

Main Project (IROS)

Conditions:

- Worth 50% of the final grade + 10% bonus.
- Solving the main task is enough to pass the course with C, **in case that** you did not solve the first two homework.
- Work in groups (up to 5 members recommended) or individually.
- Deadline: Friday 26/07/2024 – 23:45.
- For assistance, you can ask me or your friends.
- Submit your package folder on Moodle, put the name of group members as a comment inside setup.py, also submit a screen recording show that project is working.

Solving Grid World Problem

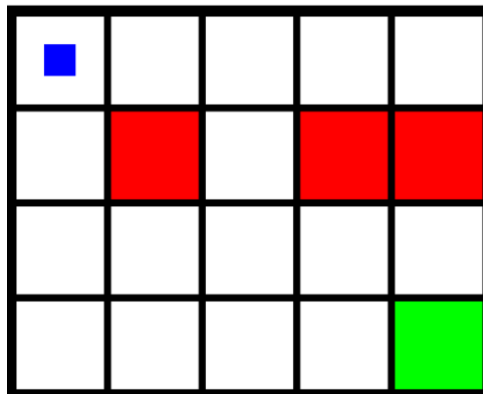
Objective:

Solving Grid World problem finishing the following subtasks:

1. Write a new node called `world_solving_node`.
 - a. Subscribe to '`world_state`' and convert the initial state of World to a graph (Use any graph representation you want).
 - b. Find the shortest path between the player initial position (upper left) and the terminal position (lower right) (You are free to use any algorithm you want).
 - c. Solve the problem by publishing the commands to "`usr_cmd`", it is important that you send each command after you received the new state.
2. Write a launch file that will start the both nodes world node and solver node.
3. You can add a delay before starting sending the first command so you will be able to record the screen from the starting of the solution.

Bonus: To be announced later.

The following is an example for the grid world with size 4x5 which is the default size:



Red blocks mean obstacles, they always have a random positions.

Green block means Terminal block and it is always at the lower right corner.

Blue small square means the player position and it always start from the upper left corner.

State matrix for the previous example:

$$state = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$