Juan Andreas, Brianna De Carvalho Kavanagh

TeamSalt

CMPS 140

# Tournament Report

We first started by implementing the provided baselineTeam code in the tournament file. We had some difficulties comprehending what each class did but found out that there were two different classes in the baselineTeam that provided code for an offensive agent and a defensive agent. As stated in the instructions.html page, the offensive agent's strategy was to find the nearest food, and the defensive agent's strategy was to intercept any intruders that it smelled, otherwise it just wanders around randomly. The first problem we noticed about the offensive agent was that, although going for the nearest food pellet was an intuitive idea, there were no strategies to avoid ghosts, and living for an extra length of time to eat up as many pellets as it could would have been a better offensive strategy. This was one factor that we wanted to implement into our offensive agent. With the defensive agent, wandering around randomly was very ineffective as there was a chance that the intruder was on one side of the maze while the defensive agent was on the other side of the maze. We discovered that getting the location of adversaries requires an agent to be a certain distance away from the opposing agents and therefore, wandering randomly is very ineffective for finding the locations of opponents.

The first thing TeamSalt did to improve their agent from the baselineTeam code was adding some new features to the defensive class, SaltProtec. We realized that it would be helpful to know where food particles were located on our side of the board. Using the getFoodYouAreDefending function from the class captureAgents, we used a similar approach to how the baseline team code finds the distance to invaders we can see. We calculated the distances to the closest food particle on our side of the board, and then added this to the weight function with a value of -10. We tested some other values until we found a value which gives the best performance of the defensive agent. When we ran our program, our defensive agent found the closest food pellet and went to sit at that pellet's location. Then when the other team's pacman inevitably came for the food, the defensive agent would eliminate the pacman. While this did improve our agent, we soon realized there were problems with using this strategy. The other team would go for the capsule on our side of the board and then proceed to eat the defensive agent which was sitting by some food. This made us realize that it was important to also protect the capsule as well as the food pellets. If the invading pacman tried to collect the capsule, it would be met by the defensive agent first. Therefore the defensive agent could not be killed by pacman because the capsule was never eaten.

In an attempt to improve the offensive agent, we tried to make our agent be able to avoid ghosts. The idea behind improving the offensive agent was to pick up any detectable positions of the opponents and create a feature to avoid ghosts. This would have been done using the same method used to calculate distances to the nearest food pellets, which was to calculate maze distances. While testing, we noticed that this made the agent unable to distinguish harmless and harmful adversaries. For example, when the baselineTeam's offensive agent would cross to our side of the maze, our saltAttac agent would avoid it even when it was not necessary because it was obviously harmless (eating it would have been better). Another thing was that it had the potential to just stop in a position for reasons that we could not figure out and also get caught in a loop where the opposing ghost and attack agent would just chase each other in one area of the maze endlessly. This was a problem because there was no way to get more points which would either make us win/lose through timeouts and it tended to lose much more often.

We also thought about implementing scared times for our offensive agent, but we ended up not having enough time because of trying to fix bugs. If we were to implement it, we would have liked to have it not be concerned with chasing ghosts when the ghosts were scared. As long as they were not posing a threat, then it would have been a better option to just let them run away as our offensive agent only prioritizes eating food pellets. However, it would have also been useful to target food areas of high densities so that it maximized its time of freedom from ghosts.

When watching other teams go up against our agent, we realized that it would have been smarter to focus on defending the borders from attackers instead of focusing on defending the capsule. Our food was getting eaten by the enemy and we should have focused more on this problem. For example, there was one team who, other than effectively chasing down intruders, had their defensive agent camp on the border and catch opposing offensive agents as soon as they crossed. We also noticed that some teams turned the defensive agents into offensive agents in certain situations which would have been a smart idea. Instead of having a defensive agent that would just sit idly, the defensive agent would try to eat as much food if it can.