Project 1: Dynamic Programming

Collaboration in the sense of discussion is allowed, however, the assignment is individual and the work you do should be entirely your own. See the collaboration and academic integrity statement here: https://natanaso.github.io/ece276b. *Books, websites, and papers may be consulted but not copied from. It is absolutely forbidden to copy or even look at anyone else's code.* ***Please acknowledge in writing people you discuss the problems with.***

Submission

Please submit the following files on **Gradescope** by the deadline shown at the top right corner.

1. **Programming assignment**: upload all code you have written for the project and a README file with a clear, concise description of the main file and how to run your code.
2. **Report**: upload your report in pdf format. You are encouraged but not required to use an IEEE conference template[1] for your report. Please refer to Slide 12 and 13 of ECE 276A Lecture 1 for the expected report structure and contents[2].

Problems

In square brackets are the points assigned to each part.

1. This project focuses on autonomous navigation in a **Door & Key** environment, shown in Fig. 1. The objective is to get our agent (red triangle) to the goal location (green square). The environment may contain a door which blocks the way to the goal. If the door is closed, the agent needs to pick up a key to unlock the door. The agent has three regular actions, *move forward* (MF), *turn left* (TL), and *turn right* (TR), and two special actions, *pick up key* (PK) and *unlock door* (UD). Taking any of these five actions costs energy (positive cost). For example, an action sequence minimizing the cost in Fig. 1 is:

$$\begin{aligned} PK \rightarrow MF \rightarrow MF \rightarrow TR \rightarrow UD \rightarrow MF \rightarrow MF \\ \rightarrow TR \rightarrow MF \rightarrow MF \rightarrow MF \rightarrow TR \rightarrow MF \end{aligned}$$

   Design and implement a Dynamic Programming algorithm that minimizes the cost of reaching the goal in two scenarios:

   (a) *"Known Map"*: you should compute a control policy on each of the 7 environments provided in the starter code and evaluate its performance on the same environment that it was computed for. Further description of the 7 environments and their functionality is provided in the accompanying starter code and README file.

   (b) *"Random Map"*: you should compute a single control policy, whose performance may be evaluated on any of the 36 random 8×8 environments. The size of the grid in the random maps is 8×8 and the perimeter is surrounded by walls. Denote the index of the top-left cell as $(0, 0)$, bottom-left as $(0, 7)$, top-right as $(7, 0)$ and bottom-right as $(7, 7)$. There is a vertical wall at column 4 with two doors at $(4, 2)$ and $(4, 5)$. Each door can either be open or locked (requires a key to open). The key is randomly located in one of three positions $\{(1, 1), (2, 3), (1, 6)\}$ and the goal is randomly located in one of three positions $\{(5, 1), (6, 3), (5, 6)\}$. The agent is initially spawned at $(3, 5)$ facing up.

[1] https://www.ieee.org/conferences_events/conferences/publishing/templates.html
[2] https://natanaso.github.io/ece276a/ref/ECE276A_1_Introduction.pdf

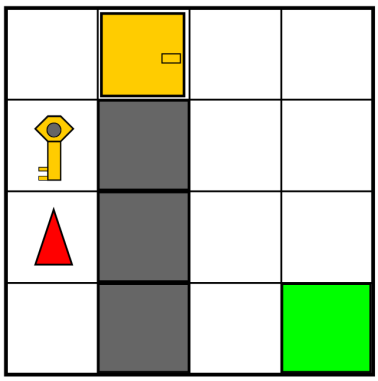

Figure 1: Door & Key Environment: An agent (red triangle) needs to navigate to a goal location (green square). The white squares are traversable, while the gray squares are obstacles that the agent cannot go through. The agent may need to pick up a key if a door along the way to the goal.

Write a project report describing your approach to the Door & Key problem. Include the following sections:

(a) [5 pts] **Introduction:** provide a short introduction to the problem, its potential application to robotics, and your approach at a high level.

(b) [25 pts] **Problem Statement:** formulate the problem as a Markov Decision Process. Clearly define the state space $\mathcal{X}$, control space $\mathcal{U}$, motion model f or p_f, the initial state $\mathbf{x}_0$, the planning horizon T, the stage and terminal costs ℓ, $\mathfrak{q}$, and any other elements necessary to make this a well defined problem.

(c) [25 pts] **Technical Approach:** describe your algorithm for obtaining an optimal control policy that minimizes the agent's cost. This should be a clear description of the algorithm that you implemented in python.

(d) [20 pts] **Results:** use your python implementation to determine an optimal control policy the the *"Known Map"* and *"Random Map"* scenarios. Your results should include visualizations of the trajectory followed by the agent with different starting positions and orientations for each environment. The results section should also include a discussion of your algorithm's performance: what worked, what did not, and why.

In addition to the report, your submission should include the code you have written to solve the Door & Key problem:

(e) [25 pts] **Code:** your code will be evaluated based on *correctness* and *efficiency*. Correctness refers to whether your code implements the algorithm you present in your report and whether the algorithm you present actually solves the problem you formulated. Efficiency refers to your code being able to execute and produce control sequences for the agent in reasonable time. The results presented in the report should be consistent with the output of your code. The code should have a clear structure and comments. **Copying, rephrasing, or even looking at anyone else's code is strictly forbidden. Writing your own code is the best way to avoid plagiarism.**