**Report for Assignment II Submission**

**Reflex Agent:**

In this agent implementation we choose the scoring in following way:

1. If it encounters a ghost in scared timer, we add a high value and divide it by distance, so that we get close to the ghost.

Else if the ghost is not scared, we will want to move away from it, so we subtract a value almost equal to half of what we gained during the scared ghost.

1. Next we look for capsules. If we are near it, grab it and we get a high score.
2. Now we look for food from previous board state, so that we see, if we have moved closer to food and give a good reward. If we are farther, we give lesser value by making it inversely proportional to distance.

**Minimax Agent:**

In this agent implementation we choose the scoring in following way:

Since we are implementing Minimax, where pacman is Max and ghosts are min, and moreover the ghosts are multiple, in this board, we evaluate the board state for all the ghosts and then based on whether it is pacman or ghost, we return maximum and minimum value respectively.

**AlphaBeta Agent:**

In this agent implementation we choose the scoring in following way:

1. As we are implementing Alpha Beta Pruning, we are supposed to prune, whether a certain sub branch of tree should be evaluated from alpha beta values.
2. We first start with pacman and start evaluating all the legal actions and pass on the values of alpha beta to each agent as they are min agents and after obtaining the value, it actually tries to prune the branch as the ghost may not choose the branch that maximizes the pacman score.
3. The min agents now try to expand the tree till the allowed depth and tries to run Minimax algorithm and in the process, if they encounter node value greater than alpha, it will prune the nodes and return.

**Statistics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm/Property** | **Reflex Agent(for all tests)** | **Minimax(for all tests)** | **AlphaBeta(for all tests)** |
| **Nodes Expanded** | **6587** | **8167** | **3993** |
| **Time(small classic)** | **1 sec** | **1 sec** | **0** |

**Conclusions:**

**As we** can see that Reflex Agent performs not so great, but when we use minimax, it performed better as its losing fewer games. But between minimax and alphabeta pruning we can see a drastic difference.

The time taken by minimax for smallclassic puzzle is around a sec and when we pruned the branches with alpha beta pruning, we see that time taken is almost zero and instantaneous. We can also see the difference in the number of nodes expanded and there by the memory required has also been reduced to almost half, which is a great savings. If a small game like Pacman can save so much, we can save a lot using alpha beta pruning.