Initial Test

Farbod Younesi

1- The provided python script parses a maze image and solves it using the A* algorithm. Here's how it works:

    1- It reads the image, applies a threshold to make the image binary, and crops the additional pixels out of the maze.
    2- The map of the maze is extracted by finding the lowest thickness of walls and tiles, then each tile and the adjacent walls or placement of walls are treated as a pixel.
    3- The entrance and the exit of the maze is extracted from the outer layer of pixels of image.
    4- The A* algorithm is used to find the path between the start and end tiles.
    5- The generated path is both displayed in both map and image of the maze

2- If the maze is placed on a table and a robot arm is used to navigate the maze, additional factors must be considered:

    1- Coordinates of the tile's centers should be considered as the points to move between.
    2- The points from the solution must be translated into real-world coordinates and then joint angles for the arm.
    3- The robot arm's degrees of freedom must be sufficient to traverse the maze.

3- For a mobile robot navigating a real 3D maze, additional considerations include:

    1- Accurate localization is required to maneuver through the maze. Heading and encoder feedback is essential.
    2- Stable controller is also needed to control the robot's distance from the center of the tiles. Distance sensors are vital for this task.