-- Messages from file: [BBN-TENEXE]<JHAVERTY>MESSAGE.TXT;1
-- Tuesday, May 23, 1978 11:38:22-EDT --

Return to Jack Haverty

Mail from MIT-DMS rcvd at 23-May-78 1110-EDT
Date: 23 May 1978 1056-EDT
From: PDL at MIT-DMS (P. David Lebling)
To: JFH at MIT-DMS
Subject: Maze Algorithm
Message-id: <[MIT-DMS].76602>

The enclosed should do the trick:

----- Enclosure #1: DSK:PDL;HIDDEN > -----

HIDDEN LINE ELIMINATION

Given a world of cubes (very similar to a maze, in fact) and an observer within the maze, the problem is to decide exactly what the observer sees in terms of lines in a perspective view from his location.

The maze consists of cubical areas that are either filled in or empty (walls and halls). The maze is not allowed to contain a four cube array that is totally empty (the maze will not be drawn correctly in this case).

To simplify the problem, the picture is vertically symmetric about the vanishing point.

The picture can be drawn totally on the basis of the square the observer is in and the three neighboring squares on each side. For the left side we have Fig 1:

```
------------------------
|         |           |
|   L2    |    R2     |
|         |           |
|         |           |
     4---------1o---------2
|         |           |
|   L1    |    R1     |
|         |           |
|         |    X | =  |   |
------------------------
   3
```

Figure 1: Top view.

Observer is in R1, facing towards R2. A similar diagram could be drawn for the blocks to the observer's right, but any algorithm that works for the left will work equally well for the right.

There are four lines that are of interest, all radiating from the "o" in the figure:

1) a vertical line ("out" of the paper)
2) a horizontal line to the right
3) a line running back past the observer
4) a horizontal line to the left.

In perspective:

```
 |1
 |
 |
 |
4-------o--------2
       /
      /
     /
    3
```

Figure 2: Perspective

Now we create a table of which lines are seen given different states of the four blocks we are interested in (obviously any possibility in which R1 is 1 (that is, a wall) is uninteresting since there could then be no observer):

Block				Line			
L1	R1	L2	R2	X1	X2	X3	X4
0	0	0	0	f	f	f	f
1	0	0	0	1	0	1	0
0	1	0	0	f	f	f	f
1	1	0	0	f	f	f	f
0	0	1	0	1	0	0	1
1	0	1	0	0	0	1	0
0	1	1	0	f	f	f	f
1	1	1	0	f	f	f	f
0	0	0	1	0	1	0	1
1	0	0	1	1	1	1	0
0	1	0	1	f	f	f	f
1	1	0	1	f	f	f	f
0	0	1	1	0	1	0	1
1	0	1	1	1	1	1	0
0	1	1	1	f	f	f	f
1	1	1	1	f	f	f	f

Table 1: Truth table.

The tabular results are converted into Karnaugh Maps:

```
X1      X2
  \ L2R2        \ L2R2
L1R1\  00 01 11 10 L1R1\  00 01 11 10

  00    f  0  0  1   00    f  1  1  0
  01    f  f  f  f   01    f  f  f  f
  11    f  f  f  f   11    f  f  f  f
  10    1  1  1  0   10    0  1  1  0


X3      X4
  \ L2R2        \ L2R2
L1R1\  00 01 11 10 L1R1\  00 01 11 10

  00    f  0  0  0   00    f  1  1  1
  01    f  f  f  f   01    f  f  f  f
  11    f  f  f  f   11    f  f  f  f
  10    1  1  1  1   10    0  0  0  0
```

Table 2: Karnaugh Maps.

Therefore:

$$X_1 = \overline{L1}\,(R2 + \overline{L2}) + R2\,L1$$

$$X2 = R2$$

$$X3 = \underline{L1}$$

$$X4 = \overline{L1}$$

The only one that is at all painful is X1 (the vertical line), but it can be reduced to the decision tree in Figure 3. In the figure a square identifier (as "L1") in parens implies a test, with the results branching as indicated. A line identifier underlined means to output that line (make it visible).

```
              (L1)
          1 /      \ 0
           /        \
     X3 __ /          \ X4 __
         /              \
    (R2)                 (R2)
      1 /    \ 0        1 /   \ 0
       /      \          /     \
   X2  __      \      X2  __         X1  __
  X1  __        \
    (L2)
         \ 0
     \
      X1  __
```

Figure 3: Decision Tree.