Subsystem test plan

Assumption for all tests as follows:

- Gazebo, Rviz and Ros2 is all installed, up to date and working.
- The Andy Package is downloaded and built inside the Ros2 workspace

Procedure for starting Gazebo, Rviz and Andy Package:

Terminal 1:

1. export TURTLEBOT3_MODEL=waffle_pi
2. ros2 launch turtlebot3_gazebo turtlebot3_world.launch.py4

Terminal 2:

1. export TURTLEBOT3_MODEL=waffle_pi
2. ros2 launch turtlebot3_navigation2 navigation2.launch.py use_sim_time:=true
3. Localise the turtlebot in RViz using the 2DPoseEstimate function.

Terminal 3:

1. colcon build --packages-select andy
2. ros2 run andy path_planner

All input for testing to be written in terminal 3 after the nodes have been activated.

Input for the current position must be in the format "position" to print a position in the format “x y” where x and y are axial coordinates in m. All future reference to x and y are axial coordinates in m unless specified otherwise.

Test 1 (P)

Requirements:

TurtleBot’s can reach goals without colliding with any stationary known obstacles.

Procedure:

Send a command to the turtlebot to move to a goal after localisation has been completed. Goal command input format "goal x y" where x and y are doubles.

Results:

Pass only if no objects were hit and the distance to goal is less than 0.3m.

Test 2 (C)

Requirements:

P and TurtleBot can recognise and navigate around newly discovered stationary objects.

Procedure:

Send a command to the turtlebot to move to a goal after localisation has been completed. Goal command input format "goal x y" where x and y are doubles. During the movement, an objects coordinates can be inputted into the terminal mimicking receiving information from the SLAM subsystem. Input must be in the format "object x y" where x and y are doubles.

Results:

Pass only if the turtlebot stops when a new object is "discovered", know and new objects are avoided and the distance to goal is less than 0.3m.

Test 3 (D)

Requirements:

C and implements own rudimentary path planning.

Procedure:

Send a command to the turtlebot to move to a goal after localisation has been completed. Goal command input format "goal x y" where x and y are doubles. During the movement, a objects coordinates can be inputted into the terminal mimicking receiving information from the SLAM subsystem. Input must be in the format "object x y" where x and y are doubles.

Results:

Pass only if standards from the credit test is passed and implements its own path planning algorithm without any use of the Nav2 package.

Test 4 (HD)

Requirements:

D and optimise to move along the shortest path.

Procedure:

Send a command to the turtlebot to move to a goal after localisation has been completed. Goal command input format "goal x y" where x and y are doubles. During the movement, a objects coordinates can be inputted into the terminal mimicking receiving information from the SLAM subsystem. Input must be in the format "object x y" where x and y are doubles.

Results:

Pass only if standards from the distinction test is passed and the terminal displays distance score for at least 3 paths in total length (m) of path. The turtlebot follows the lowest scoring path. In the instance of a new object being discovered, repeats distance scores and chooses the new shortest path.

Test 5 (Extension)

Requirements:

HD and create a safety factor to move around the objects within a range that can be set.

Procedure:

Send a command to the turtlebot to move to a goal after localisation has been completed. Tolerance of distance to objects must be inputted before the goal is sent. Input must be in the format "distance to object d" where d is the minimum distance to objects desired. Goal command input format "goal x y" where x and y are doubles. During the movement, a objects coordinates can be inputted into the terminal mimicking receiving information from the SLAM subsystem. Input must be in the format "object x y" where x and y are doubles.

Results:

Pass only if standards from the high distinction test is passed and the turtlebot prints out distance to objects when it is within 0.2m of the tolerance setting. The turtlebot stays outside of the tolerance setting. Tolerance setting does not get overwritten when new goals are sent or new objects are discovered.