CS 188:

Walker Sneaker, Thursday, 7/19: Part 1

Initially attempted to just use the given features with some weights basic weights to attempt to solve this problem. With just these basic features and some well choses weights I was able to get a 7/10 on the first try. I then spent 3-4 hours attempting to adjust weights and add new features, including ones that prioritized areas of the map that are food heavy, to take the problem that was not at 9/10 test cases passed to 10/10 test cases passed with little to no success. Decided to call it a night there.

Walker Snedaker, Friday, 7/20: Part 2-3

Looked at piazza and found that the 9/10 was good enough to pass part one so I began part 2. Spent a little under an hour setting up the feature function to look at the closest food that is not in the immediate path of our teammate and accounting for the edge case when our agent has eaten all of the food that is not in our teammates path. Using only this and the score as the metric for the heuristic this, on the second try, yielded a 10/10 passed solution. After this I looked at the part 3 and spent about an hour starting to set up a minimax evaluation process to both make efficient use of our teammates broadcast, create our own broadcast, and hopefully avoid the ghosts.

Walker Snedaker, Saturday 7/21: Part 3

Spent about three hours finishing the minimax evaluation process, debugging the evaluator, and finally setting up a feature and weight function to serve as a base for the evaluation tree. As of now the weights are not correctly balanced and for most setups it appears that the function is taking too long to evaluate, I think implementing alpha beta pruning, and possibly changing what do in the case where the there are not many values for the teammates plan, might solve this problem but as of now it’s pretty late so I’ll come back and tomorrow.

Monday 7/23: Part 3

After thinking alot about ways to increase the efficiency of my algorithm I realized that I could cut back on some time by choosing a random action for the teammate pacman when no action was given in there broadcast or when they were eaten by the ghost, making there plann invalid. This increased the speed of my program by a considerable margin. Of course is still takes a massive amount of time, upward of ten minutes to run half of the ten auto-grader tests, due to the size and number of agents on each map.

Tuesday 7/24: Part 3

After deciding that my pac man was not broadcasting enough actions, usually only one or two at best I attempted to implement alpha beta pruning to speed up the code on larger mdp trees, and thus allow myself to plan ahead maybe one or two more steps. This was not that difficult of a thing to implement but still was not able to increase the speed of my program in a large enough way to add more depth to the mdp. So I realized that if the the ghost was quite far, further than some greater than one constant multiplied by the depth of the mdp tree, its immediate actions would not matter at all so we could replace minimizing over the ghosts actions with the ghost randomly action in a legal form. This greatly increased the speed of the program but sub optimal weights on my evaluation program are still causing some non-ideal behaviors, but as of just these additions the average over the fastest 8 games is 701 which is enough below the grade cut of grade val at 750 that i may just test more before updating the evaluation program.
(tuesday after 12:30 pm)
During class I was checking something on piazza and realized that I was not filtering out reverse directions and stop actions for my packman or even for the ghost, so I filtered these actions and was able to extend the depth of my search to 3, but revealed a large problem in the evaluation heuristic of my code and how it is not weighing high enough the consumption of the food.
(later tuesday)
After running a bunch of random matches decided to reconsider the reverse action but was am still removing the stop action as a possible action for packman. I also updated the heuristic to remove all dependency on the number of time packman had already visited a spot on the map because it's adition in the heuristic was causing packman to go back and forth between two squares randomly before continuing to the next food, probably to remove the weight on spots between pacman and the closest food. I also spent a fair amount of time figuring out an ideal distance to be considering the ghosts minimaxed movement, and decided that if at any point in the minimax search if the ghost was closes then 3 times the current depth of the tree it should me minimized over. This lead to a more efficient pac man but I noticed that occasionally my code would slow down when the ghost had no limitations on its movement. So to further attempt to speed up my code I tried to remove the backward actions from my consideration of the ghosts movement. But this lead to un optimal behavior so I updated the multiple to depth to set it to 2*depth and found that this lead to faster code and the optimal behavior.

Wednesday 25 2018:

Ran a bunch of matches on the website and although it worked for the most part I have some concerns about the website. First of when one of the bots times out on a turn it just returns “Agent with index 0” or “Agent with index 1” timed out, but from what i can tell which agent is which is not always the same, if there could be some sort of key displayed on the page, like agent 1 = NameOfBot and such so that us users can know if the match we just attempted to run was crashed by us and we have faulty code or if the other bot was faulty. Otherwise I like the website. Noticed some strange behavior in my code I think but without knowing which bot is which I am not sure if i should edit my bot, since I am doing ok on most matches.

Thursday 26 2018:

Now that the staff bot is up I am running even more games and seeing for the most part my bot is ok. From what I can tell though there are a couple of super difficult maps where the maze from the pac man spawn point to the rest of the map and the ghost will just sit in that choke point causing little to no way for either pac man to enter the map making the game a loss. Also I have noticed some strange behavior in the staff bot that it will when ghosts are nearby run from the ghost all the way back to the spawn point, causing a pore score when said actions occur. I believe this has to do with either giving to great of a weight to avoiding the ghost, this could also be a problem with favoring not repeating the same square. From this behavior I am a little worried that if this is the final staff bot and only a small amount of test will be run on each bot with the staff bot there is a high chance that the random scores will be bad, right now that's my only concern. Also the website, more today than other days from what I can tell, is randomly logging me out while I am waiting for a game to load. Since this is probably the last entry and I am going to submit this soon I have some last comments about the project as a whole.

Overall the project was pretty fun especially part three when the website was working. For the most part the first two parts of the project felt rather arbitrary with solutions being super simple when found and extraordinarily arbitrary to try to get to the ten out of ten value with the somewhat reflexive agents. Part three was very enjoyable when the website was working, I have already talked about how the website could have been improved so I wont go over that again here, it was and is extremely cool and fun to run my bot against a bunch of different bots and see how other bots act and respond and compare that to my bot. The last thought about the project is it kinda sucks that this project is due friday and the next project is due the very next day on saturday, it’s just a bit annoying and stressful.