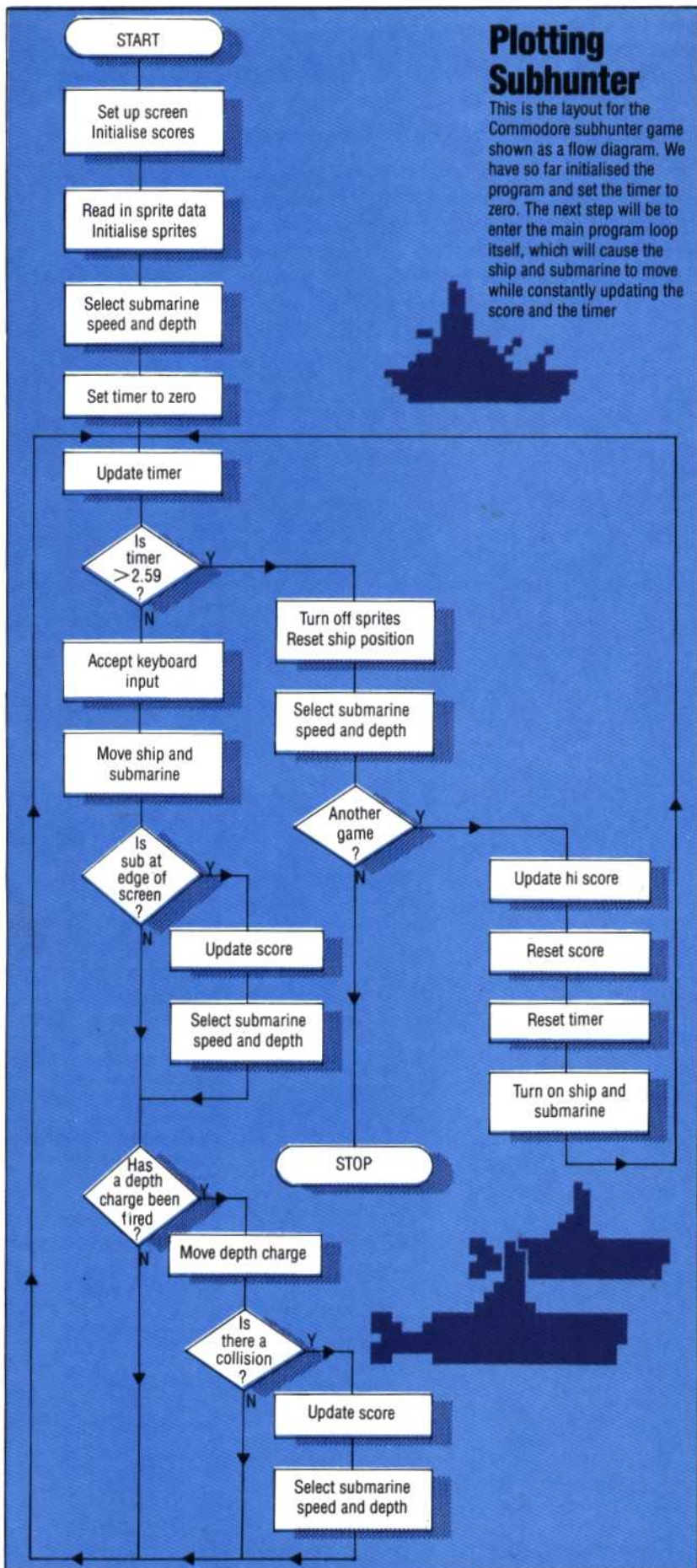


Plotting Subhunter

This is the layout for the Commodore subhunter game shown as a flow diagram. We have so far initialised the program and set the timer to zero. The next step will be to enter the main program loop itself, which will cause the ship and submarine to move while constantly updating the score and the timer



Subhunter Program

The Commodore 64 has its own internal clock that can be used to time BASIC programs. The clock has six digits, rather like a digital watch, representing hours (00-23), minutes (00-59), and seconds (00-59). The clock can be accessed from BASIC through the string variable TIS\$. The value of TIS\$ gives the time that has elapsed since the computer was turned on, but it can also be reset at any time. The following short program demonstrates how the timer works.

```

10 REM ....TIMER....
20 PRINT CHR$(147) : REM CLEAR SCREEN
30 TIS$ = "000000" : REM SET TIMER TO ZERO
40 PRINTCHR$(145);TIS$ : REM PRINT CURRENT VALUE OF TIMER
50 : REM CHR$(145)=CURSOR UP
60 GOTO40
    
```

The program runs in a continuous loop, printing the timer to the screen until you press the Run/Stop key.

The Subhunter game we are writing requires a clock to be displayed on the screen and to end the game when three minutes have elapsed. The game clock, therefore, requires only the minutes and seconds parts of TIS\$. By using the string functions we can break TIS\$ down as follows:

$$TIS\$ = HH(MM)(SS)$$

RIGHTS(TIS,2) ↓
 MIDS(TIS,3,2)

The two seconds digits can be stripped off by RIGHTS(TIS,2), and the minutes digits can be isolated by MIDS(TIS,3,2).

The main program loop of our game starts at line 200 and ends at 390. Load up the subroutine already typed in from the last section and add these lines:

```

140 TIS$="000000"
150 :
160 :
200 REM .... MAIN LOOP ....
205 :
210 REM .. TIMER ..
220 PRINTCHR$(19);:TAB(14)CHR$(5)"TIME
    ";MIDS(TIS,3,2);:":RIGHTS(TIS,2)
225 IFVAL(TIS)>259THEN400:REM END GAME
:
390 GOTO200:REM RESTART MAIN LOOP
400 END
    
```

Line 140 re-sets the clock at the start of the program. Line 220 PRINTs the current value of the clock in minutes and seconds, separated by a colon. TAB(14) causes 14 spaces to be left before PRINTing and positions the clock in the middle of the screen. CHR\$(5) will colour the characters white. Line 225 converts TIS\$ to a numeric quantity so that its value can be tested. If playing time has exceeded two minutes and 59 seconds, then the game is at an end.