Report of a Pac-Man AI Agent Using Monte Carlo Tree Search

Menmber: Xiang Luo

1 Background

Pac-Man is a classic maze-chasing game. Player control the Pac-Man moving in the maze to eat points and avoid ghosts. I'm not good at this kind of game, thus would like to create an agent to help me win the game.
Inspired by the chasing story in Genshin Impact's recent main quest, I would like to choose Rerir as the Pac-Man, and Traveler, Aino, Albedo, Flins act as ghosts. I drew these characters in pixels.

2 Solution

Monte Carlo Tree Search (MCTS) is a common method in adversary search and works well in game model that do not need to know the whole spaceof the game. Therefore, I would like to use MCTS to complete my agent.
The project contains a Pac-Man game engine (*index.html*, etc.) and agents in Python (*navi.py*). I completed MCTS agent. Other two agents (A* and DQN) are random-moving agents.
I used generative AI (Gemini) to construct the basis of the game engine. Then I modified it to better align with my agents. For the movements of ghosts, I referred this website: *https : //pacmanonline.org/?lang = manual*. As for the MCTS agent, it was totally written by me. I consulted Tom Pepels *et al*'s work in MCTS Pac-Man agent as a reference of defining different states for agent at the very beginning.

3 Requirements

VS Code or other platforms that can run Python scripts. In the next part, I use VS Code as an example.
Websockets: pip install websockets. (My version: Python: 3.11.14, websockets: 15.0.1.)
Since the game and agent work at different platform, websockets can connect them.

4 Running Process

Note: If anything goes wrong, refresh the browser or use ctl+c to stop and then restart the python.

Try the game yourself (optional):
1) Double-click the HTML file (index.HTML) to open the game in your default browser. You can also use another browser you prefer.
2) Move Rerir (light blue character at the bottom of the maze) in any direction (J: left, L: right, I: up, K: down) to avoid enemies and eat points in the maze.

3) Win the game: eat all points in the maze and be caught by enemies no more than twice. You will see a message pop up: "Rerir Werewolf!" If you are caught by enemies three times, the message will be: " YOU ARE ARRESTED! (Rerir has run out of lives.)"

Use agents:
1) Open the HTML file in a browser.
2) Run navi.py in VS Code terminal. python navi.py.
If succeed, terminal will automatically return: "AI WebSocket server started, listening on ws://localhost:8765...".
3) Back to the browser. There are two ways to test agents:

- Enemies staying in the cage: You can find the skills part on the right side of the screen. Click "Apply" to use the method.

- Enemies chasing: Move Rerir (light blue character at the bottom of the maze) in any direction (J: left, L: right, I: up, K: down) to wake the enemies. Then apply agents by click "Apply" in the skills part on the right side of the screen.

I only finished MCTS, so apply Navigation (MCTS) will use AI agent. A* and DQN will use random movements.
4) Watch the agent moving.

5 Result

The result is not thrilled. I tested the agent manually myself for about one hundred times, haven't won the game with MCTS agent. The final scores were about 1100.
Rerir can move autonomously by applying MCTS agent. During my tests, sometimes Rerir moves left and right rather than going up. Since Rerir's next movement is based on heuristic scores, if the values are the same or has little difference, Rerir will move back and forth in a small area (usually in a short line). It failed to get out of such a loop itself. To address this problem, in the simulation part, I generate a ramdom number between 0-1 first. If the number is less than 0.5, then apply random move. Otherwise use MCTS. This approach makes Rerir performs slightly better.
Once an enemies getting closer, Rerir "feels" the enemy, and may leave the loop by re-value the movements.
Rerir performs better in long path than short path or path with multiple walls. One of my assumption is the evaluation takes time. Rerir will move along one direction if no other direction is given. Suppose Rerir is moving up. When agent selects the next move, for example, turn left to another path, Rerir has already pass it and its left is now walls, which prevent the movement of left.
Another reason may because there are four enemies chasing Rerir, which make the game a little challenging. I still need to modify my code to improve the MCTS agent.

6 Reference

[1] Pepels, Tom, Mark HM Winands, and Marc Lanctot. “Real-time monte carlo tree search in ms pac-man.” *IEEE Transactions on Computational Intelligence and AI in games* 6.3 (2014): 245-257.
[2] PacmanOnline.org. “Pacman Online Game Manual.” *Pacman Online Official Website* (2024): Accessed Nov 12, 2025. https://pacmanonline.org/?lang=manual.