Reinforcement Learning for Game Playing: Multi-Agent Pac-Man Environment

Riley McKinney, Nathan Dow, Zain Karim
Department of Computer Science
University of Texas at Dallas
Richardson, TX, USA

***Abstract*—This project investigates reinforcement learning for multi-agent game environments through a custom Pac-Man simulation. The goal is to develop both a hider (runner) and a seeker agent, each trained independently with distinct reward functions under the same Q-Learning framework. The project will begin with a 10×10 grid containing randomly placed pellets and a single seeker, later progressing toward maze-based environments with multiple seekers and variable movement limits. Performance will be evaluated by tracking reward convergence, learning stability, and emergent pursuit-evasion behaviors.**

***Index Terms*—Reinforcement Learning, Q-Learning, Multi-Agent Systems, Game AI, Pac-Man**

I. Project Topic

The proposed topic is **Reinforcement Learning for Game Playing**. A custom Pac-Man environment will serve as the foundation for studying multi-agent reinforcement learning, where agents pursue conflicting goals: the runner aims to survive and collect pellets, while the seeker aims to catch the runner as efficiently as possible.

II. Team Members

This project will be conducted by **Riley McKinney, Nathan Dow, and Zain Karim**.

III. Technique / Algorithm

The planned technique is **Q-Learning**, a model-free reinforcement learning algorithm based on temporal difference updates:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)] \quad (1)$$

Both the runner and seeker will use this algorithm but will be trained on separate reward functions.

- **Runner Reward:** Encourages pellet collection and avoidance of seekers.
- **Seeker Reward:** Rewards successful capture and penalizes inefficient pursuit.

Each agent will maintain its own Q-table and policy while interacting within the same environment.

IV. Environment Details

The current environment is a 10×10 open grid with five randomly placed pellets and a single seeker. Agents can move up, down, left, or right. Rewards are currently defined as:

- **Runner (Pac-Man):** +1 per pellet, +20 for clearing all pellets, 10 if caught, 0.01 step cost.
- **Seeker (Ghost):** +10 for catching the runner, 0.01 step cost.

Future iterations will introduce maze-like structures or collections of mazes, walls to constrain movement, and multiple seekers coordinating in pursuit. The maximum number of allowed steps per episode (move cap) will be increased to accommodate the added complexity.

V. Coding Language / Tools

- **Language:** Python 3
- **Libraries:** NumPy, Matplotlib, PyGame, and PyTorch (for future DQN extension)
- **Modules:**
  - `environment.py` — Grid world logic, reward computation, and PyGame rendering.
  - `agent.py` — Q-Learning algorithm and action policy.
  - `main.py` — Simulation loop and future training integration.
  - `utils.py` — Reward visualization and logging utilities.

VI. Preliminary Results

Early random-action simulations confirm that the environment updates, scoring logic, and rendering are stable. Pac-Man collects pellets correctly, the seeker detects collisions, and reward signals behave as expected. A baseline test of 100 random episodes showed the runner achieving an average cumulative reward of approximately **-6.3**, validating the negative bias of the current environment before learning. Visualization through PyGame confirms proper step-by-step rendering of agent movements and score updates, providing a reliable platform for introducing learning agents in the next phase.

VII. Expected Results

Upcoming work will involve training both agents independently using Q-Learning and observing emergent pursuit and evasion strategies. Expected outcomes include:

- Converging Q-values for both agents under opposing objectives.
- Improved survival time and efficiency for the runner agent.
- Coordinated multi-seeker pursuit behavior in maze configurations.

ACKNOWLEDGMENT

The authors thank the course instructor for guidance and the University of Texas at Dallas for providing the project framework and computing resources.

REFERENCES

[1] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992.

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.

[3] IEEE, "Conference Templates," 2025. [Online]. Available: https://www.ieee.org/conferences/publishing/templates.html