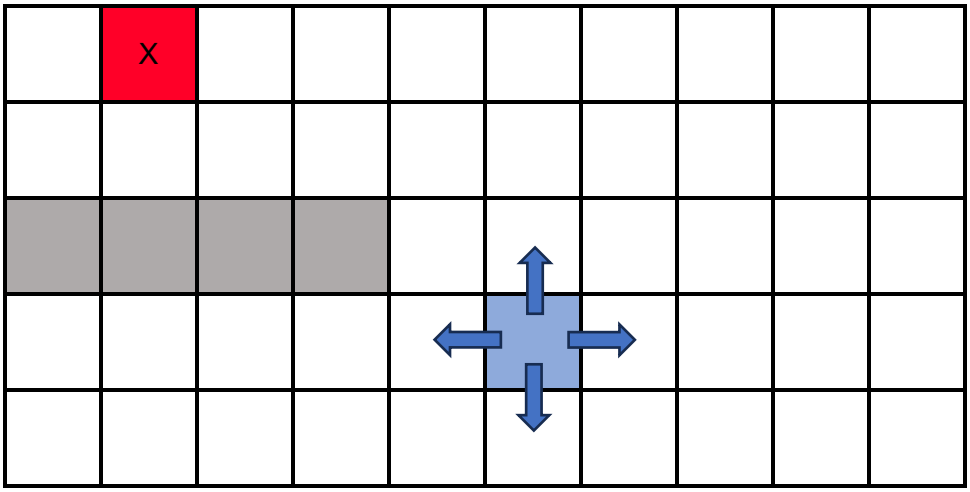
ROB 537
Learning Based Control
Fall 2025
HW #3: Reinforcement Learning
Due 11/4

Our learning environment is a 10x5 gridworld. There is:

- a door on the red square (x) with a reward of 20;
- a solid wall (gray) the agent cannot move across;
- a reward of -1 for every time the agent is in a state other than the red door state.

The agent starts at a random location and has five actions (move left, right, up, down, or stay in place), the state of the system is the location of the agent (x,y), and an episode is 20 time steps.

We provide a gridworld in python to help get you started.



Part A: Write a Q-learning algorithm to solve this problem. Run the algorithm at least 10 times. Plot the learning curves and learned Q-value tables. How did the algorithm perform? If it did well, explain why it did well. If it struggled, explain why it struggled.

Part B: Choose your own adventure. Pick option 1, 2, or 3 for Part B of your homework.

Option 1, Dynamic Environment: Modify the environment so that the door moves every even time step. Keep the initial starting location of the door the same as before. Run the algorithm at least 10 times and plot the learning curves and Q-value tables. **Discussion**: Explain how the door moves (randomly, or following a fixed path, etc). How does the algorithm perform compared to Part A? If it does better, explain why it does better. If it does worse, explain why it does worse.

(continue reading on page 2)

Option 2, Local Sensing: Modify the state representation for the agent. Rather than giving the agent its position, create a virtual sensor that gives relative information about the door and the obstacles. Run the algorithm at least 10 times and plot the learning curves. Rather than Q-value tables, plot a few paths the agent took in the environment. **Discussion**: Explain how your virtual sensor works. How does the algorithm perform compared to Part A? If it does better, explain why it does better. If it does worse, explain why it does worse.

Option 3, Reward Shaping: Modify the reward for the agent. Run the algorithm at least 10 times and plot the learning curves and Q-value tables. **Discussion**: Show us your new reward function. How does the algorithm perform compared to Part A? If it does better, explain why it does better. If it does worse, explain why it does worse.

Cite outside information you reference, tools you use, and people you work with. (Ex: Referenced Wikipedia, used github copilot w. Claude 4, got help from TA.)

Submit your report as either a PDF with your code attached as an appendix, or a zip file including a PDF of your report and your code in separate file(s).