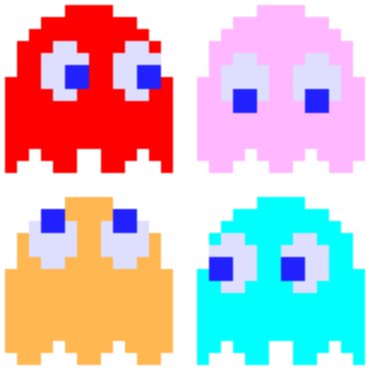
Neural N∅twork
PacMan Report



R.J. Hamilton
Farid Karadsheh
Johnathon Killeen
Ryan Eisenbarth

Project Reflection

1. What strategy(s) did you employ with your swarms?

The strategy that we programmed the AI to use was a mix between offense and defense. Initially, one bot is programmed to play offense and rush to collect food, while the second bot plays defense for the entirety of the match. After gaining the lead, the offensive bot will return to it's side of the map and perform an evaluation. After much revision our AI functions as follows:

- If they are losing in points, the offensive bot will perform the same offensive strategy and collect more food.

- If they are winning in points, both bots resort to a defensive strategy where they are prioritized to patrol the middle border where they will wait for enemies to cross.

- By utilizing the option of switching between offense and defensive based on the score heuristic, our bots have been found to be very successful.

2. What behaviors did you implement?

We designed the AI to evaluate their best course of action based on a few different heuristics.

- The first behavior that we implemented was the offensive bot's ability to calculate risk. We defined risk as *d * c*, where *d* is the distance from the safe point and *c* is the number of food being carried. The resulting behavior was a bot that collected food closest to the safe point.

- The second behavior that we implemented was the bot's ability to detect where the center of the map was. This is critical to our strategy because once our team has taken the lead, the bots need to be able to patrol near the center of the map to better defend their side.

- Another behavior that is critical to our strategy is to stop patrolling once an enemy pacman is detected on our side. Our ghosts will then proceed to chase the pacman and return back to patrol if the enemy pacman is either caught or has returned to their side.

- To enhance the early scoring of our bot, we decided to implement a feature which prioritizes the power capsule. The power capsule secures our lead in the early game which allows us to play on the defensive for the remainder.

3. What worked? What didn't?

Worked:

- Starting off with a balanced scenario where we are using an offensive and defensive agent. This strategy has the ability to change based on the score. If we are down in points, then the strategy will remain the same. If we are up in points, then the bots will both switch to a defensive mode.

- Having our bots patrol the middle while on defense provided mixed results. The issue with patrolling the center of the defending area doesn't work if the enemies stick to the edge of the map. If an enemy AI were to quickly grab a pellet and leave, our bot would have an issue responding quickly.

Didn't Work:

- Both bots performing same task

- Another thing we tried to great failure was to break the game itself. Much of the beginning time was spent trying to beat the game itself. We circulated ideas such as: spawning additional pellets, spawning more ghosts, travelling through walls, and teleporting to select locations. To our disappointment, none of these strategies panned out because of the design of the game engine. Some good did come out of this because the frame of thought that led us to being trolls led us to the idea of mimicking the opponent's AI. After finding that we could not see the enemy at all times on the map, we settled with our current strategy to play a hard defense and try to detect the enemies as soon as possible.

The ability to switch between offense and defense depending on the score seems to work quite well.

Our bots also have a tendency to follow the same path as the other bot. To fix this we have tried a couple things. We first tried to have both bots act as the offensive and defensive at the same time,an offense and defensive bot similar to the base AI, however, this did not perform with our intended strategy. Our goal with our AI was to quickly grab some points, and play defense the

whole time. Two bots that do different things did not satisfy our goal, which is why we decided to stick with a non-optimal version of our AI.

4. Speculate as to why your bot was effective against some opponents but not others.

Our bot is extremely effective against any bot that rushes the middle of the map because our patrol is set up there. This leads us to being extremely weak to bots that stick to the edges of the map since there are times at which our bots fail to detect their presence. We attempted to improve this by having each defender prioritise a single defender.

One thing to note is that, especially in the early stages, our bots are playing by their own agenda and not the one from the other AI. Our offensive bot rushes at the beginning to secure our lead and plays defense until we are behind on points. With this strategy we found over a 90% win rate against the improved team when running the simulations all night. To say that our AI is effective is true but not extremely effective.

5. What was difficult?

The most difficult part of this project was understanding the code base. The ideas on what we wanted our AI to do was simple compared to finding out the available methods and documentation that we had available. This is a big reason why we opted to take the baseline team and modify it into the AI that we have functioning currently. After this initial hurdle was crossed the next major issue was having our AI, to put it simply, do what we wanted it to. We had issues with bots following the same path, failing to score, and simply looping back and forth. To solve this we had to play with the weights and analyze the consequences of our features.

6. What was fun?

Breaking down the goals of the game into logical behaviors that could be programmed was an interesting problem to handle. It led to us having to explore multiple options concerning how an agent behaved which resulted in theories for many types of agents. Creating these agents and matching them against each other was a fun way to determine holes in behaviors, as well as an effective way to determine pitfalls our agents could fall in to. The project provides a simple goal with solutions that can be incredibly complex. Trying to predict how other team's agents will behave, and creating behaviors that work effectively against them is a challenge we enjoyed.

7. Was there anything that was particularly frustrating, and how could it be fixed in the future?

The data structures provided were vague both in documentation, and the clarity of their variable names. Often, we found ourselves struggling to determine the functions of parts of the provided

code due to how abstractly they were written. Understanding how to read the code of other programmers is important, but it felt like we worked more on understanding their code than we did on designing our agents. A small overview of some of the key features could help future classes in focusing on working with agent behavior.

8. If you tried a novel behavior and it fails spectacularly, did it accomplish what you set out to do? (e.g. zerg rushing at the start, always go for the closest food, cheesing your opponent, etc)

One behavior that we implemented was starting both agents on offense. Using this tactic the agents will rush to the other side of the map leaving their own side defenseless. Assuming that they make it back alive with some food, they will most likely have a winning score and will then switch into a defensive mode. Once they are in this defensive mode they are very successful at preventing enemy players from crossing over. The problem that we faced was that in the scenario where we don't end up grabbing any food on the initial offensive rush, the AI will continue to keep rushing until they are ahead in the score; leaving no option for a defense. This strategy had its merits but in the end we decided it was too risky to keep and ended up starting with a balanced team. This decision gave us more of an opportunity to utilize our defensive bot's more reliable skillset.