COMP341 – Assignment 2 Report

Melis Oktayoglu 0064388

Q1 *What are the features you used in your evaluation function for your reflex agent? Why did you chose them? If you have too many, limit yourself to at most 5. Do you think using the reciprocals or negatives of some values is a good idea and why?*

I used the distance of the Pacman to the foods in the list. I tried Manhattan distance, maze distance and Euclidean distance but Euclidean performed better so I sticked to it. Here, I put all the distances to food into a list and pick the minimum of the food distances and multiply by 30 so that it serves as a weight when compared to distance to ghosts. Picking the minimum of distances is I thought to be useful because first the Pacman should prioritize going to the closest food, and second taking the max or the sum of the food distances isn't informative when differentiating different game states, as it is the main goal of the evaluation function in picking the optimal move. In food-related features I also included the length of food distances, subtracted it from the total score, this was useful because the min food distance alone wasn't extremely informative as there can be vast variety of game states with same min distance to food, this resulted in Pacman thrashing. Subtracting the remaining food number, makes the Pacman prioritize the states with less food remaining. I also added the scared times of ghost to the score, since this is favorable to the Pacman and factorized by 10 thinking that this will help Pacman prioritize more eating food than picking states with scared ghost. I also, did the same distance addition of ghosts in food distances with slight alterations. I again picked the min Euclidean distance but if the Pacman and ghost is in the same location, I immediately returned minus infinity so that this is never picked. And favorized the states where the Pacman is away from the ghost by a threshold. Both these cases happen if the ghosts aren't scared. If they're scared, then Pacman goes towards the ghost by the same food distance addition but this time I multiplied by 10. I think using the reciprocal of distances may be useful but if this is picked it should be careful if the divisions by 0 give desirable functionality. Also, the negatives of distances can be used but maybe this time max of food distances can be subtracted, punishing in a sense the states where there are very distant food.

Q2 *In your tests, were you able to see any speed difference between the MinimaxAgent and AlphaBetaAgent, between pacman actions? If so, why and if not why not? Is there any situation you came across that highlights this?*

Yes, AlphaBetaAgent performs faster decisions than MinimaxAgent. This is because of the pruning AlphaBetaAgent does, as it expands less nodes it is faster than Minimax agent.

Q3 *When you were running the tests in the previous question, did your pacman behave exactly in both cases? Why?*

Yes, in both algorithms Pacman picked the same decisions. This is because in the root node they both result in the same decision just AlphaBeta pruning skips exploring some children nodes. Thus, in both algorithms they will always return the same results.

Q4 *Comment on how fast your code runs. Compare it with the MinimaxAgent and AlphaBetaAgent. Note that this comparison is trickier to do. If you are not able to conclusively see anything, write what you would have expected.*

I think through my observation ExpectiMax gave faster decisions than minimax agent, but slower than AlphaBeta pruning. Since ExpectiMax doedsnt do pruning I would expect it to be slower than alphaBeta, as it expands more nodes than alphaBeta.

Q5 *We are sure that you were able to write a better evaluation function than the one we used for the programming questions 2-4. Did you change anything from your evaluation function for the ReflexAgent? If so, what were the changes? What, if anything, is different in this case? If you have written something entirely different, comment on your new evaluation function.*

The only thing I changed in the better evaluation function was also using the getScore of the current game state and summing it in the final score. Here I also wanted to omit the stop check to which I added in the first evaluation function. This was advised no to used in the pdf, so I tried altering the weights. And, I this time subtracted the ghost distance in the case it is safely far away from the PacMan, this helps Pacman not stop and go towards the ghost, sometimes because I observed that it was prioritizing running away from the ghost too much and in stead stays far away in places with no food. In the earlier function I was using a fixed number in favoring these ghost-safe distance with a static value. This was resulting in very low scores because of the inefficient time spent. But I picked a smaller weight than food distance weight in the better evaluation function.

Q6 *For both the programming questions 1 and 5, you probably needed to tune your feature weights. If so, comment on how you selected your weights, what did you prioritize and why.*

Weights I used were for food distances, scared time of ghosts and ghost distances. I respectively picked 30, 10 and 10. By these selections I wanted the Pacman to prioritize eating the food over eating the ghosts. Because I thought eating the food is more important in wining the game.