Pac-Man Final Project Conclusion by Adam Stafford and Barnabas Fluck

In this project, we implemented advanced graph traversal algorithms to control the ghosts in Pac-Man, referencing the study Exploring the Possibilities of MADDPG for UAV Swarm Control by Simulating in Pac-Man Environment. While the paper focuses on deep learning, it establishes rule-based algorithms (BFS and A*) as critical benchmarks for navigation and pursuit, which we implemented here.

Blinky (The Chaser):
Blinky has been modified to use the Breadth-First Search (BFS) algorithm to try to reach Pac-Man. Now, Blinky uses BFS to calculate the shortest path to Pac-Man's current location. It allows Blinky to use an optimal navigation across the maze, which is more effective than the original. Blinky now updates his movements dynamically to reach Pac-Man's position. BFS has some limitations, such as high computational cost. It has a higher O than the original Blinky algorithm.

Pinky (The Interceptor):
Pinky's strategy is to target a tile 2 units ahead of Pac-Man. We achieved this by implementing an A Search (A*). A* uses heuristics (Manhattan distance) to prioritize paths that move closer to Pac-Man, making it efficient to intercept Pac-Man. Pinky's strength is that he can be really effective in open spaces to find paths to Pac-Man.

Inky (The hybrid):
Inky uses a mixed strategy that implements Blinky's and Pinky's. He chooses from the strategies based on his distance from Pac-Man. If he is outside of 150px, Inky chooses Pinky's strategy with the A* however, if he is within 150px, he chooses Blinky's strategy with the BFS. It allows Inky to potentially trap Pac-Man.

Clyde (The Feigner):
We left Clyde as is.

The strengths of our projects are efficiency. As I mentioned in the ghosts' description, the new strategies are more effective at catching Pac-Man than the originals. Furthermore, we implemented a Static Boolean Map for the wall grid to make BFS easier and to run the game smoothly.
Some limitations of this project are predictability. As noted in the source paper, rule-based algorithms (like A* and BFS) are deterministic. The new algorithms also require a higher computational cost, meaning that in scenarios such as when Pac-Man cannot be reached, we had to implement loop preventions.