CUERT Autonomous Dev. Team

Recruitment Task

Intro

In modern logistics, autonomous robots play a vital role in efficient warehouse management. These robots are tasked with navigating through the warehouse environment, picking up items, and delivering them to designated locations. To achieve this, the robots need to utilize localization, mapping, and path planning algorithms.

For commercial autonomous robots, a high-definition road map is provided in addition to accurate GPS localization, which helps greatly in the autonomous task.

But in our case such a map is rarely provided, that's why we tend to sense the environment around us and construct a local map in addition to determining our location on that map.

That's essentially what's so-called "SLAM"; Simultaneous localization and mapping.

After SLAM we need to figure out how to reach a certain goal location, that's where "Path Planning" comes into the picture.

In addition to SLAM and Path Planning, there exist Control and Perception algorithms to guide the robot through the generated path and sense other objects around it like workers and other robots, respectively.

GOAL

The goal of this task is to develop an algorithm that enables an autonomous robot to efficiently navigate through a warehouse, pick up items, and deliver them to specified drop-off points. The robot should be able to dynamically map its environment, plan paths to reach items, and avoid obstacles.

Task Description:

1. **Warehouse Environment:**

   You are provided with a warehouse represented as a grid. The grid contains various elements:

   "**.**" represents empty spaces where the robot can move.

   "**#**" represents obstacles that the robot should avoid.

   "**>**" represents the robot's initial position.

   "**P**" represents pickup points for items.

   "**D**" represents drop-off points for items.

```
>..........
...........
......P....
..#######..
..#######..
..#######..
..#######..
..#######..
...........
...........
..........D
```

2. API:

You will be provided with a **"warehouse.h"** header file containing functions to interact with the environment:

isWalkable: Checks if a cell around the robot is walkable.

moveRobot: Moves the robot one step in different directions.

getRobotPos: Retrieves the current position of the robot.

getItemPos: Retrieves the item position.

getGoalPos: Retrieves the delivery(goal) position .

pickItem: to pick the item from the ground.

3. Path Planning:

Your algorithm should implement a path planning strategy to navigate the robot through the warehouse. This **includes**:

- Avoiding obstacles and finding optimal paths to pickup points and drop-off points.
- Efficiently handling multiple items and their destinations[**Bonus**].

4. Execution:

Use the provided **"main.cpp"** file containing a simulation environment. Your code should be written in the initialize and solve functions. The initialize function sets up the environment, and the solve function implements the robot's behavior.

5. Compilation:

Compile your code using any C++ compiler and provide the necessary arguments, similar to this **example**: `g++ main.cpp -o main.exe main.exe warehouse1.txt`

Evaluation Criteria:

You will be evaluated based on your algorithm's **ability to**:

- Navigate the robot through the warehouse while avoiding obstacles.
- Efficiently pick up items and deliver them to the correct drop-off points.
- Utilize path planning techniques to optimize the robot's movement.

This task assesses your skills in combining **localization**, **mapping**, and **path planning** algorithms to achieve autonomous navigation and item delivery in a warehouse environment.

BONUS :

Solve it for multiple items with the same destination you can add a bit difficulty by changing the destination for each item but you will need to modify the API to do that in that case you are free to add or remove what you want from the API to solve it for multiple items and multiple destinations for each item.