COMP341
ASSIGNMENT2
MULTIAGENT
YAKUP ENES GÜVEN
64045

Written Q1:

I chose to trade by choosing the minimum distance to the nearest food, the minimum distance to the nearest ghost, and the number of foods remaining. using distances to food and ghost here is essential for pacma to find direction. When the number decreased depending on the number of foods used, I changed my orientation towards food, this situation had to be even more careful. While doing the operations, I used the division of the numbers to give the highest value, that is, the priority, to the closest one. This is how I got the result I wanted. I used a very high negative value to escape from the ghosts, so the ghosts expressed a very bad situation and provided distance.

Written Q2:

I examined and observed the speed difference between the Minimax and AlphaBeta algorithms in terms of the movement of the paccman and the food he ate. The Minimax algorithm runs slower than the AlphaBeta algorithm at 20 seconds. This is because the AlphaBeta algorithm does not expand the agent tree if it is apparently unnecessary. The minimax algorithm expands the agent tree at the appropriate depth as specified, that is, as entered as a command. This difference causes Minimax to run slower than AlphaBeta. We can assume this as a result of the complexity comparison we saw in the lesson.

Written Q3:

In both cases, Pac-Man behaves the same and ends the game with the same scores. AlphaBeta has the same concerns as Minimax when choosing a move but works in a slightly different way. Both algorithms aim to find the best action in the agent tree and work accordingly. AphaBeta has pruning improvement to work more optimally than Minimax, but the goal is not to find the best.

Written Q4:

I couldn't do the coding part for the Expectimax algorithm, but I think it will work at the same speed as the Minimax algorithm. Both should extend the tool the same way as all three and iterate over each leaf. I can only speak on conjecture as I can't figure out how to implement them. So I don't think there will be any change in terms of speed. On the other hand, I think it will run slower than AlphaBeta algorithm because minimax was already slower, and I think it will be even slower since this algorithm will be like minimax.

Written Q5:

We can do an even more effective process by giving ghost distances and a weight variable accordingly. An even more effective pacman movement can be achieved by properly adjusting the score exchange between ghost and food.

Written Q6:
I just added ghost distance in my code in questions 1 and 5. here I differentiated the process by giving more value. It should have worked in theory, but it didn't. my aim was to move more carefully than ghosts. the weight will be processed according to the ghost distance and food distance and should choose the appropriate one.