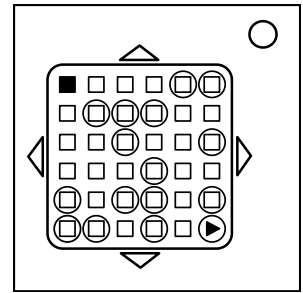


## On the Subject of Procedural Mazes

*No, this isn't a "Not" module. No, this isn't based on Generated Maze.*

The defuser must navigate the white light to the blue triangle using the arrow buttons, while avoiding the invisible walls in the maze.



Each wall is either present, absent, or undecided. The initial state of the walls is determined as follows.

1. All walls on the border of the maze are present.
2. Any remaining undecided walls around the starting position are absent.
3. The remaining walls are undecided, and will be generated as the defuser traverses the maze.

If the defuser attempts to cross a present wall, the module will strike and regenerate. The defuser can also regenerate the module, without striking, by holding any button for three seconds.

### The Initial Seed

Create a binary number consisting of six bits using the number of circles in each column of the maze, right to left. If the number of circles is even, that column's bit is a 1. Otherwise, it's a 0. This is the initial seed; the seed will change as walls are generated, but will always be six bits long.

### Generating Walls

After each move, look at the previously undecided walls around the current cell, beginning with the wall the defuser just crossed and going clockwise. For each of these, determine if the wall is now present or absent, as follows.

1. Use the table on the next page to determine a slice of the maze to use as a bitmask.
2. For each cell in the bitmask with a circle, check if there is a "1" in the same position in the current seed.
3. If there is an **even** number of such cells, the wall is **present**. Prepend a "1" to the start of the current seed and remove the rightmost bit.
4. If there is an **odd** number of such cells, the wall is **absent**. Prepend a "0" to the start of the current seed and remove the rightmost bit.

If the most recent move was...	...then the bitmask is the current...
right	column, from top to bottom.
up	row, from left to right.
left	column, from bottom to top.
down	row, from right to left.