Project Code: TCV2
Controlling a Toy Car around a Grid [Version 2]

Project Duration : 18-Aug-2024 – 07-Sep-2024
Submission Information : (via) Google Form

Objective:

In a car racing competition the driver has to control a car around a circular shaped track. You have a similar task at hand. You have to control a car around a discrete grid so that it makes a complete lap (comes back to the starting position). Presume that the car is autonomous, and it senses its state using radar signals in various directions. If the radar signal does not hit any obstacle, it returns a 1, else it returns a 0. This will help move the car.

Also, the car can sense its current position in the grid. You have only the radar signals to detect the path (no other variable for its indication). Find which algorithm is able to identify the path based on the radar sensings only and solve the task.

Your task is to build an agent using RL algorithms to help solve the problem. In particular, you shall be doing the following tasks:

1. Design an Environment Class that models the given specifications. You have to build the state space, action space for the robot and the reward structure following the structure of Gym Environment Classes. The reward structure can however change depending on the requirement of the algorithms. I will provide you with the starter code for reference.
2. Design a visualization tool for rendering how the algorithm controls the robot. The rendering function should preferably be part of the Environment Class. This will help demonstrate the agent's actions.
3. Text-based visualizations will not be accepted.
4. Your visualization should clearly illustrate various features specific to your design environment and show the current state, action and reward received.
5. Demonstrate the Value Iteration, and Temporal-Difference (TD) Learning algorithms using the Env class designed above.
6. Provide the initial, intermediate (after 0<n<N iterations) and final (after N iterations) policies for all the algorithms.
7. Compare the results from the above algorithms, and give insights if any of why the results are better or worse for one algorithm compared to others.
8. Run the corresponding algorithms from this Github repository, and compare with your results.

Note: *The program can be written in preferably Python programming language, and should run in Linux.*

Submission Details: (to be submitted through Google Form)

1. ZIPPED Code Distribution [create separate Python file for the Env Class]
2. A brief (2-4 page) report/manual of your work. Emphasize on Experiments, results, Discussions, Remarks / Observations, Machine Config. (optional).

Submission Guidelines:

1. You should preferably use Python, which should run on a Linux Environment.
2. You should include a README file in the code distribution, which describes how to run the codebase and the important files.
3. The submitted program file *should* have the following header comments:
   # Group Number
   # Roll Number (Name of member)
   # Project Number
   # Project Title
4. Submit through Google Form only.
   Link to form: https://forms.gle/G5Gts9MvWbVoRpmq5
5. You should name your file as <RollNo_ProjectCode.extension>.
   (e.g., 24CS10000_TCV2.zip for code and 24CS10000_TCV2.pdf for report)

You should not use any code available on the Web. Submissions found to be plagiarized will be awarded zero marks.

For any questions about the assignment, contact the following TA:
Somnath Hazra (Email: sommnathsh@gmail.com)