Deployment Instructions:

. 1) Framework: Pygame

. 2) Algorithm: Backtracking Algorithm

Algorithm Illustration:

- If we reach the Destination:

  It should return an array containing only the position of the destination

- else

  o Move in the forwards direction and check if there is a solution. Let's call this '1'.

  o In case there is no solution, we move down. Let's call this '2'.

  o If either of the two above options work, we add the current position to the solution obtained at either '1' or '2'. Let's call this '3'.

Maze Creation:

The code creates a random NXN maze where N is a variable that can be input by the user. The hurdles in the path are not predefined. They are basically defined randomly using black and white color.

The starting point is indicated with a red color whereas the destination point is indicated with a blue color. The color black represents a block whereas the white color represents a legit path. Using the maze solver method described above, in case there is a path from source to destination, the entire path is highlighted with a blue color.