Violetta Gkika 1115201600222

Pacman-Project 2

multiagent.py is modified as it is described in the presentation.

Q1: Reflex Agent
Evaluation function. Here we take an action and we need to return a value that evaluates this action. To do so we extract some information from the game state as shown in the given code and by finding the distance of the food and the ghosts from the position the given action leads pacman, we try to calculate a value. A better evaluation value is reached if that action leads Pacman to a position that food is closer to him and ghosts are more distant. So, in evaluation value calculation we try to give the corresponding weight to these parameters.

Q2: MinMax
MinMax makes decisions categorising agents on MAX agents that for us is, Pacman and MIN agents that for us are ghosts. For terminal states, minmax call the evaluations functions of questions1. For Pacman and ghosts, minmax calls max_value function and min_value functions correspondingly. These two functions work symmetrically. Both, explore all the possible actions an agents can make from the current position and after calculating the evaluation value of each action, choose the action that will bring the maximum or minimum (depending if its max agent or min agent) score.

Q3: Alpha Beta
Same as Min Max, but now we have added alpha and beta values and a comparison to set the corresponding values to these two new parameters, exactly as described in the presentation.

Q4: Ecpectminmax
Same as MinMax, but in this case we do not know for sure that ghosts will play optimally, so, instead of a min_value function that returns the minimum possible value we implement an expected_value function which returns a more probabilistic calculation.
In this case ghost will not go for the move that minimizes Pacman performance but will make a random move, so, to calculate the evaluations value of that action we …. the evaluation value of that move with the probability that move has to occur.
It's like, if we had a tree, the ghost nodes will be treated as chance nodes instead of min nodes.

Q5: Evaluation Function

Similar logic to the one applied on question 1, but now as we have to evaluate the whole game state and not only the action taken we consider all the features of the state. As mentioned in the comments: `The closer the food,the lesser the amount of food and capsules, the greater the distance between pacman and closest ghost, and the higher the current score the higher the evaluation value would be. So, we try to multiply each feature with the representative "weight" so as to contribute accordingly on the final value.`