPACE to continue> <G for Game>"
020 A1$=INKEY$:IF A1$="" THEN 2120
030 IF A1$=" " THEN A1$="":GOTO 2150
040 IF A1$="G" OR A1$="g" THEN RETURN
045 GOTO 2120
050 X=200:Y=230:GOSUB 980:X=X+64:GOSUB 980:X=X+64:GOSUB 980
060 LOCATE 29,9:PRINT"=$30"
070 LOCATE 9,12:PRINT"LEMON-LEMON-LEMON WINS $30"
080 A1$=INKEY$:IF A1$="" THEN 2180
090 IF A1$=" " THEN 2210
000 IF A1$="G" OR A1$="g" THEN RETURN
005 GOTO 2180
010 X=200:Y=230:X=X+128:GOSUB 1380
020 LOCATE 29,9:PRINT"=$15"
030 LOCATE 9,12:PRINT" LEMON-LEMON-BAR WINS $15"
040 A1$=INKEY$:IF A1$="" THEN 2240
050 IF A1$=" " THEN 2270
060 IF A1$="G" OR A1$="g" THEN RETURN
065 GOTO 2240
070 X=200:Y=230:GOSUB 1240:X=X+64:GOSUB 1240:X=X+64:GOSUB 1240
080 LOCATE 29,9:PRINT"=$10"
090 LOCATE 9,12:PRINT"GRAPE-GRAPE-GRAPE WINS $10"
000 A1$=INKEY$:IF A1$="" THEN 2300
010 IF A1$=" " THEN 2330
020 IF A1$="G" OR A1$="g" THEN RETURN
025 GOTO 2300
030 X=200:Y=230:X=X+128:GOSUB 1380
040 LOCATE 29,9:PRINT"=$7"
050 LOCATE 9,12:PRINT" GRAPE-GRAPE-BAR WINS $7"
055 A1$=INKEY$:IF A1$="" THEN 2360
070 IF A1$=" " THEN 2390
080 IF A1$="G" OR A1$="g" THEN RETURN
085 GOTO 2360
090 X=200:Y=230:GOSUB 1060:X=X+64:GOSUB 1060:X=X+64:GOSUB 1060
000 LOCATE 29,9:PRINT"=$5"
010 LOCATE 9,12:PRINT" STRAW-STRAW-STRAW WINS $5"
020 A1$=INKEY$:IF A1$="" THEN 2420
030 IF A1$=" " THEN 2450
040 IF A1$="G" OR A1$="g" THEN RETURN
045 GOTO 2420
050 X=200:Y=230:X=X+128:GOSUB 1380
060 X=200:Y=230:GOSUB 1240:X=X+64:GOSUB 1240:X=X+64:GOSUB 1240:X=X+64:FOR I=230 TO 300:
0010 X,I,0:DRAW X+64,I:NEXT
00 LOCATE 29,9:PRINT"=$2"
000 LOCATE 9,12:PRINT" GRAPE-GRAPE WINS $2"
000 A1$=INKEY$:IF A1$="" THEN 2490
000 IF A1$=" " THEN 2530
000 IF A1$="G" OR A1$="g" THEN RETURN
000 GOTO 2490

```



```

2530 X=200:Y=230:X=X+64:FOR I=230 TO 300:PLOT X,I,0:DRAW X+64,I:NEXT
2540 LOCATE 29,9:PRINT"=$1"
2550 LOCATE 9,12:PRINT" GRAPE
2560 FOR I=1 TO 1000:NEXT:RETURN
2570 END
2580 INK 1,26
2590 RETURN
2600 PLOT 400,350:DRAW 620,350:DRAW 620,40:DRAW 400,40:DRAW 400,350
2610 LOCATE 27,5:PRINT"B B B = $200"
2620 LOCATE 27,6:PRINT"L L L = $30"
2630 LOCATE 27,7:PRINT"L L B = $15"
2640 LOCATE 27,8:PRINT"G G G = $10"
2650 LOCATE 27,9:PRINT"G G B = $7"
2660 LOCATE 27,10:PRINT"S S S = $5"
2670 LOCATE 27,11:PRINT"S S B = $3"
2680 LOCATE 27,12:PRINT"G G = $2"
2690 LOCATE 27,13:PRINT"G
2700 LOCATE 27,24:PRINT"PRESS <SPACE>"
2710 RETURN
2720 CLS
2730 PLOT 0,0:DRAW 0,395:DRAW 635,395:DRAW 635,0:DRAW 0,0
2740 PLOT 10000,10000,3
2750 PLOT 5,5:DRAW 5,390:DRAW 630,390:DRAW 630,5:DRAW 5,5
2760 PLOT 10000,10000,1
2770 LOCATE 7,4:PRINT"You LOST all your money !!"
2780 LOCATE 15,7:PRINT"HARD LUCK !!!"
2790 LOCATE 7,15:PRINT"Would you like another go"
2800 FOR A=1 TO 500:NEXT A
2810 C$=INKEY$:IF C$<>"Y" AND C$<>"y" AND C$<>"N" AND C$<>"n" THEN 281
0
2820 IF C$="Y" OR C$="y" THEN CLS:MONEY=10:GOTO 70
2830 CLS:END
2840 ** MOVE HANDLE **
2850 FOR I=380 TO 240 STEP-2:PLOT 360,I,0:DRAW 390,I:NEXT I
2860 PLOT 340,200,1:DRAW 370,200:DRAW 370,240:DRAW 376,240:DRAW 376,19
0:DRAW 340,190
2870 REM ** BALL **
2880 PLOT 368,240:DRAW 378,240:PLOT 366,242:PLOT 364,244:DRAW 364,248:
382,244:PLOT 368,252:DRAW 378,252:PLOT 380,250:PLOT 382,248:DRAW
2890 FOR I=1 TO 1000:NEXT
2900 FOR I=252 TO 240 STEP-2:PLOT 364,I,0:DRAW 382,I:NEXT I:PLOT 1000
0,10000,1
2910 FOR I=1 TO 100:NEXT I:AS=1:GOSUB 210
2920 RETURN
2930 P=0:D=0:REM ** THEME SONG AULD LANGE SYNE **
2940 SOUND 2,638,60,5:SOUND 2,478,85,5:SOUND 2,0,5,5:SOUND 2,478,25,5:
SOUND 2,0,5,5
2950 IF AB=1 THEN RETURN
2960 SOUND 2,478,60,5:SOUND 2,379,60,5:SOUND 2,426,90,5:SOUND 2,478,30
,5:SOUND 2,426,60,5
2970 IF AB=2 THEN RETURN
2980 SOUND 2,379,60,5:SOUND 2,478,85,5:SOUND 2,0,5,5:SOUND 2,478,30,5:
SOUND 2,379,60,5:SOUND 2,319,60,5
2990 IF AB=3 THEN RETURN
3000 SOUND 2,284,175,5:SOUND 2,0,5,5:SOUND 2,284,60,5:SOUND 2,319,90,5
D 2,426,90,5:SOUND 2,478,30,5:SOUND 2,426,60,5
3010 IF AB=4 THEN RETURN
3020 SOUND 2,379,60,5:SOUND 2,478,90,5:SOUND 2,568,25,5:SOUND 2,0,5,5:
SOUND 2,568,60,5
3030 SOUND 2,638,60,5:SOUND 2,478,120,5
3040 IF AB=5 THEN RETURN
3050 SOUND 2,5,379,5:SOUND 2,60,478,5:SOUND 2,60,426,5:SOUND 2,90,478,
5:SOUND 2,30,426,5:SOUND 2,60,0
3060 DATA 379,60,478,90,568,25,0,5,568,60

```



```

530 X=200:Y=230:X=X+64:FOR I=230 TO 300:PLOT X,I,0:DRAW X+64,I:NEXT
540 LOCATE 29,9:PRINT"=$1"
550 LOCATE 9,12:PRINT" GRAPE WINS $1 "
560 FOR I=1 TO 1000:NEXT:RETURN
570 END
580 INK 1,26
590 RETURN
600 PLOT 400,350:DRAW 620,350:DRAW 620,40:DRAW 400,40:DRAW 400,350
610 LOCATE 27,5:PRINT"B B = $200"
620 LOCATE 27,6:PRINT"L L = $30"
630 LOCATE 27,7:PRINT"L L B = $15"
640 LOCATE 27,8:PRINT"G G = $10"
650 LOCATE 27,9:PRINT"G G B = $7"
660 LOCATE 27,10:PRINT"S S = $5"
670 LOCATE 27,11:PRINT"S S B = $3"
680 LOCATE 27,12:PRINT"G G = $2"
690 LOCATE 27,13:PRINT"G = $1"
700 LOCATE 27,24:PRINT"PRESS <SPACE>"
710 RETURN
720 CLS
730 PLOT 0,0:DRAW 0,395:DRAW 635,395:DRAW 635,0:DRAW 0,0
740 PLOT 10000,10000,3
750 PLOT 5,5:DRAW 5,390:DRAW 630,390:DRAW 630,5:DRAW 5,5
760 PLOT 10000,10000,1
770 LOCATE 7,4:PRINT"You LOST all your money !!"
780 LOCATE 15,7:PRINT"HARD LUCK .."
790 LOCATE 7,15:PRINT"Would you like another go"
800 FOR A=1 TO 500:NEXT A
810 C$=INKEY$:IF C$<>"Y" AND C$<>"N" AND C$<>"n" THEN 281
820 IF C$="Y" OR C$="y" THEN CLS:MONEY=10:GOTO 70
830 CLS:END
840 '** MOVE HANDLE **
850 FOR I=380 TO 240 STEP-2:PLOT 360,I,0:DRAW 390,I:NEXT I
860 PLOT 340,200,1:DRAW 370,200:DRAW 370,240:DRAW 376,240:DRAW 376,19
:DRAW 340,190
870 REM ** BALL **
880 PLOT 368,240:DRAW 378,240:PLOT 366,242:PLOT 364,244:DRAW 364,248:
PLOT 366,250:PLOT 368,252:DRAW 378,252:PLOT 380,250:PLOT 382,248:DRAW
32,244:PLOT 380,242
890 FOR I=1 TO 1000:NEXT
900 FOR I=252 TO 240 STEP-2:PLOT 364,I,0:DRAW 382,I:NEXT I:PLOT 1000
,10000,1
910 FOR I=1 TO 100:NEXT I:AS=1:GOSUB 210
920 RETURN
930 P=0:D=0:REM ** THEME SONG AULD LANGE SYNE **
940 SOUND 2,638,60,5:SOUND 2,478,85,5:SOUND 2,0,5,5:SOUND 2,478,25,5:
SOUND 2,0,5,5
950 IF AB=1 THEN RETURN
960 SOUND 2,478,60,5:SOUND 2,379,60,5:SOUND 2,426,90,5:SOUND 2,478,30
,5:SOUND 2,426,60,5
970 IF AB=2 THEN RETURN
980 SOUND 2,379,60,5:SOUND 2,478,85,5:SOUND 2,0,5,5:SOUND 2,478,30,5:
SOUND 2,379,60,5:SOUND 2,319,60,5
990 IF AB=3 THEN RETURN
1000 SOUND 2,284,175,5:SOUND 2,0,5,5:SOUND 2,284,60,5:SOUND 2,319,90,5
SOUND 2,379,25,5:SOUND 2,0,5,5:SOUND 2,379,60,5:SOUND 2,478,60,5:SOUN
2,426,90,5:SOUND 2,478,30,5:SOUND 2,426,60,5
110 IF AB=4 THEN RETURN
120 SOUND 2,379,60,5:SOUND 2,478,90,5:SOUND 2,568,25,5:SOUND 2,0,5,5:
SOUND 2,568,60,5
130 SOUND 2,638,60,5:SOUND 2,478,120,5
140 IF AB=5 THEN RETURN
150 SOUND 2,5,379,5:SOUND 2,60,478,5:SOUND 2,60,426,5:SOUND 2,90,478,
SOUND 2,30,426,5:SOUND 2,60,0
160 DATA 379,60,478,90,568,25,0,5,568,60
170 DATA 638,60,478,120,-1,-1

```



Bytes and Pieces

More useful Machine Code routines from Sydney Brown

Here are some more machine code routines which should make things faster and easier in your own basic programs. Sharp eyes would have noticed a small error in Routine #5 last month - the last line should have commenced with C9 not 69. This is, of course, in hexadecimal and you should remember that all numbers to the left of the opcodes are in this format.

Routine #1

This routine accesses the firmware routine called KM WAIT KEY which will wait until a key is pressed and then return the character or expansion taken in the A register.

Here is the source code for the routine to call from basic.

```
D5 PUSH DE Save variable address  
 on stack
```

```
CD 18 BB CALL &BB18 Call KM WAIT KEY
```

```
E1 POP HL Retrieve variable  
 address
```

```
77 LD(HL),A Put value into variable
```

```
C9 RET Return to basic
```

As with two of the routines last month the value is returned and placed into an integer variable which must have previously been used.

Here is a typical example for those people who missed the previous routines.

```
10 A%-0
```

```
20 CODE-&4000
```

```
30 CALL CODE,@A%
```

```
40 PRINT A%
```

Remember before running this program you must place the data for the machine code into memory, set CODE to equal the start and move MEMORY to below the code.

Routines #2 and #3

These two routines will enable you to scroll vertically

[1] The whole screen.

[2] A section of the screen.

[1] To scroll the whole screen we will use the routine called SSCR HW ROLL which stands for Hardware Roll.

This means that the actual start address of the screen RAM is increased or decreased by 80 bytes so in effect the visual screen is moved over the RAM so the data does not have to be moved around in memory.

ROLL WHOLE SCREEN DOWN

```
3E C8 LD A,&C8 Load colour to fill  
 cleared line
```

```
06 00 LD B,0 Select down roll
```

```
CD 4D BC CALL &BC4D Call SCR HW ROLL
```

```
C9 RET Return to basic
```

The same routine can be used to roll the whole screen up, the only change is on the second line which must read:-

```
06 01 LD B,1 Select up roll
```

The colour loaded into A is actually the encoded ink, so you can fill the area with solid or striped colour.

An attempted full explanation of the method of encoding inks was made in the last issue of this magazine and is much too lengthy and complex to repeat but you could experiment by placing any value from 0 to 255 into A in the first line, instead of &C8.

[2] The second scroll routine allows you to scroll a character block area of the screen up or down.

The program is a bit longer than the last routine as you must set up the top, bottom, left and right edges of the area to scroll.

This time we are going to use the routine SCR SW ROLL which stands for SoftWare Roll.

A software roll is a bit slower than a hardware roll as all of the information on the screen, (16,384 bytes or memory positions) must be picked up one at a time and placed 80 bytes up or down through screen RAM.

SCROLL A SECTION OF THE SCREEN DOWN

```
3E C8 LD A,&C8 Load colour to fill  
 cleared line
```

```
06 00 LD B,0 Select down roll
```

```
26 05 LD H,5 Left character column
```

```
2E 03 LD L,3 Top character row
```

```
16 0C LD D,&C Right character column
```

```
1E 0F LD E,&0F Bottom character row
```

```
CD 50 BC CALL &BC50 Call SCR SW ROLL
```

```
C9 RET Return to basic
```

The colour encoding and roll direction are controlled exactly the same as the previous routine.

The co-ordinates start from 0,0, with 0,0 being the top left corner of the screen.

The area is set as follows:-

```
TOP-3 BOTTOM-15 RIGHT-12 LEFT-5  
or &03 or &0F or &0C or &05
```


Routine #4

This routine will allow you to change the screen offset address, which is the address of the start of the screen with reference to the start of screen memory, &C000 or 49152 in decimal.

```
26 00 LD H,0      Set HI byte of offset
2E 00 LD L,0      set LO byte of offset
CD 05 BC CALL &BC05 Call SCR set offset
C9      RET      Return to basic
```

The screen offset will only move in two byte blocks, (one mode 1 character width) and to a maximum of 16k-2 or 16382.

To split the offset into the required HI and LO bytes, use the following (e.g. OFF=the offset from zero).

```
HI=INT(OFF/256);LO=OFF-256*HI;POKE
LOW,LO;POKE HIGH,HI;CALL OFFSET
```

By experimenting you will notice that poking LOW with 2 will move the screen backwards by one character.

You will find things easier to work with if you experiment or use the routine after a MODE command as this resets the offset to 0.

A related effect can be produced by using the OUT command in basic. The output port 511 will set the screen offset but only in program run mode. Once the computer returns to direct mode the screen will automatically return to its previous position.

This small program will show you how the port works:

```
10 FOR W=0 TO 80:OUT 511,W:FOR WW=1 TO
35:NEXT:NEXT:FOR W=80 TO 0 STEP -1:OUT
511,W:FOR WW=1 TO 35:NEXT:NEXT:GOTO 10
```

Other Snippets

Out of interest, another useful out command is OUT 512,0 which turns the cassette OFF and OUT 512,255 which turns the cassette ON.

CALL &BC71 also turns the cassette OFF and CALL &BC6E turns the cassette ON.

Here are some useful FIRMWARE ROUTINES which you activate by simply calling the required address.

CALL &BB48 - will disarm break key while the program is running but will be automatically reset in direct mode or waiting on an input command.

CALL &BB4E - will totally reset the text screen to clear, set paper to ink0 and reset all inks reset to initial colours.

CALL &BB9C - will swap the current pen and paper inks. (Inverse character).

CALL &BBBBA - will reset the graphics screen and colours as &BB4E does for the text screen.

CALL &BBFF - resets screen to default mode as does switch on (mode 1).

CALL &BCA7 - will reset the sound chip and clear all sound queues and enable interrupts.

CALL &BCB6 - holds all sound output as is but retains all values.

CALL &BCB6 - allows sounds to continue after being held up by the previous routine.

CALL &BD19 - waits for vertical blank pulse used in animation without flicker.

CALL &BD37 - resets the original jumpblock values you have been experimenting with redirecting the call to your own routines.

You may find some of these memory positions useful which to experiment:

&B1C8 contains the current screen mode.
&B4F4 contains value for joystick 1.
&B4F1 contains value for joystick 2.
&B4E9 contains value for the initial key delay.
&B4EA contains value for the key repeat speed.
&B1FB enables colour flashing ON=255, OFF=0.

&B187 the four positions

&B188 used to store the value for

&B189 TIME and can be reset by

&B18A poking any of them with 0.

&B1D7 set the on/off times

&B1D8 for the flash rate.

&B1D9 to &B1FA contain the encoded inks for border and inks 0 to 15. (Have fun with these).

(continued from Page 13)

organising all services to provide the end of the year party. To protect all members interests an auditor will be appointed at the end of the club year to inspect and verify all club accounts.

WINDING UP

7. In the event that for any reason it is necessary to wind up the affairs of the club, a notice of a Special General Meeting shall be sent to all members. If at the meeting a resolution is carried by 51% or more of the members present to wind up the club then:

a. All assets of the club will be sold at auction.

b. All outstanding accounts will be paid.

c. All monies raised from the auction together with all residual club funds will be evenly divided amongst all members who were present and any member who notified the secretary by registered letter received prior to the Special meeting of their inability to attend.

INDEMNITY

8. No person being a member of the club committee shall be held responsible for the actions of any club member who wittingly commits an act that defies the articles and rules of the club. Such acts deemed to be contrary to the good spirit and aims of the club shall lead to expulsion from the club. Any such member may appeal in writing against the decision but will remain suspended until such appeal is decided upon by the committee.

Bytes and Pieces - the Basic code

```
10 MEMORY 32999:GOSUB 30000:a%=0
99 STOP
30000 REM MORE MC ROUTINES
30010 REM
30020 REM WAIT FOR & RETURN CHARACTER
30030 REM
30040 REM use with CALL WAITKEY,@A%
30050 REM After key press,character
      A% value now in
30060 REM
30070 WAITKEY=33040:m=0
30080 READ d:IF d<999 THEN POKE m+waitkey,d:m=m+1:GOTO 30080
30090 DATA &d5,&cd,&18,&bb,&e1,&77,&c9,999
30100 REM
30110 REM SCROLL RECTANGULAR AREA OF
30120 REM THE SCREEN UP OR DOWN.
30130 REM poke COL with ENCODED COLOUR
30140 REM poke UPDN with 0=DOWN 1=UP
30150 REM poke TOP with TOP ROW of area
30160 REM poke BOT with BOTTOM ROW
30170 REM poke LFT with LEFT COLUMN
30180 REM poke RIT with RIGHT COLUMN
30190 REM
30200 SCROLL=33050:COL=SCROLL+1:UPDN=SCROLL+3:LFT=SCROLL+5:TOP=SC
      ROLL+7:RIT=SCROLL+9:BOT=SCROLL+11:m=0
30210 READ d:IF d<999 THEN POKE m+SCROLL,d:m=m+1:GOTO 30210
30220 DATA &3e,&c8,&06,&00,&26,&05,&2e,&03,&16,&0c,&1e,&0f,&cd,&5
      0,&bc,&c9,999
30230 REM
30240 REM SCROLL WHOLE SCREEN UP OR DOWN
30250 REM
30260 REM Poke UPDN2 with 0=down 1=up
30270 REM Poke COL2 with encoded colour
30280 REM
30290 SCROLL2=33070:COL2=SCROLL2+1:UPDN2=SCROLL2+3:m=0
30300 READ d:IF d<999 THEN POKE m+SCROLL2,d:m=m+1:GOTO 30300
30310 DATA &3e,&c8,&06,&00,&cd,&4d,&bc,&c9,999
30320 REM
30330 REM SET SCREEN OFFSET FROM &C000
30340 REM
30350 REM poke LOW , low byte of offset
30360 REM poke HIGH, high byte of offset
30370 OFFSET=33080:HIGH=OFFSET+1:LOW=OFFSET+3:m=0
30380 READ d:IF d<999 THEN POKE m+OFFSET,d:m=m+1:GOTO 30380
30390 DATA &26,&00,&2e,&00,&cd,&05,&bc,&c9,999
30400 RETURN
```


Sound and Fury

A Demonstration from Peter Campbell

The program takes up the theme of simulating animation through sequential colour changing and applies it to an explosion, accompanied by an appropriate sound. However, the program is probably of more interest because of the way in which symmetry is used to draw the circles.

All routines used are labelled with explanatory headings. The first sets the initial values for the variables used. The second draws the circles. Note how lines 180-210 plot several points simultaneously. The third changes the colour in all pens (except number 1) to black so that the colours can be rotated by the fourth routine. The final routine generates the sound. Note the use of DN SQ(2) GOSUB to replenish the sound channel. The sound is called by RELEASE 2 in line 360 and kept in synchronization by the loop containing the variable 'waits'.

How It Works:

Lines 10-110:

Set up the initial values of the variables used.

Lines 120-220:

Draw a series of 15 concentric circles.

Lines 230-240:

Change the colour of inks 2 to 15 to black.

Lines 250-350:

Rotate the ink colours to create the illusion of animation.

Line 360:

Releases the sound and synchronizes it with the action.

Lines 370-380:

Ends the program when "e" is hit

Lines 400-440:

Generate the sound. Note that the subroutine is only called if there is a vacancy in the sound queue.

Variables Used:

a-b

Variables which convert the expressions $r(\text{adius}) * \sin(j)$ and $r(\text{adius}) * \cos(j)$ to a form that can be used to plot eight points at the same time.

i-j

Control variables. "i" is used for ink colour, "j" for the angle being plotted.

s

String variable used in detecting if "e" has been pressed.

x-y

Randomly generated values used in ink colour rotation.

```
10 REM ***** REM
20 REM *** Sound and Fury *** REM
30 REM *** by P N Campbell *** REM
40 REM ***** REM
50 ,
60 REM *** Initial parameters ***
70 ,
80 DEFINT i-j,r,x-y:DEFSTR s
90 MODE 0:INK 1,13:INK 0,0:BORDER 0
100 x=1:waits=120:DEG
110 ORIGIN 320,200
120 ,
130 REM *** Draw circles ***
140 ,
150 FOR i=1 TO 15
160 FOR j=2 TO 44 STEP 0.5*i+1
170 r=12*i:a=r*COS(j):b=r*SIN(j)
180 PLOT a,b,i:PLOT -a,b,i
190 PLOT a,-b,i:PLOT -a,-b,i
200 PLOT b,a,i:PLOT -b,a,i
210 PLOT b,-a,i:PLOT -b,-a,i
220 NEXT:NEXT
230 ,
240 REM *** All pens black bar ! ***
250 ,
260 FOR i=2 TO 15:INK i,0:NEXT
270 ,
280 REM *** Explosions ***
290 ,
300 INK x,26:ON SQ(2) GOSUB 400
310 y=x-RND*5:k=RND*17+1
320 IF y<1 THEN y=y+15
330 IF x=1 THEN INK 15,0 ELSE INK x-1,0
340 IF y=1 THEN INK 15,k ELSE INK y-1,k
350 x=x+1:IF x=16 THEN x=1:CALL &B019
360 RELEASE 2:FOR i = 1 TO waits:NEXT
370 s=:!KEY$:s=LOWER$(s)
380 IF s="e" THEN END ELSE 300
390 ,
400 REM *** Sound generation ***
410 ,
420 ENV 1,1,24,1,9,-1,1,14,-1,10
430 SOUND 66,0,0,0,1,0,21
440 RETURN
```


Basic, Pascal and Fortran

An insight from Arthur Harris

The three main languages available for microcomputers are BASIC, Pascal and Fortran. Other languages available include Cobol and Forth. In this diatribe, I will discuss the first three, primarily because they are the ones that I know, but also because Cobol will not, I believe, ever be a force in the microcomputer world and Forth, although gaining in popularity, has not achieved any real standing yet.

BASIC is by far the most common language on micros. Why it was chosen in the first place has never been publicised. It was developed in about 1968, at Dartmouth College in the USA. It is an acronym for Beginners All-purpose Symbolic Instruction Code. As implied by its name, it suits beginners in programming and is easy to write. Every line must be preceded by a line number. Its simplicity as a programming language leads to the fact that it is not conducive to producing programs that are properly structured. By applying a little self-discipline, it is very easy to produce a structured program.

Being mainly implemented as an interpreted language, it runs comparatively slowly. I will explain the implications of interpreted versus compiled languages and the pros and cons of each in a future article.

BASIC is rich in numeric calculation functions and has excellent inbuilt string handling ability. In a disk-based system, with a good Disk Operating system (DOS), it usually has good file-handling capabilities. For example, under Newdos 80 on the TRS-80, there are 5 separate file structures.

The main drawback to BASIC is the fact that, during its existence, a large number of dialects have come into

being. This does not allow a program written on one machine to be run on all machines with BASIC. This feature is known as transportability.

The definition of variable names differs widely between dialects, from the PC-1211 where you are limited to a single letter, to some versions where variable names may be up to 32 characters long and all characters are significant. Under interpreted Microsoft BASIC, variable names may be any length, but only the first two characters are significant. This means that COW and COST are interpreted as the same variable, since the first two letters are the same. The restriction to two characters is not really a restriction, since this still allows at least 3828 variable names. It is not necessary to define variables in BASIC. Some dialects do not allow this option. Others allow the definition of variables as string, integer, single precision and double precision, with the ability to over-ride this definition with specific declarators.

Input/Output is easily formatted and may be neglected during the early stages of programming. Comprehensive formatting is provided and allows 'pretty' output that is easy to read.

Pascal was developed as a teaching language, in the early 1978's by Nicklaus Wirth. The strong points of Pascal include the fact that it is necessary to use structure in producing programs. It was the first language to combine structure of program with structure of data. Another strong feature of Pascal is the ability to define data types. This helps in error trapping during input. Yet another feature of this language is the fact that it incorporates pointers in the data types for files.

Pascal supports "long" variable names. All variables MUST be declared at the beginning of the program. Variables can be both global and local. A variable declared at the beginning of a program is a global variable and one declared at the beginning of a procedure is a local variable. Local variables may have the same names as global variables. This vastly increases the number of variables available during the writing of a program.

The language contains most, if not all, of the functions necessary to carry out numeric calculations. The basic language is a little weak in string handling, but contains the tools to allow procedures to be written to achieve the most sophisticated string handling. File handling capabilities are excellent. The flexibility of the language is such that any type of file handling that is not provided, can be overcome by writing a procedure. The use of pointers assists greatly in file handling. The record structure is much improved by the use of variant records to handle the occasional record that differs from the main type of record.

A procedure roughly equates to a subroutine in BASIC. Procedures may be easily incorporated into a program. Since program line numbering is not used, it is not necessary to renumber lines to add a procedure. Also the fact that variables declared within a procedure are local, means that procedures are fully transportable and create no problems when added to a different program.

The strength of Pascal used to be that it was a standard language. In the last few years a few dialects have begun to appear. This is a shame as it means that programs are not fully transportable

slower.

machine to machine. If programs are written in the basic version of Pascal they will still run on a machine that has a version with enhancements, but it means that that program does not make use of the full power of the version in the recipient machine.

Input/output is rigidly controlled but can be easily formatted. Procedures need to be written to handle string input of any sophistication. The ability to define data types leads to a very powerful feature. It is possible to define a set and perform manipulation of that set with another set. Mathematicians can pass very easily test and develop set theories. No other language that I know allows this sort of operation so easily. Almost anything that can be done in Pascal can be done in BASIC, but it may require some sneaky and sophisticated programming to achieve the result in BASIC.

All versions of Pascal are compiled. Some versions compile into a pseudo-code and others compile into "native" code. Those that compile into pseudo-code are the most transportable, since the particular machine must provide an interpreter program to run the compiled version. The other benefit of pseudo-code is that compilation is fast, although run times are somewhat

Fortran (the FORMula TRANslation language) was written to facilitate computations by scientists and engineers. It is very strong in calculation functions and very efficient in the storage of numbers within memory. String handling is a very weak aspect of Fortran as it is almost non-existent. There are a comparatively small number of versions of Fortran and all are upward compatible. That is, programs written in one version of Fortran will run on an enhanced version, again without realising the full power of the enhanced version.

File handling is very rigid but adequate. Input/output must be strictly formatted and is defined in format statements. These are numbered and may be used repeatedly by reference to their number.

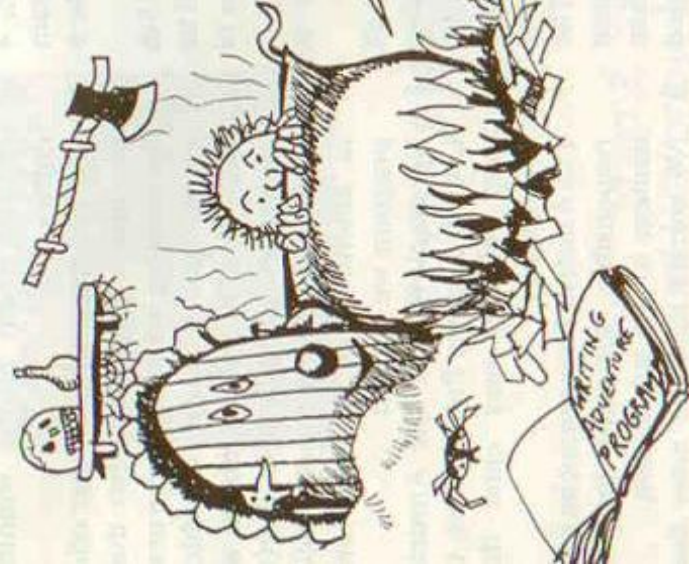
Subroutines can be written to improve the facilities of the language. Function subroutines are the equivalent of the DEF FN statement in BASIC but allow much more complicated calculations to be carried out. Lack of use of line numbering results in the easy inclusion of subroutines and function subroutines into any program. Variables used within a subroutine are local in nature which adds to the transportability of

these subroutines.

Variable names may be of the "long" variety allowing a large number to be used in a program. The convention is that variables beginning with the letters I to N, inclusive, are integer variables. Other variables are real number or string variables. This convention may be over-ridden by a declaration statement at the beginning of the program.

All versions of Fortran are fully compiled into "native" code. This limits the transportability of programs but means the fastest possible run times. Usually no run-time libraries are required, as the routines are added into the compiled program. The size of the compiled program is very dependent on the number of routines required and how often a particular routine is used within the program. These are machine language routines that perform the functions (such as SIN, ABS, +, etc.) used in the program.

This has not been intended as a definitive description of any of the languages covered, as that would require a book of considerable length. I hope that it has given some insight into the strengths and weaknesses of some of the languages that are available on computers.



Clinic At Carnegie

On 5th September, a representative of AWA-Thorn Consumer products will be on hand at the Billy Guyatts store at Carnegie (Victoria) to answer any questions concerning the availability and use of software marketed by AWA. Local users should find this useful in resolving their problems.

Starcomm Systems

The telephone number of Starcomm Systems printed in last month's issue (July 1985) in the article 'Gencom - a CP/M Utility' was incorrect. The correct number is (02) 671 2929.

•COMPETITION•

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

Class 1

Best overall program
Wins an AWA Video Recorder

Class 2

Best amusement/adventure
Wins a new DD1 disk drive

Class 3

Best educational software
Wins a new DD1 disk drive

Class 4

Best business software
Wins a new DD1 disk drive

How to enter

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

- 1 A brief summary of the program in 500 words or less.
 - 2 A clear program listing if available.
 - 3 A stamped, self addressed envelope of adequate dimensions if you would like your entry returned.
 - 4 Your name and address.
- You may make as many submissions as you want, but no entrant may win more than one prize.

Conditions of Entry

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

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

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

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

The Amstrad User cannot be held responsible for any loss or damage to any submission. No entrant may win more than one prize.

NEW GAMES

AVAILABLE



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 4 - May
Number 2 - Mar	Number 5 - June
Number 3 - Apr	Number 6 - July

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

Please send your order and remittance to:

The Amstrad User, Shop 2, 33 The Centreway,
Mt. Waverley, Victoria 3149 Tel: (03) 232 7055
(Bankcard and MasterCard accepted)



Three Adventure Games for just \$39.95

Enquiries to:

Strategy Software

Shop 2, 33 The Centreway, Blackburn Road

Mount Waverley, Victoria, 3149 Telephone: (03) 232 7055

SAVE \$450

Q. A. Q. A. Q. A.

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

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

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

There is.

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

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

By completing and returning this order.

Please send me **THE AMSTRAD USER** for 12 months

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

Payment by: Cheque Bankcard or Mastercard

Card number _____ Expiry date

Name Phone

Address

..... Postcode

Signed **Please start with Issue No**

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

(OVERSEAS PRICES ON APPLICATION TO ABOVE ADDRESS)