

Spectrum and
CBM-64 games tests

YOUR

Canada	C\$	4.00
Denmark	DKr	35.00
France	Fr	50.00
Germany	Dm	8.50
Greece	Dra	290.00
Holland	DF.	8.50
Italy	L	4200
New Zealand	NZ\$	3.70
Singapore	M\$	6.25
Spain	Pts	190.00
Switzerland	SFr	7.00
U.S.A.	\$	3.95

(D/2163)

90p

COMPUTER

OCTOBER 1984

BRITAIN'S BIGGEST-SELLING HOME COMPUTER MAGAZINE

Vol. 4 No. 10

1984

WHO'S IN CONTROL — YOU OR YOUR MICRO?

CBM-64 Jetset Jelly
Talk to your ZX-81
Spectrum Blockpaint
Amstrad Bomber
Dragon Tanks and
BBC Loona Rescue

WIN THIS NEW COMMODORE!
PLUS 4 AND 16 REVIEWS



YOUR COMPUTER

Contents

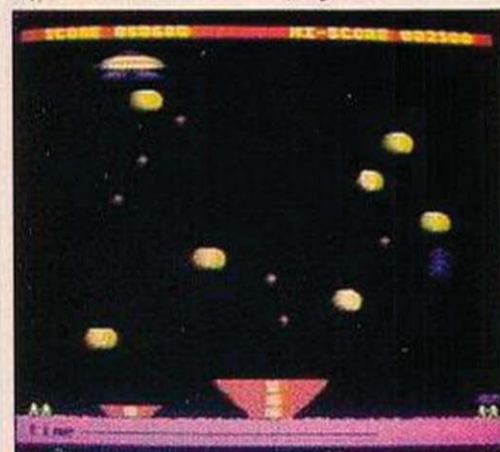
OCTOBER 1984



Commodore Plus 4 review — page 52.



Spectrum Software — page 62.



Loona rescue — page 76.



Blockpoint — page 84.



Character print: page 113.

The front cover shows a scene from Virgin Films' 1984 starring John Hurt and the late Richard Burton.

- 36** LETTERS AND EDITORIAL: Subliminal images and lots of lives.
- 38** NEWS: The American Dream and Shakespeare.
- 41** FIRST BYTES: All about Basicde and Microdrives for starters.
- 45** COMPUTER CLUB: Paul Bond rambles through the countryside.
- 47** SOFTWARE SHORTLIST: Underneath the radar, on to the magic carpet, and dogging lead on main street at high noon. We scrutinise the latest software releases.
- 51** QUEST CORNER: Hugo North reviews a game about the truly appalling prospect of a nuclear holocaust, plus Woodland Terror and Scott Adams' Hulk.
- 52** COMMODORE PLUS FOUR: No flies on this, says Glyn Moody.
- 54** COMMODORE 16: Bill Bennett takes a real close look at the CBM-16.
- 59** SEGA SC-3000H: David Aubrey-Jones walzes over Oz's fave micro.
- 62** SPECTRUM SOFTWARE: Paul Bond limps through the cosmos.
- 64** COMMODORE CBM-64 SOFTWARE REVIEW: Automania, Alice in Videoland, Flak, Potty Pigeon, Ankh. Simon Beesley's review of recent CBM-64 releases.
- 71** POLITICS AND MICROS: Politics is the only profession for which no preparation is thought necessary. Meirion Jones looks at the micro tendency.
- 76** BBC LOONA RESCUE: Evacuate the humans with David Griffiths.
- 80** CBM-64 JETSET JELLY: Whole lotta shakin' from Gary Knowles.
- 84** SPECTRUM BLOCK PAINT: Guide your character around the screen, filling in blocks. Pure machine-code from Ayal Pinkus.
- 89** ATARI KING OF THE CASTLE: Barry Prescott introduces you to Drego and Grotbags. Build a bridge and dodge the fireballs to repossess your home.
- 98** VIC-20 TRACK CHASE: Paint the track green while aliens frustrate your instincts for interior decoration. Blast off with Adrian Mardlin.
- 100** DRAGON TANK: Highly addictive, high-resolution program from Rick Ludkiewicz. Try your luck on the automated target range.
- 104** ZX-81 BUGBURST: Masses of movement and a battle through 15 infested screens. J Dave Rogers and Colin Hogg show you how to clean up.
- 109** AMSTRAD CITY BOMBER: Create a path for urban renewal with Daniel Winter's old favourite for a new machine.
- 113** SPECTRUM CHARACTER PRINT: Proportional spacing, 42 or 64 characters per line, just some of the features offered by Paul Rhodes's utility.
- 119** 64 BAR BLUES: Music for the CBM-64 by Paul Marzetti, and Marcus Davidson. Enables a three-voice melody to be played independently of program control.
- 123** COMPETITION: Your chance to win Commodore's new Plus 4. Just put the features in order of importance.
- 125** ORIC UTILITIES: Nine new commands which make a significant contribution to the power of Oric Basic, thanks to Gerard Mason.
- 131** SPECTRUM COMPACTOR: John Swales draws you a picture without using too much of your memory or wasting too much of your time.
- 135** DRAG STATS: Comprehensive program for the Dragon 32 which deals with displaying statistics and graphs. Written by John Nash.
- 143** ZX-81 VOICE RECOGNITION: This program does without extra hardware, simply using your tape recorder as an amplifier. Written by Jason Creak.
- 149** BBC KEYSTICK: Fintan Culwin puts joysticks where none has gone.
- 155** WORDS: More words from John Dawson.
- 169** SOFTWARE FILE: Ten pages of progs for most micros.
- 216** DATABASE: Paul Bond lists computer events.



Program 3.

```

1 REM Program 3
2
3 REM Machine code to move meteors and mother ship
4
10 FOR P=0 TO 3 STEP 3
20 P%=&A20
30 COPT P
40 .update
50 LDA &6A: CMP #1: BEQ ok
60 LDA &6B: EOR #1: STA &6B: CMP #1: BEQ ret
70 .ok LDA &69: CMP #1: BNE ast
80 JSR mother
90 .ast JSR asteroid
100 .ret RTS
110 \
120 \
130 .mother
140 LDA &8C: CMP #1: BNE mback
150 CLC: LDA &8E: ADC #8: STA &8E: LDA #0: ADC &8F: STA &8F
160 INC &8C: LDA &8D: CMP #68: BCC mprint
170 LDA #255: STA &8C: JMP mprint
180 .mback SEC: LDA &8E: SBC #8: STA &8E: LDA &8F: SBC #0: STA &8F
190 DEC &8D: LDA &8C: CMP #0: BNE mprint
200 LDA #1: STA &8C
210 .mprint
220 LDY #0
230 CLC: LDA &8E: ADC #&B0: STA &8E: LDA #2: ADC &8F: STA &8F
240 .nloop LDA &96C, Y: STA (&8E), Y
250 LDA &96C, Y: STA (&8E), Y
260 INY: CPY #96: BCC nloop
270 RTS
280 \
290 \
300 .asteroid LDA #1: EOR &6B: STA &6B
310 LDX #0
320 .loop CPX #8: BNE pass: JMP check
330 .pass CPX #20: BNE pass1
340 .check LDA &6B: CMP #1: BNE next
350 .pass1
360 LDA &73, X: CMP #1: BNE back
370 CLC: LDA &70, X: ADC #8: STA &70, X: LDA #0: ADC &71, X: STA &71, X
380 INC &72, X: LDA &72, X: CMP #74: BCC print
390 LDA #255: STA &73, X: JMP print
400 .back SEC: LDA &70, X: SBC #8: STA &70, X: LDA &71, X: SBC #0: STA &71, X
410 DEC &72, X: LDA &72, X: CMP #0: BNE print
420 LDA #1: STA &73, X
430 .print LDY #0
440 LDA &71, X: STA &6D
450 LDA &70, X: STA &6C
460 CLC: ADC #&8C: STA &6E
470 LDA &71, X: ADC #7: STA &6F
480 .loop1
490 LDA &900, Y: STA (&6C), Y
500 LDA &930, Y: STA (&6E), Y
510 INY: CPY #48: BNE loop1
520 .next INX: INX: INX: INX: CPY #25: BCC loop
530 RTS
540 J
550 NEXT

```

the event indirecting through &220 — that is, the memory location of the start of my event handling routine is stored in locations &220 and &221 — see line 1580 of program 5.

The reason I chose to use events is that the necessary moving characters will do so apparently automatically, leaving the Basic program running at almost normal speed to handle the setting up of the screen and the general working of the game.

Type in five programs

To get the final version of the game onto cassette it is necessary to type in five programs.

The first is the header for the game proper. Program 2 contains the data for the graphic characters of the mothership and meteors, and the procedure for storing that data in memory. Care should be taken when typing the data statements. If characters look strange it will probably be due to typing errors in this section.

Program 3 contains the assembler for the machine-code routines. Program 4 holds the definitions of the characters used in the game. Program 5 is the game itself and, due to its length, will be prone to typing errors. It may be helpful to replace line 20 with

```

20 ON ERROR MODE 7:REPORT:PRINT
   "at line";ERL:END

```

when developing the program, to trap any errors.

Program 4.

```

1 REM Program 4
2
3 REM User defined characters
4
10 VDU 23,241,200,200,200,200,200,200,200,200
20 VDU 23,242,255,63,127,31,15,15,7,3
30 VDU 23,243,254,254,248,252,224,224,192,128
40 VDU 23,254,15,63,195,195,63,15,31,195
50 VDU 23,255,192,240,12,12,240,192,48,12
60 VDU 23,246,24,24,60,90,153,60,36,102
70 VDU 23,247,16,84,50,251,56,84,16,0
80 VDU 23,138,8FF,820,86A,82F,8AA,8AA,820,8FF
90 VDU 23,139,8FF,8A0,8AA,8A6,8AA,8AA,8A0,87F
100 VDU 23,140,8FF,891,885,895,885,885,891,87F
110 VDU 23,148,0,856,854,876,854,856,0,0
120 VDU 23,149,0,898,895,898,898,898,891,0,0
130 VDU 23,157,889,84A,824,8C0,3,324,852,891

```

The following procedure should be adopted to obtain the program correctly and to store it on cassette.

1. Type in program 1 and save it on tape by, SAVE "LOONA.FESQ" (RETURN) (RETURN)
2. Type in program 2 and RUN it.
3. Type in program 3 and RUN it.
4. Type in program 4 and RUN it.
5. Save the above information with, "SAVE "LOONA.1" 900 D00 (RETURN) (RETURN)
6. Type in program 5 and save it on tape, SAVE "LOONA.2" (RETURN) (RETURN)

Although it is not necessary to save programs 2, 3 and 4 on the final cassette, it would be wise to save each program on some other cassette, and then it will be possible to check each of these as well as program 5 for typing errors, if the game does not work.

You will probably have noticed that the character definitions are in one of the tem-

(continued on page 79)

