Roboteksperimentarium
Exam description of solution

Boas Kejser Lousky fnp651, Matin Nafar qhw109,
Jonas Tamim qmh341, Mikkel Dollerup ndt926

Our solution for the robot-rally race comprises a `RobotController` class that incorporates a state machine. The robot is continuously updated within while loops to execute a sequence of commands, which include driving straight, making turns, and reading the sonar sensors. In our primary file, `main_dumb.py`, our strategy unfolds in two distinct steps. Step 1 involves pinpointing the robot's position within the coordinate system, while Step 2 entails identifying the next landmark to visit, turning towards it, and subsequently making the visitation. Throughout these actions, we have implemented a range of edge cases and fail-safes to ensure that we stay on course.
Here's a breakdown of the two steps:

1. In Step 1, we commence by locating landmark 1 and either landmark 2 or 3. If neither is initially found, we execute a slight robot movement and resume the search for landmarks 1 and either 2 or 3. Once we successfully locate these landmarks, we proceed to calculate the robot's pose using the distances to the two landmarks and the distance between them. Subsequently, we calculate the pose angle by considering these coordinates in relation to the landmark we are currently facing.

2. Moving on to Step 2, we turn the robot to face the next landmark, which is landmark 1 during the initial iteration of Step 2. We capture an image to determine the distance to this landmark and then navigate the robot towards it. Following the visit, we once again turn towards the next landmark, capture an image, and hope we spot the next landmark. In cases where our view is obstructed or we encounter errors that prevent us from spotting the next landmark, we initiate a search procedure by rotating the robot 15 degrees at a time until we complete a full 360-degree rotation. If the landmark remains elusive, we implement a ”random” move at a 45-degree angle and attempt the search once more. Once we eventually locate the next landmark, we make the necessary adjustments to the robot's angle, aligning it with the landmark's position, and then visit it. This process repeats until all the designated landmarks have been visited in the correct order.