Artificial Intelligence Course

Project 2: Multi-Agent Pacman

First name	Last name	Student number
Toni	Kuosmanen	1986123
Tapio	Kursula	2402868

Comments about the assignment (if you have)

Question 1: Reflex Agent (4 points)

Our reflex agents evaluation function gives more points to a state if it's manhattan distance to food is small. The function gives extra points if food is in the 4-neighborhood of the position of the state. If the ghosts are not scared, the function gives less points to a state if its manhattan distance to ghosts is small(It punishes getting near the ghosts). If the ghosts are scared, distance to ghosts does not matter. The function avoids hitting the ghosts at all cost, so it gives a lot of negative points to states that have the ghost in them or in the 4-neighborhood of those states.

Question 2: Minimax (5 points)

Minimax search tries to find the move that gets the best possible score for the max player while taking into account the min player working against the max player. In Pacman, the Pacman agent is the max player while the ghost are min players. The minimax search is implemented using a pair of recursive functions: minValue and maxValue. For a given game state, the maxValue returns the largest value obtained from evaluating the minValue on each of the child states of the current game state. Respectively, the minValue function returns the minimum value obtained from evaluating the maxValue on each of the game state's child states. These two functions are called in turn until a terminal state or the maximum search depth is reached. As there may be multiple ghosts agents, the minValue can be evaluated multiple times in a row. At the root node all available actions are evaluated using the minValue and the action that provides the largest score is chosen.

Question 3: Alpha-Beta Pruning (5 points)

By keeping track of the best available moves for the max and min players, the minimax search tree can be pruned. This is called alpha-beta pruning, alpha and beta being the best scores for max and min players respectively. The alpha beta pruning is be implemented by slightly modifying the minimax search. In the maxValue function each time a child node is evaluated the resulting value is compared to the beta value. If the new value is greater than the beta value it means that the min player already has a better path and the current node will not be be picked. Thus any remaining child nodes of the current node don't have to be evaluate. If the new value is smaller than the beta value, the search proceeds as usual and the alpha value is updated. The minValue functions in similar fashion, comparing each value obtained for child node to the alpha value and pruning branches that won't be picked by max player.

Question 4: Expectimax (5 points)

Minmax assumes that all agents make optimal decisions. Expectimax assumes that the decisions are stochastic in nature. Expectimax uses a probability function to determine the expected utilities. Implementation of expectimax is almost identical to minimax. Only difference is that it uses expValue instead of minValue. expValue returns the weighted average of the child states its called upon. In essence it returns the expected value of the child states.

Question 5: Evaluation Function (6 points)

Evaluation function here is the same one we used in question 1 except now we evaluate some given state instead of a successor state resulting from some action.