

Maze Game with Q-Learning

Ali Akbari

July 2025

Introduction

- Objective: Implement a Maze game using Q-Learning
- Environment: 10x10 grid with fixed walls, agent, and goal
- Tools: Python, Pygame, Pyodide (for browser execution)
- Learning: Agent learns to navigate from start to goal while avoiding walls

Environment Structure

- Grid: 400x400 pixels, divided into 10x10 cells
- Walls: Fixed (represented by 1 in the maze matrix)
- Agent: Starts at (0,0), displayed as a green square
- Goal: Located at (9,9), displayed as a red square

Q-Learning: Core Concepts

- **State:** Agent's current coordinates (x, y)
- **Actions:** Up, down, left, right
- **Rewards:**
 - Reaching the goal: +100
 - Hitting a wall: -10
 - Each valid move: -1
- **Parameters:**
 - Learning rate (α): 0.1
 - Discount factor (γ): 0.9
 - Exploration rate (ϵ): 0.1

Q-Learning Algorithm

- Q-value update formula:

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

- Action selection:
 - With probability ϵ : Random action
 - Otherwise: Action with highest Q-value
- Q-table: Stores Q-values for each state-action pair

Implementation

- **Agent Class:** Manages agent movement and valid move checks
- **Goal Class:** Defines the goal position
- **QLearningAgent Class:** Handles states, action selection, and Q-table updates
- **Graphics:** Displays walls (blue), agent (green), goal (red), and info (score, episode, steps)
- **Game Loop:** Standard Pygame loop with FPS=50 for learning

Results and Observations

- Early episodes: Agent moves randomly
- Over episodes, the agent learns to:
 - Find shorter paths to the goal
 - Avoid walls
- Score, episode number, and step count displayed in top-left corner
- Game resets after reaching the goal

Possible Improvements

- Gradually decrease ϵ for better exploitation
- Add dynamic walls for increased challenge
- Use Deep Q-Learning (DQN) for larger mazes
- Store and display the optimal path after learning

Conclusion

- Successful implementation of Maze game with Q-Learning
- Effective learning for navigating to the goal and avoiding walls
- Code compatible with Pyodide using standard Pygame loop
- Extensible project for advanced reinforcement learning experiments