This assignment consists of four parts. This is the final assignment for the course.

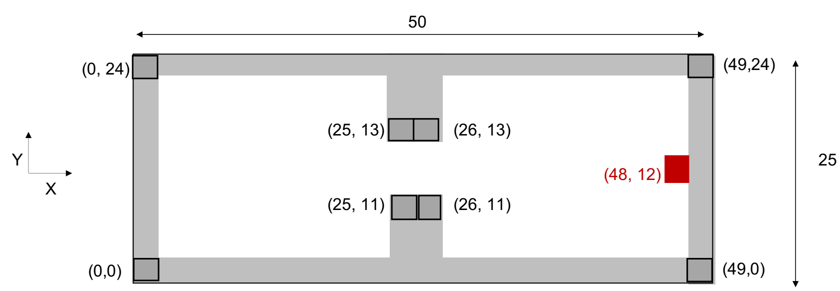
Please use Python for implementation using only standard python libraries (e.g., numpy, matplotlib etc.). Please do not use any third party libraries/implementations for algorithms.

Please prepare a report to accompany your implementation. The report should contain responses to questions and the desired plots/graphs. Briefly describe the key findings/insights for the graphs. Ensure the reproduction of graphs (modulo probabilistic execution) for your submission.

Please refer to the submission instructions on the course webpage.

1. **MDP.** Consider a mobile robot in the grid world domain. Please see the figure drawn below. Note that the figure is indicative and not drawn to scale. Some of the grid cells are drawn with the associated coordinates written along side. The other grid cells are implicit in the diagram.

   - The world is a grid of dimensions (50×25). The grid boundary that links the grid indices $(0, 0)$, $(0, 24)$, $(49, 24)$, $(49, 0)$ are walls. There is a wall in the middle portion of the grid (coordinates shown) with a gap of one grid cell between the bottom and the top half. All grid cells constituting walls are shown in *gray* colour.
   - The robot has a choice of four actions: $\uparrow, \downarrow, \rightarrow, \leftarrow$ that can move the agent to an adjacent grid cell in the *north*, *south*, *east* or *west* directions respectively.
   - The action model is such that the nominal outcome (when the robot moves in the intended direction) occurs with probability 0.8, and the other three outcomes occur with probability 0.2/3. If an action results in a collision with a grid cell part of a wall, then the robot *does not* move and remains in the same grid cell.
   - The robot receives a reward of (+100) if it transitions into the goal state. If the robot stays in the goal state, it will continue to collect a reward of (+100). In case the final state during a transition leads to a collision with a wall in the grid then the robot receives a reward of (-1). In all other cases, a reward of (0) is collected.
   - The robot intends to move to a goal located at grid cell $(48, 12)$ as shown in *red* colour.



   (a) Solve for a policy using value iteration for the setting described above. Use a discount factor of $\gamma = 0.1$ and a threshold of $\theta = 0.1$ as the *max-norm* distance in the successive value functions to determine convergence. You may simulate 100 iterations. To show the result of value iteration, generate an image, where each grid cell is a pixel in the image at the final iteration. Scale the values of the value function obtained to a gray scale value between $[0, 255]$. Show the action (e.g., as arrows) for each state (grid cell) as prescribed by the final policy.

   (b) Increase the discount factor to $\gamma = 0.99$ and plot the value function at iterations 20, 50 and 100.

   (c) Study how the *max-norm* for the successive value functions decreases over successive iterations for the discount factors $\gamma = 0.99$ and $\gamma = 0.01$. Typically, the policy is extracted once value iteration has converged. For experimentation, extract the policy after each value function iteration and study when the policy stops changing (when the policy can be considered as converged).

2. **Model-free RL.** We consider a mobile robot in the grid world domain as described in the previous question. The problem setup remains the same as the previous question with the following modifications.

   The robot receives a (+100) reward for *arriving* in the goal state. Note that once the robot has arrived at the goal state (it will remain in the goal state) and collect a reward of 0 for being in the goal state. As before, the agent receives a reward of (-1) reward for colliding with a wall, and a reward of 0 otherwise. You may assume a discount factor of $\gamma = 0.99$.

   Assume that the robot does not have access to the transition function and the reward model. The robot receives an instantaneous reward upon making a transition and the successor state obtained after taking a transition comes from the environment simulator. Note that you have access to the transition model and the reward model specified above and can simulate it, but this is not available to the robot.

   (a) Implement Q-learning to help the robot explore and learn a good policy via model-free RL. You may initialize the agent with a learning rate of $\alpha = 0.25$ and an exploration rate of $\epsilon = 0.05$ for the ϵ-greedy exploration during Q-learning. Simulate at least 4000 episodes with the robot interacting in the environment. Assume that each episode starts from a randomly selected feasible state (not the goal state) and terminates if the agent reaches the goal state or the episode reaches a maximum length of 1000 steps.

   (b) Please visualize the resulting *state-value* function for the grid world as an image. As before, generate an image by scaling the value function on a gray scale value between $[0, 255]$. Note that the Q-learner operates on the *state-action* value functions. Hence, convert the state-action value function to the state value function before plotting. Overlay the policy determined by the learner showing the prescribed action for each grid cell.

   (c) How does changing the exploration parameter affect the behavior of the Q-learner? Vary the exploration parameter as $\epsilon = 0.005$, $\epsilon = 0.05$ and $\epsilon = 0.5$ and examine the resulting value function and the estimated policy.

   (d) Plot the reward accumulated per episode against the number of training episodes. Study the plots for $\epsilon = 0.05$ and $\epsilon = 0.5$.

3. **Model-based RL.** We consider a mobile robot in the grid world domain as described in the previous question. The problem setup remains the same as the previous question with the following modifications.

   The robot receives a (+100) reward for *arriving* in the goal state. However, once the agent is in the goal state, the episode ends. Assume a maximum simulation length of an episode as 1000 steps. As before, the agent receives a reward of (-1) reward for colliding with a wall, and a reward of 0 otherwise.

   As before, assume that the robot does not have access to the transition function and the reward model. The robot receives an instantaneous reward upon making a transition and the successor state obtained after taking a transition comes from the environment simulator. Note that you have access to the transition model and the reward model specified above and can simulate it, but this is not available to the robot.

   (a) Please implement the Balanced Wandering policy. Simulate 100 episodes with this policy where the episode can start from a randomly selected feasible state and extend till termination or reaching the maximum simulation length. Plot the mean number of visits per state on the y-axis and the iteration number of episode on the x-axis. Plot the standard deviation around that mean along the y-axis. For instance, if after 20 episodes of Balanced Wandering, the agent has visited on average 10 ± 2 states, then there would be the data point in your plot at $t = 20$.

   (b) Next, use Certainty-Equivalence to infer a model from data gathered during Balanced Wandering.

   (c) Compute a policy from the model inferred from Certainty-Equivalence in the previous part using value iteration (use $\theta = 0.1$ and $\gamma = 0.99$). Visualize the value function and overlay the policy as done before. Compare this result with the value iteration result obtained from a fully known and correct model.

4. **Paper Presentation.** This part involves reading, critiquing and presenting a technical paper. Please select a paper from a prescribed list drawn from prominent conferences in the field of Robotics & AI or Embodied AI. The paper list appears at this link. Please select any one paper from the list by writing your name(s) alongside. Selection is on a first come first serve basis. Papers can be accessed here.

   Please study the selected paper. The paper would build on the basic themes discussed in class and would require some exploration to understand the work. Prepare a 15-minute presentation (max. 20 slides) organized as follows:

   (a) Problem Statement. What problem does the paper attempt to solve? Formally state the technical gap.

   (b) Baseline. What is the baseline? You may briefly state 1-2 works only.

   (c) Technical approach. Describe the key technical details for the proposed solution.

   (d) Results. What are the central results? What is the agent now able to do that it could not do before this paper?

   (e) Others. How can the work be improved or applied to a new problem?

   Use a standard Powerpoint format (16:9 format). A presentation schedule will be notified in due course. Evaluation will be on the clarity of technical ideas and not on the aesthetics/speaking aspects of your presentation. Try to present the key technical ideas through your own understanding of the material. Try to avoid a direct reproduction of descriptive text from the paper or from the author's slides. You should of course use the mathematical material, results and the overall structure from the paper.