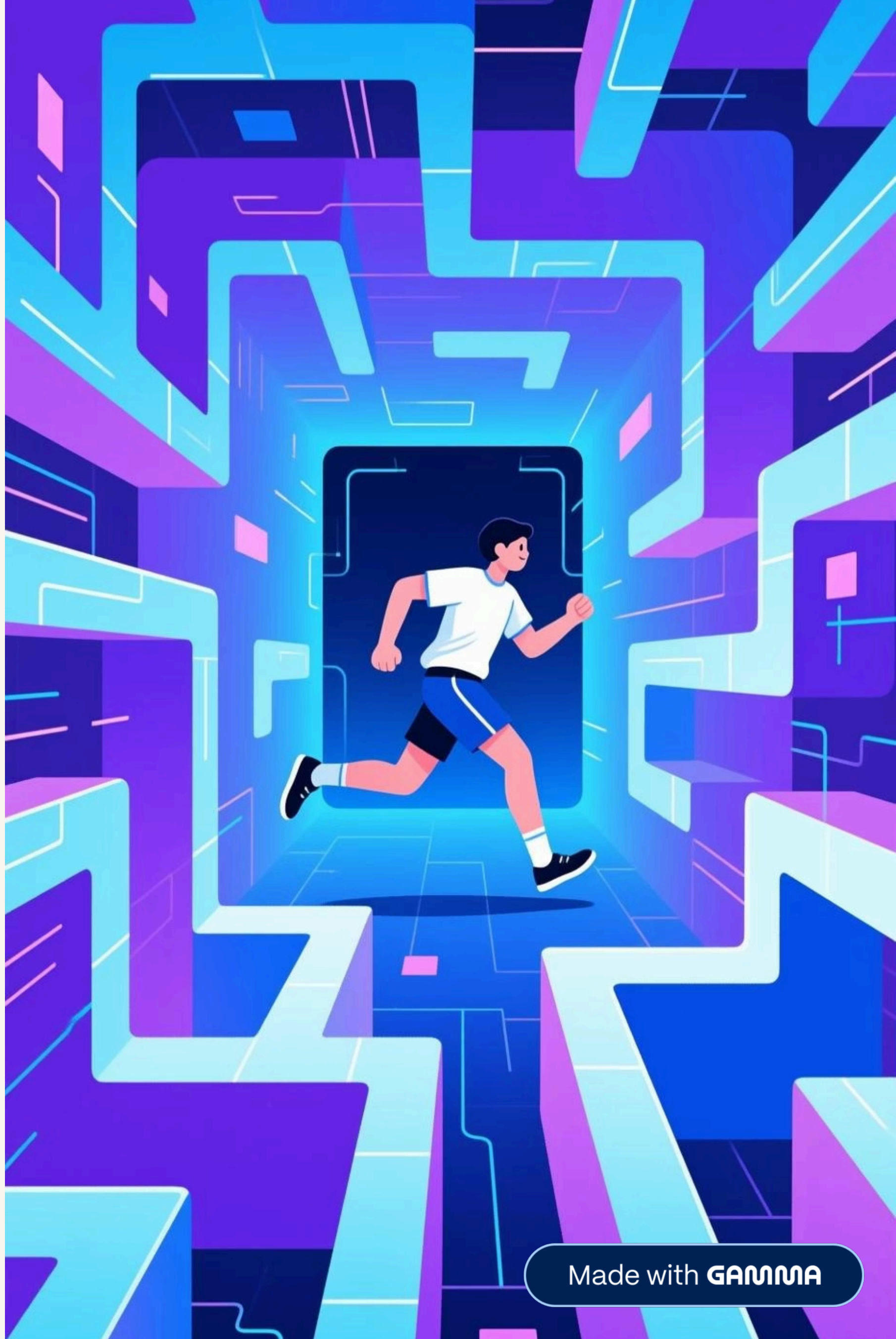
Reinforcement Learning Maze Runner:

When RL Meets Real Human Action

Name : Sriman Soma (S20230030409)



RECAP (Overview of the Project)

Environment

- 10×10 maze
- Start → Goal navigation
- Walls + Traps (Potholes)
- 100 discrete states, 4 actions

Rewards

- **+100** Goal
- **–50** Pothole (**Added this to make differ from simple grid world)**
- **–10** Wall / invalid move
- **–0.1** Normal step cost

Algorithms Implemented so far:

- **Value Iteration — as Assignment 2**
- **Monte Carlo Control — as Assignment 3**
- **SARSA — as Assignment 3**
- **Q-Learning (Final chosen) — as Assignment 3**
- **Q(λ) — as Assignment 3**
- **Linear Function Approximation (explored additionally)**

Made with GAMMA

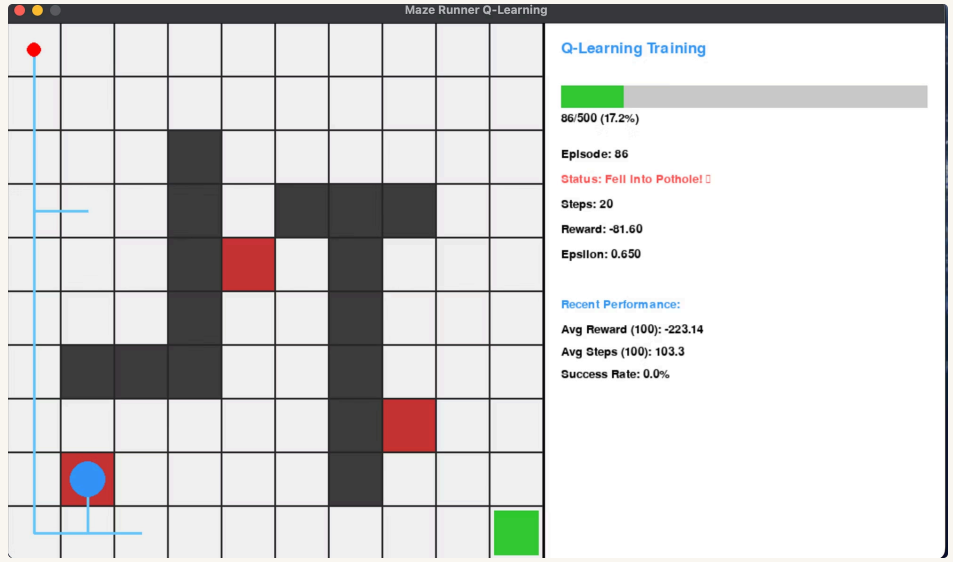
Reinforcement Learning
Learning by Experience

From Zero to Intelligence

- Q-Learning agent starts with **zero knowledge (Q-table = 0)**.
- Learns optimal policy **purely from experience — <u>no maze map.</u>**
- **<u>Bellman update:</u>**
  $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max Q(s',a') - Q(s,a)]$
  ($\alpha = 0.1$, $\gamma = 0.99$ → balances immediate vs future rewards)
- **<u>Exploration vs Exploitation</u>**
- ε-greedy policy with decay: $\varepsilon = 1.0 \rightarrow 0.01$ (× 0.995 per episode)
- **Early:** full exploration (random moves)
- **Mid:** mix of explore + exploit
- **Late:** pure exploitation (optimal path)
- **<u>Continuous Improvement</u>**
- **Ep 1–100:** Random, high variance (~200 + steps)
- **Ep 100–300:** Q-values stabilize, 60–70 % success
- **Ep 300–500:** Optimal policy, 18–20 steps, > 95 % success

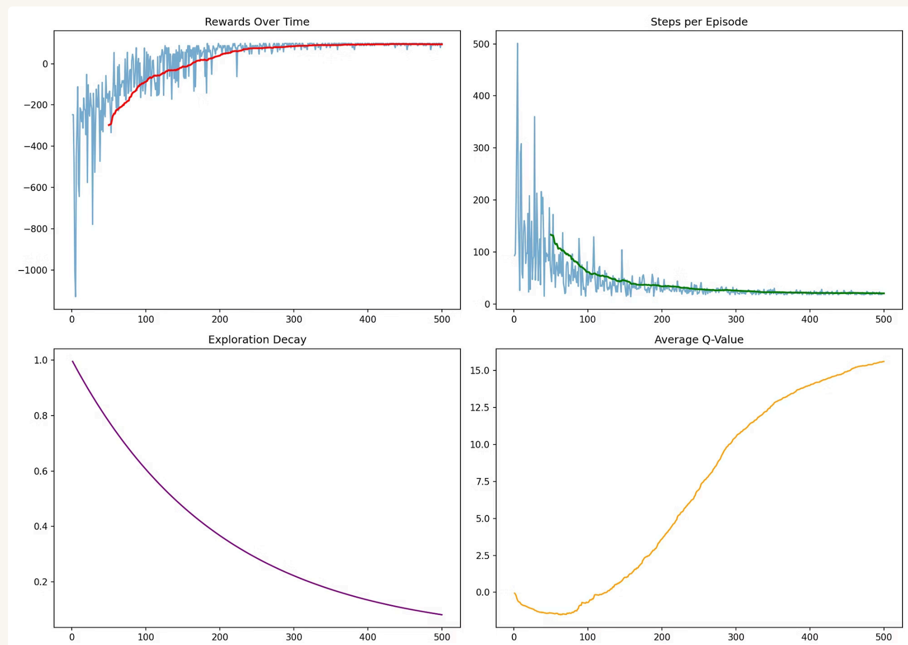
Unique features I implemented in this project:

- Pothole termination logic
- Strong negative reward −50 for traps
- Immediate episode termination on entering trap
- Agent restarts fresh next episode
- Red trap visualization in GUI
- Live status indicator (“Fell into Pothole!”)



Why this improves learning

- Avoids wasting steps inside dead zones
- Encourages safe path discovery
- Speeds up Q-value convergence
- Creates clearer patterns in training video



Why Q-Learning?

- Type: **Model-Free | Off-Policy | Value-Based**

Key Strengths

- **Model-Free:** Learns directly from experience — no need for transition model $P(s\prime \mid s, a)$$P(s' \mid s,a)$
  - **Off-Policy:** Learns optimal (greedy) policy while exploring (ε-greedy)
- **Fast TD Updates:** Updates after every step (bootstrapping) → faster & low-variance learning

Perfect for the Maze Environment

Simple Tabular Setup: 10×10 grid × 4 actions → only 400 Q-values

- **Sample Efficient (TD(0)):**
  One-step updates reuse learned values → ideal for short episodes (~500 steps)
- **Theoretical Convergence:**
  Finite state-action space
  ε-greedy exploration

Why Not Other Algorithms?

🚫 **Dynamic Programming**

- Needs full P(s'|s,a) and R(s,a) model — **not available** in our maze
- Suited for known environments, not learning tasks

🚫 **Monte Carlo**

- Must complete entire episode before update
- High variance, inefficient for long (500-step) episodes

🚫 **SARSA**

- On-policy → learns from behavior policy
- Slower convergence, overly cautious in exploration

🚫 Deep Q-Networks (DQN)

- Adds unnecessary neural-network complexity
- Risk of instability
- Tabular Q-Learning already gives exact solution

🚫 Function Approximation Makes Sense :

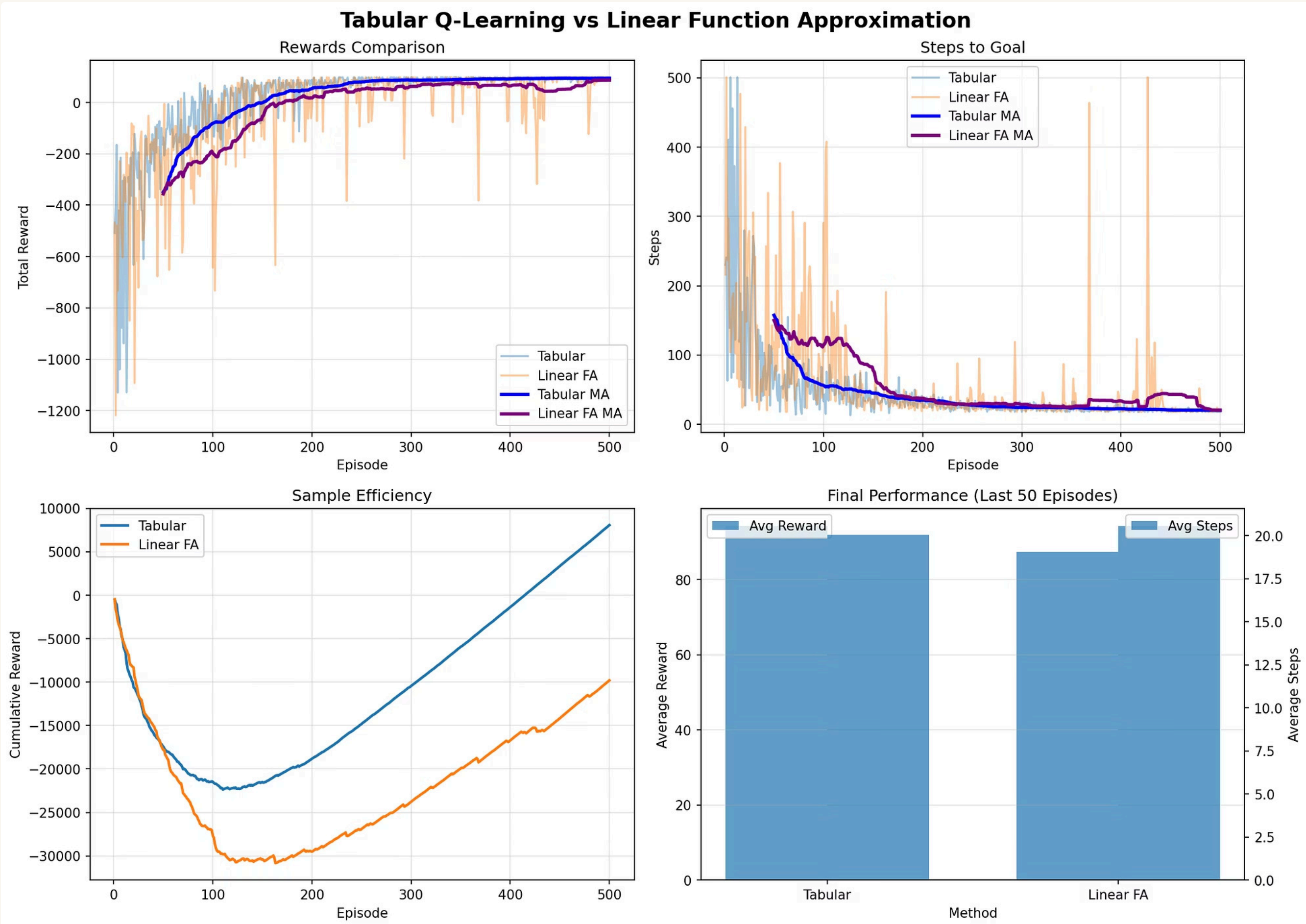
- Large/continuous state spaces: e.g., 1000×1000 maze, pixel inputs, robot joint angles
- Generalization needed: Learning from similar states
- Memory constraints: Can't store millions of Q-values

🚫 Linear Function Approximation Makes Sense :

Small & Discrete State Space → Tabular Is Exact

- Only 100 states, 4 actions → 400 Q-values total
- Tabular Q-learning gives exact values
- Linear FA introduces approximation error

Why the Linear Function approximation is not good in this project:



- Linear FA gave very unstable results and the rewards went extremely low.
- The number of steps kept jumping a lot, meaning it did not learn properly.
- It learned very slowly and needed many more episodes to improve.
- Even after full training, Linear FA performed worse than normal tabular Q-learning.

<u>TRAINING VIDEO LINK for Q-learning:</u>

<u>https://drive.google.com/file/d/1D0ycAMmUd5e0bK6NZCEm2E-uVsWJOx4I/view?usp=sharing</u>₹