Assignment: Solving a 3D Gridworld Problem using MDP & RL

Weightage: 10% (20 Marks)

Due Date: 03rd October 2025 (2359 hrs.)

Group Information: Max. of 3 students per group

1) Learning Outcomes

By the end, you should be able to:
1. Formulate a real problem as an MDP (S, A, P, R, γ).
2. Implement tabular Q-learning with ε-greedy exploration.
3. Reason about convergence criteria, exploration vs. exploitation, learning curves, and learned-policy evaluation.
4. Visualize and interpret optimal $\max_a Q(s,a)$ value functions and policies in 3D.

2) Problem Setup (3D Gridworld)

- Grid: H × W × D (height, width, depth). Default: H=W=D=6.
- States (S): each free cell (x,y,z). Obstacles are removed from the state space.
- Actions (A): {+x (East), -x (West), +y (North), -y (South), +z (Up), -z (Down)}.
- Transitions (P): With probability p go in intended direction, with probability 1-p slip uniformly to any of the 4 directions perpendicular to the intended axis. Stay in place if blocked.
- Rewards (R): step cost c_{step}=-1. Goal: +50 (absorbing). Pit: -50 (absorbing).
- Discount: γ=0.95.
- Boundaries/Obstacles: Define a list of obstacle coordinates.
- Terminals: At least 1 goal and 1 pit. Example (for 6×6×6):

  - Goal at (5,5,5), Pit at (2,2,2).
  - ~10–15% cells as obstacles, randomly placed (ensure the start and goal remain reachable).

Bellman optimality (for Q*):

$$Q^*(s,a) = \mathbb{E}\left[r + \gamma \max_{a'} Q^*(s',a')\right]$$

(Defines the fixed point you're learning toward.)

Q-learning sample update (what students implement):

$$Q(s,a) \leftarrow Q(s,a) + \alpha\Big(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\Big)$$

Greedy policy from learned Q (evaluation time):

$$\pi(s) = \arg\max_{a} Q(s,a)$$

3) Tasks

Part A — MDP & RL Formulation (10%)

Write a clear specification of (S, A, P, R, γ) for the given 3D world. Explain how slip is encoded in P.

Part B — Environment Implementation (20%)

Implement a class Gridworld3D with states, terminals, obstacles, transition probabilities, and step costs.

Part C — Q-learning Implementation (35%)

Implement Q-learning for the 3D Gridworld. Use ε-greedy exploration, update the Q-table, and train over multiple episodes. Report learning curves (reward per episode) and show convergence behaviour.

Part D — Policy Evaluation & Comparison (15%)

After training Q-learning, extract the greedy policy π(s) = argmax_a Q(s,a). Evaluate the learned policy over 100 test episodes (no exploration). Compare average returns with a baseline random policy.

Part E — Experiments & Analysis (15%)

Run experiments varying γ, slip probability, and step cost. Report convergence and interpret policy differences.

Part F — Visualization (10%)

Visualize learned value function approximations (max_a Q(s,a)) as per-slice heatmaps for at least three z-levels. Show the learned greedy policy arrows.

4) Deliverables

Submit a single zip folder per group on Nalanda containing the following:

1. Jupyter Notebook/Script with implementation and experiments.
2. PDF report (≤ 4 pages) with spec, methods, results, analysis.
3. README with run instructions and environment details.

Reproducibility: fix a random seed for obstacle placement.
