

Reinforcement Learning Techniques for Snake Game

A Comparative Study of Q-Learning, SARSA, Monte Carlo, and Temporal-Difference Learning

Name: Nguyen Tien Thanh

Date: October 11, 2024

Reinforcement Learning

- A branch of machine learning focused on how agents learn to make decisions to maximize cumulative rewards.
- Key Features:
 - Learning through trial and error
 - Interaction with dynamic environments
- Applications: Gaming, robotics, autonomous vehicles.
-



Snake Game Overview:

- **Objective: Control a snake to eat food pellets while avoiding collisions.**
- **Gameplay Mechanics:**
 - **Snake grows longer with each food item consumed.**
 - **Game ends on self-collision or wall collision.**

Project Goals

- Develop an RL agent for the Snake game.
- Compare algorithms:
 - Q-Learning
 - SARSA
 - Monte Carlo Methods
 - Temporal-Difference Learning
- Analyze: Performance, efficiency, and convergence rates.

Theoretical Background

Overview of Algorithms

Core Components of RL:

- Agent: Learner or decision-maker.
- Environment: System the agent interacts with.
- State: Current situation of the agent.
- Action: Choices available to the agent.
- Reward: Feedback signal post-action.
- Policy: Strategy mapping states to actions.

Q-Learning:

- Off-policy, learns action values using the Bellman equation.

SARSA:

- On-policy, updates based on actual actions taken.

Monte Carlo Methods:

- Learns from complete episodes.

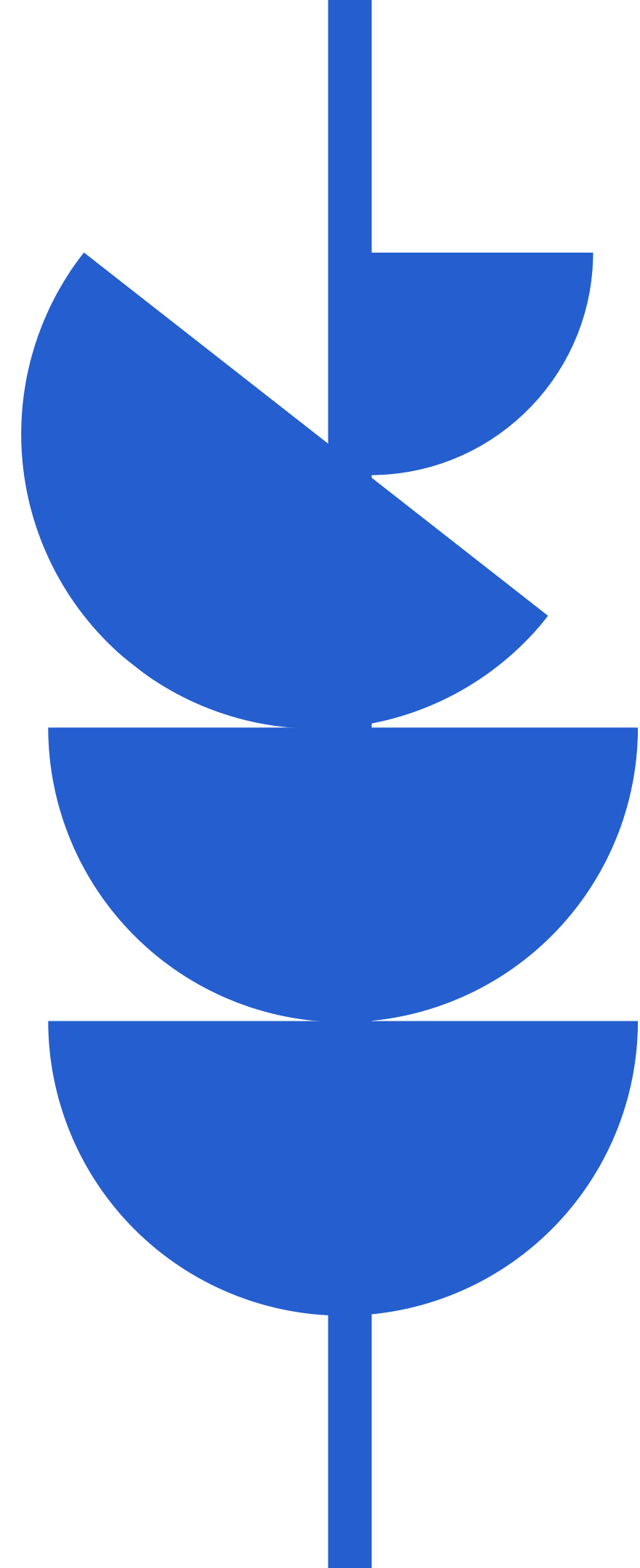
Temporal-Difference Learning:

- Combines Monte Carlo and dynamic programming for updates.

Methodology

- Problem Definition: Develop an RL agent for Snake.
- State Space: Configurations of the game board (snake position, food position).
- Action Space: Moves (up, down, left, right).
- Reward Structure:
 - Positive/negative rewards based on actions.

Results Summary



Mean Rewards after 10,000 Episodes:



Q-Learning:
118.20

TD Learning:
116.33

SARSA:
115.30

Monte Carlo:
5.60

Mean Food Eaten:



Q-Learning:
5.67

TD Learning:
5.60

SARSA: 5.57

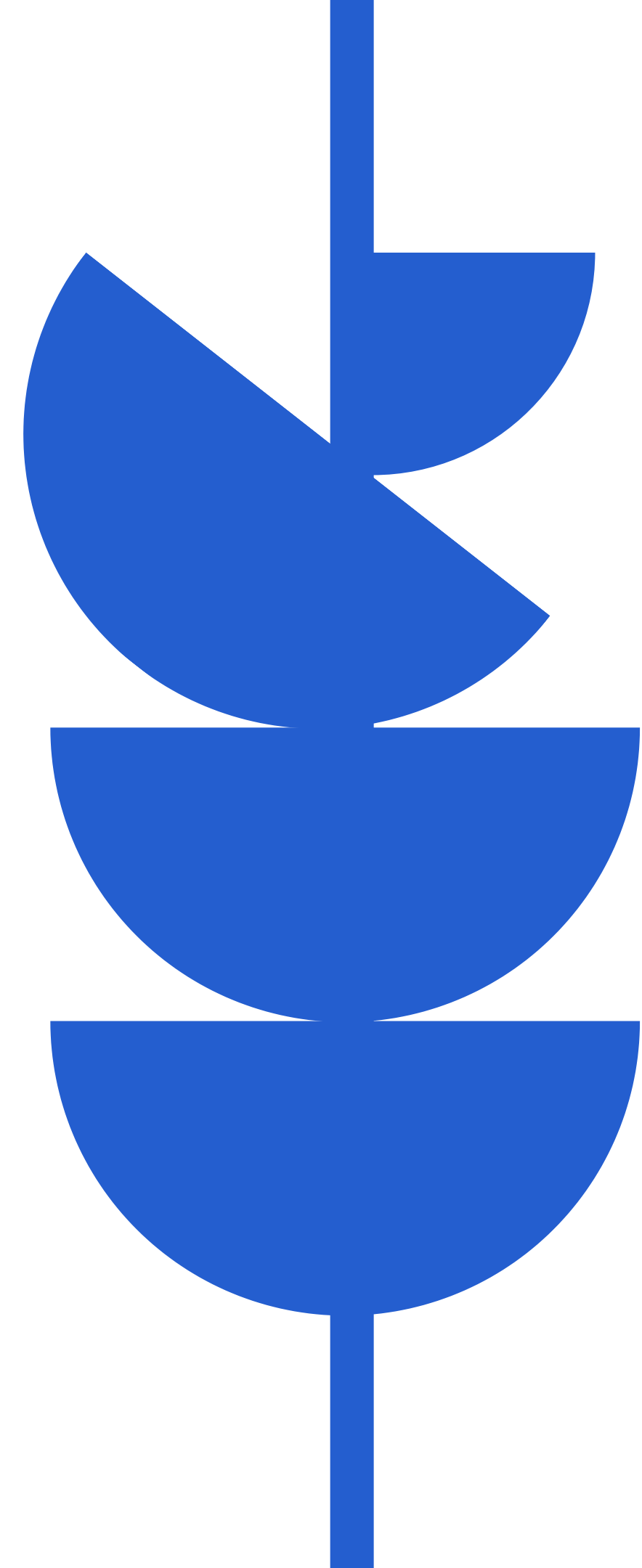
Monte Carlo:
2.11

Q-Learning: Highest performance, effective learning and exploration.

TD Learning: Competitive but slightly less effective.

SARSA: Similar to TD but slower learning due to its on-policy nature.

Monte Carlo: Less effective, reliant on complete episodes.





Key Takeaways:

Q-Learning proved to be the most effective for the Snake game.

Other methods offered insights into RL dynamics.