say, 256, two things must happen — the bit corresponding to sprite 3 in the V+16 register is set to one and the normal X co-ordinate register starts again from zero. The following table shows what happens in the registers as the submarine crosses the X = 255 boundary:

X3	V+16								V+6							
254	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0
255	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
256	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
257	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

The variable H3 will be set to one if the value of X3 exceeds 255. Correspondingly, L3 will be reset to zero if H3 becomes one. The values of L3 and H3 can then be POKEd into the registers V+6 and V+16.

FIRING THE DEPTH CHARGES

During the game, depth charges can be dropped on the submarine at any time. To make a program straightforward, we shall make the rule that once a depth charge has been fired no other depth charge can be fired until:

a) the sub has been hit; or

b) the charges have missed the sub and dropped a little way past it; or

c) the charges have missed the sub and reached the seabed.

The main loop of the program has two jobs to do in respect of depth charges: it must detect the pressing of the 'M' key, and once the depth charge has been fired it must control its vertical movement. The program must also ensure that no new depth charges are fired whilst one is in the process of dropping. This last problem can be solved by the use of a flag. This is a technique often applied in program control, signalling that a particular event has, or has not, occurred. In our program we shall use the variable FL to signal the dropping of a depth charge. Its value will be one if a charge is dropping and zero otherwise. In line 100 of the program the value of FL is initially set to zero. Line 260 accesses the 'Set Up Depth Charges' subroutine at line 3000 if 'M' is pressed and the flag is set to zero. A second subroutine at line 400 is used to move a fired depth charge sprite, and this is accessed by line 380.

The 'Set Up Depth Charges' subroutine has three functions to perform:

1) To set the flag FL to one as a signal that a charge has been fired.

2) To set the starting co-ordinates: the X co-ordinate takes its value from that of the ship and the Y co-ordinate is initially set to 95, positioning the charge just below the surface of the sea.

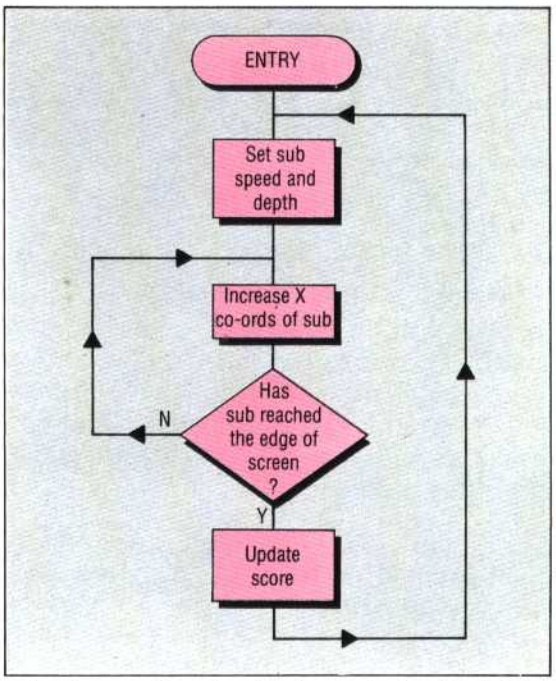
3) To turn on the depth charge sprite.

The 'Move Depth Charge' subroutine is used to move the depth charge down the screen. In addition, tests have to be made to see if:

1) The depth charge has gone past the submarine *or* reached the seabed.

2) The depth charge has hit the sub.

If the first event has occurred, the depth charge sprite can be turned off and the flag reset to zero, allowing another depth charge to be fired. The second event is tested by using another feature of Commodore 64 sprites — the sprite collision register. As with other registers of the VIC chip, this register, V+30, has one bit corresponding to each sprite. If a particular sprite is involved in a collision with another sprite then the corresponding bit in this register is set to one. Thus, if the sub (sprite 3) and the depth charge (sprite 2) are in collision, the contents of the register V+30 will be 12 (00001100 = 12). By PEEKing this register and testing its contents, we can tell if the depth charge has hit the sub. If it has, then a further 'HIT' subroutine is accessed at line 5000. This subroutine will be dealt with in the final instalment of the project, together with the instructions to update the HI SCORE and restart the game.



KEVIN JONES

Underwater Flow Chart
This simple flow chart shows how the program controls the movement of the submarine sprite. It selects a random depth and speed, being careful to make sure that the sub is under the surface and above the seabed. The sub is then moved smoothly across the screen until it reaches the other side

Subhunter's Movement Subroutines

```
130 GOSUB 2500: REM SET SUB CO-ORDS
230 GET A$
240 IF A$="Z" THEN X0=X0-1.5: IF X0<24 THEN
X0=24
250 IF A$="X" THEN X0=X0+1.5: IF X0>245 THEN
X0=245
260 IF A$="M" AND FL=0 THEN GOSUB 3000:
    REM SET DEPTH CHARGES
265 :
270 REM ** MOVE SHIP **
290 POKE V,X0
295 :
300 REM ** MOVE SUB **
310 X3=X3+DX
315 :
320 REM  ** SUB AT SCREEN END? **
330 IF X3>360 THEN DS=-1:GOSUB 5500:
GOSUB 2500
340 H3=INT(X3/256):L3=X3-256*H3
350 POKE V+6,L3: POKEV+16,PEEK(V+16)OR(8*H3)
360 IF FL=1 THEN GOSUB 4000:
 REM MOVE DEPTH CHARGE
370 GOTO 200: REM RESTART MAIN LOOP
380:
390:
2500 REM **** RESET SUB COORDS ****
2510 Y3=110+INT(RND(TI)*105)
2520 X3=0:DX=RND(TI)*3+1
2530 POKE V+7,Y3:POKE V+6,X3
2540 POKE V+16,PEEK(V+16) AND 247
2550 RETURN
2560 :
2570 :
3000 REM ***SET UP DEPTH CHARGES***
3010:
3020 REM ** SET FLAG **
3030 FL=1
3040 :
3050 REM ** SET COORDS ***
3060 Y2=95:X2=X0
3070 POKE V+4,X2:POKE V+5,Y2
3080 :
3090 REM ** TURN ON SPRITE 2 **
3100 POKE V+21,PEEK(V+21)OR4
3110 RETURN
3120 :
3130 :
4000 REM ** MOVE DEPTH CHARGE **
4010 :
4020 REM ** INCREASE Y COORD **
4030 Y2=Y2+2
4040 POKE V+5,Y2
4050 :
4060 REM **TEST BOTTOM & TURN OFF**
4070 IF Y2>Y3+25 OR Y2>216 THEN
POKE V+21,PEEK(V+21) AND 251:FL=0
4080 :
4090 REM **TEST FOR SUB HIT**
4100 IF PEEK(V+30)=12 THEN GOSUB 5000:
REM HIT ROUTINE
4110 RETURN
4120 :
4130 :
```