Capture The Flag Pacman with Reinforcement Learning

Kevin Lee[1], Yu-Hsin Weng[2], Varun Kumar[2], Siddarth Chalasani[2]
[1]Department of Mechanical and Aerospace Engineering, University of California, Los Angeles
[1]Department of Computer Science, University of California, Los Angeles

Abstract

This project develops a Capture The Flag (CTF) variation of Pacman using both greedy algorithms and machine learning (ML) approaches for agent behavior. We employ reinforcement learning (RL) algorithms like Q-Learning and Proximal Policy Optimization (PPO) to enable adaptive strategies. Integrating machine learning, game development, and animation, this interdisciplinary approach is relevant to the study of artificial life in computer graphics and vision.

Introduction

The main motivation for this project is to simulate real-world cooperation through a CTF variation of Pacman. By employing RL algorithms, our agents learn and adapt their strategies over time, providing valuable insights into multi-agent systems exhibiting life-like behaviors. This work can be applied to robotics, autonomous systems, and game development, utilizing Unity for visualization and integrating concepts from machine learning and artificial life. Our approach builds on deep reinforcement learning and multi-agent coordination, demonstrating the potential for advanced AI techniques to create intelligent and adaptive behaviors in complex environments.

Rules

Each agent starts as a ghost and becomes a Pacman after moving to the other side of the board. As Pacmen, they must capture as many pellets as possible while avoiding the opposing team's ghosts. Captured Pacmen revert to ghosts and must cross the board again to resume pellet collection. The team with the most points at the end of the game wins. Points are earned by successfully capturing and returning pellets to the home side. A game lasts for 1200 total moves or until all pellets are eaten, with strategic movement and decision-making being key to victory.

Methods

Q-Learning Algorithm

- Q-Learning is a model-free RL algorithm that aims to learn the quality of actions. It updates Q-values iteratively based on the agent's experiences.
- Utilizes the Bellman equation:
  $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$

Proximal Policy Optimization (PPO) Algorithm

- PPO optimizes the policy network by improving the likelihood ratio of new and old policies within a clipped range.
- It uses an advantage calculation to see how good our chosen action was compared to the average action taken in that state.

Approximate Q-Learning Algorithm

- Uses function approximation to generalize Q-values across similar states.
- Employs a linear function approximator where the Q-value function is represented as a linear combination of features and weights.

Agent Design and Implementation

Offensive Agent The Offensive Agent captures pellets on the enemy's side and delivers them back home using Approximate Q-Learning. Key features include:

- Bias
- Distance to Nearest Pellet
- Eats Food
- Distance to Nearest Capsule
- Eats Capsule
- Distance to Home
- Number of Ghosts Within Two Steps

The agent balances exploration and exploitation through an ϵ-greedy policy, prioritizing pellet collection and safe return while avoiding ghosts.

Defensive Agent The Defensive Agent prevents the opposing team from capturing pellets using Approximate Q-Learning. Key features include:

- Number of Invaders
- Distance to Invaders
- On Defense
- Stop and Reverse Penalties

The agent patrols its side, intercepting invaders and improving defensive tactics over time.

Cooperation Between Agents Offensive and defensive agents work together to achieve team objectives. Initially trained separately, they were later trained together for 100 episodes to ensure cooperation, balancing offense and defense through implicit communication and shared game state.

Results: Tables

To test the efficacy of our Approximate Q-Learning agents, we conducted a series of games where our team competed against a team of greedy agents. Our team won 84 out of the 100 games, tied 14 times, and lost only 2 games, indicating that our learning-based agents outperformed the greedy agents.

Table 1: Performance Metrics of Approximate Q-Learning Agents

Metric	Value
Blue Win Rate	84%
Tie Rate	14%
Red Win Rate	2%
Average Blue Team Win Margin	6.18 Pellets

Results: Game View

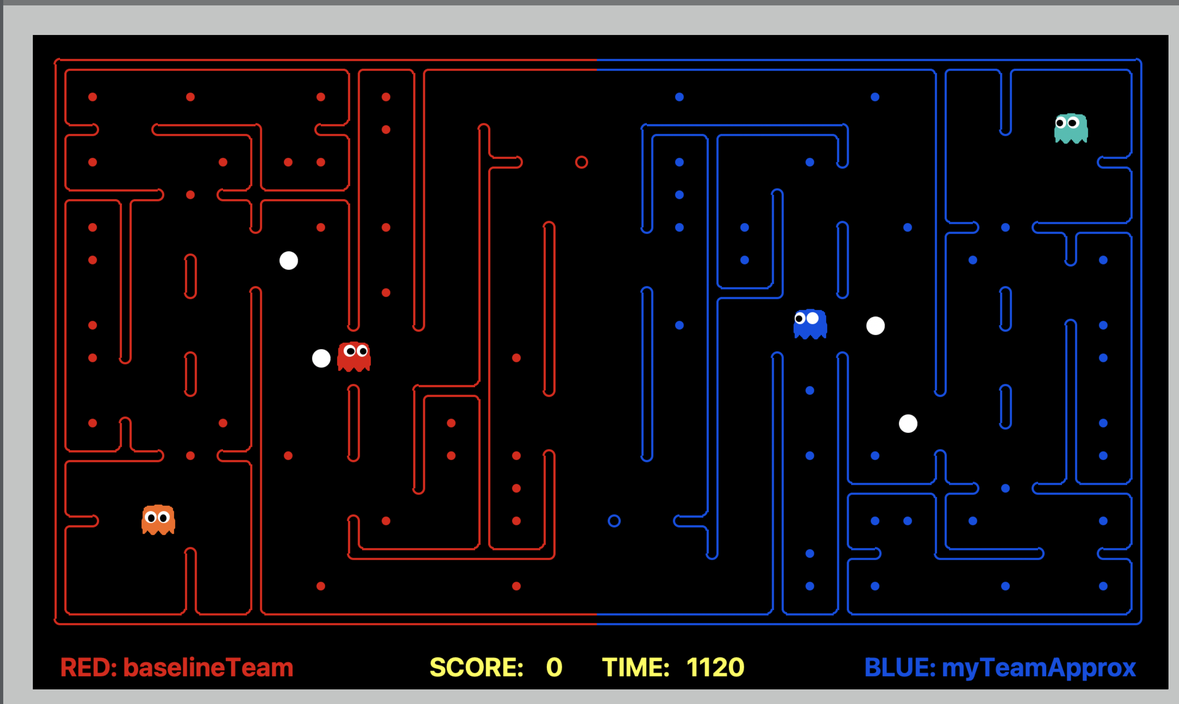

Figure 1: The Capture The Flag Pacman game layout. Each team starts with two agents in the state of ghosts. This view showcases the dynamic environment and the strategic elements of the game.

Conclusions

The Capture The Flag variation of Pacman project demonstrates the significant potential of RL algorithms in developing adaptive and intelligent game agents. By integrating Q-Learning and PPO with game development frameworks like Unity, we have created a dynamic platform that effectively showcases complex agent behaviors and interactions. The project advances our understanding of RL and its applications, setting the stage for further explorations into optimizing RL algorithms, developing more complex multi-agent interactions, and applying these techniques to a wider array of disciplines.

References

[1] Yuanhao Li, Zhiqiang Zhang, Guoliang Jiang, Lijun Li, and Qiuyi Yuan.
A survey on deep reinforcement learning.
arXiv preprint arXiv:1906.00979, 2019.

[2] Yuxi Li.
Deep reinforcement learning: An overview.
arXiv preprint arXiv:1701.07274, 2017.

[3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller.
Playing atari with deep reinforcement learning.
arXiv preprint arXiv:1312.5602, 2013.

Acknowledgements

We would like to express our gratitude to Professor Terzopoulos for his guidance and insightful feedback throughout this project. His expertise in Computer Graphics and Vision have been instrumental in shaping our work. We are grateful for the opportunity to learn under his CS275 course.

Contact Information

- Website: https:///magnaprog.github.io
- Emails:
  - yuhsin1614@ucla.edu
  - kevinlee69720@g.ucla.edu
  - vvkumar1@g.ucla.edu
  - darthsid2000@g.ucla.edu