Submissions: This week's report should contain:

(i) The ultrasound thresholds for starting down a street, stopping mid-street, or initiating a U-
Turn.

40 cm is the default distance below which the robot considers the path ahead blocked by an obstacle, using the forward-facing ultrasound sensor.

To stop mid street or during line following we chose the ultrasound threshold to be approximately 10 cm, and then 20 cm to keep going

Initiating Optional U-turn will be done for next goal.

(ii) What changes you made to the exploring logic (for blockages).

Marking Blocked Streets in the Map

When a blockage is detected we call:

- map.set_blocked(x, y, heading, True)
- This updates the map so the robot remembers which streets are blocked and avoids them in the future.

Replanning or Choosing Another Street

- If a street is blocked, the robot:
- Looks for another unblocked, unexplored, or unknown street at the intersection.
- If none are available, it may replan using Dijkstra's algorithm to find a new goal or path.
- If no path is found, exploration ends.

Blockage Checks After Moving

- After moving forward and reaching a new intersection, the robot checks again for blockages and updates the map accordingly.

Consistent Blockage Handling in All Modes (Navigation.py)

- The same blockage logic is used in:
- Exploration (autonomous_step)
- Goal-seeking (step_toward_goal)
- Initial alignment (align_to_road)

Change	Effect
Check blockage before moving	Prevents robot from driving into obstacles
Mark blocked streets in map	Robot remembers and avoids blocked streets
Replan or choose another street if blocked	Keeps exploration safe and efficient
Check blockage after moving/intersection	Updates map with new blockages
Unified blockage logic	Consistent behavior in all navigation modes

(iii) What logic you used to find a previously unknown intersection. Double-obstacle?

In our directed_exploration function, the robot uses the following logic to find and reach a previously unknown intersection:

1. Local Exploration Preference:

- At each intersection, the robot first checks for any unblocked streets that are marked as STATUS.UNKNOWN or STATUS.UNEXPLORED.
- It prefers to turn toward and drive down these streets, especially those that point most directly toward the ultimate goal (using Euclidean distance as a heuristic).

2. Global Planning with Dijkstra

If there are no local unknown/unexplored streets, the robot:

- Searches the map for all intersections that have at least one unblocked unknown or unexplored street.
- For each such intersection, it calculates the total cost as:
- total_cost = (Dijkstra cost from current position to intersection) + (Euclidean distance from intersection to goal)
- selects the intersection with the lowest total cost as the next sub-goal and uses Dijkstra's algorithm to plan a path to it.

3. Handling Double-Obstacle (Blocked Paths):

- If a street is blocked marked as blocked in both directions in the map.
- Blocked streets are never considered for exploration or as part of a path in Dijkstra.

- If all possible paths to the goal or to any unknown/unexplored intersection are blocked, the robot stops and reports that exploration is complete or the goal is unreachable.