ANIMAL MAGIC

Continuing our look at some of the entertaining programs that you can create, we present our own version of the Animals game. This game has always been regarded as fun because it gives the computer an apparent ability to think. However, the principles it employs lie behind many serious artificial intelligence programs.

Animals is a game in which the computer tries to guess the name of the animal that the player is thinking of. It does this by asking questions such as 'Can it fly?', 'Is is furry?' and so on. You are allowed to answer only 'yes' or 'no' and the computer gradually works its way to a point where it can make an 'educated' guess. Obviously, it is quite surprising, especially for people who aren't familiar with computers, that the program is able to do this. The two aspects that make the program particularly entertaining are the computer's ability to communicate in sensible English (even if your own responses are limited to 'yes' and 'no') and the vast store of knowledge that the computer can draw on to guess the animal.

Animals is actually a very simple *heuristic program* — a program that teaches itself to improve its performance as it is running. When the program is first used it 'knows' only two animal names and one question. Depending on whether or not you answer 'yes', it can attempt to guess at what your animal is. If the computer guesses wrong (which it almost certainly will do the first time), the program asks you to enter the name of your animal and a question to distinguish it from the program's guess at your animal. This information is then added to the program's database to build up a 'tree of knowledge' that it can use in the next game. Every time you play the game, the tree increases in size until finally the program is guessing most of your animals correctly and only occasionally discovering a new one.

The interesting point to remember is that the program still doesn't 'know' anything about animals. It is blindly following a guide made up from the combined knowledge of all the players who have played it. The information might as well be about different types of beer, motorcycle parts, medical complaints or your friends and family. A version of the program that allowed you to define the initial question and two answers could be used for a whole variety of different tasks. In other words, it is not the data itself that makes the program work but the way in which it is being organised.

Building the tree with a simple BASIC program is not particularly difficult. Most structures like this are held in BASIC arrays: in this case using T$() for the questions and the names of the animals, and Y() and N() for the links between particular entries in T$. These links form the path through the tree. For any one entry in T$, the corresponding entry in Y() tells the program where to look if the answer to that question is yes. Similarly, the entry in N() is the link for a negative response. At the end of the tree the text in T$() is not a question but the name of an animal. Both Y() and N() are set to 0 in this case and the program has to make a guess that the player is thinking of that particular animal.

This version of the program has been kept short and simple to show you the principles involved. If you want to enhance it, you could improve the presentation for your machine by adding colour graphics, sound and so on. A major improvement would be to give the program some way of storing its database to tape or disk. The best versions of Animals you can find are those that people have been playing for years, and which have built up a vast tree of animals, mythical animals, objects, famous people, friends and so on, all mixed up into one gigantic database. An even more impressive version would enable you to set the first question and alter and edit entries in the tree so that the program would be practical for more serious uses.

Basic Flavours

This program is written in Microsoft BASIC so it should run unchanged on most machines; the only change you might want to make is to the format of the PRINT commands, if you don't like the screen display.

On the Spectrum, all assignment statements must begin with the keyword, 'LET'. Rewrite the following lines as shown:

```
45 LET L=40 :REM No. of chars. in a question
50 DIM Y(N):DIM N(N):DIM T$(N,L)
150 LET I$=A$(1):LET P$="A "
200 IF A=30 THEN PRINT:PRINT "BYE":STOP
230 IF Y(P)=0 AND N(P)=0 THEN GOTO 290
```