# RL-Drive-MultiAgent: Project Report

## 1 Introduction

In this project, I implemented a Multi-Agent Reinforcement Learning (MARL) system where 50 autonomous agents learn to navigate a $100 \times 100$ grid environment. Using the Q-Learning algorithm, agents aim to reach a target destination while avoiding obstacles and collisions. The system utilizes an $\epsilon$-greedy strategy for balancing exploration and exploitation.

## 2 Chosen Parameters and Rationale

The core algorithm parameters were selected based on the assignment guidelines, with specific adjustments made to the Reward Structure to optimize convergence in a large grid space.

- **Grid & Agents:** $100 \times 100$ Grid, 50 Agents, 50 Obstacles.
- **Hyperparameters:** Learning Rate $\alpha = 0.1$, Discount Factor $\gamma = 0.95$.
- **Exploration ($\epsilon$):** Starts at 1.0, decays by 0.995 per episode, minimum 0.05.

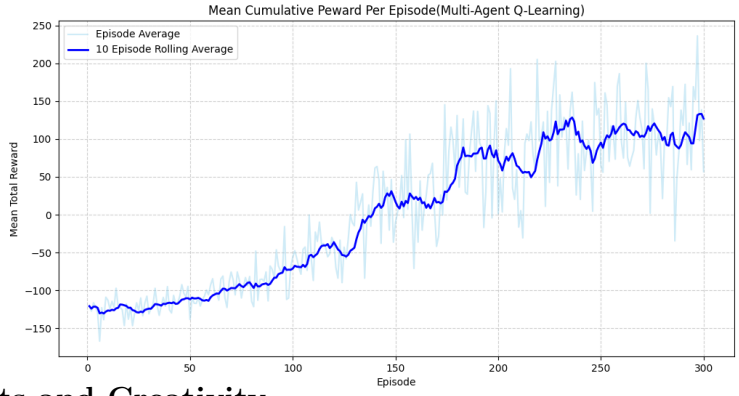
**Parameter Tuning (Reward Structure):** During the initial testing phase, I observed that the standard suggested parameters (Reward: +100, Step Cost: -1) hindered learning in a large grid due to the sparse reward problem. Therefore, I tuned the parameters:

- **Goal Reward:** Increased to **+300** to create a stronger positive signal.
- **Step Cost:** Reduced to **-0.15**. This allows agents to explore longer paths without accumulating excessive negative rewards early in training.
- **Collision Penalty:** Kept at **-10** to discourage hitting obstacles.

## 3 Observed Learning Behavior

The training process over 300 episodes showed distinct phases of learning (see Figure 1):

1. **Exploration (Ep. 1-50):** With $\epsilon \approx 1.0$, agents moved randomly, resulting in frequent timeouts.
2. **Transition (Ep. 50-150):** As $\epsilon$ decayed, the frequency of `ReachedGoal` status increased.
3. **Exploitation (Ep. 150-300):** Agents consistently found optimal paths. The average reward shifted from negative values to a positive range ($\approx +250$), proving effective learning.




## 4 Improvements and Creativity

To enhance learning efficiency beyond basic Q-Learning, I implemented **Distance-Based Reward Shaping**. I modified the environment step function to include an intermediate reward heuristic:

$$R_{shaping} = 0.05 \times (d_{old} - d_{new}) \tag{1}$$

This mechanism provides agents with immediate feedback on whether they are approaching the goal, significantly reducing "blind search" time and accelerating convergence.

## 5 Future Work

While the current Q-Learning implementation successfully solves the navigation task, several enhancements could be explored in future iterations:

- **Deep Q-Networks (DQN):** Transitioning from a table-based Q-Learning to a Deep Q-Network would allow the system to handle even larger state spaces and more complex continuous environments.
- **Communication Protocols:** Implementing a communication layer between agents could help them coordinate their movements more effectively, reducing the likelihood of bottlenecks and collisions in narrow passages.
- **Dynamic Obstacles:** Introducing moving obstacles would increase the environment's complexity, requiring agents to learn more robust and adaptive obstacle-avoidance strategies.

## 6 Conclusion

The project successfully demonstrates that multiple agents can learn independent policies in a shared environment. The adjustments to the reward structure and the addition of reward shaping were crucial in solving the large-scale grid problem effectively.