Vyorius ML Intern Task - Short Report

Project Overview

This project involved building an autonomous agent in a 2D environment using Machine Learning-specifically Q-learning, a form of Reinforcement Learning. The goal was for the agent to learn to reach a target point while avoiding obstacles, without relying on hardcoded rules.

Logic & Model

Environment

- Grid-based 2D world
- Obstacles represented as fixed rectangles
- Agent and goal placed randomly at valid positions

Agent

- Implemented using Q-Learning (from agent.py)
- State: Current grid location (x, y)
- Actions: Move Up, Down, Left, or Right
- Reward Structure:
  + +100 for reaching goal
  + -10 for hitting obstacle
  + -1 for every step (to encourage efficiency)

Training

- 2000+ episodes
- epsilon-greedy action selection (exploration vs. exploitation)
- Epsilon decayed over time to encourage learning
- Q-table updated using Bellman Equation

Performance

The agent learned to navigate successfully after several hundred episodes. Sample outputs during training:

Episode 300 completed in 30 steps

Episode 600 completed in 4 steps

Episode 750 completed in 1 steps

Episode 1200 completed in 300 steps

Episode 1700 completed in 1 steps

Episode 1950 completed in 51 steps

Challenges Faced

- Balancing exploration (epsilon-greedy) and exploitation
- Handling grid edges and ensuring valid moves
- Tuning hyperparameters like learning rate and reward structure
- Preventing the agent from getting stuck in loops

Future Improvements

- Use Deep Q-Network (DQN) with a neural net to handle continuous states
- Add dynamic obstacles that move over time
- Enable multi-agent training
- Use simulated LIDAR input for perception
- Add a logging dashboard (e.g., with Matplotlib or Streamlit)
- Improve the reward strategy with intermediate goals

Tools Used

- Python
- Pygame (for 2D visualization)
- NumPy (for matrix operations)

Outcome

The agent successfully learns to navigate a basic 2D environment using Q-learning. This simulation demonstrates key ideas in autonomous navigation, perception, and ML-based decision-making.