

bounded by walls to the south, north and east.

To save on memory space, the array could be numeric rather than string, and the four-digit number regarded as a binary number. In our example above, the cell containing north, east and south walls would contain the number 11 (1011).

All cells would start with four walls. By randomly generating the entrance from anywhere along the perimeter, the next cell could be chosen at random from any of the three adjacent cells. When that cell is chosen, the sequence continues — randomly selecting a cell from any of the three adjacent cells, disregarding the one you have just come from.

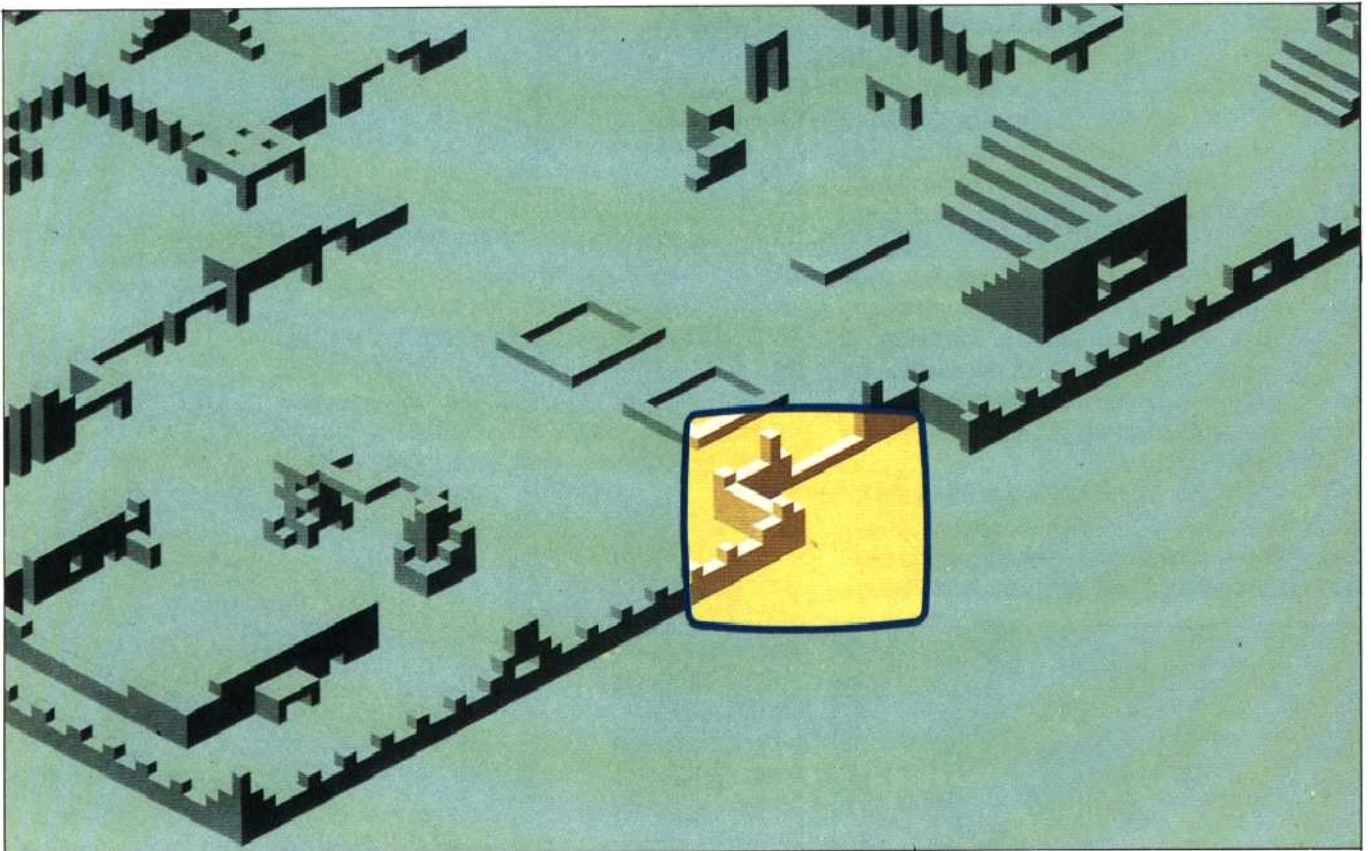
As each new direction is chosen, the appropriate 'wall' is removed from the cell you are about to leave and the cell you are about to enter. Checks must be made to ensure you don't move outside the boundaries of the maze (unless the particular perimeter cell is to be the exit point), or create closed circuits (all parts of a maze should be accessible from any other part).

sibilities), two adjacent walls (four possibilities), three adjacent walls (four possibilities).

Using the appropriate binary number (0-15) for each of these, it is possible to 'rotate' the number left or right to obtain the appropriate view faced by the player. For example, a north wall could be represented by 2 (0010), south wall by 8 (1000), east wall by 1 (0001) and west wall by 4 (0100). If the player facing north in a 'west wall only' room (4) turned to face the west, his view of the room would now be one bounded by a north wall (since facing forward in a three-dimensional display is always to the 'north'). As the player turned to his left (the west), moving the bit pattern one place to the right supplies the description we want, i.e. west wall binary 0100 (decimal 4) becomes binary 0010 (decimal 2 — a north wall!) The bits are moved in the opposite direction when turning right, twice for an about turn. It is necessary, of course, to include a system for 'wrapping around' the bits that are lost from the left- or right-hand end of the half-byte during this process, otherwise the identifying

Ant Attack

When this game is run on the Spectrum, the computer's screen acts as a window onto a large maze-like field of play. As the play moves, the screen will move, revealing more of the scene



When a cell that has fewer than four walls (i.e. a cell that has already been visited) is encountered, the program must choose another of the remaining adjacent cells. If all adjacent cells have been visited, the program has to 'step back' to the previous cell visited and take a new branch.

Another method of recording the characteristics of a cell is by a more sophisticated use of binary numbering, which is especially useful for displaying three-dimensional perspectives. There are 16 possible ways a cell can be built: no walls, walls on all sides, a single wall (four possibilities), walls on opposite sides (two pos-

characteristics of a cell will be changed each time the player turns within it. A cell originally defined as 0011, for instance (walls to the north and east), must become 0110 if the player turns to the right, and 1100 if he turns completely around.

In machine code, there are special instructions for rotating binary numbers left and right. In BASIC, a four-bit binary number expressed as a decimal number in the range 0-15 can be shifted left by multiplying the number by two, and then subtracting 15 if the result exceeds 15. To rotate right: divide by two if it is an even number, or add 15 and then divide by two for an odd number.