Ονοματεπώνυμο: Παντελεήμων Μαλέκας
A.M: 1115201600268

Comments regarding assignment 2 (Pacman)

- Q1: Reflex Agent

For this question I implemented the evaluation function for the Reflex Agent. I used the reciprocal of important values as it was noted in the assignment's presentation. First, I get the Manhattan distances from Pacman's current position to the remaining food states and then, using those distances, the reciprocal of the closest food state. The exact same thing is done for the ghost states and capsule states. To get the result we need, this evaluation function returns our current score, plus the closest food, subtracting the closest ghost distance (so Pacman can avoid it), and we also add the closest capsule. However if the ghosts are scared, there's no need to avoid them, so the function returns the other values.

- Q2: Minimax

Three functions were defined for Minimax. Minimax_Decision, which has almost the exact same body as the next function, Max_Val, simply sets the algorithm in motion. Max_Val is called for the states that Pacman is in. In Max_Val, for every successor state, the Min_Val function is called to get its value and return the action of the successor state with the greatest value. The function returns the maximum value as well as the action that brings us to this node. Finally, Min_Val is called for the ghost states. For each ghost we have two cases: If it is the last agent we have, this means every other ghost has received its Minimax value so it's time to call Max_Val so Pacman may continue. So, for every successor state, call the Max_Val function to get its value and return the action of the successor state with smallest value. Otherwise, call Min_Val again because there are ghosts that haven't received their Minimax values. Min_Val returns the minimum value as well as the action that brings us to this node. If we ever reach a terminal state, we call the evaluation function to assign a value.

- Q3: Alpha-Beta Pruning

The algorithm is almost the exact same as in Minimax, only here the alpha and beta values are taken into account. These values are passed in every function call. For Max_Val, we check if the value of the successor state is greater than beta. If so, there's no need to explore the successors, so we prune them by returning the (value, action) tuple. Also, the alpha value is updated if need be. For Min_Val, we check if the value of the successor state is less than alpha. If so, there's no need to explore the successors, so we prune them by returning the (value, action) tuple. Also, the beta value is updated if need be.

- Q4: Expectimax

The algorithm is almost the exact same as in Minimax, only here the Exp_Val function is defined for the ghost states. In Exp_Val, the value we need to return is calculated by getting the average from the values of each successor state.

- Q5: Evaluation Function

The evaluation function is quite similar to the one I implemented in the Reflex Agent. In this one, I decided to use max values since they provided better scores. Specifically, I used the reciprocal of the furthest food, ghost and capsule states. This evaluation function will return our current score as well as the max values that were calculated.