

height, flashing, blue characters on a steady white background, we use the following lines of code:

```
10 MODE 7
20 PRINT CHR$(141);CHR$(136);CHR$(132);
  CHR$(157);CHR$(135); "IN THE NAVY"
30 PRINT CHR$(141);CHR$(136);CHR$(132);
  CHR$(157);CHR$(135); "IN THE NAVY"
```

You can see that all these control codes take up a lot of program space and are time-consuming to type out. An alternative method is to chain several codes together into a single string that can then be used in the PRINT statements. For example, if we need to do a lot of PRINTing using red characters on a yellow background, we can start by creating a string (red\$) that can then be used in each PRINT statement requiring this effect. Thus:

```
10 MODE 7
20 red$=CHR$(129)+CHR$(157)+CHR$(131)
30 PRINTred$; "GAME OVER"
40 PRINTred$; "YOUR SCORE"
```

## THE END-OF-GAME PROCEDURE

Let's now look more closely at how we employ these effects in our end-of-game screen:

```
2110DEF PROCend_game
2120 IF score$>hi_score$ THEN hi_score$=score$
2130red$=CHR$(129)+CHR$(157)+CHR$(131)
2140game$="G A M E O V E R"
2150PRINTTAB(0,5);red$;CHR$(141);CHR$(136);TAB(12);game$
2160PRINTred$;CHR$(141);CHR$(136);TAB(12);game$
2170PRINT:PRINTred$;"Your Score";TAB(30);score$
2180PRINT:PRINTred$;"Hi score";TAB(30);hi_score$
2190PRINT:PRINTred$;"Time";TAB(30);time$
2200blue$=CHR$(132)+CHR$(157)+CHR$(134)
2210go$="A N O T H E R G O Y / N ?"
2220PRINT:PRINT
2230PRINTblue$;CHR$(141);CHR$(136);TAB(5);go$
2240PRINTblue$;CHR$(141);CHR$(136);TAB(5);go$
2250REM ** REPLY ? **
2260*FX 15,1
2270answer$=INKEY$(0)
2280IF GET$="N" THEN finish_flag=1
2290ENDPROC
```

Line 2120 checks to see if the score in the game just concluded was greater than the previous highest score and updates the high score if necessary.

The message GAME OVER is then printed in double-height, red flashing characters on a yellow background (lines 2130 to 2160) and the details of the scores and time are displayed (lines 2170 to 2190). The player is then asked if another game is required. If the answer is N then a variable (finish flag) is set to one.

Notice that mode 7 has not been set during this procedure. This is because the BBC Micro will not allow a mode change to be made within a procedure. An attempt to do this will result in a BAD MODE error message. We must, instead, set mode 7 in the short main program that calls the procedures. The following lines should be added to complete the program. Notice that the whole of the calling program has now been placed in a REPEAT...UNTIL loop, which will repeat until finish flag is set to one.

```
1100REPEAT
1200MODE7
1210REM ** TURN OFF CURSOR **
1220VDU23,1,0;0;0;0;
1230PROCend_game
1240UNTIL finish_flag=1
1250CLS
1260END
```

## THE ELECTRON ALTERNATIVE

Electron users may have been a little worried during our discussion of mode 7, as the Electron does not feature this mode. As an alternative, we have prepared a different procedure that uses mode 5 for the end-of-game screen. Omit line 1200 from the calling program just given and enter this procedure in place of the BBC's end-of-game procedure:

```
>L.2100,2300
2100 DEF PROCend_game
2110 IF score$>hi_score$ THEN hi_score$=score$
2120 REM ENSURE BACKGROUND YELLOW
2130 VDU19,130,3,0,0,0
2140 GCOL0,130:CLG:REM COLOUR SCREEN
2150 COLOUR1:COLOUR130:REM SET TEXT COLOURS
2160 game$="G A M E O V E R"
2170 PRINTTAB(2,4);game$
2180 COLOUR0
2190 PRINTTAB(0,8);"Your Score";TAB(15);score$
2200 PRINT:PRINT"Hi score";TAB(15);hi_score$
2210 PRINT:PRINT"Time";TAB(15);time$
2220 go$="ANOTHER GO Y/N ?"
2230 REM CHANGE COL3 TO FLASH YELL/BLUE
2240 VDU19,3,11,0,0,0
2250 COLOUR3
2260 PRINT:PRINT
2265 PRINTTAB(2)go$
2270 REM ** REPLY ? **
2275 *FX 15,1
2280 answer$=INKEY$(0)
2285 IF GET$="N" THEN finish_flag=1
2290 VDU 20:REM RESET DEFAULT COLOURS
2300 ENDPROC
```

## The Final Listing

```
1000REM *****
1010REM ** GAMES **
1020MODE5
1030REM **
1040REM *****
1050:
1060hi_score$="00000"
1070finish_flag=0
1080:
1090REM *** MAIN PROGRAM ***
1100REPEAT
1110MODE5
1120REM ** TURN OFF CURSOR **
1130VDU23,1,0;0;0;0;
1140PROCtitle_page
1150CLS
1160PROCsetup
1170:
1180PROCloop
1190:
1200MODE7
1210REM ** TURN OFF CURSOR **
1220VDU23,1,0;0;0;0;
1230PROCend_game
1240UNTIL finish_flag=1
1250CLS
1260END
1270:
1280:
1290REM ***** DEFINE PROCEDURES *****
1300DEF PROCtitle_page
1310GCOL 0,129
1320CLG
1330GCOL 3,3
1340PROCmusic
1350y=100;x=0
1360REPEAT
1370x=x+20;y=y+50
1380FOR i=1 TO 2
1390PROCmines
1400NEXT i
1410UNTIL Y>700
1420:
1430PROCmines
1440PRINTTAB(0,20)*Skill factor (0-9)?*
1450PROCmusic
1460REPEAT
1470skill=GET-48
1480UNTIL skill<1 AND skill<10
1490ENDPROC
1500:
1510DEF PROCmines
1520PLOT4,X,Y
1530REM ** LETTER M **
1540PLOT1,0,200
1550PLOT1,80,-100
1560PLOT1,80,100
1570PLOT1,0,-200
1580REM ** LETTER I **
1590PLOT0,100,0
1600PLOT1,80,0
1610PLOT0,-40,0
1620PLOT1,0,200
1630PLOT0,-40,0
1640PLOT1,80,0
1650REM ** LETTER N **
1660PLOT0,40,-200
1670PLOT1,0,200
1680PLOT1,120,-200
1690PLOT1,0,200
1700REM ** LETTER E **
1710PLOT0,160,0
1720PLOT1,-120,0
1730PLOT1,0,-200
```