SWINGTIME

KEVIN JONES

Hangman is a traditional wordgame that is easily implemented on home computers and can prove educational. Programming a Hangman game provides us with an opportunity to explore string manipulation. We discuss the structure of a simple version of the game, and give listings for the BBC Micro and the Spectrum.

Probably everybody has played a game of Hangman at some time. The object of the game is simply to guess the letters in a word. The only information given is the number of characters in the word, all of which are represented by dashes. A correctly guessed letter is displayed in its proper position, and an incorrect guess causes a part of a picture of a man on a scaffold to be drawn. In our program, there are 10 parts to this drawing, which is shown on the right-hand side of the screen. If all 10 parts are completed before you've filled in all the letters in the word, then the man is hanged and you've lost the game.

The basic principle of our program is very simple. It involves checking to see if a letter entered at the keyboard is contained in a randomly selected 'secret' word (one of 11 words held in DATA statements at the end of the program). If the guessed letter is present, it is displayed on the screen in its correct place. If the letter is not present, the program must display it anyway, to remind the player that it has already been used. In addition, of course, the program must then be made to jump to a subroutine that will draw a part of the hanged man.

The words that our programs use are stored in lines 1020 and 1030 in both versions. There is no reason why you can't add to these, using more DATA statements. But if you do add your own lexical brain-teasers, you must remember that they should not be more than 10 letters long. (Although this restriction could be lifted by altering lines 30 and 50.) Also, the total number of words in the DATA statements must be added up, and the value of N in line 20 altered accordingly.

At the beginning of the program, all of the words are read into an array. One of the elements of the array is picked at random, and a line of dashes corresponding to the length of the word is displayed on the screen. The rest of the game consists of a repetitive loop. When a letter is entered from the keyboard it is screened: if it is more than one letter, or not a character at all, then the program makes a BEEP and loops back for another input. The letter is also checked against the list of letters that have already been tried in the game. If it has been used before, then a warning is flashed on the screen and a new letter read.

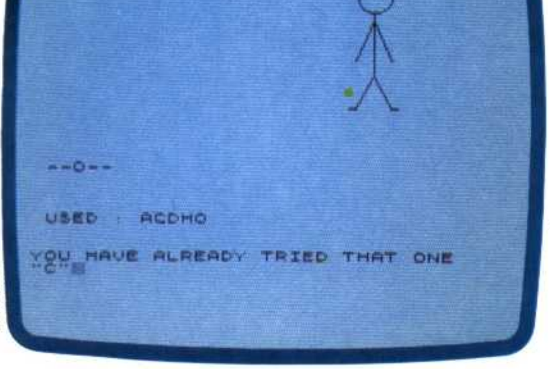
If the trial letter is acceptable, it is added to the displayed list of used characters, and then compared against each letter of the word in turn. If it matches in any position, it is put up on the screen in place of the corresponding dash. If no match is found in the whole word, a subroutine is called to draw a part of the hanging scene.

If the player has had 10 wrong guesses then the man is hanged, a short consolatory tune is played and a new word is selected. Alternatively, if all the letters of the word have been correctly placed, a congratulatory musical phrase is played.

Our two versions of the program can be easily adapted to most home micros. The subroutines to draw the man and the scaffold, of course, need to be especially adapted to the graphics capabilities of individual machines. Programmers with an interest in creating interesting screen displays may like to elaborate on the drawn result: a hanged man gently swinging in the breeze if the player fails to win, perhaps?

Other refinements can also be added to the game. We suggest that it may be a good idea to add a check at the beginning of the program to see that a selected word has not already been used. As the program stands, the selection of the word is a purely random matter, and the same word could be chosen twice in succession.

Even more helpful would be a routine to screen the inputs so that only upper case letters were accepted as trial letters. (The program as given will accept any character as a trial 'letter'.) Lower case letters, numbers, and symbols could be checked by their ASCII codes and rejected. As it is given here, however, our program does play a good game of Hangman, and offers opportunities for adventurous programmers.



```
HANGMAN

--O--

USED : ACDHO
YOU HAVE ALREADY TRIED THAT ONE
"C"
```

LIZ HEANEY

Progress Report
The game in progress, showing the development of the hanged man and the progression of the word