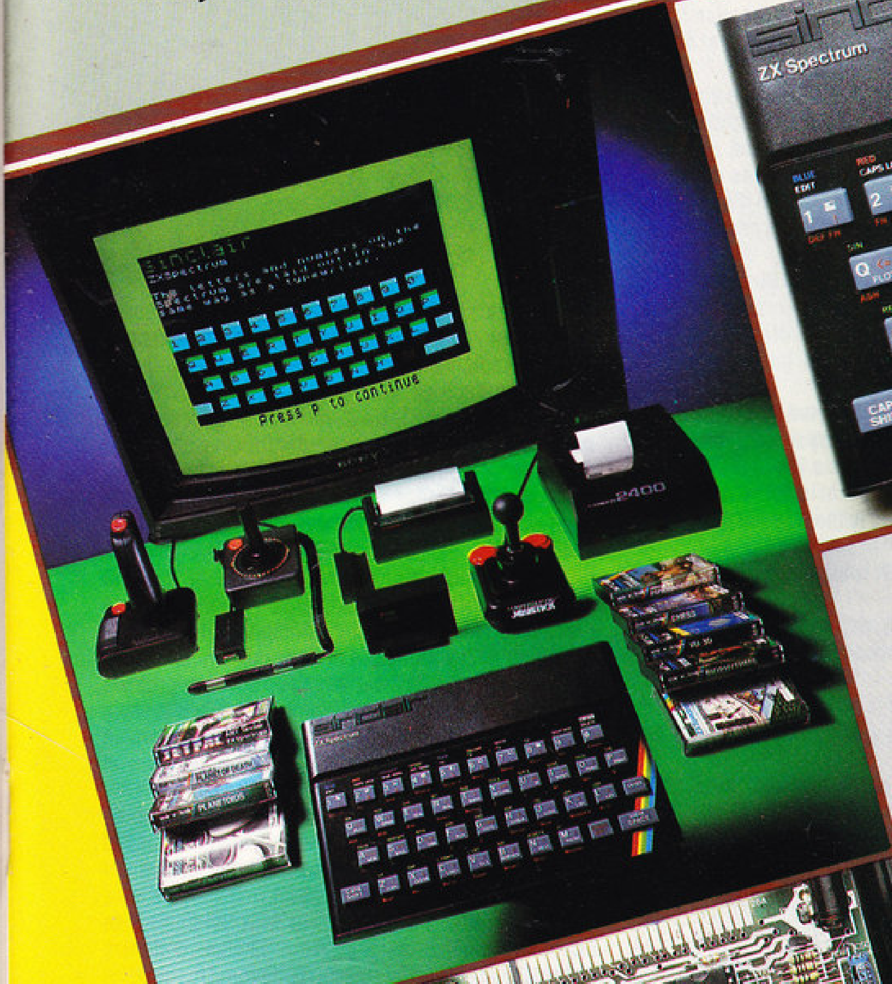
PCN micropaedia

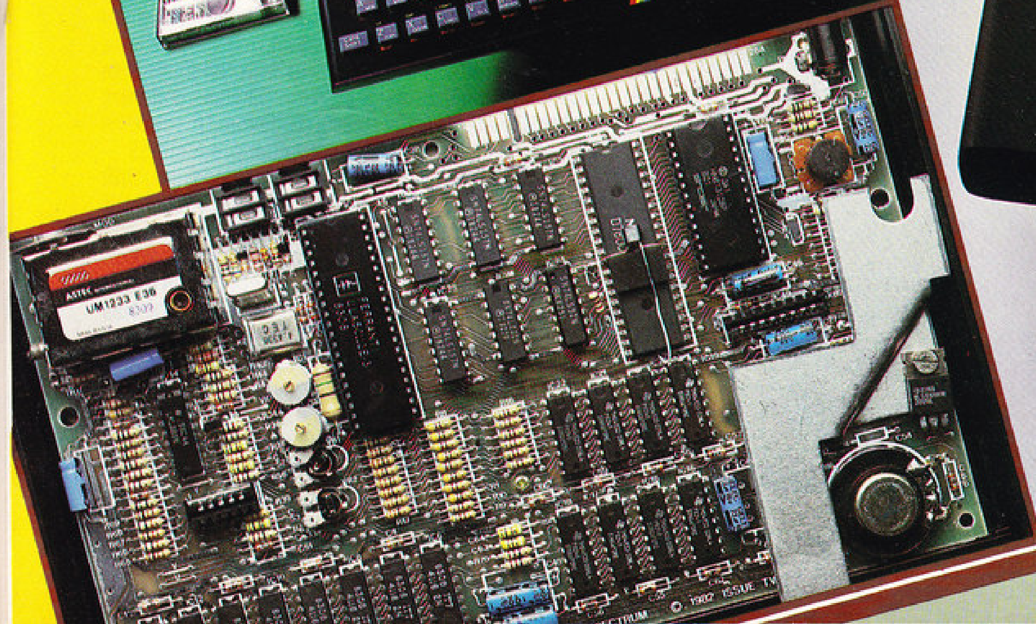
Volume 6 Part 1

PULL OUT AND KEEP

SPECTRUM

Inside, outside...





...and all round the machine.

Over the rainbow

Twenty years ago you'd buy an electric guitar, hitch to London, and try to be a rock and roll star. Nowadays the young hopeful is more likely to get a Spectrum and a ZX printer, then hit the stardust trail. There are a lot more of them, and it doesn't cost as much as a good electric guitar.

More than any other micro manufacturer, Sinclair has produced the people's computer — first with the ZX81, then with the Spectrum. Low prices have bred volume sales, which have bred excellent software support — and a whole new generation of young computer users and programmers.

It's easy to see the impact the Spectrum has had on the micro market — the machine hasn't been in existence long, and became generally available still more recently, yet a new software manufacturer like Quicksilva has been able to tap a well of programming skill no-one suspected existed.

The machine already has 'state of the art games' like Timegate and the Psion Flight Simulator to its credit, and software production in general is moving away from the Space Invaders/PacMan clones towards genuine originality.

But the fact that the machine has gone down so well with games manufacturers is only partially attributable to its wide availability.

It offers good high-resolution and user-defined colour graphics. The first reaction of the new Spectrum owner is invariably to rush ahead and design 21 different varieties of Space Invader — great fun, yes, but there's more to it than that.

It's also relatively simple to use the same facilities to redesign the Spectrum's alphabet — so you can tap (well OK, wobble) away in almost any arcane script you care to mention.

It's all too easy to get over-enthusiastic about the Spectrum's games and graphics capabilities. When the machine first came out, the pundits hailed it with cries of 'Uncle Clive's done it again.' It was seen as the best you could get for the price — and it's still a lot better than many machines at twice the price — and reviewers toyed with the possibility of it being used as a small business machine.

But much of this hinged on the Microdrives which, on passing their first birthday, were still to all intents and purposes a pious hope. (More details on the Microdrives next week).

Business software manufacturers held fire, waiting for the storage revolution, and the Spectrum spiralled off into the — still highly lucrative — ghetto of games/hobbyist machine.

But what a ghetto! The Spectrum is cheap, available and well-supported by software. Its Basic is workmanlike and thoroughly debugged, and its single-key entry system, although alien and displeasing to many a hard-bitten veteran programmer, is a joy to the novice.

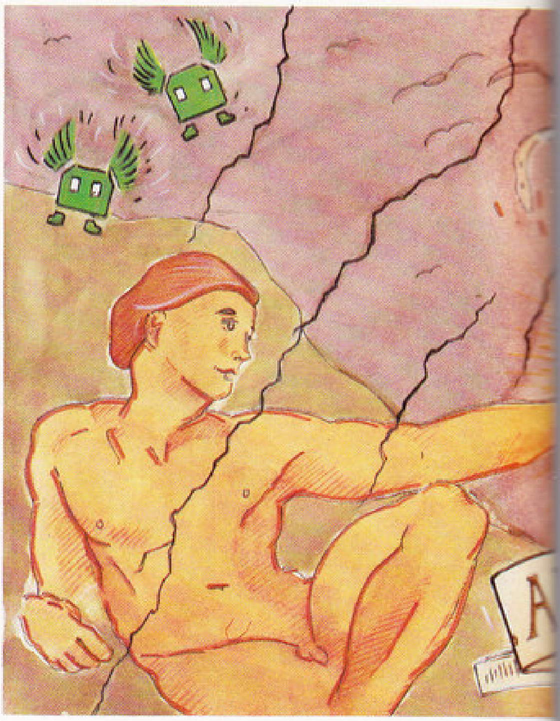
The machine's grim determination to forbid you to enter any line that is not syntactically correct is also — usually — a giant step for user-friendliness, as is the way the Spectrum automatically spaces keywords in the program listing.

So the basic machine, plus the 'cheap and cheerful' ZX printer, is all the novice needs to learn how to program in glorious Specnicolor.

Sinclair is also promising a low-cost cartridge system which, depending on the sort of software that is offered, will make a considerable difference to the machine's capabilities.

Certainly Sinclair machines *are* different, and the Spectrum is no exception, but there are now enough of them about for users to be confident that they won't be stranded with an outmoded white elephant.

Who knows, in two years' time there could be millions of people out there with souped-up Spectrums, cartridge software and the flat-tube TV.



Man be

Clive Sinclair was born with solder in his veins. At the age of 22 he founded his first company, Sinclair Radionics, which made radio and amplifier kits for the electrical hobbyist.

Within six years 'Uncle Clive' — as he affectionately became known in the computer business — had developed a reputation for innovative small business. His 'small' business in the late sixties included a palm-sized transistor radio and a range of amplifier kits.

The little radio and his 1972 release of the first pocket calculator — The Executive — confirmed his predilection for miniaturising consumer products and selling them at a miniature price in mass quantities.

By 1976 Mr Sinclair was taking on so many projects that he found himself beginning to take a few losses; chip supplies for his Black Watch digital wristwatch were drying up and his pocket TV project was in trouble.

The National Enterprise Board agreed to put some money into the kitty to help Mr Sinclair out of his project problems and the 2-inch screen pocket TV made it to the marketplace in 1977.

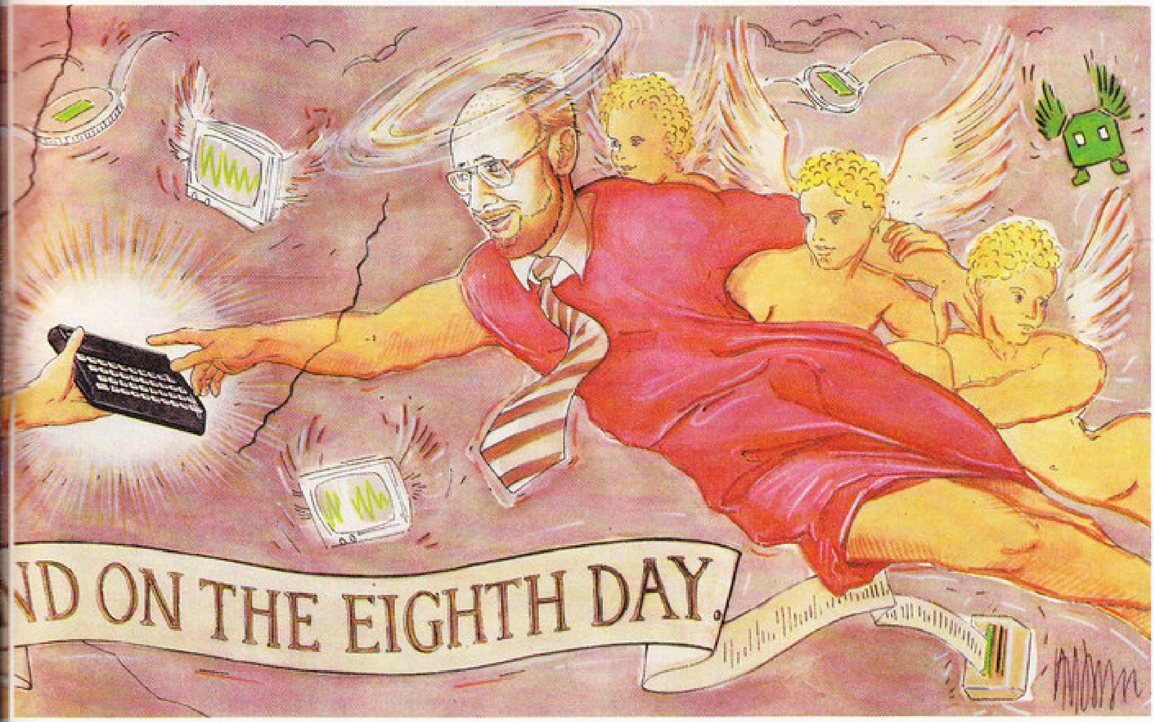
In 1979, after a disagreement with the Enterprise Board, Mr Sinclair resigned from Sinclair Radionics and established Sinclair Research, devoted to consumer electronics and personal computer production.

In 1980, Sinclair Research launched its first computer, the ZX80 — followed one year later by a modified version called the ZX81 — both were small black and white computers selling for under £100.

In April 1982, the ZX Spectrum was launched to add a cheap colour computer to Sinclair's range. It was met with rave reviews and hailed as a breakthrough in low-priced colour computing.

In an interview with *PCN* in May 1983, Mr Sinclair said that he thought his computers — at least the ZX81 model — survived because they were inexpensive.

'If the '81 were anything but highly competitive, it would cease to survive,' he said. And he added that



hind the machine

price reductions — such as the spring 1983 cut on the Spectrum's price — were made whenever possible to keep the machines competitive.

'It's our policy to reduce prices when we're able to and not wait. When costs come down we bring down our prices — we do this even if the product's selling very well, as it was in the case of the Spectrum. If we failed to reduce the prices under those conditions it would simply attract more competition.'

Because the machine is inexpensive, and it's 'soft' keyboard makes it quite bashable, Mr Sinclair said he's surprised more schools aren't using the Spectrum for teaching. 'Schools tend to like to have machines that are designed for schools even if they're not as good as other machines available,' he said.

He added that the BBC Micro was recommended for secondary schools and that primary schools were advised to follow suit in order to keep the software and interfaces consistent across schools. Mr Sinclair also said that educators and school districts were led to believe they were getting a good deal in buying BBCs, although when the price of VDUs was added in, the savings were comparatively small.

'Really everything was against the Spectrum on that one. We are most anxious to serve the needs of the educational world. We've got to rethink our strategy.'

With the incredible success of his computers, Mr Sinclair has been drawn away from a day-to-day role in the development of computers at Sinclair Research and now fulfils more of a 'policy-making' function that keeps him at the company's London office talking about plans for electric cars, flat-screen TVs and business-type micros.

He has also been looked on by the British financial community as something of a guru and has been known to give his opinions on government policy towards British computer manufacturers.

The everyday running of Sinclair's microcomputer side is now left to Sinclair Research managing director Nigel Searle (interviewed at right). And Mr Searle has taken quite an interest in further development of the Spectrum, announcing the development of plug-in cartridge software for the machine, the release of the Microdrive low-cost storage unit and possible updates to the machine's video circuitry.

Mr Searle has been with Sinclair since 1972 and for a while looked after Sinclair's operations in the US. Sinclair now has a technology licensing agreement with Timex, and all its microcomputer sales in North America are taken care of by that company.

The ZX81 is now available in North America as the Timex TS1000, while the Spectrum is to be launched there as the TS2000 — and an interim TS1500 machine which incorporates some of the features of both the machines will sell for a price inbetween them.

It wasn't always that way. When Sinclair started selling ZX81s to schools two years ago, more than 2,300 of them took advantage of its subsidised computer buying scheme. The company sold the ZX81s to schools at half-price — with a 16K RAMpack included.

As of spring 1983, more than 25,000 ZX81s had been sold to schools and Sinclair had been encouraging the development of educational software through grants and special offers to summer computer camps for children. One computer camp operator even had an offer that gave camp visitors a free ZX81 for attending a certain number of lessons.

Sinclair Research hoped for similar success with the Spectrum in schools, although statistics indicate that BBC micros make up 80 per cent of that market. Part of Sinclair's hopes were pinned on a £15 million scheme for 27,000 UK schools that would give Spectrums to one and all.

The offer gave every school that ordered a Spectrum under the government's Micros in Schools project a free ZX printer, a copy of Logo and ten discount vouchers valid until 1984. Sinclair still hopes for a bigger take-up rate among schools as this offer continues.

Looking to the future

Aside from Clive Sinclair, who is increasingly occupied by his company's other projects, Nigel Searle is the top man at Sinclair Research and has his finger on the future of the Spectrum. He spoke to *PCN* in April 1983 about what that future holds.

PCN: **Given the declining cost of colour on computers, what do you see as the future of the Spectrum?**

Searle: **I think the Spectrum will need some attention from an engineering point of view to make sure that it's designed at minimal cost and we'll look at ways to see if we can build the same product at a lower cost. But I think if we do that it will be a competitive product for a long time.**

Certainly it will still be around two or three years from now, which would give it a total life of about four years. It hasn't got a built-in display, as some other products plan to, but they will have only limited monochrome displays and all the evidence suggests that the single most popular aspect of the Spectrum is that it's a colour machine.

PCN: **Does the same argument apply to the falling price of extra RAM? Will there be a time soon when all Spectrums will come standard with 48K?**

Searle: **I think that depends on the market. Although people have to spend only £30 more to buy a 48K instead of a 16K Spectrum it's still important to have the £100 unit out because it's that price that attracts people. The 16K machine may become an insignificant proportion of the sales. On the other hand, we'll be bringing out a lot of software for the Spectrum in ROM cartridges, and that means that you no longer need to use the internal RAM space for programs. (*PCN* May 13-May 20).**

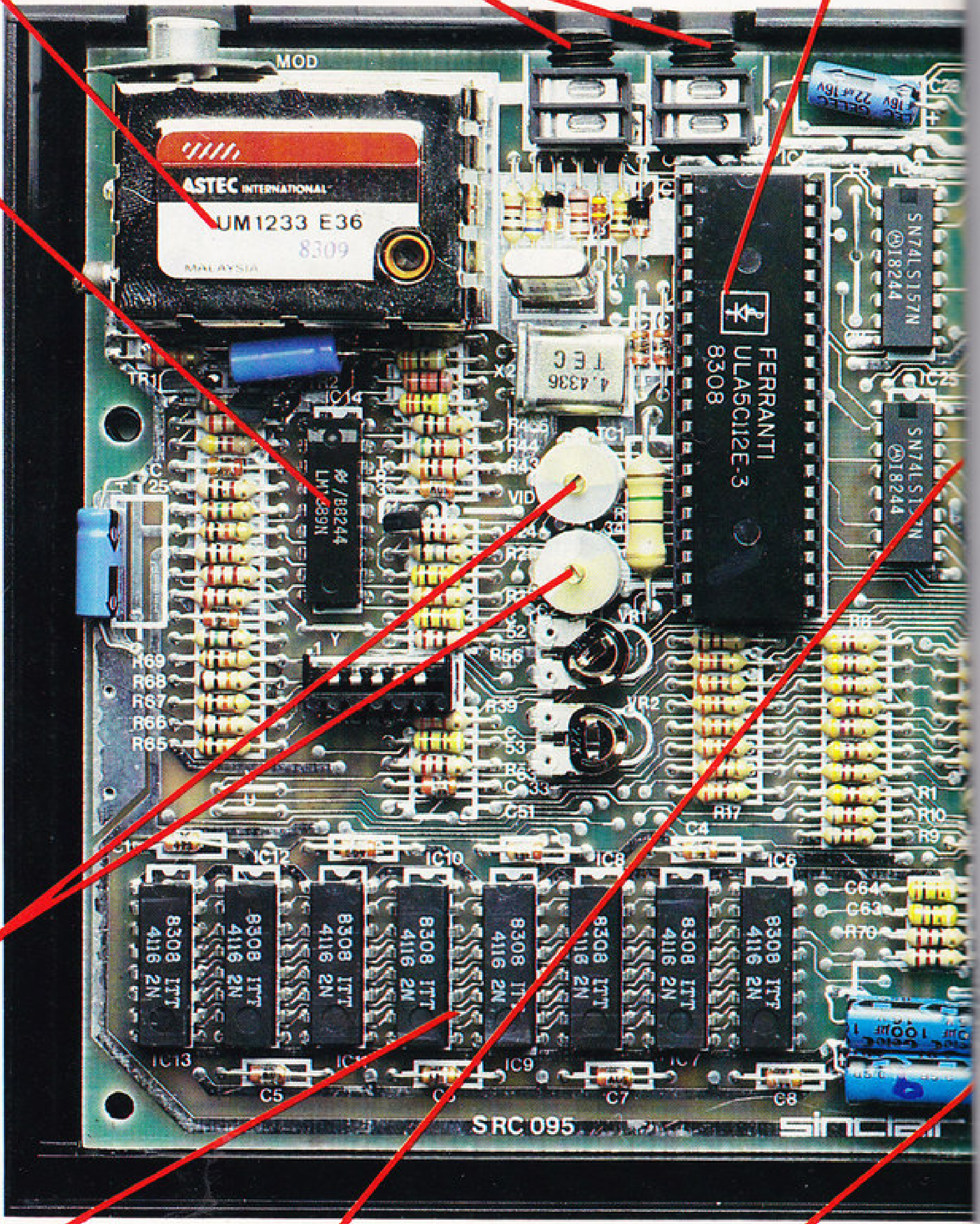
UHF or VHF modulator — A standard piece of hardware for any computer that aims to run off a domestic television. The modulator takes the video signal produced by the Sinclair's video circuitry and converts it to a UHF or VHF signal that can be read by the television. The Spectrum usually outputs a signal on UHF Channel 36 and can be tuned in from most modern televisions. A small signal adjustment on the bottom right-hand corner of the modulator can be fine-tuned.

Ear and microphone sockets — The lifeline for data coming in and out of the computer from cassette tape storage. The ear and microphone sockets are tied into the video circuitry so that when LOADing or SAVEing programs a rolling or squiggled pattern is produced on screen. Both sockets take a standard 3.5mm jack-plug. Either socket can be used as an additional sound output, although Sinclair recommends the microphone socket.

ULA (Uncommitted Logic Array) — This acts as the 'translator' for all information coming from inputs or leaving to outputs. The ULA gets information directly from keyboard and cassette inputs, and copies the screen output from the video memory to the video output circuit 50 times per second. The ULA was designed to replace the groups of smaller logic integrated circuits that characterised earlier computers.

Video circuit — The Spectrum's video circuitry is based in the LM1889N integrated circuit and accepts blue-yellow and red-yellow colour difference signals from the ULA to produce a single colour output signal. The two adjustment resistors to the right and below the LM1889N alter the relative amplification of the red-yellow and blue-yellow signals. If you alter them, you can vary the colour quality or grey scale that you see on the screen.

INSIDE THE SPECTRUM



The ULA 14Mhz clock — Adjustment of the clock is sometimes necessary to remove striations on the side of characters. By adjusting the screw underneath the screw hole at VC1, you can readjust the channel set on your television. But you must let the Spectrum warm up before you try and adjust the clock. As the crystal warms up, it expands, and this alters its frequency. What this means is you could end up tuning it up for the state it's in least of the time.

16K RAM — This is the basic supply of memory that comes with all Spectrums and resides on the left-hand side of the machine. These are the 2K chips arranged in rows of eight (8×2=16) to combine for the minimum memory. They are in locations IC14 to IC8. Although this may not seem much for a colour computer, the 16K is a considerable step up from the 1K that came standard with Sinclair's ZX81 computer.

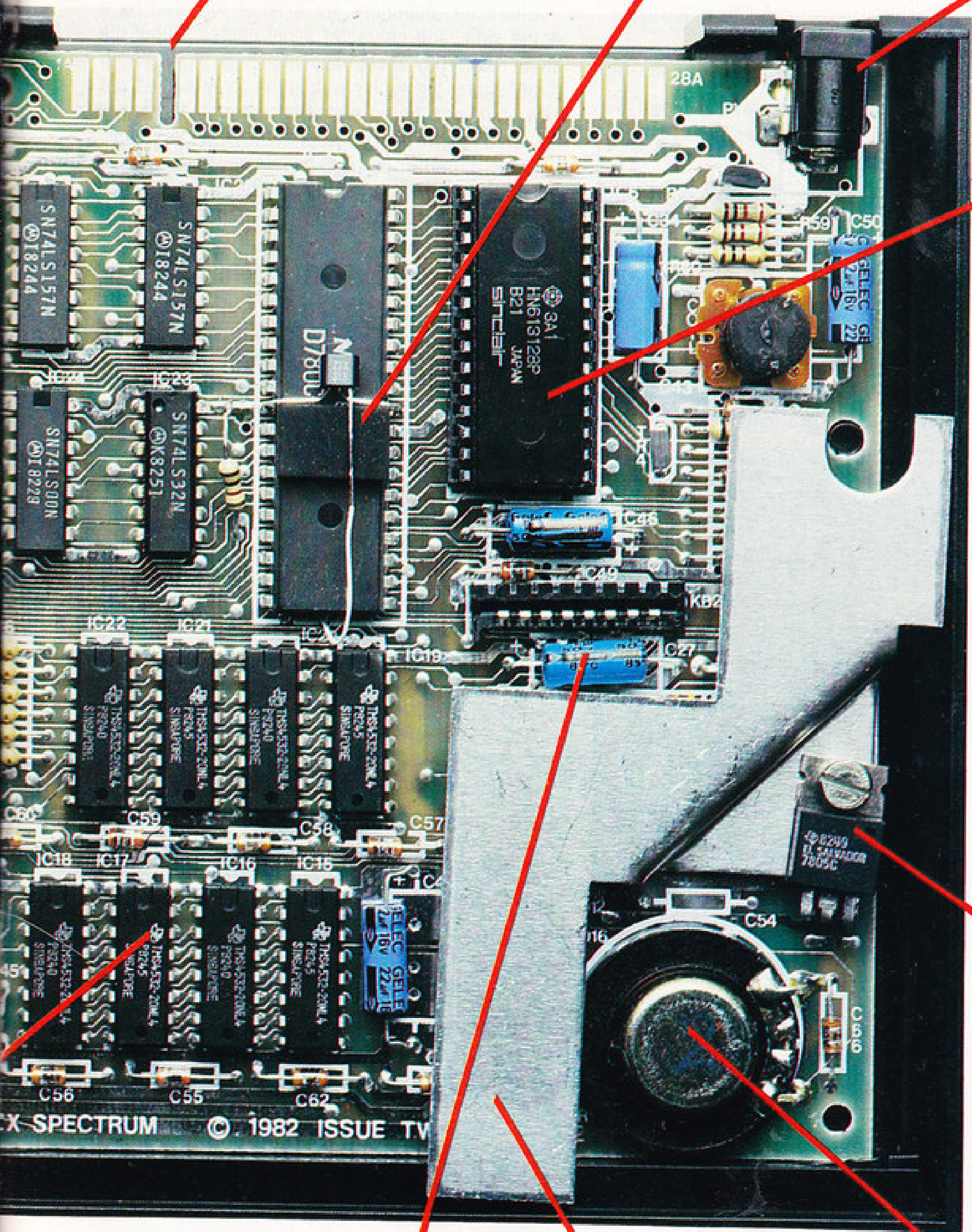
Logic gates — The chips at the middle and top of the machine are the logic gates. They perform the pedestrian function of turning switches on and off. They decide which piece of memory is going to be addressed. This being the 'basis' of how a computer 'thinks' you can see you wouldn't get very far without them. It's also worth noting that the 16K machine needs fewer of these than the 48K.

32K RAM — There are eight 4K chips in these expansion sockets, completing the upgrade from a 16K to 48K machine. They complement the 16K chips at the left of the machine and can be installed with the 48K upgrade kit — although with the falling price of memory and the increasing numbers of programs that take advantage of larger memory, fewer 16K machines have been sold than the 48K variety.

Edge connector — At the rear of the Spectrum is a 28-way double-sided connector which can be used to plug into a variety of peripherals that currently include the ZX printer, several memory expansion packs and interface modules for full-size printers and joysticks. Make sure that any connector you put on this edge fits snugly and doesn't have any wobble. A poor connection will interfere with your work, and might even blow a chip.

Z80A CPU — The Central Processing Unit. Sinclair's choice of the Z80A has given the Spectrum a fast and powerful 8-bit processor — running at up to 4Mhz as opposed to the 2.5Mhz speed of the Z80. The Z80A also has a better register set and a bigger instruction set than its competitor the 6502 — the processor used on the BBC micro. But it's slower than the 6502, and the Spectrum's Z80A can't run the CP/M business operating system.

Power socket — The Spectrum uses a 9V power supply with a current of up to 1.2 amps. The power supply plug on the Spectrum also acts as the on-off switch and can provide a hard re-set when Break just won't do. The power supply itself is an independent unit which converts normal household voltage. When that current reaches the Spectrum it must be carefully regulated (see the description of the voltage regulator below).



Basic — You'll find the chip containing Sinclair Basic at location IC5. It has 16K ROM, 28 pins with 14 address lines, eight data lines, two chip-select pins, and one output to enable data to be read by the CPU and power supply connections. (The power supply connection is at the top right-hand corner of the board). The Basic chip interprets the language in which your program is written so that the CPU can understand and carry out your instructions.

There are several different versions of the Spectrum board. This is an Issue 2 board released with a Spectrum sold in May 1983. On the Issue 1 board, the Basic and CPU chips are lined up parallel to one another, with the Basic sitting where the CPU sits in the Issue 2 and the CPU where the 32K chips are here.

In the near future the video circuitry may also be updated to allow it to work with a wider range of televisions. The televisions currently recommended for use with the Spectrum are those in the Sony Trinitron series, which seems to have the best hold on the sometimes erratic Spectrum colour signal.

The Spectrum also has no remote tape control plug-in, to stop and start the tape before and after SAVE programs, but a number of peripheral manufacturers make devices that give you that tape control through the edge connector on the back.

Voltage regulator — Because none of the Spectrum's chips use a 9V power supply, they must all be carefully regulated to receive just the right volt and amperage. The CPU and logic gates use a +5V supply, the ULA needs +12V and the video memory chips need +12V, +5V, and −5V simultaneously. The regulator starts by accepting +9V and begins its output with +5V. The other 4V is dealt with and dispersed by the heat sink.

Keyboard plug-ins — The Spectrum keyboard plugs into the main machine at two places. One of these is on the left-hand side of the keyboard, and one on the right. The keyboard itself is a five by eight grid of wires — this may come as a surprise if you're used to looking at the outside — and their crossing points are connected by pressing a key. The signal is then transmitted through the grid to the micro.

Heatsink — This is the large sideways Z-shaped aluminium plate dominating the bottom right-hand corner of the Spectrum. The heat sink's role is to absorb the 4V that the power supply doesn't need. It puts the extra power through a large resistance, which converts it to heat. This explains why your Spectrum gets so hot when you leave it running for extended periods of time. So don't worry — it's all part of the design.

Speaker — The speaker on the Spectrum is not really a speaker at all, but is actually a small electric buzzer. Because it isn't a real speaker, and because it sits face down in the machine with the sound coming out the bottom rather than the top, it is much quieter than many other similar machines. It is connected to the PCB through two small wires and hides just next to the large 'heat sink'.

Sinclair Basic is one of the most enticing, as well as one of the most controversial aspects of the Spectrum. Although it has a lot in common with other Basics, the machine's use of special 'keywords', rather than simply characters typed in through the letters on the keyboard, is more than some people can get used to.

If you have no objection to the keywords, and unless you've used another computer beforehand you're not likely to, you'll find that the Spectrum's Basic has some nice extensions.

Keywords like CIRCLE, which let you draw a circle merely by naming the x and y co-ordinates of the centre and giving the radius, PAPER which lets you change the main background colour simply by giving a different number for the PAPER, and INK which does the same for the text.

All the operating system commands such as LOAD, SAVE and PRINT can also be found as keywords.

Basically speaking . . .

With the proliferation of ZX Spectrums, Spectrum Basic is rapidly becoming one of the most widely used Basic dialects. Any high level language has a number of general features and the following is a consideration of these on the Spectrum.

Data types and data structures

The Spectrum allows for two main data types — floating point numeric, and character — and arrays may be formed of both.

Integer numbers have their own special internal format, but this is transparent to the programmer: there is no way of specifying a number as an integer, as on the Oric for example.

Numeric identifiers must start with a letter which may then be optionally followed by an arbitrary number of letters or digits. However, control identifiers (*ie* those used as an index in FOR loops) may only consist of a single letter.

Similarly, numeric arrays are identified by a single letter; they are declared by a DIM statement (one per array); they may have an arbitrary number of dimensions, and are subscripted starting at 1.

Character identifiers are a letter followed by a dollar sign, and character array names follow the same rule. They again have an arbitrary number of dimensions, but each string in the array is of a fixed size, this size being specified when the array is declared, as the final number in the DIM statement.

Binary numbers can also be referred to using BIN, but this does not constitute another data type. All data structure space may be deleted using the CLEAR instruction.

Assignments and calculations

Spectrum Basic uses the LET statement for assignments and calculations, the LET not being optional as in many Basics. Calculation expressions may also appear in many other instructions in place of a simple number, the evaluation being carried out when the instruction is obeyed.

The arithmetic operators used in numeric expressions are the familiar set:

↑ / * — +

and the priority of their evaluation is in the order that they appear above.

The plus sign may be used as a character operator, producing concatenation of strings. Strings may also be split into smaller parts using the keyword TO to slice them.

Character and string expressions may not be mixed, but of the several functions that can be used in expressions some may be used to convert from character to numeric format and vice versa.

Control statements

The Spectrum allows conditional and unconditional transfers of program control, using the IF and GO TO instructions respectively. The IF statement may have the form:

IF condition THEN action
or IF condition THEN action 1 ELSE action 2

Multiple statements separated by colons may be included after THEN, but care must be taken in the use of this format, since *all* the instructions after THEN up to the end of the line will be executed only if the condition is true.

The condition part of the instruction may consist of several complete sub-conditions separated by AND or OR, and NOT can be use for negation. The relational operators used in conditions are the following

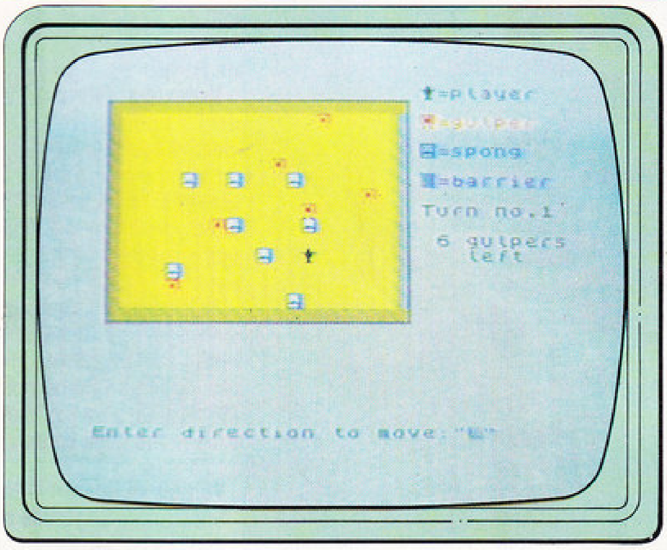
< > = <= >= <>

each being a single key depression on the Spectrum.

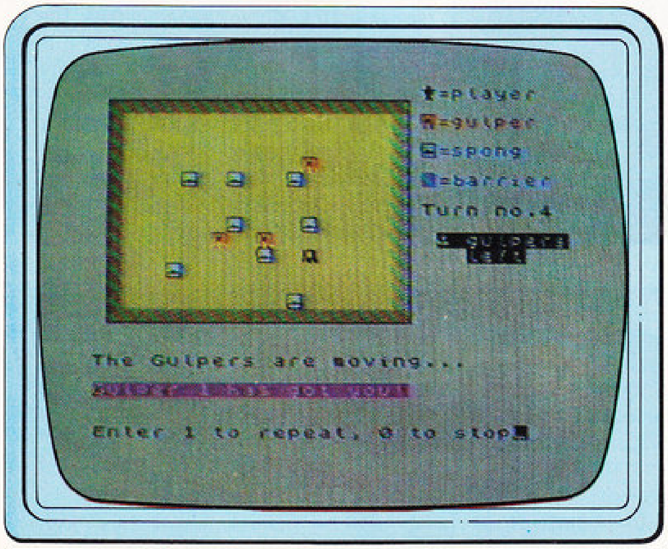
The GO TO instruction is the unconditional transfer of control but it may be followed by a numeric expression evaluating to a line number, thus making it a computed jump similar to the ON-GOTO of other machines.

Iteration

Looping is provided by the FOR instruction, and it is



The screens show the end-product of the Pursuit game listed on the following page. In the shot at right you see the Gulpers and Spongs who threaten your very existence. You must avoid them for ten turns in order to win — and when they're moving more quickly than you that task can be difficult.



The task is so difficult that the player in this screen didn't make the grade. The idea is to try and get the Spongs and Gulpers to eat one another while you avoid them — but the hero in this game wasn't fast enough.

here that the Spectrum fails to measure up to some of its competitors. Good programming can be hindered by the absence of such constructs as REPEAT . . . UNTIL or WHILE . . . DO and it is unfortunate that these are not available. The instruction has the form:

FOR control-variable = value 1 TO value 2

with the optional addition of a STEP clause at the end. There must be an accompanying NEXT instruction to mark the end of the iterative section, and the name of the control variable must be specified in the NEXT statement.

Nested iteration is also permitted, the only limit being the size of the stack (or perhaps the number of letters of the alphabet as control variables.)

Input/Output and files

The Spectrum has a very flexible INPUT statement, equivalent to combined PRINT and INPUT statements on many other microcomputers. Several items of data can be entered in response to one INPUT, and several prompts may also be included. The INKEY$ function provides a means of reading one character from the keyboard without the user pressing ENTER.

Data may also be entered using combined READ and DATA lines, and the pointer showing which item of data to read next may be reset by using RESTORE followed by the line number of a DATA statement.

PRINT sends output to the screen, and may be followed by a series of identifiers or expressions, separated by semicolons or commas, depending upon the spacing required. The positioning of the printing may be specified, either along a line using TAB, or on the whole screen with the AT clause defining the line and column co-ordinates. The way in which data is to be displayed is also given in the PRINT line, in terms of attributes such as INK and BRIGHT.

Three commands control output to the printer — COPY sends a screen of data to the printer, LLIST produces a hard-copy of the program listing, and LPRINT is the printer equivalent of PRINT.

Direct memory access is achieved by POKE and PEEK, and the processor ports may be accessed by IN and OUT.

Cassette files are provided for using variations on the SAVE and LOAD commands. As on the ZX81 these commands transfer both program and variables, so that a program may be restarted from where it was left on a previous occasion. Data structures, memory areas or screens may be transferred to and from cassette. The presence of VERIFY is a bonus for checking the transfer, and MERGE allows incoming program lines to be merged with those already in memory.

Subprograms and procedures

Again the Spectrum is not very adventurous in this realm. Subroutine jumps are available using GO SUB which may be followed by an expression as in GO TO, and the RETURN instruction causes a subroutine exit. Machine code subroutines may be accessed by USR.

User-defined functions are set up by DEF FN, and are referred to by FN and the function name (single letter or letter and $). Machines such as the BBC microcomputer score heavily against the Spectrum with their procedures enabling the programmer to adopt much more sophisticated constructs.

Intrinsic functions

Spectrum Basic has the usual set of trigonometric and scientific functions including ABS, ACS, ASN, ATN, COS, EXP, INT, LN, PI, RND, SGN, SIN, SQR and TAN. Predefined string functions include CHR$, CODE, LEN STR$, VAL, VAL$.

Sound and graphics

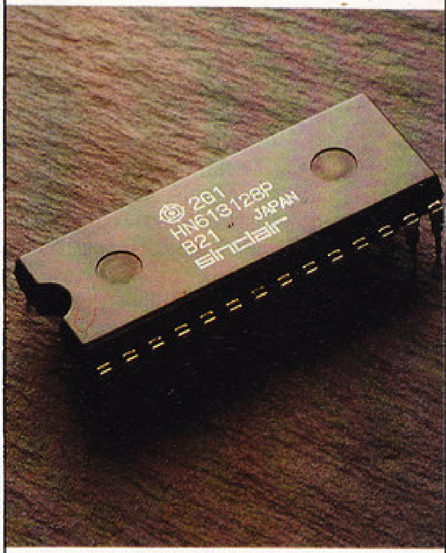
Although it is widely rumoured that the next available Spectrum peripheral will be a hearing aid, the machine does have a reasonable sound facility with the BEEP command. It does not rival the Oric either in flexibility, for musical and non-musical sounds, or in its power, but much current games software shows that a lot can be done with it.

On the other hand, Spectrum graphics are excellent, given the machine's price. The attributes of a displayed item may be specified using PAPER, INK, FLASH, BRIGHT, INVERSE and OVER, all of which can be used in a very straightforward manner in a PRINT line.

The attributes are held in a memory area parallel with the screen data, so that there are not the restrictions imposed by the serial attribute characters of the Oric, for example.

High resolution drawings with CIRCLE, DRAW or PLOT may be mixed freely with text; and the functions ATTR, POINT and SCREEN$ are methods of reading data from the screen display.

Chip that minds your language



This Sinclair Basic chip is made by Hitachi, and moves Basic commands through the keyboard and then on to the processor.

As a halfway house for all your Basic commands it does all the automatic syntax checking which prevents you from entering a line if there is an error in it. The chip also takes care of all the cassette-handling. In that 28-pin electronic centipede is an operating system that handles PRINTing, LOADing and SAVEing of programs. The basic stores numbers in floating point binary with one exponent byte ($1<=e<=255$), and four mantissa bytes m($\frac{1}{2}<=m<1$).

The unspeakable in pursuit of . . . you

If you derive pleasure from being chased around by malevolent monsters, you should enjoy Pursuit (*see overleaf*). It is particularly a game for the determined, because it is extremely difficult to win.

For unexplained reasons you find yourself in the lair of a group of Gulpers, strange creatures whose sole objective in life is to find and devour anything edible, and both humans and other Gulpers come into that category. The lair is totally surrounded by an antimatter barrier, lethal to all forms of life.

Finally there are also a few Spongs scattered around the lair. A Spong shows little signs of activity until you step on it, at which point it dematerialises you. Spongs show no discrimination, absorbing both Gulpers and humans alike.

Your objective is to avoid these many hazards for ten turns. On each turn you move one step in any direction and the Gulpers try to catch you. Wise players will attempt to lure Gulpers towards a Spong, so that there are fewer Gulpers to avoid.

A move is specified as an abbreviated direction, ie. N, S, E, W, NE, SE, NW, or SW. Use * to give up. To change the level of difficulty alter the number of Spongs, Gulpers or turns in line 10.

Initialise screen and variables

Set screen colours	(10)
Set number of turns (nt), number of Spongs (ns), number of Gulpers (ng) and number of Gulpers left (ngl)	(20)
Set up user graphic symbols	(50-70)
Display symbol key and headings	(80)
Draw lair	(90-150)
Draw ns Spongs randomly	(160 & SR500)
Set position of Gulpers in arrays x & y and display	(170 & SR500)
Set position of player (px, py) & display	(180-190 & SR500)
For turn number t from 1 to nt	(200)
Display turn number	(200)
Enter player's move m$	(205)
If m$ is "*" then	(207)
(i) Quit message & sound	(& SR900)
(ii) Go to 3(b)	

continued overleaf

Continued from page 111

If directions m$ invalid then (210 & SR600)
- (i) Message & reenter
- (ii) Go to (d)

Calculate player's new position (px, py) (220 & SR600)
If object at (px, py) then (230)
- (i) Display message that player hit gulper/spong/barrier (240-270)
- (ii) Sound (SR900)
- (iii) Go to 3(b) (270)

Display player's new position (px, py) (280)
Clear message lines (SR800)
For every Gulper: (300)
- (i) If eliminated (ie. line position = 0) then go to (vii) (305)
- (ii) Recalculate Gulpers position to move closer to player (320)
- (iii) If new position = player's position then (340)
  - (A) Death message & sound (& SR900)
  - (B) Go to 3(b)
- (iv) If new position = object then (345)
  - (A) Message that Gulper has been Sponged, or Barred & sound (350-365 & SR850)
  - (B) Decrease Gulpers left (ngl) by 1
  - (C) Set Gulper's line position x to 0
  - (D) Go to (vi)
- (v) Display Gulper & sound (370-375)
- (vi) Display number of Gulpers left (ngl) (380)
- (vii) Repeat to (i) (390)

If no Gulpers left (ngl = 0) then (400)
- (i) Message
- (ii) Go to 3(b)

Repeat to (2) (410)
End
Survived message (420)
Rerun or stop (430-450)

Routines

SR500 = finds a position (l, c) in the lair not occupied by a Spong, Gulper or barrier
SR600 = test whether m$ is a valid direction. Sets ok to 0 if invalid of 1 if valid. Calls SR700 and SR750
SR700 = calculates increment for px, based upon direction m$
SR750 = calculates increment for px, based upon direction m$
SR800 = clears message lines
SR850 = funeral sound for dead Gulper
SR900 = funeral sound for dead player
FNl = gives a random line position within the lair (2-14)
FNc = gives a random column position within the lair (2-19)
FNi = using FN b, extracts the ink colour of a position (j, k) in the lair
FNb = as above

```
   10 BRIGHT 0: FLASH 0: PAPER 7:
 BORDER 7: INK 0: CLS
   20 LET nt=10: LET ns=8: LET ng
=6: LET ngl=ng: DIM x(ng): DIM y
(ng)
   30 DEF FN l()=INT (RND*13)+2
   40 DEF FN c()=INT (RND*18)+2
   43 DEF FN b(n)=(n-64)-8*INT ((
n-64)/8)
   45 DEF FN i(j,k)=FN b(ATTR (j,
k))
   50 FOR i=1 TO 4: READ x$
   60 FOR j=0 TO 7: READ x: POKE
USR x$+j,x: NEXT j: NEXT i
   70 DATA "B",170,85,170,85,170,
85,170,85,"G",255,165,189,189,18
9,165,36,102,"P",24,24,255,60,60
,24,24,60,"S",255,129,165,129,18
9,255,129,255
   80 PRINT AT 0,22;"P=player";AT
 2,22; INK 2;"G=gulper";AT 4,22;
 INK 1;"S=spong";AT 6,22;"B=barr
ier"; INK 0;AT 8,22;"Turn no.1";
AT 10,23;ng;" gulpers";AT 11,25;
"left"
   90 INK 1: PAPER 6
  100 FOR l=2 TO 14: PRINT AT l,2
;"                  ": NEXT l
  110 BRIGHT 1
  120 PRINT AT 1,1;"BBBBBBBBBBBBBB
BBBBBBB";AT 15,1;"BBBBBBBBBBBBBB
BBBBBB"
  130 FOR l=2 TO 14: PRINT AT l,1
;"B";AT l,20;"B": NEXT l
  140 INK 0: PAPER 8
  150 PLOT 8,167: DRAW 161,0: DRA
W 0,-119: DRAW -161,0: DRAW 0,11
9
  160 FOR i=1 TO ns: GO SUB 500:
PRINT AT l,c; INK 1; PAPER 7;"S"
: NEXT i
  170 FOR i=1 TO ng: GO SUB 500:
LET x(i)=l: LET y(i)=c: PRINT AT
 l,c; INK 2;"G": NEXT i
  180 BRIGHT 0
  190 GO SUB 500: LET px=l: LET p
y=c: PRINT AT px,py;"P"
  200 FOR t=1 TO nt: PRINT AT 8,3
0;t
  205 INPUT "Enter direction to m
ove:";m$
  207 IF m$="*" THEN PRINT FLASH
1;AT 20,0;"You gave up": GO SUB
900: GO TO 430
  210 GO SUB 600: IF NOT ok THEN
INPUT "Reenter (eg. SW or N):";m
$: GO TO 210
  220 PRINT AT px,py; PAPER 6;" "
: LET px=px+ix: LET py=py+iy:
  230 IF ATTR (px,py)<64 THEN GO
TO 280
  240 IF FN i(px,py)=2 THEN LET w
$="Gulper": GO TO 270
  250 IF FN i(px,py)=1 THEN LET w
$="Spong": GO TO 270
  260 LET w$="Barrier"
  270 PRINT AT px,py; FLASH 1;"*"
; INK 3;AT 18,0;"You walked into
a ";w$;"!": GO SUB 900: GO TO 4
30
  280 PRINT AT px,py;"P": GO SUB
800: BEEP .2,20: BEEP .2,10
  290 PRINT AT 18,0;"The Gulpers
are moving..."
  300 FOR i=1 TO ng
  305 IF x(i)=0 THEN GO TO 390
  310 PRINT AT x(i),y(i); PAPER 6
;" "
```

```
  320 LET x(i)=x(i)+SGN (px-x(i))
: LET y(i)=y(i)+SGN (py-y(i))
  340 IF px=x(i) AND py=y(i) THEN
 PRINT AT px,py; FLASH 1;"G"; IN
K 3;AT 20,0;"Gulper ";i;" has go
t you!": GO SUB 900: GO TO 430
  345 IF ATTR (x(i),y(i))<64 THEN
 GO TO 370
  350 IF FN i(x(i),y(i))=1 THEN P
RINT AT x(i),y(i); FLASH 1; INK
1; PAPER 7;"S";AT 19,0; FLASH 0;
"Gulper ";i;" absorbed  ": GO
SUB 850: PRINT AT x(i),y(i); BRI
GHT 1; INK 1; PAPER 7;"S": LET n
gl=ngl-1: LET x(i)=0: GO TO 380
  360 IF FN i(x(i),y(i))=0 THEN P
RINT AT x(i),y(i); FLASH 1;"*";
FLASH 0;AT 19,0;"Gulper ";i;" me
lted  ": GO SUB 850: PRINT A
T x(i),y(i); BRIGHT 1; PAPER 6;"
B": LET ngl=ngl-1: LET x(i)=0: G
O TO 380
  365 IF FN i(x(i),y(i))=2 THEN P
RINT AT x(i),y(i); FLASH 1;"G";A
T 19,0; FLASH 0;"Gulper ";i;" ca
nnibalised": GO SUB 850: PRINT A
T x(i),y(i); BRIGHT 1; INK 2; PA
PER 6;"G": LET ngl=ngl-1: LET x(
i)=0: GO TO 380
  370 PRINT AT x(i),y(i); BRIGHT
1; INK 2;"G"
  375 BEEP .1,15: PAUSE 25: BEEP
.1,5: BEEP .1,-5
  380 PRINT AT 10,23;ngl
  390 NEXT i
  400 IF ngl=0 THEN PRINT FLASH 1
;AT 20,0;"No Gulpers left - you
win!": GO TO 430
  410 NEXT t
  420 PRINT FLASH 1;AT 20,0;"Well
 done - you have survived!"
  430 PRINT AT 10,23; FLASH 1;ngl
;" gulpers";AT 11,25;"left": PAU
SE 200: INPUT "Enter 1 to repeat
, 0 to stop"; LINE z$: IF z$="1"
 THEN RUN
  450 STOP
  500 LET l=FN l(): LET c=FN c()
  510 IF ATTR (l,c)>64 THEN GO TO
 500
  520 RETURN
  600 LET ok=0: LET lm=LEN m$
  610 IF lm<1 OR lm>2 THEN RETURN
  620 IF lm=1 THEN LET y$=m$(1):
GO SUB 700: GO SUB 750: IF ix=0
AND iy=0 THEN RETURN
  630 IF lm=2 THEN LET y$=m$(1):
GO SUB 700: LET y$=m$(2): GO SUB
 750: IF ix=0 OR iy=0 THEN RETUR
N
  640 LET ok=1: RETURN
  700 LET ix=(y$="S" OR y$="s")-(
y$="N" OR y$="n"): RETURN
  750 LET iy=(y$="E" OR y$="e")-(
y$="W" OR y$="w"): RETURN
  800 FOR l=18 TO 20: PRINT AT l,
0;"                              
  ": NEXT l
  810 RETURN
  850 FOR j=1 TO 3: BEEP 1,6*J: B
EEP .01,20: NEXT j
  860 RETURN
  900 FOR j=1 TO 4: BEEP .1,20: B
EEP 1,-20: NEXT j: BEEP 5,-20
  910 RETURN
```

This program, taken from Bob Maunder's 'Spectrum Games Companion', published by Linsac, illustrates many of the features of Spectrum Basic. The author recommends 'Pursuit' as 'a game for the determined, because it is extremely difficult to win'.

Contributors: John Lettice, Bob Maunder and Geof Wheelwright.

NEXT WEEK

Micropaedia moves from the inside of the Spectrum to peripherals next week as we look at almost every plug-in you can think of, from Microdrives to joysticks to printers. We'll also have a look at Basicare's 'stacking' expansion units for the Spectrum and some hints on using TVs and cassettes with the machine.

And in two weeks we look at software and programming. You'll get the rundown on how to convert ZX81 programs to run on the Spectrum as well as a good look at some of the classic games you can play on it.