THE AMSTRAD USER

Issue No. 18 $3.50 April 1986

- *LAST CHANCE TO WIN A RADIO/CASSETTE PLAYER*
- *GREAT GAME OF YAHTZEE FOR THE AMSTRAD*
- *CPC6128 RSX COMMANDS - MORE ON CP/M*
- *USER GROUP INFORMATION*

FOR THE NOVICE & EXPERIENCED USER

Registered by Australia Post – Publication No VBP7017

THE AMSTRAD USER

Issue No. 15
April 1986

For Tape Subscribers, the programs can be found at these approximate positions:
Side 1: JJPIC1- 10, JJPIC2- 20, LOCODEM- 30, AMSFILE3- 40, LPVII- 62
Side 2: YAHTZEE- 23

All enquiries and contacts concerning this Publication should be made to The Amstrad User, Suite 1, 33 The Centreway, Blackburn Road, Mt. Waverley, Victoria 3149, Australia. [Telephone: (03) 232 7055].

The Amstrad User is published each month 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.

In the November 1985 Editorial I made a comment about the number of queries we receive at this office from users having programming problems with their Amstrads. It would seem that the queries rise in direct proportion to the number of machines and magazines sold, and it has now got to the point where we just cannot answer them all. I am sure you will all understand that entering into personal correspondence (or phone calls) is time consuming and detracts from the main function of publishing The Amstrad User. Of course, letters of relevance to the magazine are, and always will be welcome. So, from now on we reluctantly shut the door on other queries demanding a personal reply.

After that rather negative start, you may be interested to know that a number of 464 owners who already have a DDI-1 are taking advantage of our Disc Drive offer to buy a second drive (instead of the FD-1). Naturally, they are buying the cable, discs and manual again but the price advantage outweighs the duplication.

This month's magazine is pretty well packed with something for everyone. For those suffering withdrawal symptoms, Tony Blakemore provides the third part of Amsfile and for the 'hackers', six pages of Yahtzee to key in - and a good version it is too.

And for DMP-1 (or similar) printer owners who are fed up with the lower case descender problem, Petr Lukes provides a solution. There again, you could get a DMP-2000 which will be the subject of a review in the May issue of The Amstrad User.

See you next month,

Ed

Letters

All correspondence published in this section earns a payment of five dollars.
Letters should be addressed to The Editor, The Amstrad User, Suite 1, 33 The Centreway, Mt. Waverley, Victoria, 3149.

Readers may recall my letter published in your December issue asking for the appropriate POKE command to operate the CAPS LOCK from within Basic on a 6128.

A fellow member of the Canberra User Group suggested dumping the memory contents with CAPS LOCK on and off respectively and comparing the results. After considering this for a while, I wrote a Basic program to store in an array the contents of memory locations B100(hex) to C000(hex). I then made the program pause so I could press the CAPS LOCK key, and then made it store the contents of the same memory locations in a second array.

Finally, I made the program print out the contents of the memory locations which had changed their values. Since there were quite a number, I looked only for those that had changed from 0 to 255 and vice versa. Having found two, experimentation showed that POKE 46642,255 forces upper case input on the 6128 (irrespective of the CAPS LOCK condition beforehand). Use 0 instead of 255 for lower case.

Neale Yardley, McKellar, ACT

Would you believe that we were going to publish the answer to your query in this month's issue? It was kindly phoned in by Steve Miles of Laverton. Never mind - you probably learnt more by experimenting yourself!

Recently I purchased an LP-1 Light Pen, marketed under the Amstrad banner by Trojan Products for my Amstrad CPC664. Although pleased with the quality of the hardware, I encountered several problems with the software about which you might like to inform your readers.

The first problem encountered was when I tried to carry out the instructions on converting the program from cassette to disc (page 12 User Instructions). Each time a "read fail" error occurred when I tried to save the program to disc. I subsequently tried to carry out this same operation on a CPC464 with DDI-1 disc drive, a CPC6128 and my own CPC664. In each case the program loaded and ran correctly but did not save to disc. This problem repeated itself when I endeavoured to save the screen to disc rather than the tape.

The second problem occurred when I tried to output the screen to my Amstrad recommended Seikosha SP 1000A dot matrix printer. Only one line of unintelligible characters were output before the program crashed and the computer had to be reset.

The last problem concerns the instructions given at the start of the program. They state "for menu selection etc., the routine at lines 30000 onwards should be removed and used with a small movable machine code routine that is positioned at 40000 to 40074." Upon listing the program, the last line is 40000, which is the address where the User Instructions state that execution is to occur for saving the programme to disc.

Perhaps one of your readers may be able to render some assistance in this matter, or you may be able to publish a review in the future, as this highly useful tool is severely hampered by software written too quickly.

D.C. MacKinnon, Unanderra, NSW

I subscribe to your magazine and I have heard and read that you use a Seikosha SP-1000A. I recently bought one in Perth and to my horror I found that when it listed anything it did double line spacing. I have read that people with other makes of printers have had to either cut wires or sticky tape over them. Surely AWA who distribute the machine should be able to overcome this?

My next problem is trying to understand the dip switch settings on Page 22 of the manual. As I see it, where it says Switch 2-3 you are supposed to push the 2 switch down in the set of four and the 3 switch down in the set of eight. Am I right in my assumption?

My third problem relates to the commands they give to address certain modes, in this case italic pica characters. I cannot make head nor tail of them.

Stephen Snow, Albany, WA

If your dealer was worth his salt, he should be able to answer the questions for you!

The line feed/carriage return problem may be due to you being supplied with an incorrect printer cable. The cable supplied with our printer has not caused this problem.

You seem to be confused with the dip switch settings. There are two switches - switch one having 8 on/off settings and switch two having 4 on/off settings. A new machine should always have all the settings of both switches in the off position (ie. switched down) - if they are not, then it is possible that someone has previously tampered with them.

To make sure that the correct language font is set (eg. to get a # instead of a £), the first three setting on switch one have to be on - in the up position. This is denoted by "set 1-1, 1-2 and 1-3". Once these are set it should not be necessary to touch them again.

To print in italics you need to set switch 2-1, that is move the first setting up on the second switch.

The printer will not recognise any manual setting changes if they are made while it is powered on, so you may need to power it off and on again to invoke the initialisation phase which reads the dip switch settings.

The following program is a printer utility that will make life easier for people that are annoyed with entering printer codes. These codes are written for the Super 5 EN-P1090 but can quite easily be converted to the printer in question.

Different fonts can be mixed because the program loops back, so the code selected is retained inthe buffer. To select a function, type the group of letters in brackets following the operation. Once the options have been selected, type the abort option and start printing away.

P. Mezzavia, Geelong, Vic

```
10 MODE 2
20 CLS
30 BORDER 4
40 LOCATE 25,1:PRINT " EN-P1090 COMMANDS "
50 LOCATE 5,3:PRINT "A]  DOUBLE WIDTH ELONGATED FONT (DOUBW)
60 LOCATE 5,4:PRINT "B]  COMPRESSED CHARACTER FONT (COMPCHAR)"
80 LOCATE 5,5:PRINT "C]  PICA (10 chrs. PER/INCH) FONT (PIC)"
90 LOCATE 5,6:PRINT "D]  ELITE (12 chrs. PER/INCH) FONT (ELT)"
100 LOCATE 5,7:PRINT "E]  EMPHASIZED FONT (NEAR LETTER QUALITY) (EMPH)
110 LOCATE 5,8:PRINT "F]  DOUBLE STRIKE FONT (DOUBST)"
120 LOCATE 5,9:PRINT "G]  UNDERLINE DESIGNATION (UND)"
130 LOCATE 5,10:PRINT "H]  SUPERSCRIPT DESIGNATION (SUPSC)"
140 LOCATE 5,11:PRINT "I]  SUBSCRIPT DESIGNATION (SUBSC)"
150 LOCATE 5,12:PRINT "J]  ITALIC FONT (ITAF)"
160 LOCATE 5,13:PRINT "K]  RESET PRINTER (RESPRINT)"
170 LOCATE 5,14:PRINT "  RELEASE COMMANDS  "
180 LOCATE 5,15:PRINT "L]  DOUBLE STRIKE RELEASE (DOUBSTR)"
190 LOCATE 5,16:PRINT "M]  UNDERLINE RELEASE (UNDR)"
200 LOCATE 5,17:PRINT "N]  EMPHASIZED RELEASE (EMPHR)"
210 LOCATE 5,18:PRINT "O]  SUB/SUPERSCRIPT RELEASE (SCRIPTRE)"
220 LOCATE 5,19:PRINT "P]  DOUBLE WIDTH RELEASE (DOUBWR)"
230 LOCATE 5,20:PRINT "Q]  ITALIC RELEASE (ITAFR)"
231 LOCATE 5,21:PRINT "R]  COMPRESSED CHARACTER RELEASE (COMPCHARE)"
232 LOCATE 5,22:PRINT "S]  ABORT PROGRAM (ABORT)"
240 LOCATE 1,24:INPUT "ENTER OPTION (TYPE BRACKET VALUE)";A$
250 IF A$="DOUBW" THEN 260 ELSE 270
260 PRINT #8,CHR$(27)+CHR$(87)+CHR$(1):GOTO 10
270 IF A$="COMPCHAR" THEN 280 ELSE 290
280 PRINT #8,CHR$(15):GOTO 10
290 IF A$="PIC" THEN 300 ELSE 310
300 PRINT #8,CHR$(27)+CHR$(80)+CHR$(1):GOTO 10
310 IF A$="ELT" THEN 320 ELSE 330
320 PRINT #8,CHR$(27)+CHR$(80)+CHR$(0):GOTO 10
330 IF A$="EMPH" THEN 340 ELSE 350
340 PRINT #8,CHR$(27)+CHR$(69):GOTO 10
350 IF A$="DOUBST" THEN 360 ELSE 370
360 PRINT #8,CHR$(27)+CHR$(71):GOTO 10
370 IF A$="UND" THEN 380 ELSE 390
380 PRINT #8,CHR$(27)+CHR$(45)+CHR$(1):GOTO 10
390 IF A$="SUPSC" THEN 400 ELSE 410
400 PRINT #8,CHR$(27)+CHR$(83)+CHR$(0):GOTO 10
410 IF A$="SUBSC" THEN 420 ELSE 430
420 PRINT #8,CHR$(27)+CHR$(83)+CHR$(1):GOTO 10
430 IF A$="ITAF" THEN 440 ELSE 450
440 PRINT #8,CHR$(27)+CHR$(52):GOTO 10
450 IF A$="RESPRINT" THEN 460 ELSE 470
460 PRINT #8,CHR$(27)+CHR$(64)
470 IF A$="DOUBSTR" THEN 480 ELSE 490
480 PRINT #8,CHR$(27)+CHR$(72):GOTO 10
490 IF A$="UNDR" THEN 500 ELSE 510
500 PRINT #8,CHR$(27)+CHR$(45)+CHR$(0):GOTO 10
510 IF A$="EMPHR" THEN 520 ELSE 530
520 PRINT #8,CHR$(27)+CHR$(70):GOTO 10
530 IF A$="SCRIPTRE" THEN 540 ELSE 550
540 PRINT #8,CHR$(27)+CHR$(84):GOTO 10
550 IF A$="DOUBWR" THEN 560 ELSE 570
560 PRINT #8,CHR$(27)+CHR$(87)+CHR$(0)
570 IF A$="ITAFR" THEN 580 ELSE 590
580 PRINT #8,CHR$(27)+CHR$(53):GOTO 10
590 IF A$="COMPCHARE" THEN 600 ELSE 610
600 PRINT #8,CHR$(18):GOTO 10
610 IF A$="ABORT" THEN END
```

The Learning Centre

CP/M Explored - Part Two

by Shane Kelly

Following on from the last article we are now going to investigate how CP/M communicates with the outside world. To do this we must first become familiar with a program supplied with our CP/M master disk called STAT.COM.

Insert the backup disc we made last time into drive A: and invoke CP/M with the |CP/M command from Basic. Do a directory of drive A: and ensure that STAT.COM. is there. (If not, copy it across). Now type:-

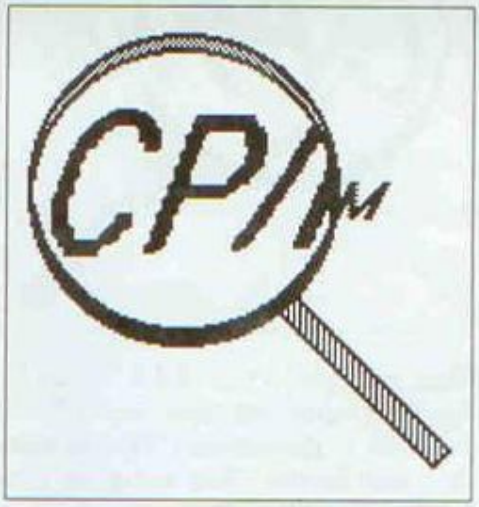
```
STAT <RETURN>
```

(In future, the return will not be shown, but it is implied)

After a bit of disc activity the display responds with:-

```
A: R/W, SPACE: <xx>K
```

(where XX is some decimal number)



Taking this information in turn we are told that we are looking at Drive A: and it is a read/write drive with <xx>K bytes remaining to store files. We discussed the meaning of 'logged drive' (or default drive) in the last article, but what does a read/write drive mean? Well, you know by your study of the DDI-1 manual that it is possible to make a disc 'read only' and so it is with a drive. If you have been experimenting with CP/M you will know that when a disc is changed you must log it in using control C or else you get the message

```
BDOS ERROR ON A:R/O
```

meaning that this drive is now read only and so protects your files from accidental overwriting by any active program should you not realise that you have changed discs. So it is with STAT.COM. you may deliberately set a drive to read only status so that any program you are working on does not try and overwrite your files on that disk. You set a drive to read only by typing:-

```
A>STAT A:=R/O
```

Note that this protection is only temporary and will be cancelled by the next warm start (CONTROL C).

STAT is a fairly versatile program and is capable of giving you all sorts of information about your disc drives and your files on those drives.

Try the following and note the results. (If you have a printer use control P to get a hardcopy of the following).

```
STAT A:
STAT A:*.COM
STAT A:*.*
```

The process goes like this:- STAT is invoked from the logged disc drive and processes the remaining commands to it. It then prints out on the screen (and printer if you used control P) all the information it can about those files. You will see that the column headings for the last two are somewhat cryptic. From left to right they mean records

(to CP/M a record is a standard unit of 128 bytes), Bytes (in 1k lots), extents (an extent is the amount of space controlled by one directory entry) and ACC stands for access attributes and, as you will realise, the R/W in this column stands for read/write. Finally we have the drive code and the filename. Without going far deeper into the workings of the CP/M than is within the scope of these articles, I will leave the explanation of file storage at that, although you will realise that there is far more to it than I have told you.

Now type the following:

```
STAT VAL:
```

The display looks like this:-

```
TEMP R/O DISK:D:=R/O
SETINDICATOR:D:FILENAME.TYP
             $R/O $R/W $SYS
             $DIR
DISK STATUS: DSK: D:DSK:
USER STATUS: USR:
IOBYTE ASSIGN:
CON: = TTY: CRT: BAT: UC1:
RDR: = TTY: PTR: UR1: UR2:
PUN: = TTY: PTP: UP1: UP2:
LST: = TTY: CRT: LPT: UL1:
```

Taking them from the top, we have the command to set a drive to temporary 'read only', as discussed above. The next item (set indicator) allows us to set our files to two different pairs of access attributes (refer to the fourth column of the printout obtained with STAT *.* or STAT *.COM). We can set any file or files to read only or read write status using the $R/O AND $R/W parameters. For instance, to set all .COM files to read only status, type this:-

```
A>STAT *.COM $R/O
```

Note that no indication is given to tell you of the success or otherwise of this action. You can reverse the above by using the $R/W parameter. When you save a file they are automatically created with the read/write attribute set so this is the default attribute. The next pair are $SYS and $DIR. To set a file to $SYS status is to make the file invisible to an ordinary directory listing. Try this:-

```
A>STAT AMSDOS.COM $SYS
```

Now do a directory of drive A: AMDOS.COM is not there. It is, to all directory listings, invisible. Now try this:-

```
A>STAT AMSDOS.COM
```

You will get the usual columns of information, but note that AMDOS.COM is now in brackets. This is STATS way of telling us that the file has been set to SYSTEM STATUS and will not be listed in the directory. The file has not been erased and you may check this by typing (after the A> PROMPT) AMSDOS. You will be unceremoniously dumped

back to Basic.) From this it should be apparent that although the file is invisible to the directory it is still there and can be run. This being so then what use is the $SYS attribute? Say for instance you had a disk that was used with a word processor. You may have many small files on it that are standard paragraphs for inclusion in other documents. If these are all set to $SYS STATUS then they will not clutter up the directory listings, but will still be available for use. The $DIR parameter is, as you will have guessed, the reverse of the above.

The next item is the way we check the drive characteristics of the disk drives attatched to our system. Type this:-

```
A>STAT A:DSK:
```

The resulting display is the information required by CP/M to write (and read) files on the disc. To explain them all would be a major work in itself and I will not go into it here but will recommend that you obtain a text book on CP/M that will go into more detail.

The next item gives us information about the current user code and the number of active files in that user code. User codes were discussed in the last article and are not really of much use in the small capacity drives that are on the standard Amstrad 464/664.

Now we come to the meat of this month's article - the IOBYTE assignment. To understand the IOBYTE concept it is necessary to delve a little deeper into the BDOS. The BDOS (Basic Disk Operating System) handles 'requests' from application programs (and the CCP) in a standard manner in all CP/M systems. (Remember - only the BIOS is rewritten for each different computer). A program may 'request' from BDOS a service to open a file, write to a file, close a file, print something on the console, find a directory entry or any of another 30-DDD services that BDOS performs. If the 'request' involves input or output to anything other than a disc file, the IOBYTE is consulted to see where the data should be directed. This is made possible because the IOBYTE is not treated as a byte, but as a bit field assignment of logical devices to physical devices. Lets define our logical devices first.

Look at the leftmost column of the printout under the heading IOBYTE assignment. The CON: RDR: PUN: and LST: are the logical devices that CP/M supports. Taken one by one, the breakdown is this:-

CON: is the logical device that functions as operator input/output or is the console function.

RDR: performs the function of input (from anything).

PUN: performs the output function (to anything).

LST: performs the listing (or printing) function.

What do all those things have in common? Yes that's right they are all functions, i.e. they are the logical functions that a computer carries out. Think about it. Console function takes commands and returns the results. RDR: (short for reader) receives information. PUN: (short for punch) transmits information. LST: (short for list but not very!) performs the archive function of storing output in (relatively) non-perishable form. The point is this:- those logical functions neither know nor care whether they are talking to a printer, modem, speech synthesiser, light pen, graphics tablet or your Aunt Mable. They are just not interested in the actual physical device with which they are communicating. That should give you a fair idea of the definition of the physical devices. That's right - anything at all you care to hang on the computer. You will note that these are only a subset of the possible physical devices allowed at any one time. They are given the names that appear to the right of the logical device names and I will expand on them now.

TTY: (TELETYPWRITER) a general purpose physical device that usually sends and receives data.

CRT: (CATHODE RAY TUBE) usually your keyboard and monitor.

BAT: (BATCH) used for batch processing but not generally implemented in most CP/M systems which usually use SUBMIT.COM and XSUB.COM for this purpose.

UC1:(?) usually another device capable fo performing the console function.

PTR: (PAPER TAPE READER) a general purpose serial input.

UR1: & UR2:(?) other serial input devices.

PTP: (PAPER TAPE PUNCH) a general purpose serial output.

UP1: & UP2:(?) other serial output devices.

LPT: (LINE PRINTER) usually the main printer device.

UL1:(?) another printer or serial output.

Right, now lets talk specifics. The 464 has a printer port (centronics) and you would rightly assume that this is the LPT: physical device. The 464 also has a screen and keyboard and again you would be right in assuming that these two together are the CRT: physical device. In addition, the BIOS also supports a serial interface comprising two channels. The physical ports are not on the Amstrad but the code is there in the BIOS to drive them. These are given the physical device names of TTY: for special I/O device zero. This is a two channel device that is capable of sending and receiving data. Special I/O device one is assigned the following names

UC1:,UR1:,UP1: and UL1: does this pose the question in your mind that is seems a little strange to give a physical device four out of a limited number of 12 names? Well, the answer is in the printout of the IOBYTE assignment.

You can see that some physical devices are only allowed to be assigned to certain logical devices. By assigning these four physical device names to only one actual device we can therefore assign that actual device to any of the logical devices we choose. Note that because special I/O device zero is given the name TTY:, we can also assign it to any logical device.

I realise that this has been quite a mouthful to swallow, but an example may help to clarify the situation. Assume the following hardware configuration:- 464, disc drive, daisywheel printer (attached to printer interface) and dot matrix printer (attached to serial interface). The normal printer to be used is the dot matrix because of it's speed and 16k print buffer. This presents a problem as the normal IOBYTE assignment is for the printer port to be known as LPT: or Primary Listing Device and if we type the following you will see that this is the case:-

```
A>STAT DEV:
```

You will see the default IOBYTE assignment for the standard system:-

```
CON: IS CRT:
RDR: IS TTY:
PUN: IS TTY:
LST: IS LPT:
```

This means that all listing is directed to the daisywheel printer and is not what we want. How do we change the IOBYTE setting? The easiest method is to simply tell STAT what we want! Type this:-

`A>STAT LST:=TTY:` (assign logical list device to serial interface)

`A>STAT DEV:` (show us the current IOBYTE setting)

```
CON: IS CRT:
RDR: IS TTY:
PUN: IS TTY:
LST: IS TTY:
```

As you can see, the default list device is now the TTY: physical device which is actually the serial interface. This is what we wanted and now all printing will take place on our dot matrix printer. To get a proof copy on the daisywheel we just reverse the assignment by typing:-

```
A>STAT  LST:=LPT:
```

and our printing will be done on the

daisywheel. Most of us will not have the serial interface connected but we can try out the following anyway, as the logical device does not care whether the serial interface is connected or not. Assign the LST: device as above i.e. to the TTY: physical device, and type this after the A> PROMPT

```
A>^P(control P);NOW IS THE
WINTER OF OUR DISCONTENT
<RETURN>
A>^P
```

After a short wait the A> PROMPT returns and nothing is printed out on the printer because it is no longer the assigned listing device.

As the final part of this article I will give you the physical names of the devices as Amstrad have assigned them on the 464.

```
TTY: SPECIAL I/O DEVICE ZERO
CRT: KEYBOARD AND SCREEN
BAT: INPUT FROM RDR: AND
     OUTPUT TO LST:
UC1: SPECIAL I/O DEVICE ONE
PTR: NULL INPUT (ACTUALLY END
     OF FILE INPUT)
UR1: SPECIAL I/O DEVICE ONE
UR2: KEYBOARD ONLY
PTP: NULL OUTPUT
UP1: SPECIAL I/O DEVICE ONE
UP2: SCREEN ONLY
LPT: CENTRONICS PORT
UL1: SPECIAL I/O DEVICE ONE
```

As usual, experimentation is the greatest aid to learning. Please play around with the device assignments as its possible only to scratch the surface in a small article such as this. Further reading is essential to greater understanding of the way CP/M communicates with the outside world.

Finally, I would welcome any comments from you on the contents of these two articles so far and any constructive criticism would also be welcomed.

(Not a bad idea - and if you feel that there is a particular area that should be covered, drop me a line - Ed.)

THE AMSTRAD USER HALL OF FAME

GAME	SCORE/TIME	ACHIEVER
Battle for Midway	8 carriers:speed 1:level 3	Steve Alatakis
Chuckie Egg	395960/45 mins	Tony Barberi
Codename Mat	870/45 mins	Gill Cherry
Combat Lynx	81450/no time specified	Steve Alatakis
Decathalon	331840/110 mins	John Farquhar
Gilligan's Gold	107403/9.75 mins	Alex Smyth
Harrier Attack	207550/10 mins	Dean Stibbe
Haunted Hedges	20110/11 mins	Samuel Yim
Hunter Killer	17/67 mins	Chris Catalfamo
Knight Lore	98%/44 mins	Umut Akcelik
Minder	$17749/no time specified	Steve Alatakis
Moonbuggy	152400/26.75 mins	Alex Smyth
Roland in the Caves	79884/4 mins	Emma Poynton
Roland goes Digging	$616.35/30 mins	Chris Catalfamo
Roland on the Ropes	736900/92 mins	Allison Pilbeam
Roland in Time	72/18 mins	Paul Azzopardi
Sorcery	91500/14 mins	Mike Nicolai
Sorcery +	126259/40.5 mins	John Evers
Star Commando	193810/133 mins	Alex Smyth
Survivor	223160/19.5 mins	Alex Smyth
Way of Exploding Fist	295600:10th Dan/41 mins	R. Schneider
Wild Bunch	10539/no time specified	Steve Alatakis
Yie Ar Kung Fu	445040:level 20/30 mins	Andrew Portbury
3-D Monster Chase	1320:7 keys/7 mins	Adam Broadway

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, along with your tips for playing the game to:
Amstrad Achievers, The Amstrad User, Suite 1, 33 The Centreway,
Blackburn Road, Mt. Waverley, Victoria 3149.

Junior Jotters

Now that the holiday period is well and truly over, Junior Jotters returns to interfere with another term of homework assignments! It shouldn't, of course, so when you have finished your allotted tasks, why not put something together for this column and earn a little in the process.

SPECIAL EFFECTS (1)
by David Fennessy

For those J.J's who like graphic demonstrations, I enclose a short program which gives quite good effects. I have a good play around with the program and come up with some really spectacular effects. This program could be used for many things - may be at the start of a program while the rest is loading. Anyway, I hope you have some fun with it.

You can make many more effects if you change the values of the numbers in line 140. Experiment with this line and you would be amazed at the different effects you can create.

HOW IT WORKS

a(n)	-	Position to be drawn from horizontally
a(m)	-	Position to be drawn to horizontally
b(n)	-	Position to be drawn from vertically
b(m)	-	Position to be drawn to vertically
t	-	Colour and number of shapes drawn
z	-	Centre of the screen vertically
p	-	Centre of the screen vertically
n	-	Number of positions within the array
m	-	Number of positions within the array to be drawn to.

Pitman's First Book of Games helped me greatly on how this program works. Most of the program is based from the book, but I have made a lot of changes to suit my own tastes.

```
140 MODE 0
150 PAPER 1:PEN 2:BORDER 0:CLS
160 FOR a=0 TO 15: INK a,a:NEXT a
    :INK 5,28
170 q=0:x=320:y=200:n=0
180 PLOT x,y,1:GOSUB 240
190 n=n+8
200 IF n<332 THEN GOTO 180
210 FOR p=1 TO 15:INK p,RND*26:NE
    XT p
220 q=0:n=n-8:GOSUB 250
230 IF n>0 THEN 220 ELSE 160
240 q=q+1
250 SOUND 1,n:PLOT x,y-n,q:PLOT x
    ,y+n,q:PLOT x-n,y,q:PLOT x+n,
    y,q:PLOT x+n/2,y+n/2,q:PLOT x
    +n/2,y-n/2,q:PLOT x-n/2,y-n/2
    ,q:PLOT x-n/2,y+n/2,q:IF q<15
     THEN RETURN ELSE q=0:RETURN
```

SPECIAL EFFECTS (2)
by Alex Smyth

Here is another demonstration, including sound, to experiment with. Alex didn't send any documentation except to say that for a different view, change the 'paper 0' in line 150 to 'paper 1'.

```
100 MODE 1:BORDER 0,3:INK 0,3:SPE
    ED INK 2,2:CLS
110 LOCATE 12,1:PRINT "By...D.G.
    Fennessy"
120 DIM a(36):DIM b(36):p=320:z=2
    00
130 FOR t=1 TO 5:FOR n=1 TO 36
140 k=n/11*PI*1000/1.2562*1.52348
150 a(n)=320+p*SIN(k):b(n)=200+z*
    COS(k)
160 MOVE a(n),b(n):NEXT n
170 FOR n=1 TO 36:m=n+12
180 IF m>36 THEN m=m-36
190 DRAW a(n),b(n):DRAW a(m),b(m)
    ,t:NEXT n:p=p/2:z=z/2:NEXT t
200 END
```

MINI CHARACTER DEFINER
by Rob Brown

This short one line program is handy because it is small enough to fit anywhere in the programcurrently under development. To use it, all you do is draw your character on a piece if paper and then run the line or type it in immediate mode. The screen will respond with: 0 1 2 3 4 5 6 7.

Now refer back to your piece of paper and copy the matrix in directly, using a "1" when a pixel is on and a "0" when the pixel is off. Make sure you remember to press return at the end of each line. After eight lines, the program prints 8 decimal numbers. Now just type in your line number and "SYMBOL" command followed by the CHR$ code of the character you want defined, and "COPY CURSOR" the row of numbers! There is no need for commas as they are placed in between the numbers already. Simple isn't it!?

```
10 PRINT CHR$(24);"01234567";CHR$(7):DIM a(7):
   FOR t=0 TO 7: INPUT " ",a$:a(t)=VAL("&X"+A$):
   NEXT: FOR t=0 to 7:PRINT
   ",";RIGHT$(STR$(a(t)),
   LEN(STR$(a(t)))-1;:NEXT
```

User Group Contact List

Please note that the following names are listed as contacts 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
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
Sue Kelly	Manangatang	(050) 35 1402
Angela Evans	Mt. Evelyn	(03) 736 1852
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	Calingunee 246 (weekendsonly)
D.F. Read	Ingham	(077) 77 8576
Tim Takken	Ipswich	(07) 202 4039
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

HINTS for high score games

Battle for Midway	-	withhold attacks until enemy fleets are close by.
Codename Mat	-	Leave cruisers alone until you have destroyed all the other ships.
Combat Lynx	-	use mines to destroy enemy reinforcements.
Harrier Attack	-	Fly low over buildings and hold the bomb release button down.
Haunted Hedges	-	get the "Demon Ice-Axes" only when the Guardians are near. Try to collect all the gold coins in the bottom half of the screen first.
Hunter Killer	-	get close to enemy then fire.
Knight Lore	-	Many objects, including spheres and flashing stars do not like you when you are a were-wolf.
Minder	-	buy in bulk and always read description of goods carefully.
Moonbuggy	-	when jumping the double set of holes, start your jump 2-3 cms before the first hole. Hold the joystick/jump button down as well as the fire button.
Roland in the Caves	-	always jump right as your first move.
Roland goes Digging	-	always stay in a safe place and dig holes around you.
Roland in Time	-	practice well on cards 1,6 and 4 to give a sound base for a good score.
Sorcery	-	Ignore the 'nasties' on every screen unless they happen to be draining an excessive amount of energy. Test all cauldrons before sitting on them by sliding across the top and watching your energy level. Don't waste Shooting Stars and Sacks of Spells - always zap two nasties with each. Save the eight Sorcerors as quickly as possible and then use the time left to zap nasties and open as many doors and plugs as possible.
Sorcery +	-	take a crown into the 2nd chapter.
Star Commando	-	shoot accurately and keep a record of all sectors so as to avoid "no data available".
Survivor	-	shoot fast and conserve porcupine bombs.
Way of Exp. Fist	-	do some flying kicks until your foot passes the computer's blocking arm then quickly do a low punch, but don't try this until you achieve 2nd Dan.
Wild Bunch	-	expect the Marshall to travel freely between towns 2 and 4.
3-D Monster Chase	-	keys 1 and 2 are on the middle floor; 3,4 and 7 are on the bottom; 5 and 6 are on the top.

What makes Locomotive run?

by Petr Lukes

Every time we turn the computer on or reset it, we are informed that LOCOMOTIVE SOFTWARE Ltd. are joint copyright holders. This applies to the resident BASIC interpreter and the operating system. Our BASIC is a special adaptation of their Mallard BASIC, which is being marketed separately (it is now supplied with the PCW8256) and has a reputation for being very fast.

If timing benchmarks are a valid indication of speed of processing, the AMSTRAD adaptation is about the fastest interpreter available on personal computers at this stage. While it is not completely bug-free (it has been said that there is no such thing as a bug-free working program), any bugs that have emerged so far are minor and rarely encountered.

Microssoft BASIC has established itself as the de-facto standard language, and Locomotive is largely compatible with it. This does not imply that it is a copy, rather that the same commands produce identical results.

The operating system of the computer consists of three distinct parts. When the machine is turned on, the operating system carries out numerous housekeeping tasks and then enters the editor mode to wait for our commands. It will wait until we press the ENTER key, and then try to make sense out of whatever we typed in (even if it is nothing).

If the line (disregarding leading spaces) does not start with characters which can be converted into a valid line number, the editor assumes that we are entering a command which requires immediate execution, and call the interpreter to do so. If the interpreter does not recognize the command, it will report an error.

A valid line number indicates that we want the line stored as a BASIC program line for later execution (by RUN). The line is not stored as we typed it, but is processed by the editor into a format which allows the interpreter to do its work faster. We will look at the format later.

When the interpreter is called by RUN, it starts at the beginning of the stored program and executes the instructions until it finishes, or is tripped up by an error, or gets interrupted by us. Obviously, this is a very complicated process which involves managing the available memory, creating storage locations for the variables and their values, as well as carrying out a certain amount of compilation, such as converting the target line numbers of GOTOs etc. into absolute addresses, as we shall see.

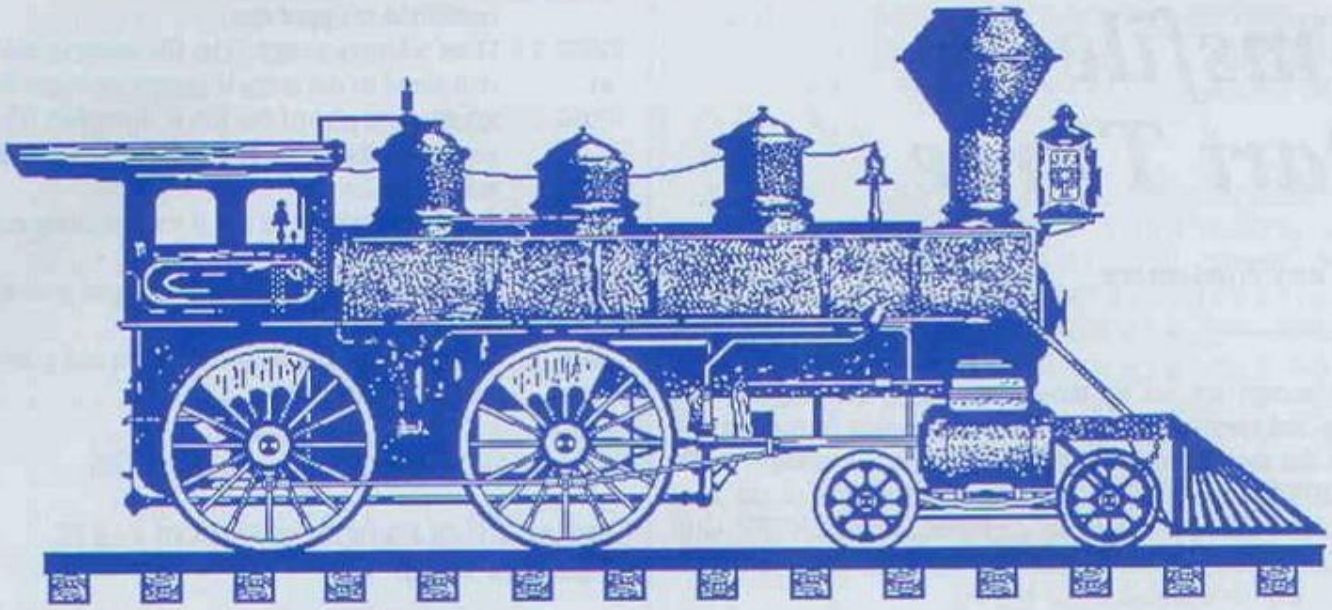
At the lowest level (i.e. closest to the hardware which produces the display and processes the data) is a collection of machine-language routines, some of which do their work without any intervention from us, and others which are called by the editor or interpreter as necessary.

The following short routine will enable us to examine the program lines as they are stored. Before entering it, reset your computer (by Ctrl/Shift/Esc), and in the direct mode (no line numbers) enter this statement:

```
a=0:?@a-6
```

The result is the starting address of the Basic program area. A standard CPC464 will return 368 (=170 hex), but the later versions of Basic may use a different address. It is also possible that sideways ROMs may reserve some low memory for their own use. Also, since Locomotive and Mallard are closely related, it is probable that much of what we discover about the structure of Locomotive will apply to Mallard, but the addresses will be very different. In Mallard, the '@a' would be replaced by 'varptr(a)'.

If the direct statement returns 368, type in the routine as is; if not, replace



the '&170' at the end of line 9020 with your value (no need to convert it to hex). Run it, and you will see it display itself. Each line starts with four housekeeping bytes: two bytes for the total number of bytes in the line, two bytes for the line number, and each line is terminated by a zero byte. The display is in hexadecimal numbers, with the character corresponding to that number immediately adjacent.

If you examine the display of the first line, you will see the identification message; this should help you to sort out the display format. After this training exercise, you can remove the "e=e+500" on the fourth line; this will stop the display on the first line of the routine. This is achieved by finding where the "id$" is stored in the program area. This topic is not, to my knowledge, explained in the CPC464 manuals, and will be dealt with later.

We can now start looking at what the editor and interpreter do with our program lines. Enter the two following lines:

```
10 GOTO 9000
20 GOTO 9000
```

and RUN. The actual codes of the first two lines (after the four housekeeping bytes) will appear as:

```
a0 20 1d 83 01 00
a0 20 1e 28 23 00
```

The number A0hex (160 decimal) is a token for GOTO. All keywords are converted by the editor into one or two tokens as each line is stored. 20hex (32 decimal) is the ASCII code for space. Now we come to the difference between the two lines: If we convert the 2328hex in the second line (very simply by ?&2328 in the command mode, but note the reverse order), we get 9000, the target line for the GOTO. Because line 20 was not executed since line 10 directed the program flow past it, the interpreter did not convert the line number into an address.

This is evident from comparison with the first line: you will have noticed that the line 9000 starts at 0184hex. The address 0183hex points to the end byte of the preceding line. Obviously the interpreter has done some compilation while executing the first line, and changed the token 1e to 1d to indicate the conversion. The absolute addresses will be changed back to line numbers when the program is edited, and before a LIST or SAVE.

You can now experiment with entering the various commands on the lines preceding the routine. A RUN 9000 will jump over them and display them without executing them.

```
9000 id$="BASIC lines LKS 860117"
9010 PRINT:PRINT CHR$(24)"Locomotive BASIC 1
     .0 on AMSTRAD CPC464":PRINT id$CHR$(24)
9020 a=@id$:e=PEEK(a+1)+PEEK(a+2)*256:a=&170
9030 e=e+500:WHILE a<e
9040 Le=PEEK(a)+PEEK(a+1)*256:Ln=PEEK(a+2)+P
     EEK(a+3)*256
9050 PRINT:PRINT USING"Le#####";Le;:PRINT US
     ING" Ln#####";Ln;:PRINT"    Start (hex)
     "HEX$(a,4)
9060 FOR b=a TO a+3:c=PEEK(b):PRINT HEX$(c,2
     )CHR$(1)CHR$(c)" ";:NEXT b:PRINT
9070 FOR b=b TO a+Le-1:c=PEEK(b):PRINT CHR$(
     24)LOWER$(HEX$(c,2))CHR$(24)CHR$(1)CHR$
     (c)" ";:NEXT b:PRINT
9080 a=b:WEND
```

Amsfile - Part Three

by Tony Blakemore

This month we set up the main part of the program that enters and corrects a record. The information is entered and when the last line is finished you will get a message "ANY CORRECTIONS (Y/N)" pressing N takes you to the next entry. Y enables a line to be corrected. Press ENTER until the line is reached and re-enter the correct line. If you press M you will be returned to the MENU.

As with most programs a couple of bugs have reared their ugly head - please correct the following lines:

```
6170 PRINT #9,FILE$(A):NEXT A
10340 WINDOW #1,11,26,6,9
10350 WINDOW #2,11,23,10,12
```

HOW IT WORKS

ENTER NEW RECORD (SUBROUTINE 2000)

2000 - Prints heading sets ink and goes to subroutine 11000 to set up file headings.

2010 - FILENO keeps track of the number of files

2020 - Erases area of the file when you go onto another entry.

2030 - to 2370 GUIDE$ prints under the entry a guide to length SNAME$ prompts for the entry (note Surname first)
LINE INPUT X$ takes the entry and holds it.
NAME$ = LEFT$(X$+PACKING$,16) X$ has 16 blank spaces added to it and the total length is cut to 16 characters. This procedure is repeated for all the entries.
FLAG is set to either 1 or 0 at the correction prompt and if set enables you to go past the lines that you do not want to correct.

2350 - Joins all the entries together into FILE$

2360 - Resets all the entry strings to nothing

2380 - Prompts for a correction. Y prints message and goes to subroutine 12000 which cuts FILE$ into entry strings NAME$ etc. It then goes into the correction sequence.

2420 - Returns to MENU

2430 - Goes to next file entry

VIEW/ALTER FILE (subroutine 3000)

3000 - Sets up heading

3010 - Full or part name is entered into SEARCH$ and converted to upper case.

3020 - to 3052 Does a binary search. The file is cut in half and compared to the entry if greater or lesser the appropriate part of the file is discarded. The process is then repeated until the file is found or not found!

3055 - Is a further check to see if the file is the correct one.

3060 - Prints message if file not found and goes to start of routine.

3070 - If file is found sets up the screen and goes to correction option.

SET UP FILE HEADING (subroutine 11000)

BREAK UP FILES INTO SEPARATE PARTS (subroutine 12000)

PRINT FILE INFORMATION (subroutine 13000)

Save the program as "AMSFILE3",a: Now merge part 1 and part 2. You can now enter and display records. Until the sort routine is entered next month, along with the final parts of the program, make sure that the entries are in alphabetical order. The binary search will only work on a sorted file and will ignore a record that is out of order. To test the program so far, you can enter a series of records starting with ,say, A and progressing through the alphabet. If you have a series of records that you wish to enter, sort them by hand first and enter them in alphabetical order. The file can now be saved.

```
2000 CLS:PEN 2:GOSUB 11000
2010 FILENO=FILENO+1:NO=FILENO
2020 CLS #1:CLS #2:CLS #3:CLS #4:L
     OCATE 8,1:PRINT MENU2$;" NO";
     FILENO
2030 LOCATE 15,19:PRINT LEFT$(GUID
     E$,16)
2040 LOCATE 10,18:PRINT SNAME$;
2050 LINE INPUT "";X$:X$=UPPER$(X$
     )
2060 IF FLAG=1 AND X$="" THEN 2100
2070 IF X$="" THEN CLS #4:GOTO 203
     0
2080 IF X$="M" THEN CLS:FILENO=FIL
     ENO-1:GOTO 200
2090 LOCATE 11,6:NAME$=LEFT$(X$+PA
     CKING$,16):PRINT NAME$
2100 CLS #4:LOCATE 15,19:PRINT LEF
     T$(GUIDE$,16)
2110 LOCATE 10,18:PRINT SADDR$;
2120 LINE INPUT "";X$
2130 IF FLAG=1 AND X$="" THEN 2150
2140 LOCATE 11,8:ADDR$=LEFT$(X$+PA
```

```
    CKING$,16):PRINT ADDR$
2150 CLS #4:LOCATE 15,19:PRINT LEF
    T$(GUIDE$,12)
2160 LOCATE 10,18:PRINT SSUBR$;
2170 LINE INPUT "";X$
2180 IF FLAG=1 AND X$="" THEN 2200
2190 LOCATE 11,10:SUBR$=LEFT$(X$+P
    ACKING$,12):PRINT SUBR$
2200 CLS #4:LOCATE 15,19:PRINT LEF
    T$(GUIDE$,12)
2210 LOCATE 10,18:PRINT SPHNE$;
2220 LINE INPUT "";X$
2230 IF FLAG=1 AND X$="" THEN 2250
2240 LOCATE 11,12:PHNE$=LEFT$(X$+P
    ACKING$,12):PRINT PHNE$
2250 CLS #4:LOCATE 15,19:PRINT LEF
    T$(GUIDE$,4)
2260 LOCATE 9,18:PRINT SSTATE$;
2270 LINE INPUT "";X$
2280 IF FLAG=1 AND X$="" THEN 2300

2290 LOCATE 31,10:STATE$=LEFT$(X$+
    PACKING$,4):PRINT STATE$
2300 CLS #4:LOCATE 15,19:PRINT LEF
    T$(GUIDE$,4)
2310 LOCATE 9,18:PRINT SPCDE$;
2320 LINE INPUT "";X$
2330 IF FLAG=1 AND X$="" THEN 2350
2340 LOCATE 31,12:PCDE$=LEFT$(X$+P
    ACKING$,4):PRINT PCDE$
2350 FLAG=0:FILE$(NO)=NAME$+ADDR$+
    SUBR$+PHNE$+STATE$+PCDE$
2360 NAME$="":ADDR$="":SUBR$="":PH
    NE$="":STATE$="":PCDE$="":
2370 LOCATE 5,21:PRINT SPACE$(30)
2380 CLS #4:LOCATE 9,18:PRINT " AN
    Y CORRECTIONS (Y/N)
2390 I$=UPPER$(INKEY$):IF I$="" TH
    EN 2390
2400 IF I$="Y" THEN FLAG=1:CLS #4:
    LOCATE 6,21:PRINT "PRESS ENTE
    R TO REQUIRED LINE":GOSUB 120
    00:GOTO 2030
2410 IF I$="M" THEN CLS:GOTO 200
2420 IF I$="N" THEN 2010 ELSE 2380
3000 PEN 2:CLS:LOCATE 8,1:PRINT ME
    NU3$
3010 LOCATE 4,4:INPUT "ENTER FULL/
    PART NAME";SEARCH$:SEARCH$=UP
    PER$(SEARCH$)
3020 LOW=1:HIGH=FILENO
3030 WHILE HIGH>LOW
3040 MID=(LOW+HIGH)\2
3050 IF LEFT$(SEARCH$,LEN(SEARCH$)
    )>LEFT$(FILE$(MID),LEN(SEARCH
    $)) THEN LOW=MID+1 ELSE HIGH=
    MID
3060 WEND
3070 IF LEFT$(FILE$(HIGH),LEN(SEAR
    CH$))=SEARCH$ THEN 3090
3080 LOCATE 4,4:PRINT " NAME IS NO
    T ON FILE":PRINT CHR$(7):FOR
    A=1 TO 4000:NEXT:GOTO 3000
3090 NO=HIGH:CLS:LOCATE 7,1:PRINT
    MENU3$;" NO";NO: GOSUB 12000:
    GOSUB 11000:GOSUB 13000:GOTO
    2370
11000 LOCATE 5,6:PRINT SNAME$
11010 LOCATE 5,8:PRINT SADDR$
11020 LOCATE 5,10:PRINT SSUBR$
11030 LOCATE 5,12:PRINT SPHNE$
11040 LOCATE 24,10:PRINT SSTATE$
11050 LOCATE 24,12:PRINT SPCDE$
11060 RETURN
12000 NAME$=MID$(FILE$(NO),1,16)
12010 ADDR$=MID$(FILE$(NO),17,16)
12020 SUBR$=MID$(FILE$(NO),33,12)
12030 PHNE$=MID$(FILE$(NO),45,12)
12040 STATE$=MID$(FILE$(NO),57,4)
12050 PCDE$=MID$(FILE$(NO),61,4)
12060 RETURN
13000 LOCATE 11,6:PRINT NAME$
13010 LOCATE 11,8:PRINT ADDR$
13020 LOCATE 11,10:PRINT SUBR$
13030 LOCATE 11,12:PRINT PHNE$
13040 LOCATE 31,10:PRINT STATE$
13050 LOCATE 31,12:PRINT PCDE$
13060 RETURN
```

Yahtzee for the Amstrad

from Alf Azzopardi

This program is based on the traditional family game of Yahtzee. Originally written (anonomously) for a now redundant computer, it proved to be such an enjoyable game that I decided to adapt it for the Amstrad.

The game can be played by one to four players and the program automatically updates the scores as they are achieved. The player with the highest score is the winner. For those not familiar with the game, I have incorporated a full set of instructions.

I hope it proves to be as much fun for others as it has for myself and my friends.

```
10 REM ********************************
20 REM ***          YAHTZEE           ***
30 REM ***      Adapted for the       ***
40 REM ***          AMSTRAD           ***
50 REM ***      by Alf Azzopardi      ***
60 REM ***         18/2/86            ***
70 REM ********************************
80 BORDER 1:MODE 0:PAPER 0
90 CLS
100 INK 1,6:INK 3,0:INK 0,11:INK 2,26
110 PAPER 2:PEN 1
120 LOCATE 5,5:PRINT "Y_A_H_T_Z_E_E"
130 LOCATE 6,14:PRINT "Adapted for"
140 LOCATE 6,17:PRINT "the Amstrad"
150 LOCATE 10,20:PRINT "by"
160 LOCATE 5,23:PRINT "Alf Azzopardi"
170 GOSUB 4290
180 GOSUB 4150
190 MODE 1
200 LOCATE 15,2:PRINT "** ";:PEN 3:PRINT"
    YAHTZEE ";:PEN 1:PRINT"**"
210 LOCATE 4,4:PRINT"Do you want any inst
    ructions (Y/N)?"
220 a$=INKEY$:a$=UPPER$(a$):IF a$="" THEN
     220
230 IF a$="Y" THEN GOSUB 4540
240 PAPER 0:CLS:PAPER 2
250 PEN 1:LOCATE 15,2:PRINT "** ";:PEN 3:
    PRINT"YAHTZEE ";:PEN 1:PRINT"**"
260 LOCATE 8,6:INPUT "Enter # of Players
    (1-4):",n
270 FOR p=1 TO n
280 PRINT:PRINT "Player #";p;:INPUT "'s N
    ame or Initials:",player$(p)
290 player$(p)=UPPER$(player$(p))
300 IF LEN(player$(p))<=4 THEN 330
310 PRINT "4 Letters maximum, please!"
320 GOTO 280
330 NEXT p
340 CLS
350 PAPER 0
360 DIM score(4,16)
370 DIM die(3,3)
380 FOR p=1 TO n
390 FOR s=1 TO 16
400 score(p,s)=1000
410 NEXT s
420 NEXT p
430 SYMBOL AFTER 157
440 SYMBOL 196,0,0,60,60,60,60,0,0
450 SYMBOL 197,0,0,0,0,0,0,0,0
460 SYMBOL 204,0,0,60,60,60,60,0,0
470 SYMBOL 205,0,0,0,0,0,0,0,0
480 SYMBOL 212,0,0,60,60,60,60,0,0
490 SYMBOL 213,0,0,0,0,0,0,0,0
500 SYMBOL 220,0,0,60,60,60,60,0,0
510 SYMBOL 221,0,0,0,0,0,0,0,0
520 SYMBOL 228,0,0,60,60,60,60,0,0
530 SYMBOL 229,0,0,0,0,0,0,0,0
540 SYMBOL 159,0,0,0,0,0,0,0,0
550 SYMBOL 158,255,255,255,255,0,0,0,0
560 CLS
570 l=16
580 PEN 0
590 i=7
600 j=4
610 READ prin$
620 GOSUB 3960
630 FOR p=1 TO n
640 i=7
650 j=(15+5*p)-1
660 prin$=player$(p)
670 GOSUB 3960
680 NEXT p
690 FOR i=9 TO 24
700 j=4
710 READ prin$
720 GOSUB 3960
730 NEXT i
740 FOR b=1 TO 3
750 LOCATE 4,b:PRINT STRING$(34,159)
760 NEXT b
770 FOR i=1 TO 5
780 LOCATE 6*i+1,2:PRINT CHR$(48+i)
790 NEXT i
800 FOR b=9 TO 24
810 FOR i=19 TO 38 STEP 5
820 LOCATE i,b:PRINT STRING$(4,159)
830 LOCATE i+1,b:PRINT CHR$(45)+CHR$(45)

840 NEXT i:NEXT b
```

```
850 PAPER 2:LOCATE 3,16:PRINT  CHR$(62)

860 PAPER 0:LOCATE 4,5:PRINT STRING$(35,3
    2)
870 LOCATE 9,7:PRINT STRING$(2,159)
880 LOCATE 17,7:PRINT CHR$(159)+CHR$(159)

890 PEN 3
900 prin$="  Key 1,2,3,4,or 5 to save dic
    e!  "
910 GOSUB 3900
920 GOSUB 4150
930 prin$="  Key 0(zero) to change mind!
      "
940 GOSUB 3900
950 GOSUB 4150
960 prin$="  Press <SPACE BAR> to roll di
    ce! "
970 GOSUB 3900
980 GOSUB 4150
990 prin$="  Press "+CHR$(1)+CHR$(11)+" o
    r "+CHR$(1)+CHR$(10)+" to move pointe
    r!   "
1000 GOSUB 3900
1010 GOSUB 4150
1020 prin$="  Press <ENTER> to compute sco
    re! "
1030 GOSUB 3900
1040 GOSUB 4150
1050 prin$="  Press any key to start game!
      "
1060 GOSUB 3900
1070 a$=INKEY$:IF a$="" THEN 1070
1080 FOR turn=1 TO 13
1090 FOR p=1 TO n
1100 FOR i=1 TO 10
1110 PAPER 0:PEN 1
1120 LOCATE (15+5*p)-1,8:PRINT STRING$(4,1
    58)
1130 IF i=10 THEN 1160
1140 LOCATE (15+5*p)-1,8:PRINT STRING$(4,1
    59)
1150 NEXT i
1160 FOR try=1 TO 3
1170 PAPER 1:PEN 2:ON try GOTO 1180,1200,1
    220
1180 LOCATE 17,7:PRINT CHR$(49)
1190 GOTO 1230
1200 LOCATE 17,7:PRINT CHR$(50)
1210 GOTO 1230
1220 LOCATE 17,7:PRINT CHR$(51)
1230 FOR d=1 TO 1000:NEXT d:PAPER 2:LOCATE
     4,5:PRINT STRING$(34,159)
1240 FOR dice=1 TO 5
1250 IF keep(dice)=1 THEN 1410
1260 RANDOMIZE TIME
1270 roll=INT(6*RND)+1
1280 PAPER 1:PEN 2
1290 count(dice)=roll
1300 FOR x=1 TO 3
1310 FOR y=1 TO 3
1320 die(x,y)=189+8*dice
1330 NEXT y
1340 NEXT x
1350 ON roll GOSUB 1720,1740,1770,1790,182
    0,1840
1360 FOR y=1 TO 3
1370 FOR x=1 TO 3
1380 LOCATE (6*dice)+y+1,x:PRINT CHR$(die(
    x,y))
1390 NEXT x
1400 NEXT y
1410 NEXT dice
1420 SOUND 2,379,50,15:SOUND 2,319,50,15:S
    OUND 2,239,50,15
1430 a$=INKEY$:IF a$="" THEN 1430
1440 IF INKEY(0)=0 THEN 1870
1450 IF JOY(0) AND 1 THEN 1870
1460 IF INKEY(2)=0 THEN 1930
1470 IF JOY(0) AND 2 THEN 1930
1480 IF a$=" " THEN 1590
1490 IF INKEY(18)=0 THEN 2250
1500 IF INKEY(6)=0 THEN 2250
1510 IF JOY(0) AND 16 THEN 2250
1520 IF a$="1" THEN GOSUB 2050
1530 IF a$="2" THEN GOSUB 2090
1540 IF a$="3" THEN GOSUB 2130
1550 IF a$="4" THEN GOSUB 2170
1560 IF a$="5" THEN GOSUB 2210
1570 IF a$="0" THEN PRINT CHR$(7):GOSUB 19
    90
1580 GOTO 1430
1590 FOR d=1 TO 500:NEXT d:IF try=3 THEN 1
    430
1600 FOR i=1 TO 5
1610 IF keep(i)=1 THEN 1620
1620 NEXT i
1630 NEXT try
1640 PAPER 0:LOCATE (15+5*p)-1,8:PRINT STR
    ING$(4,159)
1650 GOSUB 1990
1660 PAPER 0:LOCATE 3,1:PRINT CHR$(32)
1670 l=16
1680 PAPER 2:PEN 3:LOCATE 3,l:PRINT CHR$(6
    2)
1690 NEXT p
1700 NEXT turn
1710 GOTO 4020
1720 die(2,2)=188+8*dice
1730 RETURN
1740 die(1,1)=188+8*dice
1750 die(3,3)=188+8*dice
1760 RETURN
1770 die(2,2)=188+8*dice
1780 GOTO 1740
1790 die(1,3)=188+8*dice
1800 die(3,1)=188+8*dice
1810 GOTO 1740
1820 die(2,2)=188+8*dice
1830 GOTO 1790
1840 die(1,2)=188+8*dice
1850 die(3,2)=188+8*dice
1860 GOTO 1790
1870 PAPER 0:LOCATE 3,1:PRINT CHR$(32)

1880 l=l-1
1890 IF l>8 THEN 1910
1900 l=9
1910 PAPER 2:PEN 3:LOCATE 3,l:PRINT CHR$(6
    2)
1920 GOTO 1430
1930 PAPER 0:LOCATE 3,1:PRINT CHR$(32)
```

```
1940 l=l+1
1950 IF l<24 THEN 1970
1960 l=23
1970 PAPER 2:PEN 3:LOCATE 3,l:PRINT CHR$(6
     2)
1980 GOTO 1430
1990 PAPER 2:FOR i=1 TO 5
2000 keep(i)=0
2010 NEXT i
2020 LOCATE 4,7:PRINT STRING$(5,32)
2030  PAPER 2:PEN 3:LOCATE 4,7:PRINT STRIN
     G$(5,45)
2040 RETURN
2050 keep(1)=1
2060 PRINT CHR$(7)
2070 PEN 2:PAPER 1:LOCATE 4,7:PRINT CHR$(4
     9)
2080 RETURN
2090 keep(2)=1
2100 PRINT CHR$(7)
2110 PEN 2:PAPER 1:LOCATE 5,7:PRINT CHR$(5
     0)
2120 RETURN
2130 keep(3)=1
2140 PRINT CHR$(7)
2150 PEN 2:PAPER 1:LOCATE 6,7:PRINT CHR$(5
     1)
2160 RETURN
2170 keep(4)=1
2180 PRINT CHR$(7)
2190 PEN 2:PAPER 1:LOCATE 7,7:PRINT CHR$(5
     2)
2200 RETURN
2210 keep(5)=1
2220 PRINT CHR$(7)
2230 PEN 2:PAPER 1:LOCATE 8,7:PRINT CHR$(5
     3)
2240 RETURN
2250 IF score(p,l-8)=1000 THEN 2280
2260 REM
2270 GOTO 1430
2280 IF l<15 THEN 2680
2290 IF l>16 THEN 3170
2300 GOTO 2260
2310 i=1
2320 j=(15+5*p)-1
2330 PAPER 0
2340 GOSUB 3960
2350 score(p,7)=score(p,1)+score(p,2)+scor
     e(p,3)+score(p,4)+score(p,5)+score(p,
     6)
2360 IF score(p,7)<1000 THEN 2390
2370 score(p,7)=score(p,7)-1000
2380 GOTO 2360
2390 IF score(p,7)<63 THEN 2490
2400 IF score(p,8)=35 THEN 2490
2410 score(p,8)=35
2420 prin$=STR$(35)
2430 i=16
2440 j=(15+5*p)-1
2450 GOSUB 3960
2460 prin$=" * It's bonus time for "+playe
     r$(p)+" ! *   "
2470 GOSUB 3900
2480 GOSUB 4290
2490 prin$=STR$(score(p,7))
2500 j=(15+5*p)-1
2510 i=15
2520 IF LEN(prin$)=3 THEN 2570
2530 IF LEN(prin$)=2 THEN 2560
2540 prin$=prin$
2550 GOTO 2570
2560 prin$=prin$
2570 GOSUB 3960
2580 score(p,16)=score(p,7)+score(p,8)+sco
     re(p,9)+score(p,10)+score(p,11)+score
     (p,12)+score(p,13)+score(p,14)
2590 score(p,16)=score(p,16)+score(p,15)
2600 IF score(p,16)<1000 THEN 2630
2610 score(p,16)=score(p,16)-1000
2620 GOTO 2600
2630 prin$=STR$(score(p,16))
2640 IF LEN(prin$)=3 THEN 2690
2650 IF LEN(prin$)=2 THEN 2680
2660 prin$=prin$
2670 GOTO 2690
2680 prin$=prin$
2690 j=(15+5*p)-1
2700 i=24
2710 GOSUB 3960
2720 IF score(p,16)-score(p,1-8)>=200 THEN
      2770
2730 IF score(p,16)<200 THEN 2770
2740 prin$=" "+player$(p)+" just broke 200
     _ Congrats! "
2750 GOSUB 3900
2760 GOSUB 4290
2770 IF score(p,16)-score(p,1-8)>250 THEN
     2820
2780 IF score(p,16)<250 THEN 2820
2790 prin$=" * 250 *  Advanced player stat
     us! "
2800 GOSUB 3900
2810 GOSUB 4290
2820 IF score(p,16)-score(p,1-8)>=300 THEN
      2870
2830 IF score(p,16)<300 THEN 2870
2840 prin$=STR$(score(p,16))+" ** Expert p
     layer status! ** "
2850 GOSUB 3900
2860 GOSUB 4290
2870 GOTO 1640
2880 add=0
2890 FOR i=1 TO 5
2900 IF count(i)=(1-8) THEN 3020
2910 NEXT i
2920 score(p,1-8)=add
2930 IF add<3*(1-8) THEN 3040
2940 IF add=3*(1-8) THEN 3090
2950 IF add>3*(1-8) THEN 3130
2960 prin$=STR$(score(p,1-8))
2970 IF LEN(prin$)=1 THEN 3000
2980 prin$=prin$
2990 GOTO 2310
3000 prin$=prin$
3010 GOTO 2310
3020 add=add+1-8
3030 GOTO 2910
3040 prin$=" Uhoh! "+player$(p)+", down on
      your"+STR$(1-8)+"'s!   "
3050 GOSUB 3900
3060 SOUND 2,1000,30,15
```

```
3070 SOUND 2,3000,60,15
3080 GOTO 2960
3090 prin$=" You're even on your"+STR$(1-8
     )+"'s, "+player$(p)+" ! "
3100 GOSUB 3900
3110 GOSUB 4180
3120 GOTO 2960
3130 j=4:i=5:prin$=" Nice goin',"+player$(
     p)+" _ Extra"+STR$(1-8)+"'s!      "
3140 GOSUB 3960
3150 GOSUB 4420
3160 GOTO 2960
3170 yes=0
3180 add=count(1)+count(2)+count(3)+count(
     4)+count(5)
3190 FOR i=1 TO 5
3200 FOR j=(i+1) TO 5
3210 IF count(i)=count(j) THEN 3250
3220 NEXT j
3230 NEXT i
3240 GOTO 3270
3250 yes=yes+1
3260 GOTO 3220
3270 ON 1-16 GOTO 3280,3380,3390,3460,3680
     ,3740,3830
3280 IF yes>=3 THEN 3320
3290 score(p,1-8)=0
3300 prin$=" You missed your"+STR$(1-14)+"
     /kind "+player$(p)+" !    "
3310 GOTO 3050
3320 score(p,1-8)=add
3330 IF add>22 THEN 3360
3340 prin$=" You score"+STR$(add)+" for"+S
     TR$(1-14)+" of a kind!    "
3350 GOTO 3100
3360 j=4:i=5:prin$=" "+STR$(add)+" is an e
     xcellent score!       "
3370 GOTO 3140
3380 IF yes>=6 THEN 3320 ELSE 3290
3390 IF yes=4 THEN 3430
3400 score(p,11)=0
3410 prin$=" That's an empty house, "+play
     er$(p)+"!      "
3420 GOTO 3050
3430 score(p,11)=25
3440 j=4:i=5:prin$=" Good times at "+playe
     r$(p)+"'s house!      "
3450 GOTO 3140
3460 IF yes>1 THEN 3470
3470 IF yes=0 THEN 3630
3480 IF add=15 THEN 3650
3490 IF add=20 THEN 3650
3500 IF add=11 THEN 3540
3510 IF add=24 THEN 3540
3520 FOR i=1 TO 5
3530 ON count(i) GOTO 3570,3600,3580,3610,
     3620
3540 score(p,12)=30
3550 j=4:i=5:prin$=" You made your small s
     traight!  "
3560 GOTO 3140
3570 IF add<15 THEN 3650
3580 NEXT i
3590 GOTO 3540
3600 IF add<=20 THEN 3650 ELSE 3580
3610 IF add>15 THEN 3650 ELSE 3580
3620 IF add>21 THEN 3650 ELSE 3580
3630 IF add=17 THEN 3650
3640 IF add=18 THEN 3650 ELSE 3540
3650 score(p,1-8)=0
3660 prin$=" That straight's pretty crooke
     d !"
3670 GOTO 3050
3680 IF yes=0 THEN 3690 ELSE 3650
3690 IF add=15 THEN 3650
3700 IF add=20 THEN 3710 ELSE 3650
3710 score(p,13)=40
3720 j=4:i=5:prin$=" You landed the big on
     e, "+player$(p)+"!  "
3730 GOTO 3140
3740 IF yes=10 THEN 3780
3750 score(p,14)=0
3760 prin$=" Yahtzee is hard to roll, "+pl
     ayer$(p)+"    "
3770 GOTO 3050
3780 score(p,14)=50
3790 prin$=" "+player$(p)+" has achieved t
     he ultimate!!"
3800 GOSUB 3900
3810 GOSUB 4290
3820 GOTO 2960
3830 score(p,15)=add
3840 IF add>22 THEN 3360
3850 IF turn<7 THEN 3880
3860 prin$=" You score"+STR$(add)+" for Ch
     ance!       "
3870 GOTO 3100
3880 prin$=" Too early for taking Chance!
      "
3890 GOTO 3050
3900 PAPER 2:PEN 3:j=4
3910 i=5
3920 FOR k=1 TO 5
3930 LOCATE j,i:PRINT prin$
3940 NEXT k
3950 GOTO 3980
3960 PAPER 2:PEN 3
3970 LOCATE j,i:PRINT prin$
3980 RETURN
3990 DATA "-----   roll:",aces.....3,tw
     os.....6,threes.....9,fours.....
     12,fives.....15
4000 DATA sixes.....18,Top Total..63,BONU
     S.....35,3 of kind.all,4 of kind.all
     ,full house.25
4010 DATA small str..30,large str..40,Yaht
     zee....50,Chance....all,TOTAL SCORE..
4020 b=MAX(score(1,16),score(2,16),score(3
     ,16),score(4,16))
4030 IF b=score(1,16) THEN p=1
4040 IF b=score(2,16) THEN p=2
4050 IF b=score(3,16) THEN p=3
4060 IF b=score(4,16) THEN p=4
4070 LOCATE 4,5:PRINT STRING$(34,159)
4080 prin$=" * Highest score was"+STR$(b)+
     " by "+player$(p)+" *"
4090 GOSUB 3900
4100 GOSUB 4150
4110 PEN 1:LOCATE 4,25:PRINT "Would you li
     ke another game? (Y/N)"
4120 an$=INKEY$:IF an$="" THEN 4120
4130 IF an$="Y" OR an$="y" THEN RUN
```

```
4140 END
4150 FOR d=1 TO 2000
4160 NEXT d
4170 RETURN
4180 SOUND 2,253,20,15
4190 SOUND 2,284,20,15
4200 SOUND 2,319,50,15
4210 SOUND 2,284,20,15
4220 SOUND 2,319,20,15
4230 SOUND 2,358,50,15
4240 SOUND 2,319,20,15
4250 SOUND 2,358,20,15
4260 SOUND 2,379,50,15
4270 SOUND 2,358,50,15
4280 RETURN
4290 SOUND 2,319,20,15
4300 SOUND 2,253,20,15
4310 SOUND 2,213,50,15
4320 SOUND 2,253,20,15
4330 SOUND 2,213,20,15
4340 SOUND 2,253,50,15
4350 SOUND 2,319,20,15
4360 SOUND 2,253,20,15
4370 SOUND 2,213,50,15
4380 SOUND 2,253,20,15
4390 SOUND 2,213,20,15
4400 SOUND 2,253,50,15
4410 RETURN
4420 SOUND 2,100,25,15
4430 SOUND 2,200,25,15
4440 SOUND 2,300,25,15
4450 SOUND 2,400,25,15
4460 SOUND 2,200,25,15
4470 SOUND 2,300,25,15
4480 SOUND 2,400,25,15
4490 SOUND 2,500,25,15
4500 SOUND 2,400,25,15
4510 SOUND 2,500,25,15
4520 SOUND 2,600,25,15
4530 RETURN
4540 LOCATE 2,7:PRINT"Five dice are used.A
     ny number of people"
4550 LOCATE 2,8:PRINT"from two to four may
      play."
4560 LOCATE 2,10:PRINT"There are twelve ro
     unds,and each player"
4570 LOCATE 2,11:PRINT"has one turn in eac
     h round. A turn con-"
4580 LOCATE 2,12:PRINT"sists of a maximum
     of three rolls of"
4590 LOCATE 2,13:PRINT"the dice, as in Pok
     er dice; the player"
4600 LOCATE 2,14:PRINT"may at any time sta
     nd, or may pick up"
4610 LOCATE 2,15:PRINT"and reroll any of t
     he dice until all"
4620 LOCATE 2,16:PRINT"three rolls have be
     en played."
4630 LOCATE 2,18:PRINT"At the end of each
     turn,the player must"
4640 LOCATE 2,19:PRINT"designate his/her f
     ive dice to count in"
4650 LOCATE 2,20:PRINT"one of the twelve c
     ategories shown on"
4660 LOCATE 2,21:PRINT"the Scorepad."
4670 PEN 3:LOCATE 8,24:PRINT"Press any key
      to continue";:PEN 1:PRINT"!"
4680 a$=INKEY$:IF a$="" THEN 4680
4690 PAPER 0:CLS:PAPER 2
4700 LOCATE 15,2:PRINT "** ";:PEN 3:PRINT"
     YAHTZEE ";:PEN 1:PRINT"**"
4710 PEN 3
4720 LOCATE 31,4:PRINT"PLAYERS
4730 LOCATE 3,5:PRINT"HAND"
4740 LOCATE 18,5:PRINT"MAXIMUM"
4750 LOCATE 28,5:PRINT"A"
4760 LOCATE 32,5:PRINT"B"
4770 LOCATE 36,5:PRINT"C"
4780 LOCATE 40,5:PRINT"D"
4790 PEN 1:LOCATE 2,7:PRINT"Yahtzee":LOCAT
     E 21,7:PRINT"50"
4800 LOCATE 2,8:PRINT"Large Straight":LOCA
     TE 21,8:PRINT"40"
4810 LOCATE 2,9:PRINT"Small Straight":LOCA
     TE 21,9:PRINT"30"
4820 LOCATE 2,10:PRINT"Four of a Kind":LOC
     ATE 21,10:PRINT"29"
4830 LOCATE 2,11:PRINT"Three of a Kind":LO
     CATE 21,11:PRINT"29"
4840 LOCATE 2,12:PRINT"Full House":LOCATE
     21,12:PRINT"25"
4850 LOCATE 2,13:PRINT"Chance":LOCATE 21,1
     3:PRINT"30"
4860 LOCATE 2,14:PRINT"Sixes":LOCATE 21,14
     :PRINT"30"
4870 LOCATE 2,15:PRINT"Fives":LOCATE 21,15
     :PRINT"25"
4880 LOCATE 2,16:PRINT"Fours":LOCATE 21,16
     :PRINT"20"
4890 LOCATE 2,17:PRINT"Threes":LOCATE 21,1
     7:PRINT"15"
4900 LOCATE 2,18:PRINT"Twos":LOCATE 21,18:
     PRINT"10"
4910 LOCATE 2,19:PRINT"Aces":LOCATE 21,19:
     PRINT"5"
4920 PEN 3:LOCATE 8,24:PRINT"Press any key
      to continue";:PEN 1:PRINT"!"
4930 a$=INKEY$:IF a$="" THEN 4930
4940 PAPER 0:CLS:PAPER 2
4950 LOCATE 15,2:PRINT "** ";:PEN 3:PRINT"
     YAHTZEE ";:PEN 1:PRINT"**"
4960 LOCATE 2,4:PRINT"For example, player
     A will have twelve"
4970 LOCATE 2,5:PRINT"turns. After each tu
     rn he/she must pick"
4980 LOCATE 2,6:PRINT"a category and enter
      in the column"
4990 LOCATE 2,7:PRINT"under his/her name t
     he score (it may be"
5000 LOCATE 2,8:PRINT"zero) that his/her f
     inal five-dice com-"
5010 LOCATE 2,9:PRINT"bination in that tur
     n counts in that"
5020 LOCATE 2,10:PRINT"particular category
     ."
5030 LOCATE 2,12:PRINT"No category may be
     used more than once"
5040 LOCATE 2,13:PRINT"in the course of th
     e game. The scoring"
5050 LOCATE 2,14:PRINT"is as follows:"
5060 PEN 3:LOCATE 2,16:PRINT"1";:PEN 1:PRI
     NT"'s,";:PEN 3:PRINT"2";
```

```
5070 PEN 1:PRINT"'s,";:PEN 3:PRINT"3";:PEN
     1:PRINT"'s,";
5080 PEN 3:PRINT"4";:PEN 1:PRINT"'s,";:PEN
     3:PRINT"5";:PEN 1
5090 PRINT"'s,";:PEN 3:PRINT"6";:PEN 1:PRI
     NT"'s : Score the"
5100 LOCATE 2,17:PRINT"Total of the number
     s that fit  in the"
5110 LOCATE 2,18:PRINT"category. Thus,if "
     ;
5120 PEN 3:PRINT"2";:PEN 1:PRINT"'s are ch
     osen, then"
5130 LOCATE 2,19:PRINT"the combination ";:
     PEN 3:PRINT"6-5-4-2-2 ";
5140 PEN 1:PRINT"would score ";:PEN 3:PRIN
     T"4"
5150 PEN 1:LOCATE 2,20:PRINT"for the two "
     ;:PEN 3:PRINT"2";
5160 PEN 1:PRINT"'s. The maximum score in"
5170 LOCATE 2,21:PRINT"the ";:PEN 3:PRINT"
     2";
5180 PEN 1:PRINT"'s category is,of course,
     ";
5190 PEN 3:PRINT"10";:PEN 1:PRINT" and in"
5200 LOCATE 2,22:PRINT"the ";:PEN 3:PRINT"
     6";
5210 PEN 1:PRINT"'s category, ";:PEN 3:PRI
     NT"30."
5220 LOCATE 8,24:PRINT"Press any key to co
     ntinue";:PEN 1:PRINT"!"
5230 a$=INKEY$:IF a$="" THEN 5230
5240 PAPER 0:CLS:PAPER 2
5250 LOCATE 15,2:PRINT "** ";:PEN 3:PRINT"
     YAHTZEE ";:PEN 1:PRINT"**"
5260 PEN 3:LOCATE 2,4:PRINT"Full House ";:
     PEN 1:PRINT"may be chosen only when t
     he"
5270 LOCATE 2,5:PRINT"roll represents thre
     e of a kind and a"
5280 LOCATE 2,6:PRINT"pair;it scores the t
     otal of the numbers"
5290 LOCATE 2,7:PRINT"which make up the ha
     nd,so that"
5300 PEN 3:LOCATE 2,8:PRINT"5-5-5-3-3 ";:P
     EN 1:PRINT"scores ";
5310 PEN 3:PRINT"21 ";:PEN 1:PRINT"for a F
     ull House.":PEN 3
5320 LOCATE 2,10:PRINT"Four of a kind ";:P
     EN 1:PRINT"also scores the total of"
5330 LOCATE 2,11:PRINT"the numbers on the
     dice, provided they"
5340 LOCATE 2,12:PRINT"include four of a k
     ind. If on the last"
5350 LOCATE 2,13:PRINT"roll a player has o
     nly the four of a "
5360 LOCATE 2,14:PRINT"kind category unuse
     d, and his roll is":PEN 3
5370 LOCATE 2,15:PRINT"5-5-5-4-2, ";:PEN 1
     :PRINT"then there would be no score"
5380 LOCATE 2,16:PRINT"at all.  If the roll
      were ";:PEN 3
5390 PRINT"5-5-5-5-2";:PEN 1:PRINT", the"
5400 LOCATE 2,17:PRINT"score would be ";:P
     EN 3:PRINT"22. ";
5410 PEN 1:PRINT"If the roll were ";:PEN 3
     :PRINT"5-5"
5420 LOCATE 2,18:PRINT"-5-5-5 ";:PEN 1:PRI
     NT"then this would count as a four"
5430 LOCATE 2,19:PRINT"of a kind and would
      score ";:PEN 3:PRINT"25."
5440 LOCATE 2,21:PRINT"Three of a kind ";:
     PEN 1:PRINT"would play the same ex-"
5450 LOCATE 2,22:PRINT"cept the limiting n
     umber would be three"
5460 LOCATE 2,23:PRINT"instead of four."
5470 PEN 3:LOCATE 8,25:PRINT"Press any key
      to continue";:PEN 1:PRINT"!"
5480 a$=INKEY$:IF a$="" THEN 5480
5490 PAPER 0:CLS:PAPER 2
5500 LOCATE 15,2:PRINT "** ";:PEN 3:PRINT"
     YAHTZEE ";:PEN 1:PRINT"**"
5510 LOCATE 2,4:PRINT"A ";:PEN 3:PRINT"Sma
     ll Straight ";
5520 PEN 1:PRINT"is ";:PEN 3:PRINT"1-2-3-4
     -5 ";:PEN 1:PRINT"and"
5530 LOCATE 2,5:PRINT"scores ";:PEN 3:PRIN
     T"30.";
5540 PEN 1:PRINT" A Large Straight is ";:P
     EN 3:PRINT"2-3-4-5-"
5550 LOCATE 2,6:PRINT"6 ";:PEN 1:PRINT"and
      scores ";:PEN 3:PRINT"40."
5560 LOCATE 2,8:PRINT"YAHTZEE ";:PEN 1:PRI
     NT"is any five of a kind and it"
5570 LOCATE 2,9:PRINT"scores ";:PEN 3:PRIN
     T"50."
5580 LOCATE 2,11:PRINT"Chance ";:PEN 1:PRI
     NT"is designed to give a player lee"
5590 LOCATE 2,12:PRINT"way for a bad roll.
      It counts the total"
5600 LOCATE 2,13:PRINT"on the dice. A play
     er who has already"
5610 LOCATE 2,14:PRINT"filled the ";:PEN 3
     :PRINT"6";
5620 PEN 1:PRINT"'s and ";:PEN 3:PRINT"Ful
     l House ";:PEN 1:PRINT"categor-"
5630 LOCATE 2,15:PRINT"ies, and who in goi
     ng out for a four of"
5640 LOCATE 2,16:PRINT"a kind and gets ";:
     PEN 3:PRINT"6-6-6-5-5, ";
5650 PEN 1:PRINT"may score it"
5660 LOCATE 2,17:PRINT"as ";:PEN 3:PRINT"C
     hance ";
5670 PEN 1:PRINT"and score ";:PEN 3:PRINT"
     28.":PEN 1
5680 LOCATE 2,19:PRINT"The categories ";:P
     EN 3:PRINT"1";:PEN 1:PRINT"'s,";
5690 PEN 3:PRINT"2";:PEN 1:PRINT"'s,and ";
     :PEN 3:PRINT"3";:PEN 1:PRINT"'s are u
     sed"
5700 LOCATE 2,20:PRINT"to cover bad rolls,
      as when a player"
5710 LOCATE 2,21:PRINT"goes out for one of
      the straights and"
5720 LOCATE 2,21:PRINT"misses.The player m
     ay as well enter the"
5730 LOCATE 2,22:PRINT"result in ";:PEN 3:
     PRINT"1";
5740 PEN 1:PRINT"'s,even if there is not a
     n"
5750 LOCATE 2,23:PRINT"ace in the roll. Wh
     en all players have"
5760 LOCATE 2,24:PRINT"had their turns,the
      highest score wins!"
5770 PEN 3:LOCATE 8,25:PRINT"Press any key
      to continue";:PEN 1:PRINT"!"
5780 a$=INKEY$:IF a$="" THEN 5780
5790 RETURN
```

Adventurer's Attic

"You enter a room full of letters, each written by a frustrated Adventurer and in a biologically treated ink which means instant death if touched -what next Ed?"
GET GLOVES
"Good thinking - what next?"
OPEN LETTERS
"They all have the same theme, start an Adventurer's column or your gloves will disintegrate - what next?"
GET ..um.. MOVING
"Wise move Ed"

It was somewhere in the middle of last year that the idea of such a column was mooted. Perhaps it was because most Amstrad users were new to computers and still going through their "games period" that the response was poor. Over the last few months however, many letters have been received from Adventurers who seem to be destined to roam the earth for eternity unless they can get a clue to save themselves.
Other readers have come to their rescue and by including this article, I hope The Amstrad User is instrumental in preventing a number of suicides. It is now up to all you Adventurers to keep the column going!

SOME ANSWERS

Tony Scott of Maffra had got stuck in the "Gems of Stradus" (December 1985), unable to get past the Alien.
The answer (from Andrew Ferguson) is to attack the Alien with the sword after you have sharpened it. He also advises to beware of the Mad House. K.F. Lane of Bracken Ridge adds that when past the Alien, make sure you have a bucket of water, lamp and matches.
Again, in the December issue, A.F. Williams of Windsor was unable to enter the manor in the game "The Secret of Bastow Manor". Rod Anderson of Camperdown says "once through the manor gates and inside the shed you are confronted with a square object that resembles a large box. To find out the name of this object, type the command 'L' and this will give a list of the objects available to the eye's field of vision. The object in the shed that has to be moved is the 'case'. MOVE CASE will reveal a trap-door to enter the manor. Don't forget to have the torch switched on."
Another tip from K.F. Lane is to have the branch to get over the pit.
Patrick Cahill of Forster went a little further. He advises that the articles inside the case should be with you before going through the trap-door. After GO TRAPDOOR you enter a room secured by a beam. You must REMOVE BEAM which the game responds with "I USE THE CROWBAR". This is a mistake, because even if you don't have the crowbar (which is found in a tool box beyond the pit full of snakes) tou can remove the beam. You then OPEN DOOR and GO DOOR and you will find yourself in the manor.

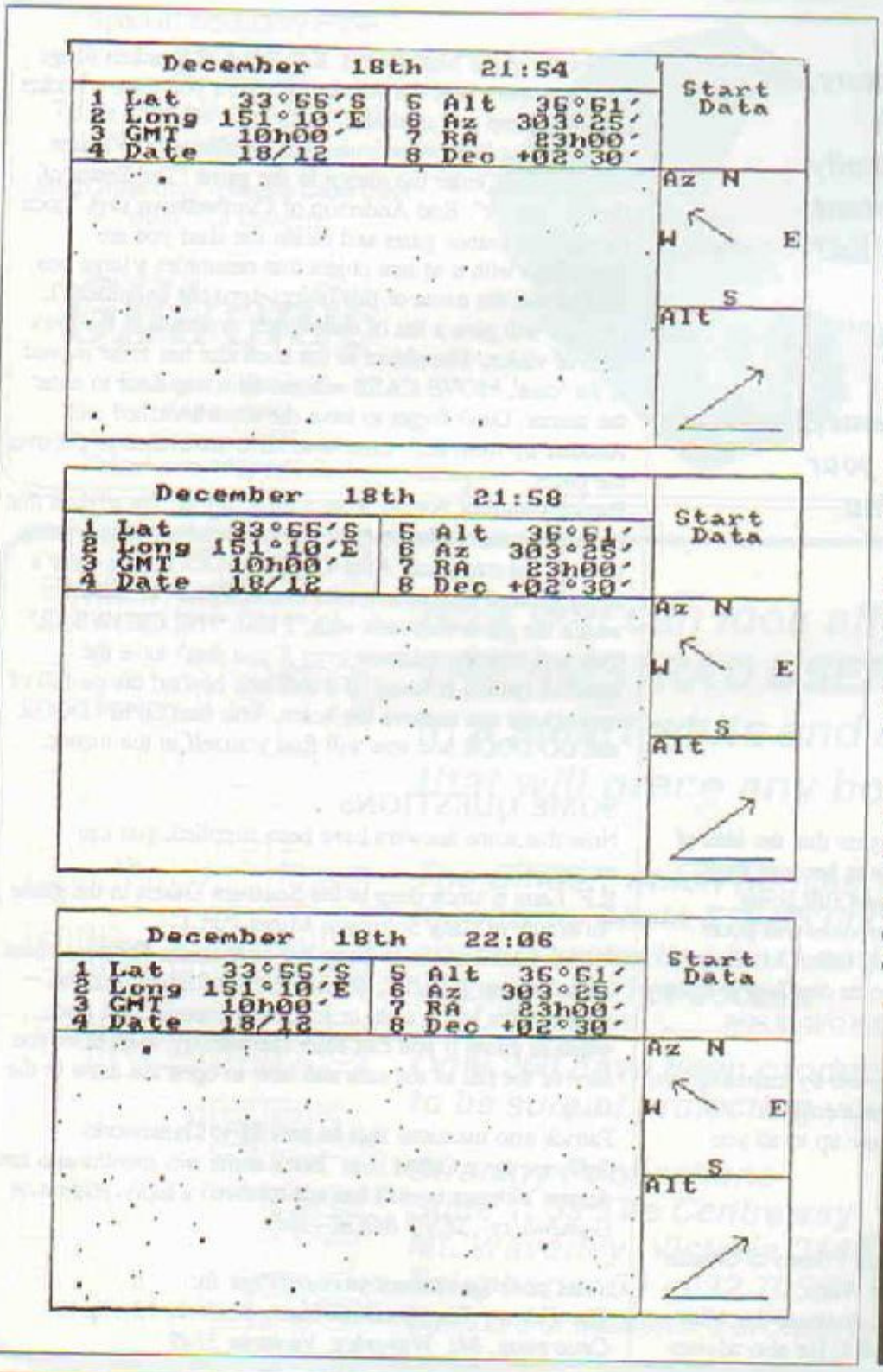
SOME QUESTIONS

Now that some answers have been supplied, you can reciprocate.
K.F. Lane is stuck deep in the Southern Desert in the game "In search of King Solomons Mines Part 1"
Patrick Cahill wants to know the code to the Treasure chest. So far he has 'IIE', 'VI', 'Right 3 Left 7 Right 2' and the message 'the left is right or is the right wrong'. He also wants to know if you can enter the balcony, even after you survive the fall of the safe and how to open the draw in the study.
Patrick also mentions that he sent $1 to Gameworks Software for a Coded Hint Book some two months ago and despite a chaser he still has not received a reply. *(Come on Gameworks - SEND BOOK - Ed).*

Send your Questions/Answers/Tips to:
The Editor, The Amstrad User, Suite 1, 33 The Centreway, Mt. Waverley, Victoria 3149

Halley-luiah!

by K.L. Webber



It had always been my ambition to observe Halley's comet on its return in 1985/6, this being my only opportunity.

I finally got hold of some binoculars in December and dialled 11622. The information given was to observe 65 deg. W of N and 30 deg. up from the horizon at 10.30 p.m.

Did I see the comet? No way!

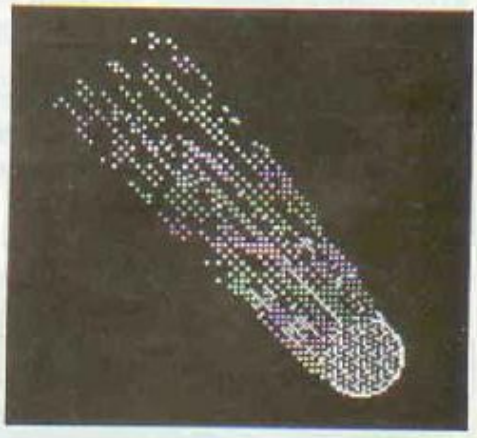
The following night I decide to put Starwatcher to work using a published graph of approximate comet position on various dates. Being December 18th, the indicated celestial co-ordinates were R.A. 23 hr and Dec. +4 deg. After setting up the co-ordinates at centre screen and making many trips between screen and sky to identify star patterns, I finally spotted what, at first, seemed like a smudge on the lens of the binoculars. The smudge, however, remained in the same place in the sky when I moved the binoculars. Using Cursor Mode, I determined the co-ordinates of the smudge to be R.A. 23 hr and Dec. +2 deg. 30 min.

The next day, Sydney Observatory confirmed the comet's position to be R.A. 22 hr 59 min. and Dec. +2 deg. 30 min. on the 18th.

Good enough! So, with Starwatcher to guide me, I had fulfilled my ambition.

I am now looking forward to the time when the comet emerges from behind the sun knowing Starwatcher will guarantee certain viewing.

The attached printouts show the section of sky observed with the comet's position at the centre of the display.



by Mark Godden

This month we will look at a powerful feature of the Amstrad computers concentrating on the 6128 in particular. This feature was demonstrated last month using the command |SCREENCOPY,n,n.

This command is just one of the many available through what are known as Resident System Extensions (RSX's).

I will begin by explaining what these are. Firstly, inside your computer is a chip called an INTERPRETER which acts like a dictionary to the computer; you type in the command CLS, the interpreter looks up the command, translates it, then executes a routine that is written in its own language (Machine Code) that has the same function as the command.

We can add new commands to the computer's 'dictionary' through the use of RSX's which are loaded from disc into RAM (Random Access Memory). Hence the name RESIDENT (they are now in memory) SYSTEM (part of the system) EXTENSIONS (have added new commands to the computer).

So when you switch on your computer they are not resident. Try it. Type in the command

|SCREENCOPY,1,2,

and you will find you get a message BAD COMMAND. The computer tried to execute the command but was unable to translate or find a routine for it. You may ask why didn't Amstrad add these commands to the INTERPRETER, well the answer is simple. If they had, your INTERPRETER would be different from the 664 and those terrific games your mates got for the 664 would probably not run on your 6128. With the use of RSX's, the 6128 owner has the power to utilize the second bank of 64k without losing compatibility with the 664.

If you are with me so far, we will now look specifically at the 6128's RSX commands. The list below shows all the RSX commands available with a brief description of each, and are broken into two sections. This is only an introduction to the 6128's RSX commands not CP/M RSX commands.

SCREEN COMMANDS

|SCREENCOPY,sn1,sn2

This command copies one complete screen (16k) to second bank, where sn1 is the destination screen number and sn2 is the source screen number.

|SCREENSWAP,sn1,sn2

This command swaps screens rather than copy them. It takes your original screen sn1 and replaces it with the screen held in sn2 (not overwrite it like the |SCREENCOPY,n,n command).

Note: the second bank is numbered in the following way:

Second Bank

0 - - - - - - - - - - - - - 65536

Normal screen No.1	16k No.2	16k No.3	16k No.4	16k No.5

There is an optional screen area parameter which we will introduce in next month's issue. It is fairly complex and requires an understanding of the screen memory - so read up if possible.

FILE COMMANDS

The file RSX commands create and use what is called a RAMfile. This is an area of memory (the second bank) where data is organized in a similar fashion to data on a floppy disc (ie - files) and because the operating system does not have to start the drive and move the reading head across the disc, data is retrieved at a very fast transfer rate.

To be able to understand these commands one needs to understand the basics of a file, record, fields & data.

Let's take a look at a basic file containg two records:

RECORD 1:

Field Names	Data
NAME	:M.C.GODDEN:
COMPUTER	:AMSTRAD 6128:

RECORD 2:

Field Names	Data
NAME	:C.P.UNIT :
COMPUTER	:AMSTRAD 664 :

So you can see that although the structure of each record is the same and the field names, size and position are also the same, the data can contain different information.

|BANKOPEN, record length

The length parameter can have a value of 2 - 255 and is a fixed length. The example above would use 22 as the length (ie - 10 for the NAME and 12 for the COMPUTER giving a record length of 22). This command also initialises the RAM-file by setting the

record number to 0.

|BANKWRITE,@status variable, string expression, record number

Function - writes data to second bank.

The status variable is given a value after returning from the write procedure (ie: -1 if no more memory available). The string expression may be a variable (NAME$) or a literal ("M.C. Godden") and last, the record number (if record number is not present the current record is written).

|BANKREAD,@status variable, string expression, record number

Function: reads data from second bank.

The status variable has the record number passed to it on return from the read procedure (-1 for above reasons), the string expression can only be a variable (NAME$) as you cannot read information into a literal, and record number is the same as the above except it is read from current record.

|BANKFIND,@status variable, search string, start record #, finish record #

Function: tries to find data in the second bank

The status variable upon return will have a value of -1 for the above reasons or -3 indicating no match was found. If a match is found it will contain the record number. The search string may be a variable (NAME$) or a literal ("FIND ME") and first and last record numbers define the area of RAMfile you wish to search. If omitted then the bank is searched from the current record number to end of file (EOF).

Note: The status variable can return from a procedure with the value -2. This means that a Bank Switching error has occurred, but using the famous words of the manual "THIS SHOULD NEVER HAPPEN".

That completes the introduction of the 6128's RSX commands. They are very easy to use but sticking to a few strict rules will avoid any data being corrupted - we will look at these rules in the coming months.

CP/M+

Well this month there isn't any but soon as I hear from readers and users who have had problems with CP/M+ or need some information about a TPA then my word processor will fire up.

TRIVIA

I await your discoveries . . . . . .

It's early days yet for this column, but if you have any information you feel would be of help to other 6128 owners, or wish to take advantage of Mark's offer to help with problems, then address your comments to the 6128 Segment. Mark will do his best to answer as many as possible through this column, but regrets that he cannot enter into personal correspondence.

Unihammer Printer Utility

by Petr Lukes

This routine shows how to produce true lower case characters. The trade-off is greatly increased printing time on an already slow printer.The program is written for the Tandy LPVII which uses control codes to toggle between the character mode and the graphic mode, and the eighth bit to indicate a graphic byte. The DMP-1 uses escape codes to toggle the graphic mode and the bit values are different, but the principle is the same.

The characters are formed in a 6 column by 7 row dot matrix. Each column is sent out as one byte, its value indicating which dot in the column is set. For the Tandy printer, bit 7 must be set to indicate a graphic column, and the bits are numbers from the top.

The sample printout was done on my ancient LP VII, through the home-made serial interface and serial driver, to make it possible to send out the necessary eighths bit.

I believe that for printers using the

The Seikosha printers are being sold under many different names, e.g. Commodore, Tandy LP VII/DMP 100, and as the AMSTRAD DMP-1 printer. They all have an abysmal lower case font, but they are cheap and quite adequate for listings and similar non-critical use.

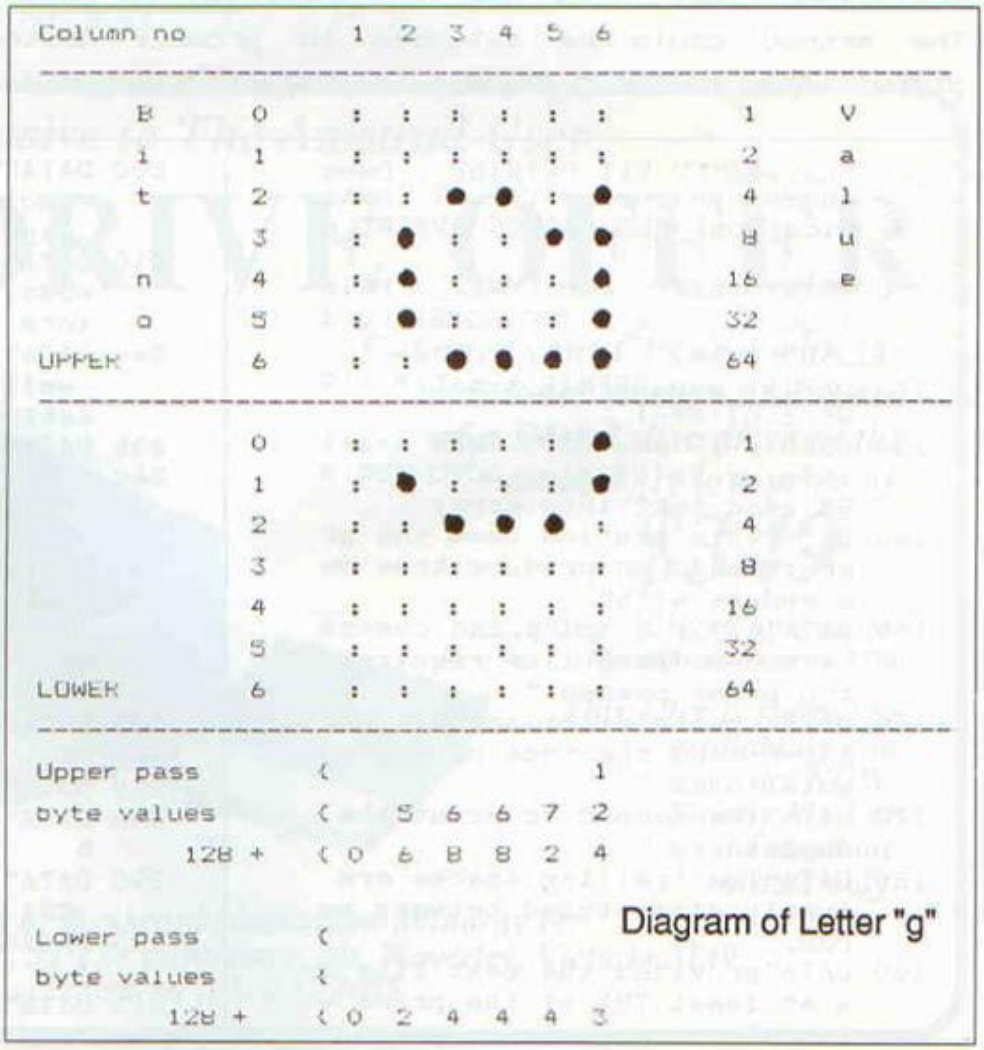


Diagram of Letter "g"

Epson-type escape codes (the AMSTRAD) the following changes should work:

The printer is normally in character mode, therefore `CHR$(30)` is not necessary.

For a special character, replace

```
PRINT#8,CHR$(18)z$(a);
```

by

```
PRINT#8,CHR$(27)"K"
CHR$(6)CHR$(0)z$(a);
```

For justification spaces:

```
PRINT#8,CHR$(27)"K"CHR$
(g)CHR$(0)STRING$(g,0);
```

The line feed between passes must be set to 9 lines/inch (no gap between them): replace

```
PRINT#8,ug;CHR$(13)uc;
```

by

```
PRINT#8,CHR$(27)"J"CHR$
(13)CHR$(13);
```

The bits are numbered from bottom up: the lowest one has value 1, the topmost one value 64. It is not necessary to add 128 to the bit sum (the interface will ignore it). The manual and a little experimentation should solve the problem. The changes work on my GEMINI-10x which uses the "standard" codes.

It should not be too difficult to patch the routine into any BASIC text editor. It will not replace daisy wheels, but the printouts become more acceptable.

Sample Printout

```
LP VII PRINTER : Descenders and proportional justification. LKS 850308
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
This routine uses the graphic mode to produce true descenders with
g,j,p,q and y, and custom a and u. Each line requires two print passes,
one to print normal characters and the tops of the special ones,
the second to print the descenders.
The trailing spaces are evenly distributed between words,
provided the text line is at least 75% of the print width.
The three lines above are not justified to avoid wide gaps between words.
The method could be extended to produce custom characters and provide
proportional printing as well as proportional justification.
```

```
100 CLS:w$="LP VII PRINTER : Desc
    enders and proportional justi
    fication. LKS 850308":PRINT w
    $
110 PRINT:PRINT"Petr LUKES,":PRIN
    T"26 Noll St., TOOWOOMBA, Q 4
    350":PRINT
120 DEFINT a-p:DEFSTR t-z:DIM t(9
    9),z(6):n=-1
130 READ z:IF z<>"END" THEN n=n+1
    :t(n)=z:PRINT t(n):GOTO 130:R
    EM read text into array
140 DATA"This routine uses the gr
    aphic mode to produce true de
    scenders with"
150 DATA"g,j,p,q and y,and custom
     a and u. Each line requires
    two print passes,"
160 DATA"one to print normal char
    acters and the tops of the sp
    ecial ones,"
170 DATA"the second to print the
    descenders."
180 DATA"The trailing spaces are
    evenly distributed between wo
    rds,"
190 DATA"provided the text line i
    s at least 75% of the print w
    idth."
200 DATA"The three lines above ar
    e not justified to avoid wide
     gaps between words."
210 DATA"The method could be exte
    nded to produce custom charac
    ters and provide"
220 DATA"proportional printing as
     well as proportional justifi
    cation."
230 DATA"END"
240 PRINT:PRINT"Assembling specia
    l characters":RESTORE 240
250 FOR a=0 TO 6:READ x:PRINT x:F
    OR b=1 TO 5:READ c:z(a)=z(a)+
    CHR$(c+128):NEXT b:z(a)=CHR$(
    128)+z(a):NEXT a:REM Read spe
    cial characters into array,ad
    ding a leading blank
260 DATA"a, top of g,q",56,68,68,
    72,124
270 DATA"top of j",0,0,0,4,125
280 DATA"top of p",124,72,68,68,5
    6
290 DATA"u, top of y",60,64,64,64
    ,124
300 DATA"descender of g,y",2,4,4,
    4,3
310 DATA"descender of p",7,0,0,0,
    0
```

```
320 DATA"descender of q",0,0,0,0,
    7
330 INPUT"ENTER to print";z:pr=79
    :ug=CHR$(18):uc=CHR$(30):ub=C
    HR$(128):y="agjpquy":PRINT w:
    PRINT#8,uc;w:PRINT#8,STRING$(
    pr,":"):REM ug=toggle graphic
     mode, uc=toggle character mo
    de, ub=graphic blank, y=list
    of special characters
340 FOR a=0 TO n:z=LEFT$(t(a),pr)
    :PRINT z:L=LEN(z):b=pr-L:j=(b
    <FIX(pr/4)):IF j THEN p=0:FOR
     c=1 TO l:x=MID$(z,c,1):p=p-(
    x=" "):NEXT c:q=b*6+p:j=(q>0)
     AND (p>0):REM For each line
    of text:q=graphic columns for
     proportioning
350 FOR b=-1 TO 0:e=q:f=p:FOR c=1
     TO l:x=MID$(z,c,1):IF j AND
    x=" " THEN g=FIX(e/f):f=f-1:e
    =e-g:PRINT#8,ug;STRING$(g,ub)
    ;uc;:GOTO 390:REM For each pa
    ss:for each character:if spac
    e then print proportioning bl
    ank columns
360 k=INSTR(y,x):IF k=0 THEN PRIN
    T#8,ub;uc;:IF b THEN 390 ELSE
     x=" ":GOTO 390 ELSE PRINT#8,
    ug;:REM If a normal character
    ,toggle character mode:if upp
    er pass,print normal characte
    r else (lower pass) print spa
    ce else (special character) t
    oggle graphic mode
370 IF b THEN IF k=1 OR k=2 OR k=
    5 THEN x=z(0) ELSE IF k=3 THE
    N x=z(1) ELSE IF k=4 THEN x=z
    (2) ELSE IF k=6 OR k=7 THEN x
    =z(3):REM Special character,u
    pper pass
380 IF NOT b THEN IF k=2 OR k=3 O
    R k=7 THEN x=z(4) ELSE IF k=4
     THEN x=z(5) ELSE IF k=5 THEN
     x=z(6) ELSE x=STRING$(6,ub):
    REM Special character,lower p
    ass
390 PRINT#8,x;:NEXT c:PRINT#8,ug;
    CHR$(13);uc;:NEXT b:NEXT a:PR
    INT#8,STRING$(pr,":"):REM Pri
    nt character. At end of each
    pass a graphic line feed, at
    end of line a character line
    feed
400 STOP
```